On the Complexity of Problems on Order Types and Geometric Graphs

Doctoral Thesis
at
Graz University of Technology

submitted by

Alexander Pilz

February 2014

Institute for Software Technology
Graz University of Technology
A-8010 Graz, Austria

Advisor: Assoc.-Prof. Dipl.-Ing. Dr. techn. Oswin Aichholzer

To my parents for their love and constant support.

Abstract

In this thesis we consider several related topics in the field of computational geometry. We are concerned with algorithmic problems fully determined by the order type of a point set, that is, on the equivalence class defined by the orientation of point triples.

In the first part, we consider edge exchange flips in triangulations of point sets and simple polygons. In particular, we are interested in the question whether the length of the shortest sequence of flips for transforming one given triangulation into another can be determined in polynomial time. We show that the corresponding optimization problem is APX-hard for triangulations of point sets. For triangulations of simple polygons, we give a reduction showing NP-completeness of the decision version of the problem. The two proofs are fundamentally different, but use a common sub-structure, the well-known double chain, whose properties with respect to flip graphs we investigate in detail.

In the second part, we focus on algorithms that do not operate on geometric graphs, but solely on point sets. We are interested in the algorithmic power of sidedness queries, that is, of using only the predicate indicating whether an ordered triple of points is oriented clockwise or counterclockwise. In this context, we show that we can find an extreme point of a point set in linear time using only sidedness queries, and prove that the algorithm also works for abstract order types (solving a long-standing open problem). Further, we define a counterpart to the concept of vertical lines in line arrangements in the more general setting of pseudo-line arrangements. We show that an intersection of a given rank on that counterpart can be selected in linear time. We apply this tool to the classic linear-time ham-sandwich cut algorithm for bi-chromatic point sets, and thereby obtain a deterministic linear-time ham-sandwich cut algorithm that uses only sidedness queries and works also for abstract order types. Finally, we consider a recently introduced generalization of convexity, the so-called k-convexity. We address the problem of deciding whether a set of points can be polygonalized to obtain a k-convex polygonization, a question that can be answered using only sidedness queries. We show that the problem is solvable in polynomial time for $k = 2$, while the problem is NP-complete for $k \geq 3$.

Kurzfassung

In dieser Arbeit werden diverse verwandte Themen aus dem Bereich der algorithmischen Geometrie behandelt. Wir betrachten algorithmische Probleme, die vollständig durch den Ordnungstypus einer Punktmenge definiert sind, d.h. durch die Äquivalenzklasse, die durch die Orientierung geordneter Punkt-Tripel definiert ist.

Im ersten Teil betrachten wir das Austauschen von Kanten in Dreiecksnetzen auf Punktmengen und simplen Polygonen, die sog. Flip-Operation. Im Speziellen behandeln wir die Frage, ob die Anzahl von Flip-Operationen, die benötigt werden, um ein gegebenes Dreiecksnetz in ein anderes zu transformieren, in polynomieller Laufzeit berechnet werden kann. Wir beweisen, dass das Problem für Dreiecksnetze auf Punktmengen APX-schwer ist, und zeigen NP-Vollständigkeit des entsprechenden Entscheidungsproblems für Dreiecksnetze in simplen Polygonen. Obwohl sich beide Beweise grundlegend voneinander unterscheiden, verwenden sie die gleiche Teilkonstruktion, die sog. Double-Chain, deren Eigenschaften bezüglich der Flip-Operation wir detailliert untersuchen.

Im zweiten Teil konzentrieren wir uns auf Algorithmen, die ausschließlich auf Punktmengen – und nicht auch auf geometrischen Graphen – arbeiten. Unser Interesse liegt in der algorithmischen Aussagekraft von Prädikaten, die für ein gegebenes Punkt-Tripel angeben, ob dieses im oder gegen den Uhrzeigersinn orientiert ist. In diesem Zusammenhang zeigen wir, dass ein Extremalpunkt einer Punktmenge in linearer Laufzeit und nur unter Verwendung dieses Prädikats gefunden werden kann, und beweisen auch, dass der Algorithmus für abstrakte Ordnungstypen funktioniert (und lösen damit ein seit Langem offenes Problem). Des Weiteren definieren wir ein Pendant zum Konzept einer vertikalen Linie in Linien-Arrangements für die Verallgemeinerung letzerer durch Arrangements von Pseudo-Linien. Wir zeigen, dass der Schnitt mit einem vorgegebenem Rang in der Reihenfolge, in der die Pseudo-Linien des Arrangements dieses Pendant schneiden, in linearer Zeit ausgewählt werden kann. Diese Erkenntnis lässt sich auf den klassischen Ham-Sandwich-Cut-Algorithmus für zweigefärbte Punktmengen anwenden, und wir erhalten dadurch einen deterministischen Algorithmus mit linearer Laufzeit, der nur die Orientierung von Punkt-Tripeln als Informationsquelle verwendet, und der auch für abstrakte Ordnungstypen korrekt ist. Zum Abschluss widmen wir uns einer erst vor Kurzem vorgestellten Verallgemeinerung des Konvexitätsbegriffes, der sog. k-Konvexität. Wir behandeln die Frage, ob eine Punktmenge zu einem k-konvexen Polygonzug verbunden werden kann. Auch diese Eigenschaft wird nur durch den Ordnungstypus der Punktmenge definiert. Wir beschreiben einen Algorithmus mit polynomieller Laufzeit für $k = 2$ und zeigen, dass das Problem für größere Werte von k NP-vollständig ist.

Acknowledgments

This thesis would of course not have been possible without the assistance of many people. I am deeply grateful to my supervisor Oswin Aichholzer for introducing me to the field of computational and combinatorial geometry, and for all the effort he spent on me and the foundation of my academic career. This allowed me to work in this fascinating field, to gain experience, and to meet that many nice and impressive people during these years. I highly appreciate having a supervisor who was always there for discussing questions, open problems, and any other concerns, or also sometimes just for some general conversation.

I also want to thank my colleagues and office mates. With Thomas Hackl and Birgit Vogtenhuber, I had countless discussions on research problems, and they were always ready to help me in any way they could, from the time I started working on my Master's thesis project up to coping with the perils of the final typesetting of this document. I am looking forward to future joint results, coffees, and enjoyable evenings spent during workshops. I highly appreciated the company of Wolfgang Aigner as an office mate during the first year of my employment. I not only profited from his experience with the CGAL library and the more algebraic side of computational geometry, but also from his cheerful attitude. I am also grateful for having benefited from Franz Aurenhammer's impressive knowledge of literature, which he was kindly willing to share. In this connection, I also want to thank the people who made my long-term visits comfortable, namely Günter Rote and the work group in Berlin as well as Pedro Ramos and David Orden in Alcalá.

After attending my first workshop and visiting an international conference for the first time, I was impressed by the friendliness of the community. During the last years, I met many fellow researchers who generously shared their experience with me as an unexperienced beginner, who joined for pleasant and productive research sessions and discussions over dinner. Many became coauthors or even friends. Many thanks go to my coauthors, with whom I obtained many interesting results — some of them are presented in this thesis — and who helped me in improving my skills.

The majority of the results presented in this thesis have already been published. In addition to my coauthors, I want to thank the anonymous referees of these publications for helping in improving the results and their presentation. Also, Oswin, Thomas, and Birgit helped me a lot by proof-reading, and by providing precise formulations on call. I would also like to thank Pedro Ramos and Günter Rote for helpful suggestions on improving the presentation of the results presented in Chapters 2 and 3, and Chapter 6, respectively.

From 2009 to 2010, I was supported by the *Austrian FWF National Research Network 'Industrial Geometry' S9205-N12.* The work on this thesis between 2011 and 2014 was supported by a DOC-fellowship of the *Austrian Academy of Sciences.* I would like to thank the team of the *Verwaltungsstelle für Stipendien und Preise* at the Austrian Academy of Sciences for their friendly support in the course of the fellowship. I am also indebted to the former and current members of the administrative staff of the Institute for Software Technology for their assistance during this time. Part of my work was done during two long-term visits to foreign institutions, in particular the *Work Group Theoretical Computer Science* at *Freie Universität Berlin* in 2011, and the *Departamento de Física y Matemáticas* at *Universidad de Alcalá* in 2012, an opportunity I highly appreciate. I am also indebted to the people that made numerous shorter research visits possible: Alfredo García and Javier Tejel at *Universidad de Zaragoza*, Emo Welzl and Michael Hoffmann at *Eidgenössische Technische Hochschule Zürich*, Ferran Hurtado, Matias Korman, Carlos Seara, and Rodrigo Silveira at *Universitat Politècnica de Catalunya*, Jean Cardinal and Stefan Langerman at *Université Libre de Bruxelles*, Wolfgang Mulzer at *Freie Universität Berlin*, and Stefan Felsner at *Technische Universität Berlin* (in approximately chronological order).

Finally, let me express my gratitude to my family and friends for their support. In particular, I want to thank my parents, who always encouraged me and assisted me whenever they could.

TU Graz
Graz University of Technology
Senat

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..
date (signature)

Contents

1 Introduction **1**
- 1.1 Thesis Overview . . . 3
- 1.2 Definitions and Notation . . . 4

I Flips in Triangulations **7**

2 Triangulations, Flips, and Double Chains **9**
- 2.1 Introduction . . . 9
- 2.2 A Single Double Chain . . . 12
- 2.3 Multiple Double Chains . . . 19
- 2.4 A Note on Double Chains and Pseudo-Triangulations . . . 20
- 2.5 Chapter Summary . . . 21

3 Flipping in Triangulations of Point Sets **23**
- 3.1 Introduction . . . 23
- 3.2 The Reduction . . . 24
  - 3.2.1 Gadgets . . . 25
    - 3.2.1.1 Edge Cores . . . 25
    - 3.2.1.2 Crossings . . . 26
    - 3.2.1.3 Wirings . . . 26
  - 3.2.2 Analysis . . . 27
  - 3.2.3 An Improved Bound on the Performance Ratio . . . 31
  - 3.2.4 A Comparison to Lubiw and Pathak's Reduction . . . 31
- 3.3 On the Coordinate Representation . . . 33
  - 3.3.1 Placing the Points of the Convex Polygon . . . 33
  - 3.3.2 A Sufficiently Small Value . . . 34
  - 3.3.3 Tunnel Construction . . . 35
  - 3.3.4 Points in the Tunnels . . . 37
  - 3.3.5 Points for the Wiring . . . 39
  - 3.3.6 Concluding Remarks on the Embedding . . . 39
- 3.4 Chapter Summary . . . 40

4 Flipping in Triangulations of Simple Polygons 41
- 4.1 Introduction 41
- 4.2 The Rectilinear Steiner Arborescence Problem 42
- 4.3 Reducing RSA to Polygon Triangulation Flip Distance 43
  - 4.3.1 The Construction 44
  - 4.3.2 Chain Paths 45
  - 4.3.3 From an RSA to a Short Flip Sequence 46
  - 4.3.4 From a Short Flip Sequence to an RSA 46
- 4.4 A Note on Coordinate Representation 52
- 4.5 Chapter Summary 53

II On the Complexity of Some Problems on Point Sets 55

5 Combinatorial Problems on Point Sets 57
- 5.1 Point Set Classification 57
  - 5.1.1 Circular Sequences 58
  - 5.1.2 Duality and Pseudo-Line Arrangements 61
  - 5.1.3 Semispaces, Order Types, and Stretchability 64
    - 5.1.3.1 From Circular Sequences to Order Types 64
    - 5.1.3.2 Arrangements in the Projective Plane 66
    - 5.1.3.3 Abstract Order Types and Pseudo-Line Stretchability . . . 69
    - 5.1.3.4 Axioms for Abstract Order Types 72
  - 5.1.4 Generalized Configurations of Points 73
- 5.2 Motivation 76
  - 5.2.1 Robust Implementations 76
  - 5.2.2 Mechanically Proving the Correctness of Algorithms 78
  - 5.2.3 Geodesic Order Types 78
  - 5.2.4 Generation of Order Types and Abstract Extension 79
  - 5.2.5 Algorithmic Properties 80
- 5.3 Chapter Summary 81

6 Extreme Point Search in Abstract Order Types 83
- 6.1 Introduction 83
- 6.2 Realizable Sets 85
- 6.3 A General Proof of the Time Bound 87
- 6.4 Chapter Summary 90

7 Pseudo-Verticals and Ham-Sandwich Cuts for Abstract Order Types 91
- 7.1 Introduction 91
- 7.2 Levels at a Crossing 93
  - 7.2.1 Properties of a Pseudo-Vertical 94

7.2.2 Ordering Pseudo-Verticals 96
7.3 Linear-Time Pseudo-Line Selection 98
7.3.1 An Oracle for an Arrangement 98
7.3.2 Selecting a Pseudo-Line 99
7.3.2.1 Pruning the Pseudo-Lines Below the Crossing 100
7.3.2.2 Pruning the Pseudo-Lines Above the Crossing 101
7.3.2.3 Using ε-Approximation for Pruning 101
7.3.2.4 Analysis 103
7.4 Revisiting the Ham-Sandwich Cut Algorithm 103
7.4.1 Obtaining Intervals 104
7.4.2 Properties of a Trapezoid-Like Structure 106
7.4.3 A Note on the Intervals 107
7.5 Connections with Extreme Point Search and the Two-Line Partitioning Problem 107
7.6 Chapter Summary 108

8 Algorithmic Aspects of *k*-Convex Point Sets 111

8.1 Introduction 111
8.2 Preliminaries and Basic Properties 112
8.3 Deciding 2-Convexity of Point Sets 113
8.3.1 The Structure of 2-Convex Polygons 113
8.3.2 Deciding 2-Convexity of Polygons 114
8.3.3 Outline of the Algorithm 115
8.3.4 Observations and Lemmas 115
8.3.4.1 Pocket Triples 116
8.3.4.2 Mighty Pockets 118
8.3.5 Putting Things Together 124
8.4 Deciding *k*-Convexity of Point Sets 124
8.4.1 Fixed Edges 125
8.4.2 Gadgets 125
8.4.2.1 Variables 126
8.4.2.2 Literals 127
8.4.2.3 Clauses 128
8.4.3 Necessity of Satisfiability 128
8.4.4 General Point Sets 130
8.4.5 Point Set Separators 132
8.4.6 Adaption of the Gadgets 132
8.5 Abstract Order Types and 2-Convexity 135
8.5.1 Star-Shaped 2-Convex Polygonizations 135
8.5.2 Helly's Theorem for Pseudo-Lines 136
8.5.3 A Conceptually Simple Extension Algorithm 138

8.5.4 Discussion of the Extension 139
8.6 Chapter Summary 140

9 Conclusion **143**

Bibliography **145**

List of Figures

2.1 A triangulation of a point set and of a simple polygon. 9
2.2 Edge flips in triangulations. 10
2.3 A double chain. 13
2.4 Two (partial) triangulations of the double chain with a flip distance of at least $(n-1)^2$. 13
2.5 An illustration of the labeling argument for the lower bound. 14
2.6 The polygon P_D and the hourglass H_D of a double chain D. 14
2.7 An extra point in the flip-kernel of D allows flipping one triangulation of P_D to the other in $4n-4$ flips. 16
2.8 Mapping a triangulation to a local triangulation of a double chain. 17
2.9 The different possibilities for the triangle to the right of l_iu_j in the triangulation T. 18
2.10 An example illustrating why a flip cannot affect more than one local triangulation. 20
2.11 Two pointed pseudo-triangulations of a double chain that can be augmented to extreme triangulations of the double chain. 20

3.1 An embedding of a graph with the circular arcs at each edge ending at a fixed distance around each vertex. 25
3.2 The double chain at the center of an edge with the source and the target triangulation. 26
3.3 A wiring with its initial and final triangulation and a triangulation that allows to quickly perform a transformation of the edge cores. 27
3.4 Five points on the unit circle. 34
3.5 Construction to obtain bounds for δ. 35
3.6 Construction for the tunnel endpoints. 37
3.7 Choosing rational points for edge cores by Farey approximation. 38
3.8 Construction of tunnel crossings. 39
3.9 Construction of the wiring points. 40

4.1 The slide operation. 43

4.2 Placing the sink gadget. 44
4.3 A triangulation of P_D^+ and its chain path. 46
4.4 Triangulations of D_s in P_D^* with $\Delta_s = \Delta$, and with Δ being an ear and Δ_s an inner triangle. 47
4.5 Obtaining T^+ and T_s from T^*. 48
4.6 Examples of how boundary sides are added to a trace. 50
4.7 Parts of traces to be modified. 50
4.8 Construction of a small double chain for a sink. 53

5.1 Sketch of the setting analyzed by Perrin. 58
5.2 The orthogonal projection of a configuration of points on a rotating line gives its circular sequence. 60
5.3 A configuration of points in the primal and its dual arrangement of lines. . . 62
5.4 A wiring diagram. 64
5.5 Two point sets having the same order type. 65
5.6 Two point sets having the same order type but a different circular sequence. 67
5.7 Sweeping a line arrangement in $\mathbb{P}^2$ by a line rotating through ψ, shown in the sphere model of $\mathbb{P}^2$. 68
5.8 A wiring diagram representing the same abstract order type as the one in Figure 5.4. 72
5.9 Illustrations for Axioms 4 and 5 for CC systems. 73
5.10 A generalized configuration of points and the corresponding wiring diagram. 75

6.1 An arbitrary matching of edges that partition S, and m, the result of BasicMin. 85
6.2 The order on U defined by removing vertices of the intersected convex hull edge. 86
6.3 The edge $u_a u_b$ allows to prune half of the upper points. 87
6.4 A pseudo-line χ' (not part of the arrangement) witnesses that no element is to the left of z' in the primal. 89

7.1 Ordering the crossings along a vertical line. 92
7.2 Local definition of a pseudo-vertical γ_{pq}. 94
7.3 The first and the last pseudo-lines of an arrangement defining γ_{pq}. 94
7.4 A pseudo-vertical γ_{pq} in a pseudo-line arrangement. 94
7.5 A pseudo-vertical γ_{pq} in an arrangement of straight lines. 95
7.6 Four pseudo-verticals (two northbound and two southbound parts) meeting in a common cell. 97
7.7 Partitioning the pseudo-lines in $L(pq)$ along p by a pseudo-line r. 101

8.1 Lines stabbing a polygon. 113
8.2 A pocket of a 2-convex polygon. 114

8.3 The order of the vertices defined by the inflection edges of a pocket. 116
8.4 A mighty pocket. 118
8.5 Possible conflicts in intermediate polygonizations. 119
8.6 The inflection edges of the mighty pocket are on the other side of ℓ from the neighbors of the tangent points. 120
8.7 A conflict induced by the inflection line ℓ. 121
8.8 Moving a neighbor of t to the reflex chain either preserves the conflict or results in a non-simple polygonization. 123
8.9 Placement of some gadget, replacing a temporary point t, in order to prevent a line passing through three gadgets. 126
8.10 A variable gadget. 126
8.11 The interaction between literals and their variable. 127
8.12 The clause gadget and its expected polygonizations. 128
8.13 A separator produces a beam of 6-stabbers between the two dotted lines. . . 129
8.14 The whole configuration representing a formula. 129
8.15 A chain of at least ten points inside a triangle defined by any three lines implies an edge between two of the points. 131
8.16 Bunches defining hyperbeams. 131
8.17 A separator constructed using points. 132
8.18 An example of an unintended behavior of a separator gadget. 133
8.19 A separator array preventing unintended paths from passing through it. . . 133
8.20 A point set variable gadget and a literal gadget. 133
8.21 A clause gadget. 134
8.22 The order type of the vertices does not encode whether a polygon is star-shaped. 136
8.23 An extension of a configuration for which Theorem 8.32 does not apply. . . 137
8.24 An extension of a configuration of points by two points and the corresponding wiring diagram. 138

Chapter 1

Introduction

The topic of this thesis belongs to the field of computational geometry. In a general phrasing, this field is concerned with the algorithmic aspects of geometric problems [122, Chapter 8]. Often, these problems are settled in the field of discrete geometry, where we are concerned with geometric entities like points, lines, polygons, circles, and subdivisions of the plane [122, 135, 145]. The term "computational geometry" for this field was coined by Shamos [145] in the seventies of the last century. As stated in Shamos' historical account, there were many developments in the history of geometry before the advent of computer science that are related to algorithms and complexity considerations, and he points out that such results need to be brought to an efficient algorithmic form [145, p. 2]: Euclid's axiomatic approach to geometry involves the algorithmic description of constructions and the proof of their correctness, as well as the definition of allowable instruments and a set of legal operations. However, some problems remained unsolvable within these constraints (like, e.g., constructing regular polygons only with ruler and compass); it was the introduction of coordinates by Descartes and the algebraic expression of geometric problems that allowed to solve them [145, p. 7].

Discrete (and combinatorial) geometry has been a highly active field in mathematics during the last century. This work laid a rich foundation for the development and analysis of geometric algorithms, but also questions from computational geometry triggered research on combinatorial problems [54, p. VII]. A notable example is the seminal work by Goodman and Pollack on the combinatorial classification of configurations of points and line arrangements by their circular sequence and their order type, as well as the abstraction of these concepts to generalized configurations of points and pseudo-line arrangements. One of their first papers on that topic, discussing circular sequences, was motivated by the question on how a computer could generate a finite number of point sets of a given size such that every point set of the same size is "essentially the same" as one of the generated ones [77, p. 221]. For the equivalence of point sets defined by their order type, this goal was achieved by Aichholzer, Aurenhammer, and Krasser [7], at least for sets of up to eleven points [13]. While the study of circular sequences goes back to Perrin [130] in 1882, and work on pseudo-line arrangements goes back to Levi [110] in 1926, the interaction between computational and combinatorial geometry triggered the development of many interesting results within the last 35 years. In this thesis, we present algorithmic problems that have a very close relation to classic topics in combinatorial geometry. All problems presented herein are algorithmic problems on finite point sets and geometric graphs. In all the problems, we do not address, e.g., metric properties, but only the order type of a point set.

The first part of the thesis is concerned with triangulations of point sets and simple polygons. Triangulations have well-known applications in the representation of surfaces, for example in computer graphics, computer-aided design, geographic information systems, and for finite element methods [93]. Besides these direct applications, triangulations are an important data structure used in computational geometry [135, p. 234], classic examples being point location queries and shortest path calculations inside polygonal domains. Also, triangulations have been used to answer combinatorial questions, for example in Fisk's [70] simple proof of Chvátal's Art Gallery Theorem. Triangulations of convex polygons have particularly interesting combinatorial properties in combination with edge exchange flips; they are one instance of the so-called Catalan structures (a large family of combinatorial structures [152, p. 173, pp. 219–229]). See [50] for a monograph devoted entirely to triangulations in two and higher dimensions. We consider the problem of transforming one triangulation to another by a sequence of edge exchange flips, in particular showing computational intractability of the problem of determining the shortest sequence of such flips in triangulations of point sets and simple polygons. Whether a geometric graph is a triangulation of a point set is a property that is completely defined by the order type of the point set, i.e., by the orientation of each point triple. For an extensive discussion of this topic, also for the higher-dimensional setting, see [50]. In contrast to a point set given by Cartesian coordinates, the order type does not encode information like the distance between two points. In particular, while flips are used in connection with metric properties of triangulations (see, e.g., [31, 93, 109] for applications), the flip distance problem solely depends on the source and target triangulation and on the order type of the underlying point set. However, the presented hardness proofs are connected to the coordinate representation of the input. Like for many hardness proofs in computational geometry, our reduction is based on representing an instance of a problem on graphs by a geometric embedding. A proof of the embeddability of the construction in polynomial time is needed. Further, an ordinary representation of the problem is required for the hardness proof to be meaningful, which, in our case, involves giving the point set in terms of Cartesian coordinates, each being a rational number.

In the second part of this thesis, we move, to some extent, in the other direction. While during the development of geometric algorithms we usually assume to be given the points involved by their coordinates, we present results where we restrict ourselves to only using sidedness queries on the point set that defines our geometric entities, i.e., querying the triple orientations, which determine the order type of the set. In terms of the initial paragraph, we are restricting our set of legal operations. While, in the light of the huge advance of geometry triggered by Descartes introduction of coordinates, it might seem pointless to give up on such a powerful tool and to back down to a more basic axiomatic setting, we give an overview of various reasons for this. These reasons include, briefly, robustness and correctness of algorithms, their applicability to more general settings like geodesic order types, and the possibility to exhaustively enumerate all different order types (within computational limits). Further, as pointed out by Knuth [105], restricting ourselves to the tools necessary to solve a problem might increase the insight into its core.

It is an open question whether a problem whose solution is completely determined by the order type of the point set admits an algorithm using the coordinate representation that is asymptotically faster than any algorithm using sidedness queries [64].

In connection with algorithms in this restricted setting, Knuth [105] asked in 1992 whether one can select an extreme point of an abstract order type in linear time using only sidedness queries. We answer this question in the affirmative. The resulting algorithm is then used in the succeeding chapter, where we consider the problem of finding a ham-sandwich cut of a bi-chromatic point set. The classic optimal algorithm uses (like many other algorithms) the dual representation of the problem as a set of lines and the intersection points of these lines with additional vertical lines. These intersection points are not completely determined by the order type of the point set, and we show how to use an abstraction of a vertical line that can be used by the algorithm in the same way as in the classic version, making it an optimal approach even in our restricted setting. We also consider a purely discrete generalization of the concept of convexity of a point set, the so-called k-convexity introduced by Aichholzer et al. [5]. Just like a straight line intersects a convex polygon in at most one connected component, a line intersects a k-convex polygon in at most k connected components. We consider the problem of whether a point set admits a k-convex polygonization. Again, k-convexity of a point set is determined only by its order type. We show the existence of a polynomial-time algorithm for deciding 2-convexity of a point set and show NP-hardness of deciding 3-convexity.

1.1 Thesis Overview

In the first part of this thesis, we consider flips in triangulations of point sets and simple polygons. Chapter 2 gives an introduction to this class of problems and then examines a special configuration, the so-called double chain, that will be used in the remainder of that part as an essential building block for our constructions. In Chapter 3, we use these results to show that the problem of determining the flip distance between two triangulations of a point set is not only NP-complete, but as hard to approximate as the Minimum Vertex Cover problem. For simple polygons, a different approach has to be taken to show NP-completeness of the flip distance problem. The reduction, also depending on double chain constructions, is presented in Chapter 4. While the first part of this thesis is concerned with geometric graphs on point sets, the second part issues algorithmic properties of the point sets themselves. All problems are defined in terms of order types of point sets, and the algorithms are generalized for abstract order types. In Chapter 5, we give an introduction to the notion of combinatorial equivalence of point sets, to order types, and to the abstraction of the concept of point sets in the plane. Further, the research on algorithms using only sidedness queries, as well as on abstract order types, is motivated. Chapter 6 presents our first result in this abstract setting, obtaining an extreme point of an abstract order type in linear time. While being of interest in its own right, the result will be reused in Chapter 7, where we present an abstraction of a vertical line in a line arrangement, allowing the classic linear-time algorithm for finding a ham-sandwich cut of a bi-chromatic point set to use only sidedness queries. Finally, in Chapter 8, we consider a possible generalization of convexity, the so-called k-convexity of a point set. We show that determining 3-convexity of a point set is an NP-hard problem, while 2-convexity can be decided in polynomial time. We first present an algorithm that uses the actual embedding of the point set in the plane, and then show how the algorithm can be modified to obtain an algorithm working with sidedness queries.

Parts of this thesis have been presented at conferences [15, 16, 124], and published in journals [5, 14, 133]. The content presented in Chapter 7 is currently prepared for publication [67].

1.2 Definitions and Notation

Our definitions mainly follow the classic textbook by Preparata and Shamos [135]. All problems in this thesis are, to some extent, concerned with finite sets of points in the plane. Let S be such a set of n points in $\mathbb{E}^2$ (the Euclidean plane). We say that S is in *general position* if there are no three points in S that are on a common line. Throughout this thesis, all finite point sets are assumed to be in general position, unless otherwise stated. A *geometric graph* is a graph where each vertex is represented by a point, and each edge by a straight line segment between the points representing its vertices. We will make no difference between the vertices and the finite set S of points representing them, and also not distinguish edges and the corresponding line segments. Since the underlying point set S is considered to be in general position, the only points of S on an edge are the endpoints representing the vertices connected by the edge. Two edges are *crossing* if they share a point that is not an endpoint. A graph G is a *spanning graph* of a point set S if S is the union of the endpoints of the edges of G. A *polygonal chain* is given by a sequence $\langle s_1, \ldots, s_n \rangle$ of n distinct points, and is a geometric graph with the edge set $\{(s_i, s_{i+1}) : 1 \leq i < n\}$. A *polygonal cycle* is obtained by adding the edge (s_n, s_1) to a polygonal chain. A polygonal chain or cycle is *simple* if no two edges cross (recall that all the points in the sequence are distinct). A *simple polygon* is the closed finite region in $\mathbb{E}^2$ bounded by a simple polygonal cycle denoted by ∂P. We call ∂P the *boundary* of P, and call $P \setminus \partial P$ the *interior* of P. The vertices and edges of P are the vertices and edges of ∂P, respectively. We call a vertex v of P *convex* if the angle in the interior of P between the two edges of ∂P incident to v is less than 180°, and *reflex* otherwise (recall that we assume general position of the vertices). A simple polygon is *convex* if all its vertices are convex. The *convex hull* of a point set S is denoted by $\mathrm{CH}(S)$ and is the smallest convex polygon that contains S.[1] Note that the vertices of $\mathrm{CH}(S)$ are elements of S. A simple polygon P is *star-shaped* if there exists a point z such that for all points p in P the segment zp is contained in P. The *kernel* of P is the union of all such points z. A polygon with non-empty kernel is *star-shaped*. Given a simple polygon P, a line intersecting ∂P in exactly one connected component is called a *supporting line* of (the boundary of) P. The supporting line of two points is the line that contains these two points. The supporting line of an edge is the supporting line of its endpoints. Given two points a and b, we will denote both the supporting line of a and b as well as the line segment (or edge) spanned

---

[1] Note that this definition differs, e.g., from the one by Preparata and Shamos [135, p. 18], who define the convex hull as the boundary of this region, and, in essence, also define simple polygons equivalently to our definition of simple polygonal cycles.

by these two points by ab; it will be clear from the context whether the line or the segment is addressed. We say that a point is left of a directed edge ab if the point is to the left of the directed supporting line ab (i.e., the supporting line directed from a to b).

Part I

Flips in Triangulations

Chapter 2

Triangulations, Flips, and Double Chains

This first part of the thesis presents two hardness results on triangulations. Triangulations are ubiquitous in computational geometry and related fields [93]. There is a considerable amount of work on locally transforming triangulations, in particular by the repeated exchange of single edges, so-called edge flips. This chapter gives preliminary definitions and discusses the so-called double chain, a sub-structure that we will use as the main building block for proving our results in Chapter 3 and Chapter 4. The content of this chapter (apart from Section 2.4) presents parts of two papers that have already been presented in [16] and published in [133].

2.1 Introduction

In this chapter, we consider both triangulations of point sets and of simple polygons. Given a finite set S of points in the Euclidean plane, a *triangulation of* S is a maximal straight-line crossing-free graph on S. Analogously, given a simple polygon P, a *triangulation of* P is a maximal straight-line crossing-free outerplanar graph whose outer face is the complement of P. For both S and P, a triangulation is a tessellation into triangles, in one case of $\mathrm{CH}(S)$ and in the other of P; see Figure 2.1.

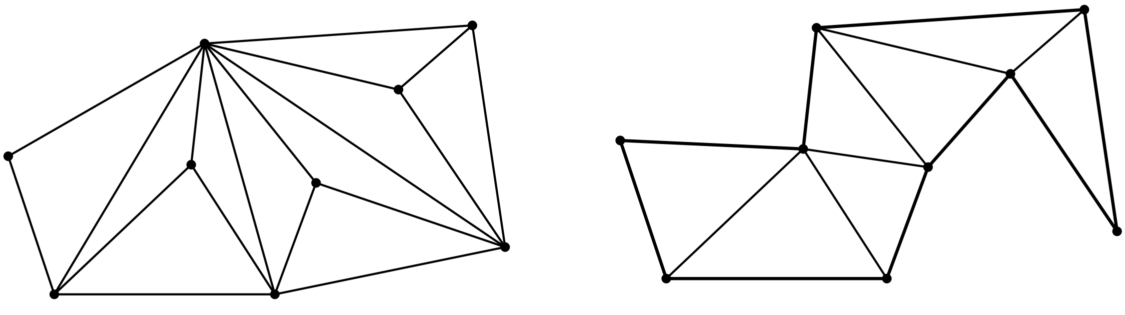


Figure 2.1: A triangulation of a point set (left) and of a simple polygon (right).

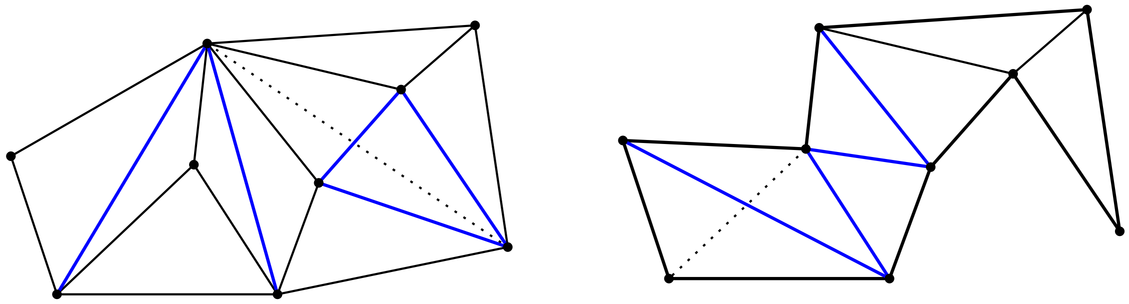


Figure 2.2: Edge flips in triangulations. The replaced edge is shown dotted, flippable edges are blue.

In the most general setting, an edge flip is the operation of removing one edge of a graph of some predefined class and replacing it by a new one such that the resulting graph is again of the same class. Edge flips were supposedly first considered by Wagner [160] for the class of maximal planar graphs.[1] He showed that every maximal planar graph on n vertices can be transformed into any other by a sequence of $O(n^2)$ edge flips. This bound was later reduced to $O(n)$ by Komuro [106]. Bose and Hurtado [36] give an extensive survey on the flip operation within various classes of graphs. In our work, we consider only flips in triangulations of point sets and polygons. Let T be a triangulation of a point set or a simple polygon. An *edge flip* is the operation of removing an edge e of T and adding a different edge f such that the resulting graph $\tilde{T}$ is again a triangulation.[2] This requires the two empty triangles incident to e to form a convex quadrilateral, which is the same as the one formed by the triangles incident to f in $\tilde{T}$. Such edges are called *flippable*. In particular, only the *diagonals* of a triangulation of a simple polygon (i.e., the edges not on the boundary) can be flippable. Further, no edge on the convex hull boundary of a point set is flippable. Since we are defining the operation on geometric graphs, a flip on a triangulation of a point set S always results in a triangulation of S, and also a flip on a triangulation of a simple polygon P results again in a triangulation of P. See Figure 2.2 for an illustration. For every point set and every simple polygon, the flip operation defines the graph $\mathcal{G}$ of triangulations, also called the *flip graph*. For a given point set S or a polygon P, the vertex set of $\mathcal{G}$ is the set of all triangulations of S or P, respectively. Two vertices in $\mathcal{G}$ are adjacent if the corresponding triangulations can be transformed into each other by a single edge flip (observe that the flip operation is reversible). A path σ in $\mathcal{G}$ is called a *flip sequence*; i.e., σ is a sequence of triangulations such that two adjacent triangulations can be transformed into each other by exactly one flip. The *flip distance* between two triangulations is their distance in $\mathcal{G}$ (i.e., the length of a shortest flip sequence between them.

---

[1] This class is sometimes also referred to as "triangulations" since every face in a plane straight-line embedding of such a graph is a triangle. In contrast to triangulations (as geometric graphs), a straight-line plane drawing of the edge added by a flip operation might not be possible. Throughout this work, a triangulation is always understood to be a geometric graph.

[2] Note that the triangulation of a point set and the triangulation of a simple polygon are different classes of graphs. Since for both classes our definitions and reasoning are virtually the same, we only point out the difference when it is not clear from the context or of particular importance.

Lawson [108] showed that $\mathcal{G}$ is connected for any point set S with diameter $O(n^2)$, where $n = |S|$. Hurtado, Noy, and Urrutia [95] proved that this bound is tight by giving a lower bound construction that holds for both point sets and simple polygons. They also showed that the diameter of the flip graph of a simple polygon with n vertices of which k are reflex is in $O(n + k^2)$ (an $O(n^2)$ upper bound on the flip distance for triangulations of simple polygons was probably first obtained by Bern and Eppstein [31] in terms of constrained Delaunay triangulations).

Flips in triangulations are used for enumeration and as a local operation to generate meshes of good quality according to a predefined criterion [36]. For example, Lawson [109] showed that one can always obtain the Delaunay triangulation after $O(n^2)$ locally improving flips. The same result can be obtained for polygons when considering the edges of the polygon as fixed edges in a constrained Delaunay triangulation [31]. The Delaunay triangulation optimizes several criteria. Also, heuristic methods for improving other properties of triangular meshes may apply local optimization using flips in combination with techniques like simulated annealing. See [31, 93] for information on the topic of mesh optimization. Another reason for the continuing interest in flips in triangulations is the bijection between binary trees and triangulations of convex polygons. There, a flip corresponds to a rotation in the binary tree. Properties of the flip graph for convex polygons were studied in the landmark paper of Sleator, Tarjan, and Thurston [149]. They show that, for $n > 12$, the flip distance between two triangulations is at most $2n - 10$ and that, for sufficiently large n, this bound is tight. In a recent preprint, Pournin [134] shows a general lower bound construction for convex polygons, implying that the bound $2n - 10$ is tight for all $n > 12$.

Interestingly, the flip distance problem is still open for point sets in convex position (or equivalently, convex polygons), regardless of the intensive investigation of that structure within the last 25 years. The problem was apparently first considered by Culik and Wood [49] in 1982. Efforts were made in solving special cases and approximating the flip distance in polynomial time. The results by Sleator et al. [149] lead to an algorithm to obtain an approximation of the flip distance within a factor of 2. Li and Zhang [111] give an algorithm that approximates the flip distance within a factor depending on the maximal vertex degree Δ in source and target triangulation, obtaining a performance ratio bound of $2 - 2/(4(\Delta - 3)(\Delta + 4) + 1)$. Cleary and St. John [45] show that the problem is fixed-parameter tractable in the flip distance. Bose et al. [37] most recently considered edge-labeled triangulations, i.e., triangulations in which each edge has a distinct label, and, after a flip, the new edge gets the label of the removed edge. For the flip distance problem, not only the edges but also their labels are given for the target triangulation. They show that, in this setting, the flip distance can be $\Theta(n \log n)$ in the worst case, and gave an $O(\log n)$-factor approximation algorithm for computing the flip distance between two edge-labeled triangulations.

For general point sets, Hanke, Ottmann, and Schuierer [89] show that the length of a shortest path between two triangulations in $\mathcal{G}$ can be bounded from above by the number of crossings between the edges of the two triangulations. They deviate from the common proof pattern of flipping to a canonical graph within a certain number of flips (which was demonstrated already by Wagner [160]) by instead showing that the number of crossings in the union of the two triangulations can be reduced by a single flip. Eppstein [60] gives a polynomial-time algorithm for computing a lower bound; note that the point sets for which Eppstein's result is tight must not contain empty convex 5-gons. This property requires that more than two points are placed on a common line if the set has 10 or more points (see, e.g., [1]). For both of our hardness results, we make the common assumption that the points of a point set and the vertices of a polygon are in general position.

Despite these results, the complexity of determining the flip distance between two triangulations has been unknown both for point sets and simple polygons. In Chapter 3, we show that the problem is APX-hard for triangulations of point sets (and actually is at least as hard to approximate as Minimum Vertex Cover), which sheds light on a "fundamental open issue" [36] in the study of flip graphs. It has been addressed as an open problem in [89] already in 1996, and, recently, in a monograph by Devadoss and O'Rourke [52, p. 71]. NP-completeness of the problem has simultaneously and independently been shown by Lubiw and Pathak [115]. However, their reduction is from the Planar Cubic Vertex Cover problem, for which a PTAS exists [25, 28] (see also [23, p. 369]), and the reduction can therefore not be adapted directly to show APX-hardness. Their reduction also uses the double chain and some of its properties; see Section 3.2.4 for a sketch of their approach and a comparison with ours. For triangulations of simple polygons, we present a reduction to show that the problem is NP-complete in Chapter 4 (covering joint work with Oswin Aichholzer and Wolfgang Mulzer that has already been presented in [16]).

In this connection, it seems also worth mentioning flips in pseudo-triangulations. A *pseudo-triangle* is a simple polygon with exactly three convex vertices. A *pseudo-triangulation* of a point set is the partition of the convex hull into pseudo-triangles without additional vertices. A pseudo-triangulation is *pointed* if every vertex is incident to an angle larger than 180°. See [139] for a survey and detailed definitions. Since a triangle is also a pseudo-triangle, pseudo-triangulations can be seen as a generalization of triangulations. Comparable to flips in triangulations, any two pseudo-triangulations can be transformed into each other by a sequence of operations comprising the removal, addition, and exchange of edges (see [4] for details, where it is shown that a linear number of such operations always suffices). A particularly interesting case are flips in pointed pseudo-triangulations. There, every inner edge is flippable and the flip graph has diameter $O(n \log n)$ [30]. In contrast to triangulations, no non-trivial lower bound is known [139]. Also, the problem of determining the shortest flip sequence between two pointed pseudo-triangulations is open.

2.2 A Single Double Chain

The construction shown by Hurtado, Noy, and Urrutia [95] to have a quadratic flip graph diameter is the so-called double chain. The double chain was probably first used by García, Noy, and Tejel [71] for giving lower bounds on the number of crossing-free geometric graphs on given point sets. (The first written use of the name "double chain" for this class of point sets seems to be in a paper by Santos and Seidel [141] on an upper bound for the number of triangulations.) It plays a central role in both reductions in the following two chapters. For this purpose, we provide the relevant properties of double chains in this section.

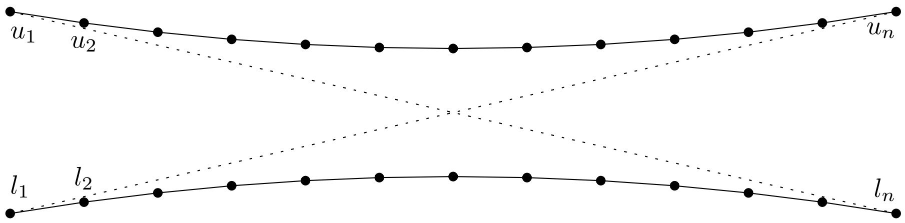


Figure 2.3: The points are divided in an upper and lower chain, each chain being in convex position in a way that every point of the lower chain "sees" every vertex of the convex hull of the upper chain, and vice-versa.

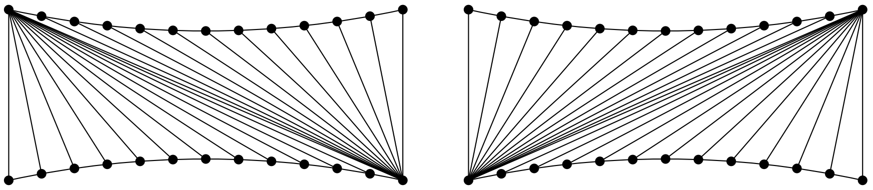


Figure 2.4: Two (partial) triangulations of the double chain with a flip distance of at least $(n-1)^2$.

See Figure 2.3. A *double chain* D is a point set of $2n$ points, n on the *upper chain* and n on the *lower chain*. Let these points be $\langle u_1, \ldots, u_n \rangle$ and $\langle l_1, \ldots, l_n \rangle$, respectively, ordered from left to right. Any point on one chain sees every point of D on the convex hull boundary of the other chain (i.e., the interior of the straight line segment between these two points does not intersect the convex hulls of the two chains), and any quadrilateral formed by three points of one chain and one point of the other chain is non-convex. Hurtado, Noy, and Urrutia [95] show that the flip graph of the double chain has quadratic diameter. Let P_D be the polygon $\langle l_1, \ldots, l_n, u_n, \ldots, u_1 \rangle$. The edges $u_i u_{i+1}$ and $l_i l_{i+1}$ for $1 \leq i < n$ have to be part of every triangulation of D since there does not exist a straight-line segment between two points of D that crosses any of them (such edges are called *unavoidable*). Therefore, we only need to consider the triangulation inside P_D for the following result, and hence the result, while stated for D, is also valid for P_D.

Theorem 2.1 (Hurtado, Noy, Urrutia [95]). *Consider any triangulation T_1 of D where u_1 is adjacent to each of $l_1, \ldots, l_n$, and any other triangulation T_2, where l_1 is adjacent to $u_1, \ldots, u_n$. The flip distance between T_1 and T_2 is at least $(n-1)^2$.*

See Figure 2.4 for the relevant parts of the two triangulations. In their proof, Hurtado et al. [95] label the triangles inside P_D that have two points on the upper chain with `1` and the ones with two points on the lower chain with `0`. Consider a horizontal line ℓ that separates the two chains. The triangles crossed by ℓ define, from left to right, a sequence σ of $(n-1)$ elements labeled `0` and $(n-1)$ elements labeled `1`, see Figure 2.5. Note that there are no triangles of a third type stabbed by ℓ. Further note that we do not care about the triangulation of the convex hull of either chain; the lower bound on the flip distance stems from the part stabbed by ℓ. It is easy to see that only an edge adjacent to two differently labeled triangles can be flipped in the stabbed part. This corresponds to

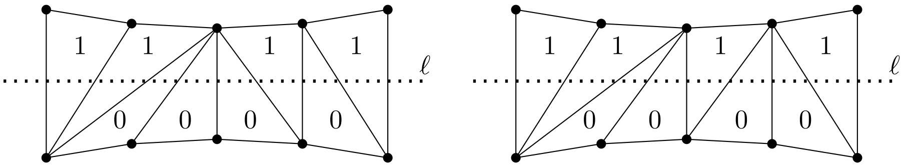


Figure 2.5: An illustration of the labeling argument for the lower bound. By the flip, the sequence changes from ⟨11000101⟩ to ⟨11001001⟩.

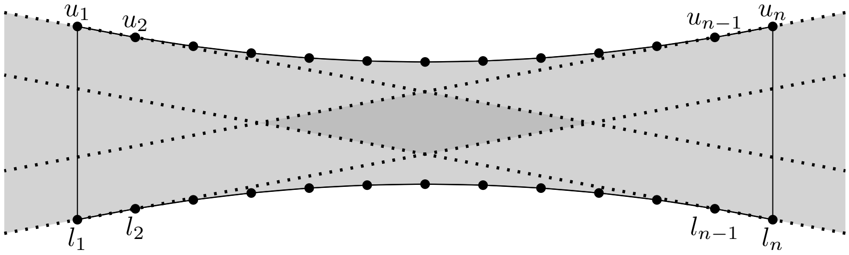


Figure 2.6: The polygon P_D (bounded by solid lines) and the hourglass H_D (gray) of a double chain D. The diamond-shaped flip-kernel can be stretched arbitrarily by flattening the bend of the chains.

exchanging an adjacent pair of `0` and `1`. Flipping the first triangulation to the second one corresponds to transforming the sequence $\sigma_1 = \langle(\mathtt{0})^{n-1}(\mathtt{1})^{n-1}\rangle$ to $\sigma_2 = \langle(\mathtt{1})^{n-1}(\mathtt{0})^{n-1}\rangle$, which leads to the desired bound. We call these two triangulations (shown in Figure 2.4) the *extreme triangulations* of D. The triangulation T_u of P_D where u_1 has maximum degree is called the *upper extreme triangulation* (observe that this triangulation is unique.) The triangulation T_l of P_D where l_1 has maximum degree is called the *lower extreme triangulation*.

In Section 4.3.2, we will observe a correspondence between double chain triangulations and paths on the integer grid that also allows for rather intuitive proofs of lower bounds, which will be useful for the construction used for triangulations of simple polygons.

Our next step will be to gain more insight into the way the flip graph is altered by the addition of points. For the following definition refer to Figure 2.6.

Definition 2.1. *Let D be a double chain of $2n$ points, and consider the convex hulls of the upper and the lower chain. Let H_D be the subset of the plane such that for any point $p \in H_D$ there exist some i, j, $2 \leq i, j \leq n-1$, with the triangle pu_il_j being interior-disjoint with the convex hulls of the upper and lower chain. We call H_D the* hourglass *of the double chain. The* flip-kernel *of a double chain D is the subset of the plane such that, for all i, j, $1 \leq i, j \leq n$, and every point p in the flip-kernel, the segments pu_i and pl_j are both interior-disjoint with the convex hulls of the upper and lower chain.*[3]

---

[3] Note that the flip-kernel of D may not be completely inside the polygon P_D (but no point in the flip-kernel is outside the hourglass of D). This is in contrast to the common use of the term "kernel" in visibility problems for polygons.

Observe that the flip-kernel is the intersection of the open half-planes below u_1u_2 and $u_{n-1}u_n$, as well as above l_1l_2 and $l_{n-1}l_n$. The hourglass is an unbounded region defined by the edges of P_D and the rays defined by the first and the last vertex pair of each chain.

Let us add a point v inside the flip-kernel of D. From any triangulation of the resulting set $D \cup \{v\}$, we can flip the edges between the chains such that they are incident to v. Reaching this canonical triangulation only requires a linear number of flips. This fact is well-known folklore, see, e.g., [157] for a printed description. Consider the case where v is placed outside P_D but inside the flip-kernel of D (observe that the flip-kernel can be stretched by flattening the bend of the chains). Add edges from v to u_n and l_n to again have a triangulation, as shown in Figure 2.7. Then, for flipping all possible edges to be incident to v, we need at most $2n-2$ flips.

For our result on point sets in Chapter 3, we are only interested in this upper bound of $4n-4$. In Chapter 4, we will also make use of the following lower bound.

Lemma 2.2. *Let P be a polygon that contains P_D and has $\langle l_1, \ldots, l_n \rangle$ and $\langle u_n, \ldots, u_1 \rangle$ as part of its boundary. Further, let T_1 and T_2 be two triangulations that contain the upper extreme triangulation and the lower extreme triangulation of P_D as a sub-triangulation, respectively. Then T_1 and T_2 have flip distance at least $4n-4$.*

Proof. We slightly generalize a proof by Lubiw and Pathak [115] for double chains of constant size.

The triangulation T_1 has $2(n-1)$ triangles with an edge on the upper or the lower chain of D. For each such triangle, the point not incident to that edge is called the *apex*. For each triangle with an edge on the upper chain, the apex must move from l_n to l_1, and similarly for the lower chain. There are three types of flips: (1) exchange an edge between the upper and the lower chain by another edge between the two chains; (2) exchange an edge between the two chains by an edge between a vertex of D and a point outside D, or vice versa; and (3) a flip where less than three of the four points involved are in D. A flip of type (1) moves the apex of two triangles by one, a flip of type (2) moves the apex of one triangle from D to a point outside D or back again, and a flip of type (3) does not move any apex along a chain or between D and a point not in D. Hence, we can disregard flips of type (3). If moving an apex involves at least one flip of type (2), then we can charge at least two flips to the corresponding triangle, one to move the apex to a point not in D and one to move it back again. If moving an apex uses no flip of type (2), then the corresponding triangle needs at least $n-1$ flips. Each such flip moves the apex of one other triangle. Thus, we can charge $(n-1)/2$ flips to each such triangle. Hence, for $n > 5$, the cheapest method is to use flips of type (2). This yields the claimed bound. □

The following result, which shows that the quadratic lower bound holds if no point inside the hourglass is used to shorten the flip sequence, gives the main property of the double chain we will use.

Theorem 2.3. *Let D be a double chain of $2n$ points and let $S \subset \mathbb{E}^2 \setminus P_D$ be a finite point set. Let T_1 and T_2 be two triangulations of $S \cup D$ such that P_D is triangulated with one extreme triangulation of D in T_1 and with the other extreme triangulation of D in T_2. Further, let σ be a flip sequence from T_1 to T_2. Assume that, throughout σ, no edge incident to a point of $S \cap H_D$ intersects the interior of P_D. Then $|\sigma| \geq (n-1)^2$.*

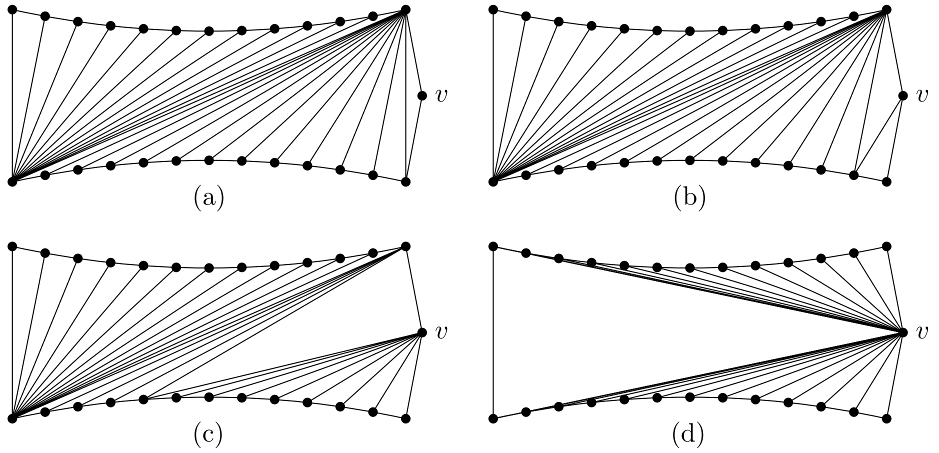


Figure 2.7: An extra point v in the flip-kernel of D allows flipping one triangulation of P_D (a) to the other in $4n-4$ flips. Note that an edge common to source and target triangulation is temporarily flipped (b).

While the theorem is stated in terms of triangulations of point sets, one can observe throughout the proof that it holds as well for triangulations of any simple polygon with vertex set $S \cup D$.

In order to prove Theorem 2.3, we consider a mapping L from the set of triangulations of $S \cup D$ in σ to the set of triangulations of the polygon P_D. When flipping an edge in a triangulation T of σ, at most one edge is flipped in the corresponding triangulation $L(T)$ of P_D. Observe throughout the description that, informally, the mapping corresponds to continuously introducing the edges of the chains along the arrows drawn in Figure 2.8, while continuously sliding the edges of T accordingly.

Consider any triangulation T of $S \cup D$ in σ. If all edges of P_D are present, $L(T)$ equals the triangulation of P_D in T (note that this is also the case for the triangulations T_1 and T_2 of Theorem 2.1). Otherwise, consider the following construction (see Figure 2.8 for an example). For any edge e of T that intersects the hourglass of D and does not have any endpoint in the interior of the hourglass, we draw an edge e' of $L(T)$ in the following way. If one of the endpoints of e is on a vertex of P_D, then also one endpoint of e' is on that vertex. If e passes through an edge $u_i u_{i+1}$ or $l_j l_{j+1}$, then the corresponding upper or lower endpoint of e' is set to u_{i+1} or l_{j+1}, respectively. If e passes through one of the rays defining the hourglass, then the corresponding endpoint of e' is mapped to the endpoint of the chain defining the ray; for example, if e' passes through the ray through u_1 (starting at u_2) but not through the edge $u_1 u_2$, then the upper endpoint of e' is placed at u_1, such that e' is contained in P_D. If an edge of T does not intersect the hourglass of D or has an endpoint in the interior of the hourglass, it is ignored by the mapping.

Let $T' = L(T)$ be the graph induced by the new edges, and let the edges of T that pass through the hourglass but do not have an endpoint in D be called *wide* edges. We call the construction T' the *local triangulation* of D when T is clear from the context. The following lemmata show that T' actually is a triangulation of P_D.

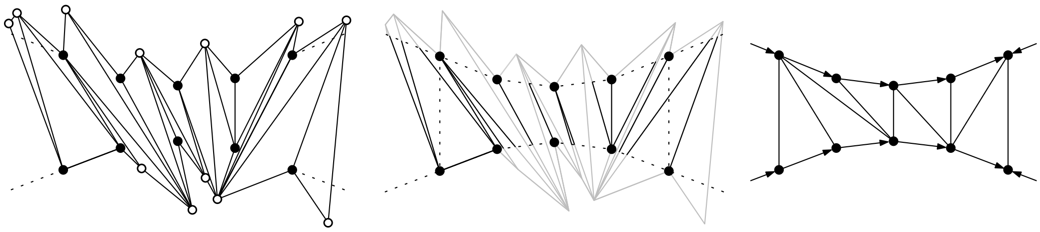


Figure 2.8: Mapping a triangulation to a local triangulation of a double chain. To the left, all triangles intersecting the hourglass of D are shown, the points of S are white. Visually, one can think of "cutting" the edges at the boundary of the hourglass (middle) and moving (and merging) the endpoints to the next point (right).

Lemma 2.4. *For every wide edge $e \in T$ that is mapped to $e' \in T'$, there is a different edge $\tilde{e} \in T$ that is also mapped to e' and that has an endpoint $p \in D$.*

Proof. Let e' be $u_i l_j$. Consider first the case where both endpoints of e' are on the same side of (the directed line supporting) e. Consider the empty triangle t of T incident to e that has its apex a on the same side of e as e'. If a is outside the hourglass of D, then another wide edge f of t is also mapped to e'. In that case we continue the argument with f, as e and f are both mapped to the same edge. If a is not outside the hourglass, then a equals either u_i or l_j, as otherwise t would contain one of them (recall that no edge of t is incident to a point of S inside the hourglass). Hence, one of the edges of t incident to a is also mapped to e'.

For the case where the two endpoints of e' are on different sides of e (i.e., one of the endpoints of e' is u_n or l_n), the argument is almost the same. Without loss of generality, let $i = n$ and l_j be to the right of e (note that j may be n). Therefore, u_n is to the left of e. Again, consider the empty triangle t of T incident to e with apex a to the right of e. Again, if a is outside the hourglass of D, there is another wide edge f of t that is also mapped to e'. If a is not outside the hourglass, then $a = l_j$; this follows from the construction of D and the fact that the lower endpoint of e is outside the hourglass. Hence, an edge of t incident to a is also mapped to e'. □

Lemma 2.5. *Every point p of D is incident to at least one edge e of T such that e disconnects the hourglass of D.*

Proof. This follows directly from the construction of D. Suppose there is no such edge, and recall that there is also no edge incident to a point in the interior of the hourglass. Then there is an angle larger than π incident to p, and the wedge defined by this angle contains points. This contradicts the fact that T is a triangulation. □

Lemma 2.6. *T' is a triangulation of P_D.*

Proof. We have to prove that T' is crossing-free and maximal in P_D.

Lemma 2.4 allows us to only consider non-wide edges. With all relevant remaining edges of T being incident to a point in D, the fact that T' is crossing-free follows from T

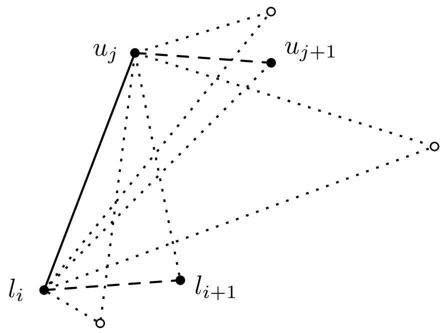


Figure 2.9: The different possibilities for the triangle to the right of l_iu_j in the triangulation T.

being crossing-free, as the mapping only "moves" the endpoints of the edges of T to the next point of D.

If T' were not maximal, there would exist a quadrilateral q inside P_D that is spanned by points of D and whose interior does not intersect any edge. If q is not convex, this would mean that no edge of T is incident to the reflex vertex of the quadrilateral. But this cannot happen due to Lemma 2.5 (an edge at that vertex in T that dissects the hourglass is mapped to an edge with the same property). If q is convex, it is of the form $l_il_{i+1}u_{j+1}u_j$. If an edge of T would have passed through the side l_iu_j, the quadrilateral would not be empty of edges. Hence, l_iu_j must have been a part of T. See Figure 2.9. Since there are points to the right of the edge l_iu_j, there has to be a triangle of T adjacent to l_iu_j having its third vertex to the right of that edge. If the third vertex of the triangle is to the right of l_il_{i+1} or to the left of u_ju_{j+1}, one side of the triangle is mapped to a diagonal of q or the triangle would contain l_{i+1} or u_{j+1}. However, if the third vertex of the triangle is to the left of l_il_{i+1} and to the right of u_ju_{j+1}, it is inside the hourglass of the double chain. Hence, there is no empty quadrilateral in P_D, which completes the proof. ☐

At first sight, it might be conceivable that a flippable edge e of T is mapped to a non-flippable edge e' and that flipping e to an edge f results in an illegal flip of e' in the mapped triangulation $L(T)$. Recall, however, that the flip operation is defined as removing one edge of a triangulation and replacing it by another one. Since the previous lemma proves that before and after the flip we have a triangulation given by mapping each edge, we know that if flipping e changes $L(T)$, then e' must be flippable as well. (Note, however, that if flipping e does not change $L(T)$, there is another edge mapped to e', and e' may or may not be flippable; this will be discussed in Lemma 2.8.)

Since any flip in T results in at most one edge being flipped in T', the lower bound construction holds: a shorter flip sequence with points outside the hourglass would immediately imply a shorter flip sequence between T_1 and T_2 in the proof of Theorem 2.1. This completes the proof of Theorem 2.3.

Theorem 2.3 will be used in Chapter 3 and also in Chapter 4. For the reductions for simple polygons, we actually only need the following, more restricted variant of the result.

Corollary 2.7. *Let P be a polygon that contains P_D and has $\langle u_n, \ldots, u_1, l_1, \ldots, l_n \rangle$ as part of its boundary. Let T_1 and T_2 be two triangulations that contain the upper and the* *lower extreme triangulation of P_D as a sub-triangulation, respectively. Consider any flip sequence σ from T_1 to T_2 and suppose there is no triangulation in σ containing a triangle with one vertex at the upper chain, the other vertex at the lower chain, and the third vertex at a point in the interior of the hourglass of P_D. Then $|\sigma| \geq (n-1)^2$.*

2.3 Multiple Double Chains

Theorem 2.3 is, however, of little use when we try to construct a point set that contains many double chains and try to argue that the flip distance between two triangulations of the set is bounded by the sum of the distances between the local triangulations of these double chains. One could imagine that a flip in the overall triangulation leads to changes in the local triangulations of several double chains. In this section, we prove that this is not possible. Keep in mind that it is a necessary condition that, for any double chain D, all other double chains are outside the hourglass of D and their polygons do not intersect.

Lemma 2.8. *Let e be a flippable edge of any triangulation T of $D \cup S$ that is mapped to the edge e' in the corresponding local triangulation T' of a double chain D. Then flipping e changes the local triangulation only if no other edge is mapped to e'.*

Proof. Suppose e is not the only edge mapped to e'. If we remove e from T, the graph on D defined by the mapping is still the local triangulation T'. If we add the new edge f after the removal of e, f must also be mapped to some existing edge f' in T' (which might not be e') or is not mapped at all, as otherwise T' would not be a triangulation. □

Note that because of Lemma 2.8, flipping an edge that is wide for a double chain does not change the local triangulation of that double chain. Therefore, a flip can only change at most four local triangulations. Actually, we can prove the following more accurate result.

Lemma 2.9. *Let D_1 and D_2 be two double chains in a point set S. If each of D_1 and D_2 is outside the hourglass of the other and $P_{D_1} \cap P_{D_2} = \emptyset$, each flip in a triangulation of S affects at most one of the two local triangulations.*

Proof. If the flipped edge e or its replacement f do not both have an endpoint in the same double chain D, then at least one of e or f either does not dissect the corresponding hourglass or is a wide edge of D. It follows from Lemma 2.8 that such a flip does not influence the local triangulation of D. Hence, in the only remaining case there is a quadrilateral that has two adjacent points in D_1 and two adjacent points in D_2 and contains a flippable edge. Let the quadrilateral be $abcd$. Without loss of generality, let a and b be part of D_1 and $e = ac$. See Figure 2.10. Suppose, for the sake of contradiction, that we flip the edge ac and the flip changes the local triangulation of D_1. Then ac has to dissect the hourglass of D_1. Then, however, ad, too, dissects the hourglass and crosses the same edge of P_{D_1} as ac (since the triangle acd is empty). Hence, ac and ad are mapped to the same edge in the local triangulation of D_1, a contradiction due to Lemma 2.8. □

Corollary 2.10. *If a point set consists of m double chains, each of size $2n$, and for every double chain all other points are outside its hourglass, then the flip graph diameter of the whole set is in $\Omega(mn^2)$.*

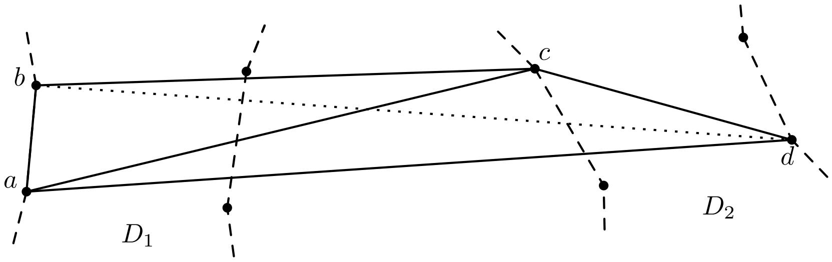


Figure 2.10: An example illustrating why a flip cannot affect more than one local triangulation. The edge *ac* is mapped to the same edge as *ad* in the local triangulation of D_1.

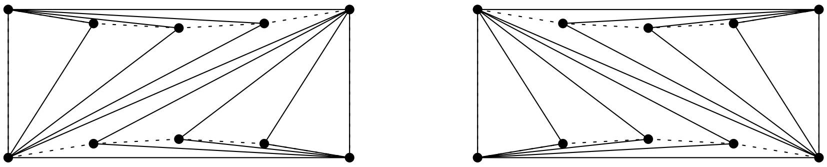


Figure 2.11: Two pointed pseudo-triangulations of a double chain that can be augmented (using the dotted edges) to extreme triangulations of the double chain.

2.4 A Note on Double Chains and Pseudo-Triangulations

In the introduction of this chapter, we also discussed pseudo-triangulations as a generalization of triangulations. There is, as already mentioned, no non-trivial known lower bound for the flip graph diameter of pointed pseudo-triangulations of a point set. However, we can give such a bound in a restricted setting. Kettner et al. [99] showed that for every point set there exists a pointed pseudo-triangulation where each pseudo-triangle has at most four vertices, i.e., the face degree is bounded by four. While it is not known whether the flip graph for such pointed pseudo-triangulations is connected at all (connectedness has been shown for a combinatorial counterpart [10]), we can show that, for any constant face degree bound, there exists a pair of pointed pseudo-triangulations that have a flip distance of $\Omega(n^2)$ by the following argument.[4] Consider the two pointed pseudo-triangulations of the double chain with maximal face degree four in Figure 2.11. Observe that we can add edges to the pointed pseudo-triangulations to obtain two extreme triangulations of the double chain. Hence, we know that their flip distance is $\Omega(n^2)$ due to Theorem 2.1. Suppose there exists a flip sequence of $o(n^2)$ flips between these two pointed pseudo-triangulations in which no pseudo-triangle has more than k vertices, for some fixed k. Then in every flip, at most $2k-2$ vertices are involved. In a triangulation that contains the initial pointed pseudo-triangulation as a subgraph, we therefore only need a constant number of flips to obtain a triangulation that has the succeeding pointed pseudo-triangulation as a subgraph. We therefore would obtain a flip sequence of length $o(n^2)$ between two extreme triangulations, a contradiction.

---

[4] This result has been obtained in joint work with Oswin Aichholzer.

2.5 Chapter Summary

In this chapter, we gave an introduction and preliminaries on the problem of finding short flip distances between triangulations of both point sets and simple polygons. In particular, we described properties of the double chain. These will be used in the following two chapters for our hardness results. As a by-product, Corollary 2.10 revealed an interesting lower bound on distances in the flip graph. Further, we showed that the quadratic lower bound for the triangulation flip distance can be transcribed to a lower bound for pointed pseudo-triangulations whose face degree is bounded by a constant.

Chapter 3

Flipping in Triangulations of Point Sets

In the previous chapter, we prepared the results on double chains which we will use in the current chapter to examine the problem of determining the flip distance between two triangulations of a point set. We give a reduction from the MINIMUM VERTEX COVER problem to the flip distance problem, showing that it is not only NP-complete (a result simultaneously and independently obtained by Lubiw and Pathak [115], using a reduction from a planar variant of the MINIMUM VERTEX COVER problem; see Section 3.2.4), but also hard to approximate within a factor of 1.36. The content of this chapter has already been published in [133].

3.1 Introduction

While we know due to Lawson [108] that there is always such a flip sequence, For the formal use throughout this chapter, we define the problem as follows.

Problem 1 (POINT SET TRIANGULATION FLIP DISTANCE)**.** *Given two triangulations* T_1 *and* T_2 *of a point set* S *of size* n*, choose a flip sequence* σ *from* T_1 *to* T_2 *such that* $|\sigma|$ *is minimized.*

APX-hardness (see, e.g., [23, p. 261]) is defined for *NP optimization problems*, which are the problems fulfilling the following three properties (see [23, p. 27]).

- The set of instances is recognizable in polynomial time.
- There exists a polynomial q such that, given an instance X, for any Y that is a feasible solution for X, $|Y| \leq q(|X|)$ and, besides, for any Y such that $|Y| \leq q(|X|)$, it is decidable in polynomial time whether Y is a feasible solution for X.
- The measure function (i.e., a function providing a positive integer that quantifies the quality of the solution [23, pp. 22–23]) is computable in polynomial time.

Clearly, the problem instances for POINT SET TRIANGULATION FLIP DISTANCE are trivially recognizable in polynomial time for any reasonable input representation. In the most natural formulation, a *feasible solution* for an instance of the flip distance problem is a flip sequence between T_1 and T_2, over which we optimize, and the measure function gives the length of these sequences. However, there is a lower bound of $\Omega(2.4317^n)$ on the number of triangulations of any point set of size n [146]. Still, all "interesting" solutions will have a length in $O(n^2)$ and we can therefore virtually add this additional length constraint to our problem definition to formally fulfill the requirements of the class of NP optimization problems (a direct analysis of Lawson's algorithm gives an upper bound of $2\sum_{i=4}^{n}(i-3) = n^2 - 5n + 6$ on the flip graph diameter). The *performance ratio* of an approximate solution to a minimization problem is the value of the measure function applied to the approximate solution (in our case $|\sigma|$) divided by the optimal value (the flip distance).

The complexity class APX consists of the NP optimization problems for which, for some constant r, there is a polynomial-time algorithm that guarantees to find a solution with performance ratio at most r [23, p. 93]. APX-hardness of the problem implies that no polynomial-time approximation scheme (PTAS) exists (i.e., there is no polynomial-time algorithm that approximates the flip distance by a ratio of at most $1+\epsilon$ for every constant $\epsilon > 0$), unless P = NP. However, as we are not aware of any constant-factor approximations of the problem, we do not know whether it is also APX-complete.

To show APX-hardness, we first use an AP-reduction from the well-known MINIMUM VERTEX COVER problem. A formal definition of this kind of reduction will be given during the analysis of the construction (Definition 3.1). Until we have to analyze the details of the construction, the reader not familiar with that concept may use the usual NP-hardness reductions as a mental model.

After having shown APX-hardness, we show how the parameters of the reduction can be refined to prove a stronger result. POINT SET TRIANGULATION FLIP DISTANCE is at least as hard to approximate as MINIMUM VERTEX COVER. This means it cannot be approximated by a factor less than $1.36-\varepsilon$ for any constant $\varepsilon > 0$ in polynomial time, unless P = NP. This bound is even $2-\varepsilon$ if the Unique Games Conjecture is true. (The *Unique Games Conjecture* was originally formulated in terms of so-called 2-prover-1-round games by Khot [101]. Khot and Regev [102] give an equivalent, more combinatorial description as a labeling problem.)

In Section 3.2, we present the gadgets used in our reduction and analyze the construction. In that section, we only present a rough overview on how the points of the set are placed, a more detailed description of how to calculate their coordinates is given in Section 3.3.

3.2 The Reduction

Now we have gathered enough knowledge about double chains as sub-configurations in order to use them as the main building blocks in a reduction. We reduce from MINIMUM VERTEX COVER, which is known to be APX-complete [129].[1]

---

[1] A previous version of our proof used a reduction from MINIMUM VERTEX COVER on 3-regular graphs, which is also known to be APX-complete [18]. However, as pointed out by an anonymous referee for the

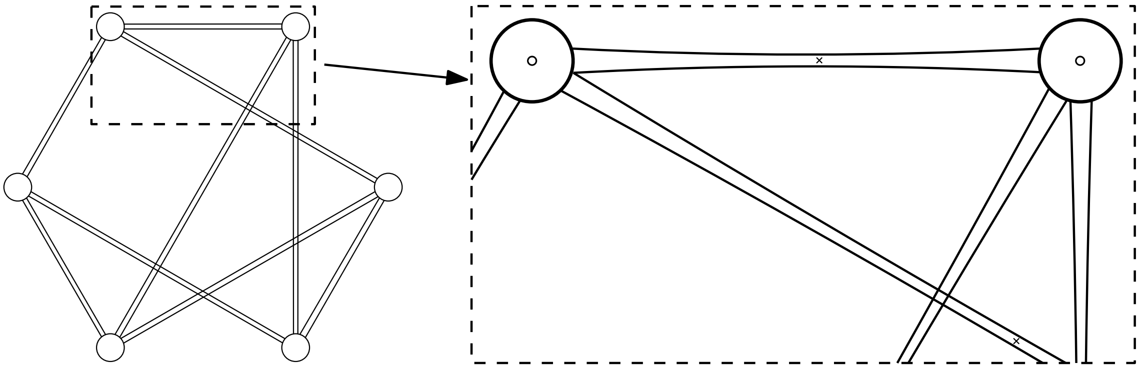


Figure 3.1: An embedding of a graph with the (almost straight) circular arcs at each edge ending at a fixed distance around each vertex.

Problem 2 (Minimum Vertex Cover)**.** *Given a simple graph $G = (V, E)$, choose a set $C \subset V$ such that every edge in E has at least one vertex in C and such that $|C|$ is minimized.*

We follow the common approach of embedding the graph G and transforming its elements to geometric gadgets. The gadgets consist of points together with the corresponding edges in the source triangulation T_1 and in the target triangulation T_2. We give the overall idea of how to embed the gadgets; for a detailed description on how to exactly place the points with rational coordinates having a representation bounded by a polynomial in the input size using polynomial time see Section 3.3.

3.2.1 Gadgets

Given a graph $G = (V, E)$ for which we have to solve the Minimum Vertex Cover problem, with $n = |V|$ and $m = |E|$, we place the elements of V as the vertices of a convex n-gon and draw the straight-line edges between them (where the edges will not be part of the final construction). Hence, we can consider G being a geometric graph in the remainder of this section. For each edge e mark a point $c_e \in e$ that is not on a crossing. Let $\vec{t}$ be a vector perpendicular to e of sufficiently small length (which will be specified in Section 3.3). Make two copies of e and translate them by $\vec{t}$ and $-\vec{t}$, respectively, to obtain the *tunnel* of the edge, i.e., the quadrilateral defined by the two copies of e. Then slightly "bend" the copies towards the (geometric) midpoint of e to obtain two circular arcs A_e and A'_e. The endpoints of the original edge e have to see any point on A_e and A'_e. See Figure 3.1.

3.2.1.1 Edge Cores

Instances of the double chain are the main ingredient in our reduction. They are contained in the gadgets representing the edges of G. See Figure 3.2 for an illustration of the construction. Let e be a straight-line edge of G, drawn between the points v and v'. In a close

---

version in [133], reducing from the general version gives a better lower bound on the performance ratio without any substantial changes to the reduction.

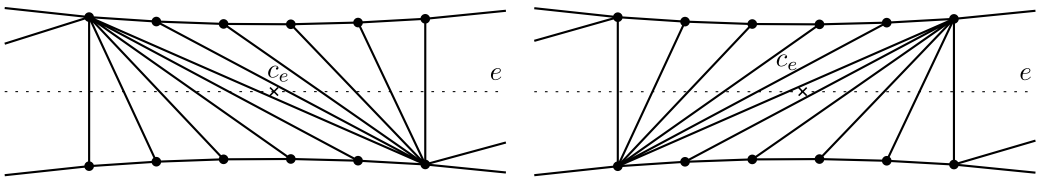


Figure 3.2: The double chain at the center of an edge with the source and the target triangulation.

neighborhood of c_e, place a double chain D_e, the *edge core*, of $2d$ points (we will fix the value of d later) along A_e and A'_e such that the two chains are separated by the supporting line of e. Note that the endpoints v and v' of e are the only points that are not outside the hourglass of D_e, and they are also in the flip-kernel of D_e (remember that A_e and A'_e can be chosen sufficiently flat). The edge cores are the only gadgets that have different edges in the source and in the target triangulation. Draw the edges that define the polygon P_{D_e} in both T_1 and T_2. Then triangulate the interior of P_{D_e} with one extreme triangulation of D_e in T_1 and with the other extreme triangulation in T_2. We refer to the process of flipping edges that are incident to an edge core as *transforming an edge core*.

3.2.1.2 Crossings

If two straight-line edges e and f of G cross, also their corresponding circular arcs cross. The four circular arcs define a region bounded by four pieces of the original arcs. Place one point at each of the four crossings of the arcs (we will actually place the points not exactly on the crossings, but close, see Section 3.3). In both source and target triangulation draw the edges connecting two points that are consecutive on any circular arc, which results in a crossing being represented by a convex quadrilateral, to which we add an arbitrary diagonal. Note that the crossing gadgets do not overlap with the edge core gadgets, as the edge cores are placed in the neighborhood of c_e, which was chosen not to be at a crossing.

3.2.1.3 Wirings

Wirings are gadgets that represent the elements of V. See Figure 3.3 for an illustration. Consider any vertex v of G and a small circle C with v in the embedding as its center. This *point* v is part of the triangulated point set. Place points on the crossings of C with the arcs of the edges incident to v in G. Since the graph is embedded on a convex n-gon and due to the small length of the vector $\vec{t}$, these points occupy strictly less than half of C. This allows us to place two chains L and R, each of $w-1$ points (the value of w is to be defined later) on C in a way that any line between one point of R and one point of L separates v from the remaining construction. In both the source and target triangulation draw the edges between consecutive points on C. Draw a zig-zag path through the points of L, R, and the first and last point where C crosses the arcs of the edges (giving $2w$ points in total). We call these edges the *zig-zag edges* of the wiring. Connect v to the first point of L and to the first point of R. The remaining part may be triangulated arbitrarily.

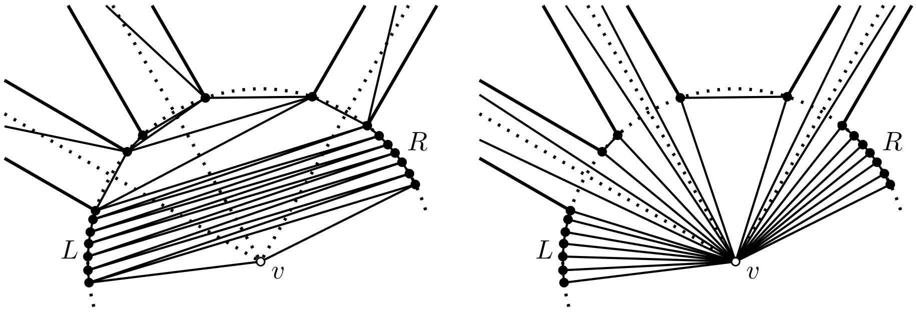


Figure 3.3: Left: A wiring with its initial and final triangulation (solid). Right: A triangulation that allows to quickly perform a transformation of the edge cores. The parts of the auxiliary construction shown in Figure 3.1 are dotted.

The remaining faces in the two plane graphs we obtained so far are triangulated arbitrarily, however in a way that the resulting triangulations T_1 and T_2 have the same edges except at the edge cores.

3.2.2 Analysis

The basic idea of the construction is that a flipping algorithm that gives the shortest flip distance or a good approximation of it has to choose which wirings to flip (requiring $4w-2$ flips each for flipping the zig-zag edges of a wiring away and back again) in order that the triangulation of an edge core can be transformed using the point in its flip-kernel at the chosen wiring. Also, the at most $4x+2$ edges between and at the crossings need to be flipped away. We will fix the values of w and d to force this behavior of any flipping algorithm that uses fewer flips than a trivial upper bound. Every edge of G will be covered; using a vertex of G for covering corresponds to flipping the zig-zag edges in the corresponding wiring.

Let v and v' be any two adjacent vertices in G. The exact number of edges in T_1 or T_2 intersected by the segment vv' in the drawing may differ with the choice of v and v' because (i) the number of crossings of each edge of G may differ, and (ii) the triangulation of the wiring gadget at the region where the edge gadgets enter it is not completely symmetric. Let x be the maximum number of crossings of a single edge in G. For every wiring, the number of edges that are intersected by the segment vv' in addition to the zig-zag edges is at most $2n-3$ (the remaining part is a $2n$-gon, see Figure 3.3). We denote the sum of these numbers over all wirings by τ; we have $\tau \in O(n^2)$.

The following lemma shows how to deduce a flip sequence in our construction from a vertex cover of size k. Note that we do not claim that this is the optimum if k is optimal.

Lemma 3.1. *If there exists a vertex cover of size k in G, then there exists a flip sequence between T_1 and T_2 of length at most*

$$\delta_k = 2(k(2w-1) + m(4x+2d) + \tau) \ .$$

Proof. Let C be a vertex cover of G with $k = |C|$. Let $v \in C$ be a vertex used to cover an edge. We use v to transform the edge cores of the adjacent edges in G (if they have not already been transformed). We need to flip all zig-zag edges in the wiring to v, which takes $2w-1$ flips. Then we need at most $2n-3$ flips (counted by τ) for the remaining wiring edges, as well as two further flips for the edges before the first crossing and two flips for the first crossing itself. All in all, with this method we need up to $4x+2$ flips for the crossing gadgets to make the first edge of the edge core visible to v. Then, we need $2d-2$ flips to make the edges incident to v (see Figure 2.7 (d)). Flipping in the desired way we need at most δ_k flips. □

On the other hand, a flip sequence should define a vertex cover. For the following lemma, we fix

$$w > \frac{c(m(4x+2d)+\tau)+1}{2}$$

for any constant $c > 1$; further, we choose d such that $(d-1)^2 > \delta_n = 2(n(2w-1) + m(4x+2d)+\tau)$ (note that since the term to the right is linear in d, such a value of d clearly exists and is polynomial in the problem size).

Lemma 3.2. *If there exists a flip sequence between T_1 and T_2 of length δ, then there exists a vertex cover of size at most*

$$k = \left\lfloor \frac{\delta}{4w-2} \right\rfloor \quad . \tag{3.1}$$

In particular, for the flip distance δ_{opt} between T_1 and T_2 and a minimum vertex cover of size k_{opt}, we have

$$k_{\text{opt}} = \frac{\delta_{\text{opt}} - R}{4w-2} \tag{3.2}$$

for some positive $R < \frac{4w-2}{c}$.

Proof. We argue that the choice of d forces an effective algorithm to flip the zig-zag edges of wirings (which corresponds to covering vertices), and that the choice of w allows to transform the number of flips to the size of the corresponding vertex cover.

If $\delta \geq (d-1)^2$, then the choice of d implies that $k \geq n$ in (3.1), which trivially implies that the lemma is true in that case. We therefore assume that $\delta < (d-1)^2$. If, for any edge core, we do not use the corresponding central points v or v' of a wiring, we need at least $(d-1)^2$ flips due to Theorem 2.3. Now suppose that we want to transform an edge core D using a point v. Then we need to flip all zig-zag edges in the wiring to v (as in the proof of Lemma 3.1), taking $2w-1$ flips. Note that this is optimal since only one of the zig-zag edges can be removed with each flip. The values of d and w have been chosen in a way that flipping the edges of all wirings, crossings, and edge cores to the corresponding central point and back, as described, uses fewer flips than transforming one edge core, due to the bound of Lemma 3.1. For any algorithm, this means that flipping all edges at wirings and crossings twice and transforming the edge cores with a point at the wiring is cheaper than transforming one edge core without a point at a wiring. Due to Theorem 2.3 we know that we need a point at a wiring for each edge core to be transformed in fewer than $(d-1)^2$ flips, as, for each edge core, the points at the two wirings are the only ones inside the hourglass of the edge core. Therefore, we know that the (optimal) flip distance δ_{opt} is given by

$$\delta_{\text{opt}} = k_{\text{opt}}(4w-2) + R \text{ for some } R > 0 \quad . \tag{3.3}$$

Equation (3.3) shows how to deduce k_{opt} from δ_{opt}: Lemma 3.1 gives us an upper bound on the flip distance, and hence $R \leq 2(m(4x + 2d) + \tau)$. Note that if $R < 4w - 2$, the size of the minimum vertex cover can be calculated from the flip distance by

$$k_{\mathrm{opt}} = \left\lfloor \frac{\delta_{\mathrm{opt}}}{4w - 2} \right\rfloor \ .$$

We actually require $cR < 4w-2$, for a given constant $c > 1$ (which is used for the reasoning about approximation ratios later in this section). This requirement can be fulfilled by choosing w under consideration of the bound $R \leq 2(m(4x + 2d) + \tau)$, i.e., such that $2(m(4x + 2d) + \tau) < (4w - 2)/c$. Thus, we have chosen w such that, in an optimal flip sequence, flipping the zig-zag edges of one wiring needs more flips than c times the number of all flips of edges not in a wiring.

No matter how well an algorithm performs, it has to flip the zig-zag edges of at least k_{opt} wirings when using less than $(d - 1)^2$ flips, and Lemma 3.1 tells us that c times the number of flips of the edges not in a wiring are in total fewer than the number of the zig-zag edges flipped for one wiring when the algorithm is optimal. ☐

To show APX-hardness of the flip distance problem, we show that we have an AP-reduction [23, pp. 256–261] from Minimum Vertex Cover using the previous lemmata. Let k_{opt} be the size of a minimum vertex cover for G and δ_{opt} be the flip distance between T_1 and T_2. See [23, pp. 257–258] for the following definition (note that r is a bound on the performance ratio of the approximate solution of the problem we reduce to, and that α is a factor in the bound for the performance ratio of the solution to the initial problem).

Definition 3.1 (AP-reduction). *Let P_1 and P_2 be two NP optimization problems. P_1 is AP-reducible to P_2 if two functions f and g and a constant $\alpha \geq 1$ exist such that:*

1. *For any instance X of P_1 and any rational $r > 1$, $f(X, r)$ is an instance of P_2.*
2. *For any instance X of P_1 and any rational $r > 1$, if there is a feasible solution of X, then there is a feasible solution of $f(X, r)$.*
3. *For any instance X of P_1 and any rational $r > 1$, and for any Y that is a feasible solution of $f(X, r)$, $g(X, Y, r)$ is a feasible solution of X.*
4. *f and g are computable by two algorithms whose running time is polynomial for any fixed rational r.*
5. *For any instance X of P_1 and any rational $r > 1$, and any feasible solution Y for $f(X, r)$, a performance ratio of at most r for Y implies a performance ratio of at most $1 + \alpha(r - 1)$ for $g(X, Y, r)$.*

In our case, f corresponds to the construction of the point set and the two triangulations. Requirements 1 and 2 follow from our construction. A vertex cover can be extracted from a flip sequence Y from the zig-zag edges flipped at the wirings; this corresponds to g, and requirement 3 is therefore fulfilled. Both f and g are polynomial-time algorithms, as demanded by requirement 4 (the parameter r is actually not used by either of these two algorithms, but will be used in the analysis).

Intuitively, Lemmata 3.1 and 3.2 give evidence that the reduction described so far fulfills also requirement 5 of Definition 3.1. However, because of the remainder term R, the performance ratio of an approximation of the flip distance does not directly give the performance ratio of the resulting approximate vertex cover; we have to show that R was chosen small enough and therefore the performance ratio of the approximate vertex cover stays within the bounds required by Definition 3.1. Let δ be an approximate solution for the flip distance such that $\delta \leq \delta_{\text{opt}} r$. Further, let R' be the remainder produced by the floor function in (3.1) of Lemma 3.2, that is, in the expression $k = \left\lfloor \frac{\delta}{4w-2} \right\rfloor$. By Lemma 3.2, we get

$$k \leq \frac{\delta - R'}{4w-2} \leq \frac{\delta_{\text{opt}} r - R'}{4w-2} \quad .$$

Let R be the remainder term for the optimal solution δ_{opt} as in (3.2) of Lemma 3.2, that is, in the expression $k_{\text{opt}} = \frac{\delta_{\text{opt}} - R}{4w-2}$. Then introducing the term $rR - rR$ in the numerator of the previous upper bound for k yields

$$k \leq r\frac{\delta_{\text{opt}} - R}{4w-2} + \frac{rR - R'}{4w-2} = rk_{\text{opt}} + \frac{rR - R'}{4w-2} \leq rk_{\text{opt}} + \frac{rR}{4w-2} < rk_{\text{opt}} + \frac{r}{c} \quad , \tag{3.4}$$

where the equality and the last inequality are due to Lemma 3.2. Let $\alpha = 4$ and $c = 2$. Suppose first that $r - 1 = \epsilon \geq \frac{1}{2k_{\text{opt}}+1}$. Then

$$rk_{\text{opt}} + \frac{r}{2} = k_{\text{opt}} + \epsilon k_{\text{opt}} + \frac{1}{2} + \frac{\epsilon}{2} = k_{\text{opt}} + \alpha\epsilon k_{\text{opt}} + \frac{1}{2} - \epsilon\left(3k_{\text{opt}} - \frac{1}{2}\right) \quad . \tag{3.5}$$

To get rid of the last part we use

$$\epsilon\left(3k_{\text{opt}} - \frac{1}{2}\right) \geq \frac{3k_{\text{opt}} - 1/2}{2k_{\text{opt}} + 1} > \frac{1}{2} \quad ,$$

which, by (3.4) and (3.5), implies

$$k \leq k_{\text{opt}} + \alpha\epsilon k_{\text{opt}} \quad .$$

On the other hand, suppose that $r - 1 = \epsilon < \frac{1}{2k_{\text{opt}}+1}$. Then from (3.4), we get

$$\begin{aligned} k < rk_{\text{opt}} + \frac{r}{2} &= k_{\text{opt}} + \epsilon k_{\text{opt}} + \frac{1}{2} + \frac{\epsilon}{2} = k_{\text{opt}} + \epsilon\left(k_{\text{opt}} + \frac{1}{2}\right) + \frac{1}{2} \\ &< k_{\text{opt}} + \frac{k_{\text{opt}} + 1/2}{2k_{\text{opt}} + 1} + \frac{1}{2} = k_{\text{opt}} + 1 \quad . \end{aligned}$$

Since the solutions to vertex cover are integers, this implies that $k = k_{\text{opt}}$ and therefore $k \leq k_{\text{opt}} + \alpha\epsilon k_{\text{opt}}$ holds. Hence, in both cases $k/k_{\text{opt}} \leq 1 + \alpha(r-1)$ and our reduction fulfills all properties of an AP-reduction from Minimum Vertex Cover.

Theorem 3.3. *The problem of determining a shortest flip sequence between two triangulations of a point set is APX-hard.*

3.2.3 An Improved Bound on the Performance Ratio

The previous reduction did not use the performance ratio bound r. As pointed out by an anonymous referee of the version in [133], a different choice of w actually allows to prove a better lower bound on the tractable performance ratios. This reduction selects c (the constant used in Lemma 3.2) according to r (recall that r is considered a constant). Hence, this is an example of a reduction that actually uses the bound r as a parameter. It is known that approximating Minimum Vertex Cover by any constant factor less than $10\sqrt{5} - 21 \approx 1.36$ is NP-hard [53], and, if the Unique Games Conjecture is true, even obtaining a performance ratio within any constant less than 2 is NP-hard [102]. However, there exist approximation algorithms achieving a ratio of $2 - o(1)$ [94, 98].

Let b be the bound for the performance ratio that a polynomial-time algorithm can guarantee for Minimum Vertex Cover (note that b is between 1.36 and 2, unless P = NP). Suppose we can approximate the flip distance by a performance ratio less than $b - \varepsilon$ for some constant ε. Due to (3.4), we can guarantee a performance ratio of at most $(b-\varepsilon)+\frac{b-\varepsilon}{k_{\text{opt}}c}$ for Minimum Vertex Cover. Hence, if $\varepsilon > \frac{b-\varepsilon}{k_{\text{opt}}c}+\varepsilon'$, then the performance ratio bound for Minimum Vertex Cover is better than $b-\varepsilon'$. This is fulfilled for $c > \frac{b-\varepsilon}{\varepsilon-\varepsilon'}$. In particular, this requires $\varepsilon = \kappa\varepsilon'$ for a constant $\kappa > 1$. Note, however, that κ cannot be 1. The reason for this is that, in (3.4), R' can be smaller than R. For example, there may exist a 2-approximation for Minimum Vertex Cover for which the corresponding flip sequence is less than twice the optimum. Still, we obtain the following result.

Theorem 3.4. *For any given constant $\varepsilon > 0$, it is NP-hard to approximate the flip distance between two triangulations by a factor less than $10\sqrt{5} - 21 - \varepsilon$, and, if the Unique Games Conjecture is true, by a factor less than $2 - \varepsilon$.*

3.2.4 A Comparison to Lubiw and Pathak's Reduction

As already mentioned, Lubiw and Pathak [115] independently gave an NP-hardness reduction for the decision version of the flip distance problem.[2] We sketch their reduction to point out the similarities and differences between their approach and the one presented herein.

Their reduction is from the vertex cover problem on 3-connected cubic (i.e., 3-regular) planar graphs. Let G be the graph of such a problem instance. G can be embedded in the plane such that the resulting drawing is a plane geometric graph in which every bounded face is strictly convex [39] (where strictly convex means that the angle at all vertices inside that face are strictly less than 180°). The vertices at the unbounded face are replaced by three vertices, one of degree 3 and two of degree 2, in a way that all angles incident to vertices of degree 3 are less than 180°. The reduction uses double chains with seven points on each chain, which are called *channels*. Similar to the edge cores in our reduction, they are placed on the edges of the initial graph, and triangulated with different extreme triangulations (called "left-inclined" and "right-inclined" triangulations) in the source and the target triangulation, in the same way as in the proof by Hurtado

---

[2] The proof of Lemma 2.2 in Chapter 2 is, as indicated, a generalization of a proof they give, but this lemma is used in Chapter 4 only. The result presented in Chapter 4 (already published in [16]), were obtained later than the main ideas of the result presented in this chapter.

et al. [95]. For their analysis of the construction, they also have to show a lower bound of 24 flips for flipping from one extreme triangulation to the other using a point in the flip kernel (called the "narrow mouth"); see also Lemma 2.2. The vertex gadgets consist of polygonal regions, triangulated in the same way in the source and target triangulation. For vertices of degree 3, the vertex gadget contains three different points such that each is in the kernel region of one channel; Further, no point of the vertex gadget is inside the hourglass (called the "wide mouth") of the channel. A vertex of degree 2 is represented by a simpler variation of the gadget having the same properties. For each vertex gadget, there is a special edge, called the *lock*. If the triangulation of a channel has to be transformed from one extreme triangulation to the other in less than 36 flips, one of the incident locks has to be flipped. (Hence, the lock is a single edge fulfilling a purpose similar to the zig-zag path in our wirings.) Selecting a vertex for the vertex cover then corresponds to flipping the lock of the corresponding vertex gadget. To assure that the edges of the gadgets are flipped only in the intended way, the edges at the boundary of the resulting structure are "repeated" n^2 times (which requires placing additional points close to the ones on the boundary of the gadgets). It is argued that it would require too many flips to "dismantle" this construction, and therefore only the edges of the gadgets are flipped in an optimal solution. The details on these additional edges are deferred to the full version of the paper (which was not available at the time of writing this thesis).

Both reductions use the properties of the double chain of [95, 157], placing only two points in the kernel region of a double chain and no further points in its hourglass. Selecting vertices for the vertex cover corresponds to flipping certain edges. The main difference between the reductions is that our reduction does not require the initial graph to be planar. The analysis of Lubiw and Pathak's construction requires an exact counting of the flips, and therefore also a tight lower bound, which they provide. In our construction, we only relied on the lower bound given by the zig-zag path in the wiring gadgets. This, however, requires a construction of the gadgets that heavily depends on the size of the initial graph (e.g., for the choice of the value of w). Still, this coarser counting of the necessary flips easily allowed for handling the crossings in the initial graph using the crossing gadgets. There is no direct way to use the reduction to show APX-hardness of the problem when reducing only from the planar variant of MINIMUM VERTEX COVER, since, as already mentioned, there exists a PTAS for these instances (see Section 2.1 for the corresponding references). Another difference between the reductions is that the extension with the n^2 additional edges is not required in our proof. The reasons for this are twofold. First, our detailed analysis of the double chain, presented in Chapter 2, shows that points outside the hourglass can be disregarded for the quadratic lower bound on the flip distance. Second, we do not have to argue about a lower bound for the case when a point inside the flip kernel is used. (We use such a lower bound in the next chapter. There, we do not have to "protect" the construction by additional edges since this is fulfilled by the boundary of the polygon.) Their line of arguments implicitly relies on a concept equivalent to local triangulations, which, in their case, is almost trivial due to the placement of the additional edges around the construction, and is left to the reader. (Without further considerations it would be conceivable that a vertex of such an additional edge could somehow be used to shorten the flip distance. We know that this is not the case due to Theorem 2.3.)

3.3 On the Coordinate Representation

Section 3.2 already contained a description of the gadgets we used in our reduction. However, the validity of gadget-based reductions when proving NP- or APX-hardness for problems on point sets requires that the coordinates of the points used can be calculated in polynomial time.

The reader may have noticed that our high-level construction involves points placed at the crossing of circular arcs, which, in general, leads to irrational coordinates, even if the circular arcs are defined by rational points. We will give a construction that slightly varies from the one described that uses only rational coordinates, with both the numerator and denominator bounded by a polynomial in the input size.

One way to strengthen the result is to show that the problem remains APX-hard for triangulations of point sets in general position. The gadgets in our reduction do not make use of collinear points. However, we did not explicitly mention how to avoid three points on a line when describing the construction. In this appendix we give an explicit construction of the point set in general position, i.e., that no three points are collinear.

Note that the construction may not be "economical" in the sense that the construction may be possible with coordinates having a smaller binary representation. We will always prefer constructions that are easy to prove. We will place the points on and close to the unit disc (meaning that a coordinate will never exceed $1 + \epsilon$, for some small $\epsilon > 0$); therefore, we can specify the size of a coordinate in terms of the size of its denominator.

3.3.1 Placing the Points of the Convex Polygon

As a first step, we give a simple construction of a convex n-gon for placing the central points of the wiring gadgets with all vertex coordinates being rational and the denominators being in $O(n^{10})$. Further, we want to assure that no three diagonals cross in the same point. For doing so, we will first choose n^5 *candidate points* on the unit circle and then select n points out of them.

Rational points on the unit circle are known to be given by $\left(\frac{1-t^2}{t^2+1}, \frac{2t}{t^2+1}\right)$ with $t \in \mathbb{Q}$, see, e.g., [40]. We define a sequence K of candidate points with $t = i/n^5$ for the integers $1 \leq i \leq n^5$. (For consistency with later parts and ease of presentation therein we choose the candidate points from the upper-right quadrant in counterclockwise ascending order; hence, the value of t is between 0 and 1.) Now we select n points out of K such that there are no three diagonals that cross at a single point. We choose the first five points of our final set from the candidate points. Suppose we have chosen $j \geq 5$ points such that no three diagonals cross at a single point. We have $n^5 - j$ points in K to choose the next point from. Consider all $\binom{j}{5}$ combinations of five points among the already chosen ones. Each combination gives exactly five points on the unit circle that cannot be chosen, and none of these is among the j already chosen candidate points. Hence, we have $5\binom{j}{5} + j$ "forbidden" points (which may not all be among the candidate points). See Figure 3.4. We have, however, $n^5 \geq j^5 > 5\binom{j}{5} + j$ candidate points to choose from, and therefore we for sure can choose point number $(j + 1)$. We denote this set of points by P_V; the elements of P_V are the points representing the vertices of the input graph.

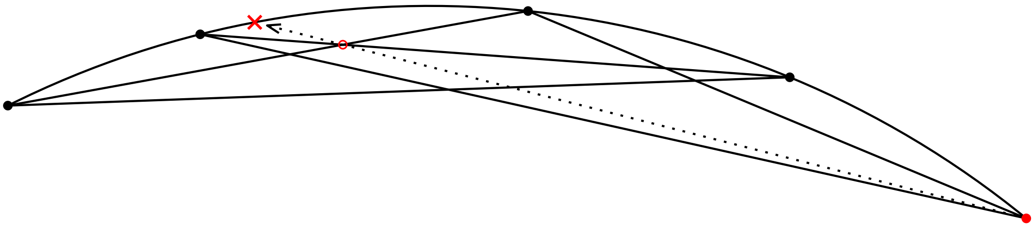


Figure 3.4: Five points on the unit circle; a point at the (red) cross would introduce a supporting line through a crossing of two other supporting lines and is therefore forbidden. The image is rotated for representational reasons, our method chooses all points from the upper-right quadrant.

Proposition 3.5. *A point set of n points in convex position with all coordinates rational having their denominators in $O(n^{10})$ and no three diagonals crossing at the same point can be found in polynomial time.*

Note that the facts that no three diagonals of the resulting n-gon cross and that the coordinates are bounded also give us a lower bound on the distance between intersection points and other diagonals, which we will use in the next part.

3.3.2 A Sufficiently Small Value

In this section, we will define four values $\delta_{\mathrm{e}}, \delta_{\mathrm{v}}, \delta_{\mathrm{n}}$, and δ_{r} that will give sufficiently small upper bounds on the construction of the gadgets. For any point p, let x_p and y_p denote its x- and y-coordinate, respectively.

For the definition of δ_{e}, find the minimum squared distance from each of the $\binom{n}{4}$ crossings of the diagonals of the n-gon to the diagonals not involved in the corresponding crossing. Let the actual distance be δ_{e}. Since the squared distance δ_{e}^2, is given by $(x_a - x_b)^2 + (y_a - y_b)^2$ between two points a and b, we can set $\delta'_{\mathrm{e}} = |x_a - x_b|$ to obtain a "small", rational and positive distance $\delta'_{\mathrm{e}} \leq \delta_{\mathrm{e}}$ (at least one of the horizontal or vertical distances is non-zero, in particular, up to here no two points can have the same x- or y-coordinate). When we construct the tunnels that are formed around an edge of the drawing of the input graph, we can choose, say, $\delta'_{\mathrm{e}}/3$ as an upper bound for the distance between the edge and the edges defining the tunnel. Then the intersection of any three tunnels is always empty. (Our actual tunnels will be even narrower.)

The vertex gadgets used "small" circles around each point in P_V. Let u, v, w be a triple of consecutive vertices on the n-gon defined by P_V. Let δ_{v}^2 denote the smallest squared distance between v and the line through u and w for every choice of the triple. As with the tunnels, we can choose a rational $\delta'_{\mathrm{v}} \leq \delta_{\mathrm{v}}$ by choosing only the horizontal or vertical distance between v and the closest point on the supporting line of u and w.

Again, let v be a vertex on the n-gon. Let ℓ_v be the line through v that is perpendicular to the line ov, where o is the origin. Consider the distances from u and w to ℓ_v. Let δ_{n}^2 be the smallest squared distance for all choices of v (and corresponding u and w), and choose a rational $\delta'_{\mathrm{n}} \leq \delta_{\mathrm{n}}$ as before. Further, let δ_{r} be the smallest horizontal or vertical distance between two points in P_V (which is non-zero by construction). See Figure 3.5. We define

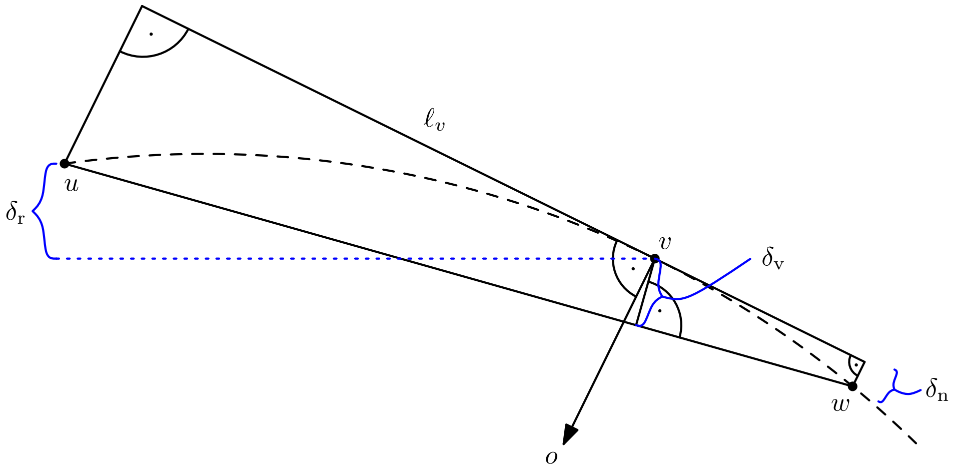


Figure 3.5: Construction to obtain bounds for δ.

$\delta = \min\{\delta'_{\mathrm{e}}/3, \delta'_{\mathrm{v}}, \delta'_{\mathrm{n}}, \delta_{\mathrm{r}}\}$. If we now choose the radius of the cycle centered at each vertex by $r_V = \delta/6$, then no two circles intersect (there is actually a distance of at least $4r_V$ between two circles), and each circle only intersects the edges of the input graph that are incident to the vertex it is centered at. Further, no circle intersects the convex hull of two other circles.

3.3.3 Tunnel Construction

For each edge e of the input graph connecting two vertices v and w, we now give the construction of the tunnels. Let C_v and C_w be the circles around v and w, respectively. The tunnel for the edge between v and w is given by two segments, each having one endpoint on C_v and one endpoint on C_w. We want to get rational points on C_v and C_w. Since these circles are not only defined by a rational center point, but also have a rational radius, the problem boils down to finding a rational point on the unit circle, or, equivalently, a (possibly irrational) angle α such that $\sin(\alpha)$ and $\cos(\alpha)$ are rational, within some interval given by quadratic irrationals. Sines with this property are called *rational sines*, and correspond with the parametrization of the unit circle that we already used before. Canny, Donald, and Ressler [40] give an algorithm for finding a rational sine for a parameter $t = p/q$ such that $|p/q - x| < \epsilon$, for given x and ϵ (we will use an extended method for non-rational radii later). Their algorithm gives a denominator q in $O(1/\epsilon)$, and the running time is polynomial in q. However, the input x is an approximation as well, and their goal is to get rational sines with small binary representation. Our angle intervals, however, are given by rational points and their relative position to the circle center. For finding a point within this interval, the Farey approximation as used by Canny et al. [40] for $t = p/q$ is sufficient and easy to apply for our setting, as we do not need an explicit approximation of the angle and the interval as input (this algorithm searches a point inside the interval in the fashion of binary search, computing the mediant $\frac{a+c}{b+d}$ of two rational values $\frac{a}{b}$ and $\frac{c}{d}$ in each step). We, however, need an upper bound on the denominator q derived from the points defining the angle.

Now we show how to use the results by Canny et al. [40] for our needs. Consider the unit circle and two points a and b. Let $\angle a$ and $\angle b$ be the polar angles of these points, and, without loss of generality, let $\angle a < \angle b$. We describe only the case where both angles are within $[0, \dots, \pi/2]$, the other cases are similar (and can easily be distinguished); in our setting we simply have to rotate the plane orthogonally. To approximate an angle between $\angle a$ and $\angle b$ using a rational number t, we reason about the (possibly irrational) values t_a and t_b. For t_a and $\angle a$ we define

$$\sin(\angle a) = \frac{2t_a}{t_a^2 + 1} \ ,$$

which, when choosing the appropriate root, gives

$$t_a = \frac{1}{\sin(\angle a)} - \sqrt{\frac{1}{\sin^2(\angle a)} - 1} \ .$$

The sine of $\angle a$ is given by $a_y / \sqrt{a_x^2 + a_y^2}$. The values of $\angle b$ and t_b are defined analogously. We therefore need to find a rational number t with $t_a \leq t \leq t_b$. The Cauchy bound (see [161]) for an algebraic number g being the root of a polynomial $\sum_{i=0}^{m} c_i x^i$ with rational coefficients c_i is given by

$$|g| \geq \frac{|c_0|}{|c_0| + \max\{|c_1|, \dots, |c_m|\}} \ .$$

The difference $|t_a - t_b|$ is therefore bounded from below by a rational that has a denominator polynomial in the problem size. Using Farey approximation, we can find a rational t whose denominator exceeds the denominator of the bound only by a polynomial factor.

Since we can choose rational points on the unit circle inside an interval (and therefore on instances of the unit circle that are translated and scaled by rational values), we now have the tools to choose the endpoints of the tunnels. For two vertices v and w, let these be called p_v and q_v (placed on C_v), as well as p_w and q_w (placed on C_w). Hence, a tunnel between v and w consists of the quadrilateral $p_v q_v q_w p_w$. In order to prevent collinear triples of points, we again select a set of candidate points on C_v and C_w and choose the four points among them. Note that this results in tunnels that may not be exactly rectangular, but this is irrelevant for our final construction. See Figure 3.6 for an accompanying illustration.

We place the points in the following way. Without loss of generality, suppose that $x_v < x_w$. To obtain the set of candidate points for p_v, consider the segment between w and the point $(x_w, y_w + r_V)$, where r_V is the radius of the circles, which we call the *upper spoke* of w. Let the *lower spoke* of w be defined analogously between v and the point $(x_w, y_w - r_V)$. Find the two parameters t_1 and t_2 for rational points p_{t_1} and p_{t_2} on C_v such that the line through v and p_{t_1} intersects the upper spoke of w at a point above $(x_w, y_w + 7r_V/8)$ and the line through v and p_{t_2} intersects the upper spoke between $(x_w, y_w + r_V/2)$ and $(x_w, y_w + 5r_V/8)$. We can now select our set K_v of candidate points from the interval $[t_1, t_2]$. The same can be done for two parameters t_3 and t_4, with the roles of v and w interchanged. We select a point $p_v \in K_v$ and a point $p_w \in K_w$ as the endpoints of one side of the tunnel gadget between v and w.

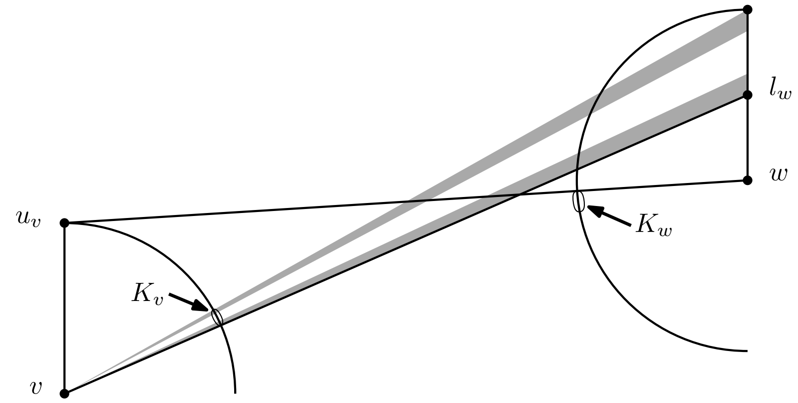


Figure 3.6: Construction for the tunnel endpoints: The two extremal candidate points for p_v are chosen inside the two gray wedges. Note that v and w in the drawing do not fulfill the required vertical distance since the drawing would get too small.

Let us now argue the correctness of this construction. Note that we do not need to require the sides to be parallel to the supporting line of v and w (we could do so by increasing the number of candidate points). The crucial property of the points we need is that $p_v v w p_w$ forms a convex quadrilateral and we therefore have to prove that p_v is always left of the directed line through v and p_w (and, analogously, that p_w is right of the directed line through w and p_v). Let $l_w = (x_w, y_w + r_V/2)$ and $u_v = (x_v, y_v + r_V)$. The diagonals vl_w and $u_v w$ of the trapezoid $u_v v w l_w$ intersect each other at a ratio of $(r_V/2)/r_V$, i.e., at two thirds of the interval $[x_v, x_w]$. Recall that the radius r_V was chosen in a way that the disc centers have a horizontal distance of at least $6r_V$. Hence, the segments intersect outside C_w; the topmost candidate point on C_w is below the line through v and the lowest candidate point on C_v, and vice versa. Note that since the candidate points on C_v are chosen inside the convex hull of C_w and v, no two tunnels from v can intersect.

It remains to find the correct number of candidate points. Suppose we already constructed all but one tunnel point. Since at every circle there are at most $2(n-1)$ tunnel points there are at most $\binom{n(2n-1)}{2}$ lines on which we are not allowed to place a point. Every line intersects the circle on which we place the last point at most twice. Hence, if we choose more than twice the number of points as we have lines, we can always choose a point such that the resulting point set is in general position.

3.3.4 Points in the Tunnels

For each tunnel, we construct two circular arcs, one for each segment defining the tunnel, on which we place the points of the edge core. The crucial property of such an arc is that for two wire centers v and w, these points are the only ones in the flip-kernel. Let q_v and q_w be the two endpoints of a tunnel edge, such that q_w is to the right of q_v and the interior of the tunnel is above the line $q_v q_w$. See Figure 3.7. The constructed arc will start at q_w and end at q_v. We consider four rays, namely the ones that leave q_v to the right in an angle of $0, (-\pi/4)$, and $\pi/4$ with the x-axis and the upward vertical ray at q_v. Let r be the one that opens the smallest positive angle α with $q_v q_w$. If the angle between q_v and q_w and $q_v w$

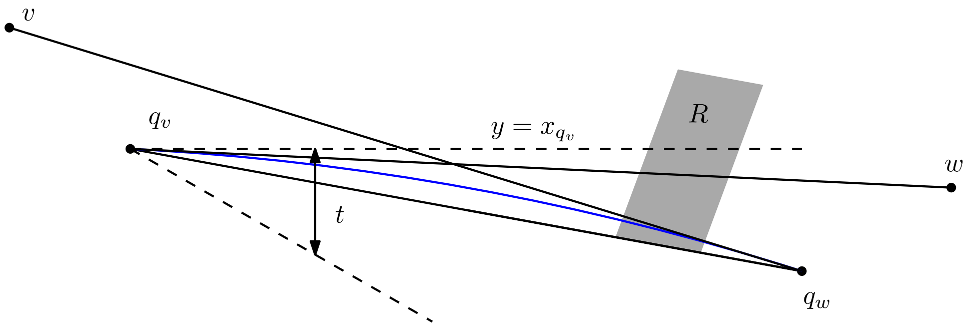


Figure 3.7: We want to choose rational points on the (blue) arc inside the gray region by Farey approximation on the slope t. Note that the gray region actually is, for presentational reasons, drawn too close to w.

is smaller than α, then let s be the ray through w starting at q_v; otherwise, let $s = r$. We perform the analogous operation (i.e., with the plane being mirrored horizontally) at q_w, obtaining a ray s'. Without loss of generality, let the angle between $q_v q_w$ and s be smaller than or equal to the one between $q_w q_v$ and s'. Construct the circle A that passes through both q_v and q_w such that A is tangent to the supporting line of s. The coordinates of the center of A are still rational. It is well-known that, when given any rational point p on A and a line ℓ with rational slope that intersects A at p and a second point p', the point p' is rational as well, see, e.g., [96, p. 5]. Hence, we need to appropriately choose lines through a point p. The crossings of the segments that define all the tunnels identify the region where the edge core should be placed. Let R be the region we have to place the points in (marked gray in Figure 3.7). By the choice of r, we constructed A in a way that we can mirror and rotate the plane orthogonally such that the intersection of A and R is within an angle of 0 and $\pi/4$ from q_v. This means that any line ℓ through q_v and this intersection will have a slope t between 0 and 1. This reasoning is similar to the one of Burnikel [38] to adapt the techniques of [40] for such rational circles (i.e., circles given by three rational points). As before, we can use, e.g., Farey approximation for the slope t of ℓ. At each iteration, we check whether the second intersection of ℓ with A is inside the quadrilateral R, and, if not, on which side it is. Since the denominators of the coordinates of the points defining A and R are polynomial, there is a polynomial lower bound on the difference between the (possibly non-rational) parameters for the two points where A enters and leaves R (as for the construction of the tunnel endpoints). Hence, after a polynomial number of steps, we have a rational slope for ℓ such that ℓ passes through A inside R; therefore, also this intersection point has rational coordinates and its denominator is polynomial in the problem size. To obtain a second such point, the process can be continued. Now we have two points in the intersection of A and R which define two slopes of lines through q_v. Any line through q_v with a slope in the interval between these two slopes gives a rational point on $A \cap R$. Hence, we can choose our candidate points by dividing that interval.

The remaining problem is the one of choosing the points for the crossing gadgets. Two arcs in crossing tunnels will, in general, cross at a point that does not have rational coordinates. The crucial property of the points of the crossing gadgets, however, is that they are outside the hourglasses of the edge cores (recall Definition 2.1) and that the edges

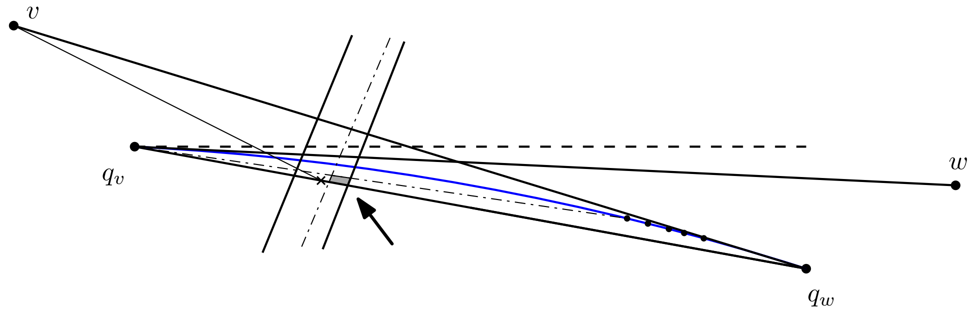


Figure 3.8: Construction of tunnel crossings. The drawing shows the lower part of a tunnel between v and w and a part of another tunnel (indicated by the near-vertical strokes).

between them can "quickly" be flipped to the center of the corresponding wiring gadget. Placing the points for the crossing gadgets at the crossings of the segments that define the tunnels would satisfy these constraints, but would lead to collinear triples. So we have to slightly perturb each point p to obtain a point p' without loosing these properties. See Figure 3.8. Between every consecutive pair of crossing points on a tunnel segment $q_v q_w$ we can choose the rational midpoint. If the perturbed point p' remains on the same side of the line through the wire center and the midpoint as p, the order around the wire center is maintained. Further, the perturbed points have to remain on the same sides of the lines that define the hourglasses of the edge cores involved. Together with the tunnel edges, these constraints give a convex region from which we can choose our perturbed point. We may again place a circular arc inside this region (marked gray in Figure 3.8) on which we select a sufficiently large number of candidate points, analogously to the construction of the other gadgets.

3.3.5 Points for the Wiring

Finally, we place the points at the wiring gadgets that allow us to draw the wiring edges, see Figure 3.9. The circles for the wiring gadgets are scaled versions of the unit circle. For a vertex v, let ℓ_v be the line through v that is perpendicular to the supporting line of the origin o and v. Since the coordinates of v are rational sines, the intersection points of ℓ_v with the circle C_v are rational as well. Due to the choice of δ'_n, all points on C_v that define tunnels are on the same side of ℓ_v as o. We are given two intervals, each between two rational sines, i.e., between the "extremal" tunnel endpoints on C_v and the intersection points of ℓ_v with C_v. Therefore, we can choose a sufficient number of rational candidate points on C_v to choose the points for the wiring from.

3.3.6 Concluding Remarks on the Embedding

The crucial part throughout the whole embedding procedure is that each (intermediate) point that is not a candidate point is constructed using only a constant number of other points. The candidate points were constructed with polynomial parameters. Hence, all denominators are polynomial in the input size. In particular, note that even though some intervals were defined by points with algebraic coordinates, a lower bound on the interval

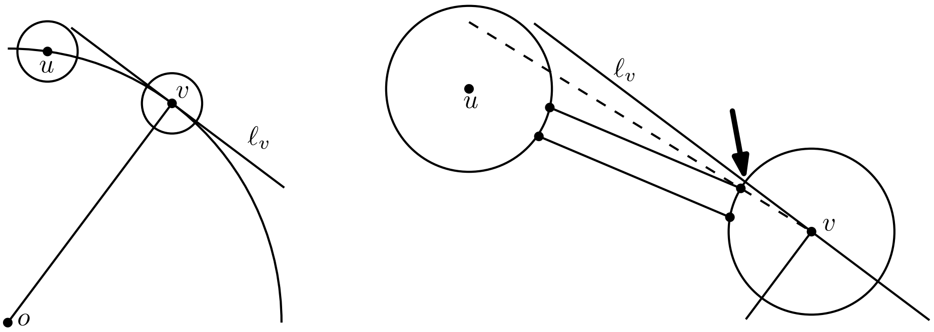


Figure 3.9: Construction of the wiring points. The small gap (indicated by the arrow) on the circle C_v of v between the intersection point with ℓ_v and the neighboring tunnel endpoint can be used for the candidate points.

can be given in terms of the other, rational coordinates that were used in the construction. This allowed us to find rational points with polynomial denominators within these intervals.

3.4 Chapter Summary

In this chapter, we showed that it is APX-hard to minimize the number of flips to transform two triangulations T_1 and T_2 of a point set S into each other, using properties of the double chain presented in Chapter 2. Further, we gave a detailed description on how the points for the construction can actually be embedded. The techniques used can be applied similarly to the exact constructions of other hardness reductions for problems on point sets, as those presented in Chapter 4 and Chapter 8.

We are not aware of any constant-factor approximation of the flip distance. For the upper bound given by Hanke et al. [89], it is easy to construct examples (like the one in Figure 2.7) where the bound is quadratic while the flip distance is linear. From an algorithmic point of view, proving the existence or non-existence of a constant-factor approximation algorithm is consequently a next step. Another next step, of course, is to simplify the problem setting. As we will argue in the next chapter, problems on triangulations are often easier for simple polygons than for point sets. Still, we obtain an NP-completeness result for the flip distance problem for triangulations of simple polygons in the next chapter.

Chapter 4

Flipping in Triangulations of Simple Polygons

While in the previous chapter we showed that the flip distance problem for triangulations of point sets is hard to approximate, we consider the flip distance problem for triangulations of simple polygons in this chapter. We show that the problem is NP-complete, using again the results on double chains from Chapter 2. However, this time, our reduction does not allow for a similar APX-hardness result. The content of this chapter has already been presented in [16].

4.1 Introduction

Some triangulation-related problems can be solved faster for simple polygons than for point sets. One example would be the problem of constructing a minimum weight triangulation, which can be solved in $O(n^3)$ time using dynamic programming [74, 104], while it is NP-hard for general point sets [127]. Another example would be the problem of counting the number of triangulations; practically the same dynamic programming approach can be applied for simple polygons, but there is no known polynomial-time algorithms for point sets, but only heuristics (see [3, 20]). However, the flip distance problem is neither concerned with constructing a special triangulation, nor is it an actual counting problem. This difference is emphasized by the hardness result we show for the flip distance problem.

In contrast to Chapter 3, we aim for an NP-completeness proof and therefore formulate our problem as a decision problem.

Problem 3 (POLYGON TRIANGULATION FLIP DISTANCE). *Given two triangulations T_1 and T_2 of a simple polygon P, and an integer l, decide whether T_1 can be transformed into T_2 by at most l flips.*

The problem is obviously in NP. To show NP-hardness, we give a polynomial-time reduction from RECTILINEAR STEINER ARBORESCENCE. RECTILINEAR STEINER ARBORESCENCE was shown to be NP-hard by Shi and Su [147]. In Section 4.2, we describe the problem in detail, and in Section 4.3, we describe our reduction and prove that it is correct.

4.2 The Rectilinear Steiner Arborescence Problem

Let S be a set of N points in the plane whose coordinates are nonnegative integers. The points in S are called *sinks*. A *rectilinear tree* T is a connected acyclic collection of horizontal and vertical line segments that intersect only at their endpoints. The *length* of T is the total length of all segments in T (cf. [97, p. 205]). The tree T is a *rectilinear Steiner tree* for S if each sink in S appears as an endpoint of a segment in T. We call T a *rectilinear Steiner arborescence* (RSA) for S if (i) T is rooted at the origin; (ii) each leaf of T lies at a sink in S; and (iii) for each $s = (x, y) \in S$, the length of the path in T from the origin to s equals $x + y$, i.e., all edges in T point north or east, as seen from the origin [136]. In the *RSA problem*, we are given a set of sinks S and an integer k. The question is whether there is an RSA for S of length at most k. Shi and Su showed that the RSA problem is strongly NP-complete; in particular, it remains NP-complete if S is contained in an $n \times n$ grid, with n polynomially bounded in N, the number of points [147].[1]

We recall an important structural property of the RSA. Let A be an RSA for a set S of sinks. Let e be a vertical segment in A that does not contain a sink. Suppose there is a horizontal segment f incident to the upper endpoint a of e. Since A is an arborescence, a is the left endpoint of f. Suppose further that a is not the lower endpoint of another vertical edge. Take a copy e' of e and translate it to the right until e' hits a sink or another segment endpoint (this will certainly happen at the right endpoint of f); see Figure 4.1. The segments e and e' define a rectangle R. The upper and left side of R are completely covered by e and (a part of) f. Since a has only two incident segments, every sink-root path in A that goes through e or f contains these two sides of R, entering the boundary of R at the upper right corner d and leaving it at the lower left corner b. We reroute every such path at d to continue clockwise along the boundary of R until it meets A again (this certainly happens at b), and we delete e and the part of f on R. In the resulting tree we subsequently remove all unnecessary segments (this happens if there are no more root-sink paths through b) to obtain another RSA A' for S. Observe that A' is not longer than A. This operation is called *sliding e to the right*. If similar conditions apply to a horizontal edge, we can *slide it upwards*. The *Hanan grid* for a point set P is the set of all vertical and horizontal lines through the points in P. In essence, the following theorem can be proved constructively by repeated segment slides in a shortest RSA.

Theorem 4.1 (Rao et al. [136])**.** *Let S be a set of sinks. There is a minimum-length RSA A for S such that all segments of A are on the Hanan grid for $S \cup \{(0, 0)\}$.*

We use a restricted version of the RSA problem, called YRSA. An instance (S, k) of the YRSA problem differs from an instance for the RSA problem in that we require that no two sinks in S have the same y-coordinate.

Theorem 4.2. YRSA *is strongly* NP*-complete.*

Proof. Due to the Hanan grid property, the YRSA problem is in NP, as the RSA problem [147]. We show how to reduce RSA to YRSA. Let (S, k) be an instance for an

---

[1] Note that a polynomial-time algorithm was claimed [156] that later has been shown to be incorrect [136].

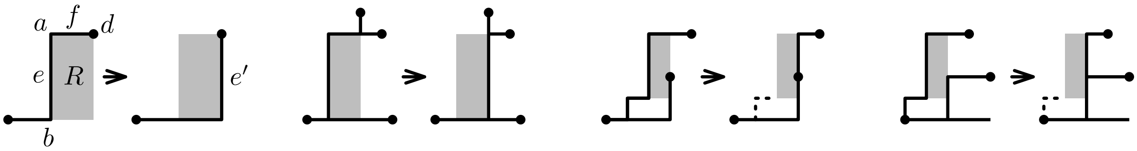


Figure 4.1: The slide operation. The dots depict sinks; the rectangle R is drawn gray. The dotted segments are deleted, since they do no longer lead to a sink.

RSA problem, and label the sinks as $S = \langle s_1, s_2, \ldots, s_N \rangle$ in an arbitrary fashion. For $i = 1, \ldots, N$, let (x_i, y_i) be the coordinates of s_i and define $s'_i := (x_i N^4, y_i N^4 + i)$. Set $S' := \{s'_1, s'_2, \ldots, s'_N\}$. Note that the y-coordinates of the sinks in S' are pairwise distinct.

Now let A be a rectilinear Steiner arborescence for S of length at most a. We can scale A by the factor N^4 and draw a vertical segment from each leaf to the corresponding sink in S'. It follows that there exists an RSA A' of length $a' \leq aN^4 + N^2$.

Suppose there exists an RSA B' of length at most b'. Due to Theorem 4.1, we can assume that B is on the Hanan grid. We can replace every y-coordinate y_s of every segment endpoint in B' by $\lfloor y_s / N^4 \rfloor N^4$ (ignoring possible segments of length 0). Then this results in an arborescence B for S that was scaled by N^4 (because the resulting drawing remains connected, every path to the origin remains monotone and no cycles are produced since the segments are on the Hanan grid). Any arborescence on the Hanan grid is a union of N paths changing directions at most N times, and every vertical part of such a path is stretched by at most N by the way we changed the y-coordinates. This gives a (very conservative) bound of $bN^4 \leq b' + N^3$ for the length b of B.

Hence, S has an arborescence of length at most k if and only if S' has an arborescence of length at most $kN^4 + N^3$, provided that $N^4 > 2N^3$, that is, $N > 2$. Since the instance $(S', kN^4 + N^3)$ can be computed in polynomial time from (S, k), and since the coordinates in S' are polynomially bounded in the coordinates of S, it follows that the YRSA problem is strongly NP-complete. □

4.3 Reducing RSA to Polygon Triangulation Flip Distance

We reduce YRSA to Polygon Triangulation Flip Distance. Let S be a set of N sinks on an $n \times n$ grid with root at $(1, 1)$ (recall that n is polynomial in N). We construct a polygon P^*_D and two triangulations T_1, T_2 in P^*_D such that a shortest flip sequence from T_1 to T_2 corresponds to a shortest RSA for S. To this end, we will describe how to interpret any triangulation of P^*_D as a *chain path*, a path in the integer grid that starts at the origin and uses only edges that go north or east. It will turn out that flips in P^*_D essentially correspond to moving the endpoint of the chain path along the grid. We choose P^*_D, T_1, and T_2 in such a way that a shortest flip sequence between T_1 and T_2 moves the endpoint of the chain path according to an Eulerian traversal of a shortest RSA for S. To force the chain path to visit all sites, we use the observations from Chapter 2: the polygon P^*_D contains a double chain for each sink, so that only for certain triangulations of P^*_D it is possible to

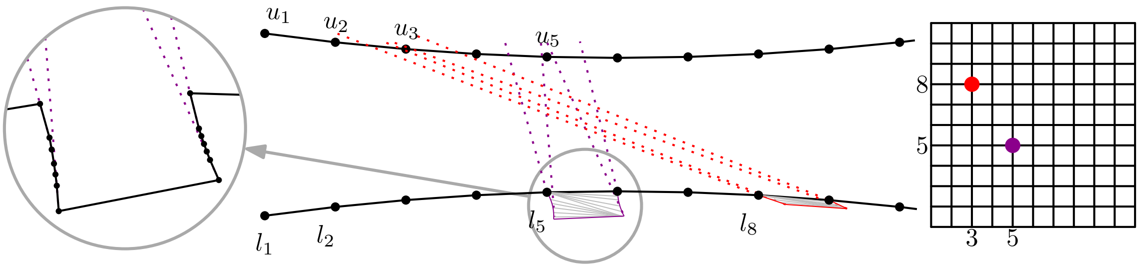


Figure 4.2: The sink gadget for a site (x, y) is obtained by replacing the edge $l_{\beta y}l_{\beta y+1}$ by a double chain with d vertices on each chain. The double chain is oriented such that $u_{\beta x}$ is the only point inside its hourglass and its flip-kernel. In our example, $\beta = 1$.

flip the double chain quickly. These triangulations will be exactly the triangulations that correspond to the chain path visiting the appropriate site.

4.3.1 The Construction

Our construction has two integral parameters, β and d. With foresight, we set $\beta = 2N$ and $d = nN$. We imagine that the sinks of S lie on a $\beta n \times \beta n$ grid, with their coordinates multiplied by β. We use the following auxiliary definition.

Definition 4.1. *Let D be a double chain of n vertices whose flip-kernel contains a point v to the right of the directed line $l_n u_n$. The polygon P_D^v is given by the sequence $\langle l_1, \ldots, l_n, v, u_n, \ldots, u_1\rangle$. The upper and the lower extreme triangulation of P_D^v contain the edge $u_n l_n$ and otherwise are defined in the same way as for P_D.*

We take a double chain D with βn vertices on each chain such that the flip-kernel of D extends to the right of $l_{\beta n}u_{\beta n}$. We add a point z to that part of the flip-kernel, and we let $P_D^+ = P_D^z$ be the polygon defined by $\langle l_1, \ldots, l_{\beta n}, z, u_{\beta n}, \ldots, u_1\rangle$. Next, we add double chains to P_D^+ in order to encode the sinks. For each sink $s = (x, y)$, we remove the edge $l_{\beta y}l_{\beta y+1}$, and we replace it by a (rotated) double chain D_s with d vertices on each chain, such that $l_{\beta y}$ and $l_{\beta y+1}$ correspond to the last point on the lower and the upper chain of D_s, respectively. We orient D_s in such a way that $u_{\beta x}$ is the only point inside the hourglass of D_s and so that $u_{\beta x}$ lies in the flip-kernel of D_s; see Figure 4.2. We refer to the added double chains as *sink gadgets*, and we call the resulting polygon P_D^*. For β large enough, the sink gadgets do not overlap, and P_D^* is a simple polygon. Since the y-coordinates in S are pairwise distinct, there is at most one sink gadget per edge of the lower chain of P_D^+. The precise placement of the sink gadgets is flexible, so we can make all coordinates polynomial in n; see Section 4.4 for details.

Next, we describe the source and target triangulation for P_D^*. In the source triangulation T_1, the interior of P_D^+ is triangulated such that all edges are incident to z. The sink gadgets are all triangulated with the upper extreme triangulation. The target triangulation T_2 is similar, but now the sink gadgets are triangulated with the lower extreme triangulation.

To get from T_1 to T_2, we must go from one extreme triangulation to the other for each sink gadget D_s. By Corollary 2.7, this requires $(d-1)^2$ flips, unless the flip sequence creates a triangle that allows us to use the vertex in the flip-kernel of D_s. In this case, we say that the flip sequence *visits* the sink s. For d large enough, a shortest flip sequence must visit each sink, and we will show that this induces an RSA for S of similar length. Conversely, we will show how to derive a flip sequence from an RSA. The precise statement is given in the following theorem.

Theorem 4.3. *Let $k \geq 1$. The flip distance between T_1 and T_2 with respect to P_D^* is at most $2\beta k + (4d-2)N$ if and only if S has an RSA of length at most k.*

We will prove Theorem 4.3 in the following sections. But first, let us show how to use it for our NP-completeness result.

Theorem 4.4. POLYGON TRIANGULATION FLIP DISTANCE *is* NP-*complete.*

Proof. As mentioned in the introduction, the flip distance in polygons is polynomially bounded, so POLYGON TRIANGULATION FLIP DISTANCE is in NP. We reduce from YRSA. Let (S, k) be an instance of YRSA such that S lies on a grid of polynomial size. We construct P_D^* and T_1, T_2 as described above. This takes polynomial time (see Section 4.4 for details). Set $l = 2\beta k + (4d-2)N$. By Theorem 4.3, there exists an RSA for S of length at most k if and only if there exists a flip sequence between T_1 and T_2 of length at most l. □

4.3.2 Chain Paths

Now we introduce the *chain path*, our main tool to establish a correspondence between flip sequences and RSAs. Let T be a triangulation of P_D^+ (i.e., the polygon P_D^* without the sink gadgets, cf. Section 4.3.1). A *chain edge* is an edge of T between the upper and the lower chain of P_D^+. A *chain triangle* is a triangle of T that contains two chain edges. Let $e_1, \dots, e_m$ be the chain edges, sorted from left to right according to their intersection with a line that separates the upper from the lower chain. For $i = 1, \dots, m$, write $e_i = (u_v, l_w)$ and set $c_i = (v, w)$. In particular, $c_1 = (1, 1)$. Since T is a triangulation, any two consecutive edges e_i, e_{i+1} share one endpoint, while the other endpoints are adjacent on the corresponding chain. Thus, c_{i+1} dominates c_i and $\|c_{i+1} - c_i\|_1 = 1$. It follows that $c_1 c_2 \dots c_m$ is an x- and y-monotone path in the $\beta n \times \beta n$-grid, beginning at the root. It is called the *chain path* for T. Each vertex of the chain path corresponds to a chain edge, and each edge of the chain path corresponds to a chain triangle. Conversely, every chain path induces a triangulation T of P_D^+; see Figure 4.3. In the following, we let b denote the upper right endpoint of the chain path. We now investigate how flipping edges in T affects the chain path.

Observation 4.5. *Suppose we flip an edge that is incident to z. Then the chain path is extended by moving b north or east.*

Observation 4.6. *Suppose that T contains at least one chain triangle. When we flip the rightmost chain edge, we shorten the chain path at b.*

Finally, we can flip an edge between two chain triangles. This operation is called a *chain flip*.

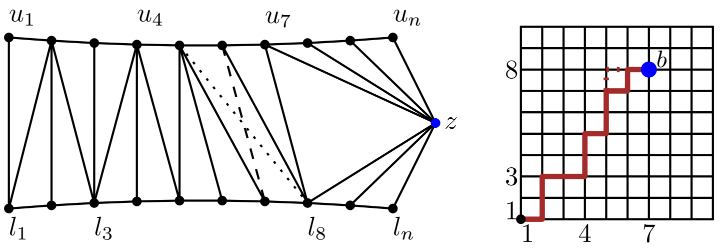


Figure 4.3: A triangulation of P_D^+ and its chain path. Flipping edges to and from z moves the endpoint b along the grid. A flip between chain triangles changes a bend.

Observation 4.7. *A chain flip changes a bend from east to north to a bend from north to east, or vice versa.*

Proof. If a chain edge $u_i l_j$ is incident to two chain triangles and is flippable, then the two triangles must be of the form $u_i u_{i-1} l_j$ and $l_j l_{j+1} u_i$, or $u_{i+1} u_i l_j$ and $l_{j-1} l_j u_i$. Thus, flipping $u_i l_j$ corresponds exactly to the claimed change in the chain path. ☐

Corollary 4.8. *A chain flip does not change the length of the chain path.*

We summarize the results of this section in the following lemma:

Lemma 4.9. *Any triangulation T of P_D^+ uniquely determines a chain path, and vice versa. A flip in T corresponds to one of the following operations on the chain path: (i) move the endpoint b north or east; (ii) shorten the path at b; (iii) change an east-north bend to a north-east bend, or vice versa.*

4.3.3 From an RSA to a Short Flip Sequence

Using the notion of a chain path, we now prove the "if" direction of Theorem 4.3.

Lemma 4.10. *Let $k \geq 1$ and A an RSA for S of length k. Then the flip distance between T_1 and T_2 with respect to P_D^* is at most $2\beta k + (4d - 2)N$.*

Proof. The triangulations T_1 and T_2 both contain a triangulation of P_D^+ whose chain path has its endpoint b at the root. We use Lemma 4.9 to generate flips inside P_D^+ so that b traverses A in a depth-first manner. This needs $2\beta k$ flips.

Each time b reaches a sink s, we move b north. This creates a chain triangle that allows the edges in the sink gadget D_s to be flipped to the auxiliary vertex in the flip-kernel of D_s. The triangulation of D_s can then be changed with $4d - 4$ flips; see Lemma 2.2. Next, we move b back south and continue the traversal. Moving b at s needs two additional flips, so we take $4d - 2$ flips per sink, for a total of $2\beta k + (4d - 2)N$ flips. ☐

4.3.4 From a Short Flip Sequence to an RSA

Finally, we consider the "only if" direction in Theorem 4.3. Let σ_1 be a flip sequence on P_D^+. We say that σ_1 *visits* a sink $s = (x, y)$ if σ_1 has at least one triangulation T that contains the chain triangle $u_{\beta x} l_{\beta y} l_{\beta y+1}$. We call σ_1 a *flip traversal* for S if (i) σ_1 begins and ends

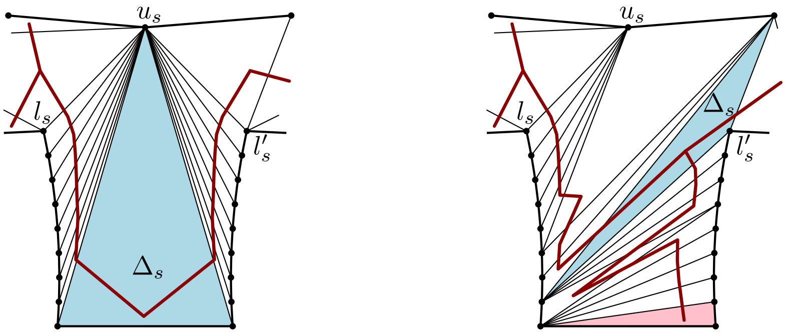


Figure 4.4: Triangulations of D_s in P_D^* with $\Delta_s = \Delta$ (left), and with Δ being an ear (red) and Δ_s an inner triangle (right). The fat tree indicates the dual.

in the triangulation whose corresponding chain path has its endpoint b at the root and (ii) σ_1 visits every sink in S. Lemma 4.11 will show that every short flip sequence in P_D^* can be mapped to a flip traversal. The basic idea behind the flip traversal is the same as for local triangulations. In the proof of Lemma 4.11, we map the triangulations of P_D^* to triangulations of P_D^+ and to triangulations of polygons defined for each sink gadget.

Lemma 4.11. *Let σ be a flip sequence from T_1 to T_2 with respect to P_D^* with $|\sigma| < (d-1)^2$. Then there is a flip traversal σ_1 for S with $|\sigma_1| \leq |\sigma| - (4d-4)N$.*

Proof. We show how to obtain a flip traversal σ_1 for S from σ. Let T^* be a triangulation of P_D^*. A triangle of T^* is an *inner triangle* if all its sides are diagonals. It is an *ear* if two of its sides are polygon edges. By construction, every inner triangle of T^* must have (i) one vertex incident to z (the rightmost vertex of P_D^+), or (ii) two vertices incident to a sink gadget (or both). In the latter case, there can be only one such triangle per sink gadget. The weak (graph theoretic) dual of T^* is a tree in which ears correspond to leaves and inner triangles have degree 3.

Let D_s be a sink gadget placed between the vertices l_s and l'_s. Let u_s be the vertex in the flip-kernel of D_s. We define a triangle Δ_s for D_s. Consider the bottommost edge e of D_s, and let Δ be the triangle of T^* that is incident to e. By construction, Δ is either an ear of T^* or is the triangle defined by e and u_s. In the latter case, we set $\Delta_s = \Delta$. In the former case, we claim that T^* has an inner triangle Δ' with two vertices on D_s: follow the path from Δ in the weak dual of T^*; while the path does not encounter an inner triangle, the next triangle must have an edge of D_s as a side. There is only a limited number of such edges, so eventually we must meet an inner triangle Δ'. We then set $\Delta_s = \Delta'$; see Figure 4.4. Note that Δ_s might be $l_s l'_s u_s$.

For each sink s, let the polygon $P_{D_s}^{u_s}$ consist of the D_s extended by the vertex u_s (cf. Definition 4.1). Let T^* be a triangulation of P_D^*. We show how to map T^* to a triangulation T^+ of P_D^+ and to triangulations T_s of $P_{D_s}^{u_s}$, for each s.

We first describe T^+. It contains every triangle of T^* with all three vertices in P_D^+. For each triangle Δ in T^* with two vertices on P_D^+ and one vertex on the left chain of a sink gadget D_s, we replace the vertex on D_s by l_s. Similarly, if the third vertex of Δ is on the right chain of D_s, we replace it by l'_s. For every sink s, the triangle Δ_s has one vertex at a

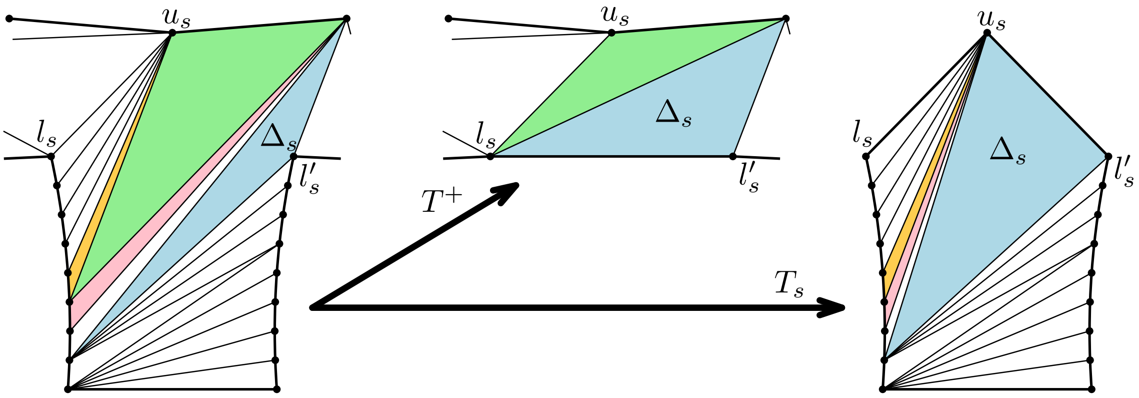


Figure 4.5: Obtaining T^+ and T_s from T^*.

point u_i of the upper chain. In T^+, we replace Δ_s by the triangle $l_s l'_s u_i$. No two triangles overlap, and they cover all of P_D^+. Thus, T^+ is indeed a triangulation of P_D^+.

Now we describe how to obtain T_s, for a sink $s \in S$. Each triangle of T^* with all vertices on $P_{D_s}^{u_s}$ is also in T_s. Each triangle with two vertices on D_s and one vertex not in $P_{D_s}^{u_s}$ is replaced in T_s by a triangle whose third vertex is moved to u_s in T_s (note that this includes Δ_s); see Figure 4.5. Again, all triangles cover $P_{D_s}^{u_s}$ and no two triangles overlap.

Eventually, we show that a flip in T^* corresponds to at most one flip either in T^+ or in precisely one T_s for some sink s. We do this by considering all the possibilities for two triangles that share a common flippable edge. Note that by construction no two triangles mapped to triangulations of different polygons $P_{D_s}^{u_s}$ and $P_{D_t}^{u_t}$ can share an edge (with $t \neq s$ being another sink).

Case 1. We flip an edge between two triangles that are either both mapped to T^+ or to T_s and are different from Δ_s. This flip clearly happens in at most one triangulation.

Case 2. We flip an edge between a triangle Δ_1 that is mapped to T_s and a triangle Δ_2 that is mapped to T^+, such that both Δ_1 and Δ_2 are different from Δ_s. This results in a triangle Δ'_1 that is incident to the same edge of $P_{D_s}^{u_s}$ as Δ_1, and a triangle Δ'_2 having the same vertices of P_D^+ as Δ_2. Since the apex of Δ_1 is a vertex of the upper chain or z (otherwise, it would not share an edge with Δ_2), it is mapped to u_s, as is the apex of Δ'_1. Also, the apex of Δ'_2 is on the same chain of D_s as the one of Δ_2. Hence, the flip affects neither T^+ nor T_s.

Case 3. We flip the edge between a triangle Δ_2 mapped to T^+ and Δ_s. By construction, this can only happen if Δ_s is an inner triangle. The flip affects only T^+, because the new inner triangle Δ'_s is mapped to the same triangle in T_s as Δ_s, since both apexes are moved to u_s.

Case 4. We flip the edge between a triangle Δ of T_s and Δ_s. Similar to Case 3, this affects only T_s, because the new triangle Δ'_s is mapped to the same triangle in T^+ as Δ_s, since the two corners are always mapped to l_s and l'_s.

Thus, σ induces a flip sequence σ_1 in P_D^+ and flip sequences σ_s in each $P_{D_s}^{u_s}$ so that $|\sigma_1| + \sum_{s \in S} |\sigma_s| \leq |\sigma|$. Furthermore, each flip sequence σ_s transforms $P_{D_s}^{u_s}$ from one extreme triangulation to the other. By the choice of d and Corollary 2.7, the triangulations T_s have to be transformed so that Δ_s has a vertex at u_s at some point, and $|\sigma_s| \geq 4d - 4$. Thus, σ_1 is a flip traversal, and $|\sigma_1| \leq |\sigma| - N(4d - 4)$, as claimed. □

In order to obtain a static RSA from a changing flip traversal, we use the notion of a trace. A *trace* is a domain on the $\beta n \times \beta n$ grid. It consists of *edges* and *boxes*: an edge is a line segment of length 1 whose endpoints have positive integer coordinates; a box is a square of side length 1 whose corners have positive integer coordinates. Similar to arborescences, we require that a trace R (i) is (topologically) connected; (ii) contains the root $(1, 1)$; and (iii) from every grid point contained in R there exists an x- and y-monotone path to the root that lies completely in R. We say R is a *covering trace* for S (or, R *covers* S) if every sink in S is part of R.

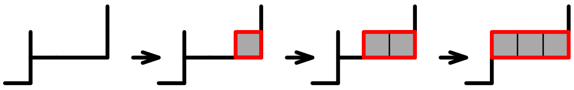
Let σ_1 be a flip traversal as in Lemma 4.11. By Lemma 4.9, each triangulation in σ_1 corresponds to a chain path. This gives a covering trace R for S in the following way. For every flip in σ_1 that extends the chain path, we add the corresponding edge to R. For every flip in σ_1 that changes a bend, we add the corresponding box to R. Afterwards, we remove from R all edges that coincide with a side of a box in R. Clearly, R is (topologically) connected. Since σ_1 is a flip traversal for S, every sink is covered by R (i.e., incident to a box or edge in R). Note that every grid point p in R is connected to the root by an x- and y-monotone path on R, since at some point p belonged to a chain path in σ_1. Hence, R is indeed a trace, the unique *trace of* σ_1.

Next, we define the *cost* of a trace R, $\text{cost}(R)$, so that if R is the trace of a flip traversal σ_1, then $\text{cost}(R)$ gives a lower bound on $|\sigma_1|$. An edge has cost 2. Let B be a box in R. A *boundary side* of B is a side that is not part of another box. The cost of B is 1 plus the number of boundary sides of B. Then, $\text{cost}(R)$ is the total cost over all boxes and edges in R. For example, the cost of a tree is twice the number of its edges, and the cost of an $a \times b$ rectangle is $ab + 2(a + b)$. An edge can be interpreted as a degenerated box, having two boundary sides and no interior.

Proposition 4.12. *Let σ_1 be a flip traversal and R the trace of σ_1. Then cost$(R) \leq |\sigma_1|$.*

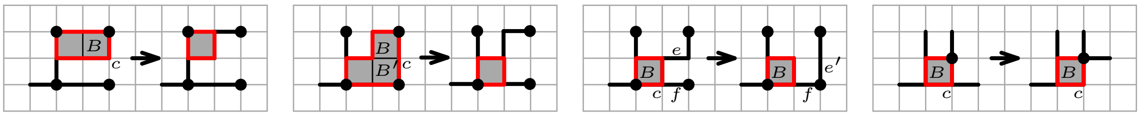
Proof. We argue that every element of R has unique corresponding flips in σ_1 that account for its cost. Let e be an edge of R. Then e corresponds to at least two flips in σ_1: one that extends the chain path to create e, and one that removes e (because the chain path starts and ends in a single point). Next let B be a box in R. The interior of B corresponds to at least one chain flip in σ_1. Moreover, when adding the box for a chain flip to the trace, we either transform edges to boundary sides or make boundary sides disappear from the boundary of the new trace. See Figure 4.6 for examples. However, when a chain flip adds a new box B to a trace, B is adjacent to at least two already existing elements (edges or boundary sides). Hence, by induction, the new boundary edges of a box add at most the cost that the box removes. □

Now we relate the length of an RSA for S to the cost of a covering trace for S, and thus to the length of a flip traversal. Since each sink (s_x, s_y) is connected in R to the root by a path of length $s_x + s_y$, traces can be regarded as generalized RSAs. In particular, we make the following observation.



(a) (b) (c) (d)

Figure 4.6: Examples of how boundary sides (red) are added to a trace. To a trace of cost 16 (a) a box (gray) is added (b), which transforms two edges in boundary sides and adds two boundary sides, resulting in an overall cost of 17. The next box removes one boundary side and one edge and adds three boundary sides (c), the cost becomes 18. A box might also remove more than two elements (d), reducing the overall cost to 17.



(a) (b) (c) (d)

Figure 4.7: Parts of traces to be modified; the boundary sides are shown in red. (a) A box that has a corner c with no incident elements can be removed. (b) Two adjacent boxes that have a shared corner c without any incident elements can be removed. (c) Replacing a single edge. (d) Sliding an edge.

Observation 4.13. *Let R be a covering trace for S that contains no boxes, and let A_{σ_1} be a shortest path tree in R from the root to all sinks in S. Then A_{σ_1} is an RSA for S.*

If σ_1 contains no flips that change bends, the corresponding trace R has no boxes. Then, R contains an RSA A_{σ_1} with $2|A_{\sigma_1}| \leq \text{cost}(R)$, by Observation 4.13. The next lemma shows that, due to the size of β, there is always a shortest covering trace for S that does not contain any boxes.

Lemma 4.14. *Let σ_1 be a flip traversal of S. Then there exists a covering trace R for S in the $\beta n \times \beta n$ grid such that R does not contain a box and such that cost$(R) \leq |\sigma_1|$.*

Proof. There exists at least one trace of cost at most $|\sigma_1|$, namely the trace of σ_1. Let $\mathcal{R}_1$ be the set of all covering traces for S that have minimum cost. If $\mathcal{R}_1$ contains a trace without boxes, we are done. Otherwise, every covering trace in $\mathcal{R}_1$ contains at least one box.

Let $\mathcal{R}_2 \subseteq \mathcal{R}_1$ be those covering traces among $\mathcal{R}_1$ that contain the minimum number of boxes. Let $Q \in \mathcal{R}_2$, and let B be a *maximal* box in Q, i.e., Q has no other box whose lower left corner has both x- and y-coordinate at least as large as the lower left corner of B. We investigate the structure of Q. Note that the property of being a trace is invariant under mirroring the plane along the line $x = y$; in particular, the choice of B in Q as a maximal box remains valid.

Observation 4.15. *Every corner c of B is incident either to a sink, an edge, or another box.*

Proof. If not, we could remove c and B while keeping the sides of B not incident to c as edges, if necessary; see Figure 4.7(a). The resulting structure would be a trace with smaller cost, contradicting the choice of Q. □

Observation 4.16. *Suppose B shares a horizontal side with another box B'. Let c be the right endpoint of the common side. Then c is incident either to a sink, an edge, or another box.*

Proof. Suppose this is not the case. Then we could remove B and B' from Q while keeping the sides not incident to c as edges, if necessary; see Figure 4.7(b). This results in a valid trace that has no higher cost but less boxes than Q, contradicting the choice of Q. □

Observation 4.17. *Let c be the lower right corner of B. Then c has no incident vertical edge.*

Proof. Such an edge would be redundant, since c already has an x- and y-monotone path to the root that goes through the lower left corner of B. □

Now we derive a contradiction from the choice of Q and the maximal box B. Note that since β is even, all sinks in S have even x- and y-coordinates. We distinguish two cases.

Case 1. There exists a maximal box B whose top right corner c' does not have both coordinates even. Suppose that the x-coordinate of c' is odd (otherwise, mirror the plane at the line $x = y$ to swap the x- and the y-axis). By Observation 4.15, there is at least one edge incident to the top right corner of B (it cannot be a box by the choice of B, and it cannot be a sink because of the current case). Recall the slide operation for an edge in an arborescence. This operation can easily be adapted in an analogous way to traces. If there is a vertical edge v incident to c', it cannot be incident to a sink. Thus, we could slide v to the right (together with all other vertical edges that are above v and on the supporting line of v). Hence, we may assume that c' is incident to a single horizontal edge e; see Figure 4.7(c). By Observation 4.15, the bottom right corner c of B must be incident to an element. We know that c cannot be the top right corner of another box (Observation 4.16), nor can it be incident to a vertical segment (Observation 4.17). Thus, c is incident to an element f that is either a horizontal edge or a box with top left corner c. But then e could be replaced by a vertical segment e' incident to f, and afterwards B could be removed as in the proof of Observation 4.15, contradicting the choice of Q.

Case 2. The top right corner of each maximal box has even coordinates. Let B be the rightmost maximal box. As before, let c be the bottom right corner of B. The y-coordinate of c is odd; see Figure 4.7(d). By the choice of B, we know that c is not the top left corner of another box: this would imply that there is another maximal box to the right of B. We may assume that c is not incident to a horizontal edge, as we could slide such an edge up, as in Case 1. Furthermore, c cannot be incident to a vertical edge (Observation 4.17), nor be the top right corner of another box (Observation 4.16). Thus, B violates Observation 4.15, and Case 2 also leads to a contradiction.

Thus, the choice of Q forces a contradiction in either case. No trace of minimum cost contains a box, so every minimum trace is an arborescence. This completes the proof of Lemma 4.14. □

Now we can finally complete the proof of Theorem 4.3 by giving the second direction of the correspondence.

Lemma 4.18. *Let $k \geq 1$ and let σ be a flip sequence on P_D^* from T_1 to T_2 with $|\sigma| \leq 2\beta k + (4d-2)N$. Then there exists an RSA for S of length at most k.*

Proof. Trivially, there always exists an RSA on S of length less than $2nN$, so we may assume that $k < 2nN$. Hence (recall that $\beta = 2N$ and $d = nN$),

$$2\beta k + 4dN - 2N < 2 \times 2N \times 2nN + 4nN^2 - 2N < 12nN^2 < (d-1)^2,$$

for $n \geq 14$ and positive N. Thus, since σ meets the requirements of Lemma 4.11, we can obtain a flip traversal σ_1 for S with $|\sigma_1| \leq 2\beta k + 2N$. By Lemma 4.14 and Observation 4.13, we can conclude that there is an RSA A for S that has length at most $\beta k + N$. By Theorem 4.1, there is an RSA A' for S that is not longer than A and that lies on the Hanan grid for S. The length of A' must be a multiple of β. Thus, since $\beta > N$, we get that A' has length at most βk, so the corresponding arborescence for S on the $n \times n$ grid has length at most k. ☐

4.4 A Note on Coordinate Representation

Since it is necessary for the validity of the proof that the input polygon can be represented in size that is bounded by a function polynomial in the size of the YRSA instance, we give a possible method on how to embed the polygon with vertices at rational coordinates whose numerator and denominator are polynomial in N. We only sketch an approach for the construction, the main techniques were already presented in-depth in Section 3.3.

As in Section 3.3, one can use the parametrization of the unit circle, choosing n points with rational coordinates inside the hourglass defined by the two common tangents of two instances of the unit circle for the upper and lower chain. Given these points, we now construct the small double chains for each sink. See Figure 4.8. Recall that, since β is a multiple of two, there are no small double chains on neighboring positions on the lower chain. Hence, for each sink we can define an orthogonal region within which we can safely draw the small double chain; we call this region the *bin* of the sink (outlined gray in Figure 4.8). Consider a sink (i, j). We first partition the segment $l_j l_{j+1}$ into thirds to obtain two points a and b, which again have rational coordinates; note that these points are not part of the polygon but "helper points" for our construction. Let t_a and t_b be the lines through u_i and a, and through u_i and b, respectively. The lines t_a and t_b intersect the bin at the points p_a and p_b, respectively. These two points will be the endpoints of the two chains. In addition to t_a the supporting lines of $u_{i-1}l_j$, as well as $l_j p_b$ and $p_a l_{j+1}$ define the triangular region $l_j p_a x_b$ (shaded gray in Figure 4.8) wherein we may place the chain incident to l_j (note that, e.g., only one of t_a and $p_a l_{j+1}$ will bound the triangular region). The chain incident to l_{j+1} is constrained analogously. We place a circular arc C through p_a and l_j inside $l_j p_a x_a$. C can be chosen to be tangent to either $p_a x_a$ or $x l_j$, in order to be contained in $l_j p_a x_a$. It is well-known that, for a line with rational slope through a point with rational coordinates on C that intersects C in a second point, this second point has rational coordinates as well. Suppose that C is tangent to $p_a x_a$. Then

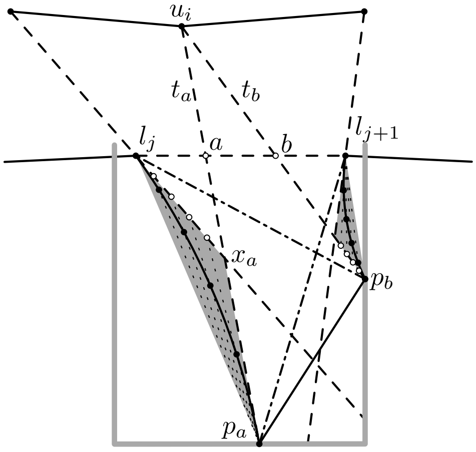


Figure 4.8: Construction of a small double chain for a sink.

we divide the line segment $l_j x_a$ into $(d-1)$ parts (where d is the number of elements on a small chain). A line through a point defined by this subdivision and p_a gives a rational point on C. Likewise, we can choose the points if C is tangent to $l_j x_a$. The points for the second chain are chosen analogously.

The coordinates are rational, and since every point can be constructed using only a constant number of other points, the numerator and denominator of each point are polynomial.

4.5 Chapter Summary

In this chapter, we showed that determining the shortest flip distance between two triangulations of a simple polygon is an NP-complete problem. In contrast to the analogue result on triangulations of point sets, our reduction does not give any hint whether there exists, say, a PTAS for the problem. There exits a PTAS for the Rectilinear Steiner Arborescence problem [114]. Given the general techniques for developing approximation algorithms for Traveling-Salesman-like problems [22, 125], it seems unlikely that an APX-hardness proof can be obtained using a similar approach with paths on the integer grid. A possibly existing approximation algorithm would also have to be valid for the convex case. But even for convex polygons, there has not been a breakthrough up to now, as discussed in Chapter 2. Determining the complexity of the flip distance problem for convex polygons is definitely a very interesting and supposedly challenging open problem.

Part II

On the Complexity of Some Problems on Point Sets

Chapter 5

Combinatorial Problems on Point Sets

This second part of the thesis is concerned with algorithms for combinatorial problems on point sets and their abstractions. In particular, we are interested in properties that can be described using sidedness queries. For a point set in the plane, a *sidedness query*[1] asks whether a point r lies to the left, on, or to the right of the directed supporting line of two points p and q. In this part of the thesis, we are interested in algorithms that only use these sidedness queries. In combinatorial geometry, it is a common concept to classify the infinite number of sets of points in $\mathbb{E}^2$ into a finite number of equivalence classes. In this chapter, we provide an introduction and preliminaries on combinatorial properties of point sets and their abstractions, aiming to provide a basis for the presentation of our algorithms in the following chapters. In addition, we discuss some motivation for developing algorithms that use only combinatorial properties and point out related work.

5.1 Point Set Classification

The aim of this section is to give a compact introduction to the aspects of combinatorial properties of point sets that are relevant for the results presented in this part of the thesis, with the goal of giving a self-contained basis for the following chapters while simultaneously introducing a common notation and putting emphasis on certain properties. For more details, the reader is referred to the work by Goodman and Pollack, in particular [82] and [84], where most of the concepts we use herein are presented, as well as to Edelsbrunner's book [54] on combinatorial and computational geometry. Knuth also provides a self-containing monograph [105] on the topic from a different point of view. We do not cover any of the results and the terminology of the closely related field of oriented matroids; the interested reader is referred to [32]. A self-contained overview is also given by Krasser [107], aiming at the enumeration of different order types. This section can be considered as a (rather informal) collection of the relevant concepts, where we restrict

---

[1] The term was probably introduced by Erickson and Seidel [63] for determining the orientation of a simplex in arbitrary dimensions.

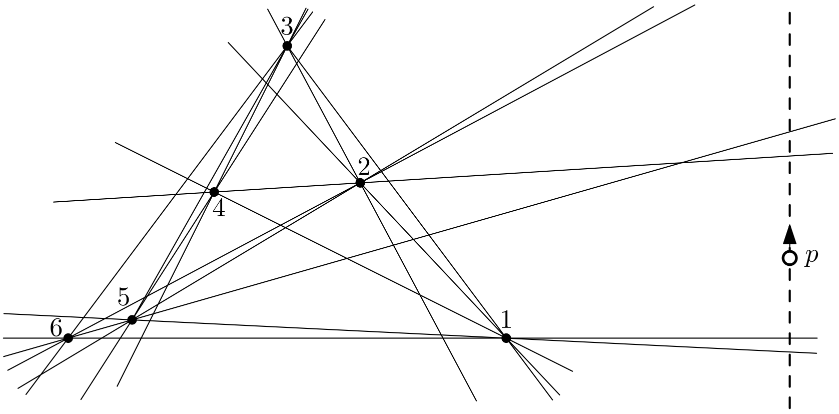


Figure 5.1: Sketch of the setting analyzed by Perrin.

our considerations to dimension 2, even though some of the results mentioned are valid in arbitrary dimensions and are presented in this way in the original publications.

5.1.1 Circular Sequences

Historically, work on the circular sequence of finite sets of points in the plane has already been published in 1882 by Perrin [130]. For a point set in the plane, Perrin considered an additional point p moving along a line that is sufficiently far from the point set (meaning that all points as well as the crossings of the supporting lines of all point pairs are on one side of the line); see Figure 5.1. For simplicity, we call these the *supporting lines of the point set*. The radial order of the points around p changes every time p traverses a supporting line of two points. He observed that these two points on this supporting line change their relative position in the circular order around p, and that these points are adjacent in the circular order around p at that time. In 1980, Goodman and Pollack [77] again considered this sequence of permutations, based on which they defined the combinatorial equivalence of point sets.

A *configuration of n points* is an ordered n-tuple of distinct labeled points in the plane.[2] A configuration of points is *non-degenerate* if no three points are collinear and no two pairs define two parallel lines. Let ℓ be a line not orthogonal to the supporting line of any two points of a configuration $\mathcal{C}$ of points. The orthogonal projection of $\mathcal{C}$ on ℓ gives a permutation of the labels of the points when traversing the line. When rotating ℓ in counterclockwise direction, ℓ will at some point be orthogonal to a supporting line of two points; right after that, the labels of these two points will have switched their position in the permutation. After having rotated ℓ by 180°, the initial permutation of the points

---

[2] In the computational geometry community it is also common to refer to finite point sets instead of to configurations of points (as is also done in the first part of this thesis). Without the order of the n-tuple being defined, there is no real formal difference between a configuration of points and a point set. However, the former term will be useful when looking at generalized configurations of points. We therefore will refer to a finite point set as a configuration when considering properties that have a dual interpretation in arrangements of lines, as suggested by Edelsbrunner [54, p. 13]. We also follow Goodman and Pollack [77, 82, 84] in requiring that the points of a configuration are labeled, usually from 1 to n.

will be inverted, and after a rotation of 360°, we again obtain the initial permutation; see Figure 5.2. We obtain a doubly-infinite sequence of permutations with period $2\binom{n}{2}$. This sequence is called the *circular sequence* of $\mathcal{C}$. The circular sequence of $\mathcal{C}$ fulfills the following two properties [77]:

1. Successive permutations differ only by two adjacent labels that have been switched.
2. Any $\binom{n}{2}$ consecutive permutations make use of all $\binom{n}{2}$ possible switches in passing from each to the next.

The second property implies that, after some pair (i, j) is switched, all other pairs of elements are switched before i and j are switched again (see also [82]). An infinite periodic sequence of permutations that satisfies these two properties is called an *allowable sequence of permutations*.[3] Hence, a circular sequence is an allowable sequence. Goodman and Pollack [77] show that an allowable sequence is determined by its sequence of ordered switches. If a configuration of points is mirrored, the sequence of ordered switches in its circular sequence is inverted. Inverting the sequence of ordered switches of any circular sequence to obtain a different one is called *reflection*. Two allowable sequences are *combinatorially equivalent* if one can be transformed into the other by relabeling its elements, by reflection, or by both. Two configurations of points are *combinatorially equivalent* if their circular sequences are combinatorially equivalent.[4]

Combinatorial equivalence allows us to partition the infinite number of point sets of size n into a finite number of combinatorially equivalent classes (such a class is also called the *combinatorial type* of a configuration [84, p. 104]). Perrin [130] claimed not only that every circular sequence fulfills the properties that define allowable sequences, but also that every allowable sequence is also the circular sequence of some configuration of points. Goodman and Pollack [77] gave a counterexample of five elements for this claim. Hence, while allowable sequences provide a means of enumerating all combinatorially different configurations of points, not all allowable sequences can be realized by a point configuration. An allowable sequence is called *realizable* if it is the circular sequence of some configuration of points.

For simplicity, we restricted the explanation of circular sequences to non-degenerate configurations of points. However, the theory extends in a natural way to configurations containing collinear triples of points and parallel supporting lines [82]. We will consider degenerate cases only when necessary. The reader is referred to the literature mentioned at the beginning of this section for explanations of the degenerate setting.

Concerning the number of combinatorially equivalent allowable sequences, the following bound exists (see also [76, pp. 117–118]).

---

[3] In [77], such sequences were called “allowable circular sequences”. Goodman and Pollack later switched to the term “allowable sequences” [82].

[4] We follow the definition in [77]. Note that, e.g., in [84] two labeled point sets are defined to be combinatorially equivalent if their circular sequence is equivalent (respecting the labeling), and two unlabeled point sets are combinatorially equivalent if there is a labeling such that the circular sequences are equivalent (after possibly mirroring one point set). Also in [82], equivalence is actually only defined by strictly respecting the labeling. Since in this work it appears to be more useful to not strictly rely on a given labeling, we follow [77].

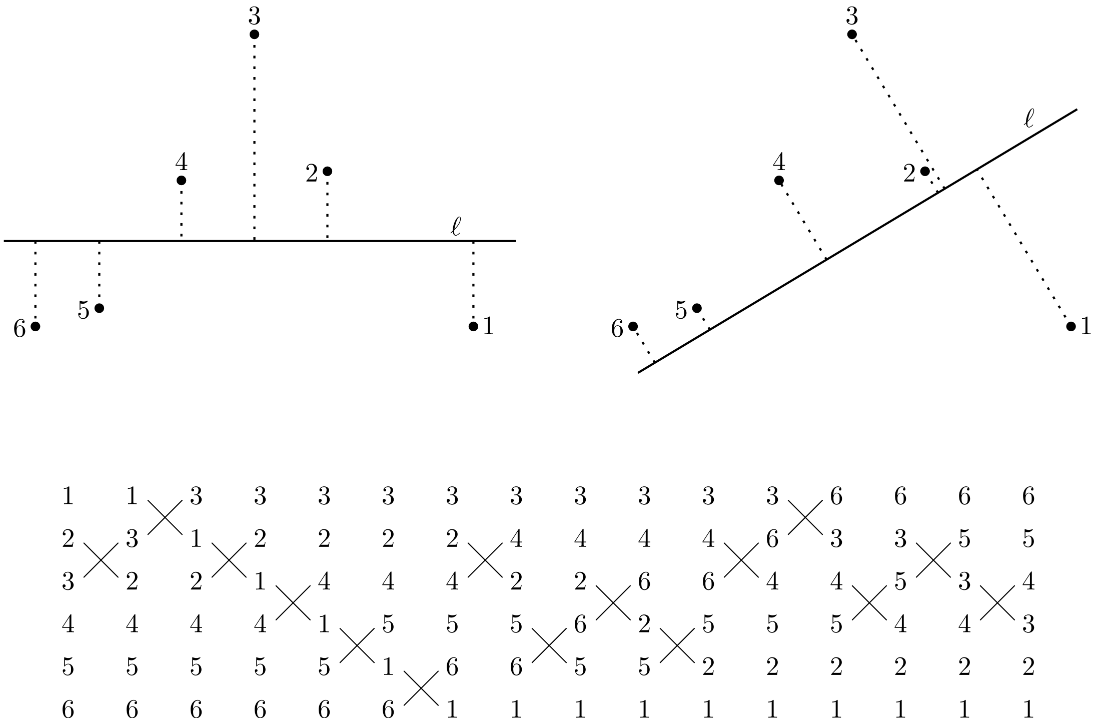


Figure 5.2: The orthogonal projection of a configuration of points on a rotating line gives its circular sequence.

Theorem 5.1 (Stanley [151]). *The number of simple allowable sequences on* $1, \dots, n$ *containing the permutation* $(1, \dots, n)$ *is given by*

$$\frac{\binom{n}{2}!}{1^{n-1}3^{n-2}\dots(2n-3)^1} .$$

Consider the extreme points of a configuration $\mathcal{C}$ of points. It is easy to see that (the label of) any extreme point h_1 has to occur at the first position in at least one of the permutations of the circular sequence of $\mathcal{C}$. When looking at the subsequent permutations, h_1 will at some instant of time be replaced by another point h_2 at the first position. We observe that h_2 is the counterclockwise neighbor of h_1 on the boundary of the convex hull of $\mathcal{C}$. Therefore, the circular sequence of $\mathcal{C}$ gives us the vertices of CH($\mathcal{C}$). If we consider an allowable sequence that is not the circular sequence of any set of points, we can also obtain such a sequence of extremal elements, just as we did for $\mathcal{C}$. So in some way, these are the vertices of the convex hull of a point set that does not even exist. We will encounter this kind of abstraction again when discussing order types, and we will see that there are settings where this abstraction actually makes sense. However, we used the classic definition that the convex hull of a point set is the convex polygon of smallest area that contains all points of the set. The concept of area is no longer applicable to a configuration that is not realized as a point set. It is therefore sometimes more convenient to handle the convex hull as the cycle of points defining its boundary, i.e., the convex hull then is a polygonal cycle and not a polygon.

The circular sequence does not only give us the convex hull of a configuration of points. We can actually obtain more general information from it. Consider two points u and v, and the two permutations where they switch such that u first precedes v and then succeeds v. The two permutations tell us that the points preceding that pair lie on one side of the directed line uv, whereas the points succeeding the pair lie on the other side. Hence, the orientation of each (ordered) triple of points is given by the circular sequence, and is therefore the same for all combinatorially equivalent configurations of points (up to mirroring the point set). Now consider two points s and t that are disjoint from u and v and that are switched next in the sequence. If we modify the sequence such that s and t are switched before u and v, we get a different allowable sequence that still has the same orientations of triples. We will revisit this observation when defining order types and their abstraction.

5.1.2 Duality and Pseudo-Line Arrangements

For the following discussion of duality, we make use of the projective plane for the first time herein. Similarly to the Euclidean plane, there is a synthetic definition of the projective plane by a set of axioms for points and lines. However, we use the analytic realization as it probably suits our needs best. See [29, Chapter 2] for definitions that concisely cover both the formal definition by axioms and the corresponding analytic embodiment. An *affine plane* is a plane in which any two points determine a unique line, and through any point not on a line ℓ, there is a unique line parallel to ℓ [29, Chapter 1]. Since we restrict ourselves to planes coordinatized using real numbers, we will, in general, not distinguish between the affine and the Euclidean plane. Suppose that we add, for any maximal set of parallel lines, a point (not having any coordinates), called a *point at infinity*, where these lines are defined to cross. The union of these points is called the *line at infinity*. The *real projective plane* $\mathbb{P}^2$ is the extension of the (coordinatized) affine plane by the line at infinity [29, p. 43]. It will be useful to consider the following definition. The points in the real projective plane can be defined as lines in $\mathbb{E}^3$ that pass through the origin; a line in $\mathbb{P}^2$ corresponds to a plane in $\mathbb{E}^3$ containing the origin [29, pp. 42–43]. Note that by intersecting the lines in $\mathbb{E}^3$ with a plane A, we obtain points in an affine plane. Observe that it can be shown that the real projective plane is non-orientable: When performing a transformation in $\mathbb{P}^2$ by continuously rotating A around the origin, each point on a small oriented closed curve on A will vanish at infinity and return to the plane (at the “opposite side”); however, the orientation of the curve has changed (see, e.g., [57, pp. 28–29]). Another convenient model of $\mathbb{P}^2$ is by points on the unit sphere; each point is identified with its antipodal point, and lines become great circles. This is equivalent to the intersection of the unit sphere with the lines and planes through the origin in the previous model (see, e.g., [29, p. 42]).

The concept of duality is known since around 200 years in connection with the axiomatic definition of the projective plane: a proposition remains true if one interchanges the terms “point” and “line” as well as “joint” and “intersection” [47, pp. 15–16] (where “joint” denotes the supporting line of a pair of points).[5] In the real projective plane, such a duality can be obtained by a construction that may be explained by the model of $\mathbb{P}^2$ in $\mathbb{E}^3$ the

---

[5] This duality should not be confused with the graph theoretic dual used in Chapter 4.

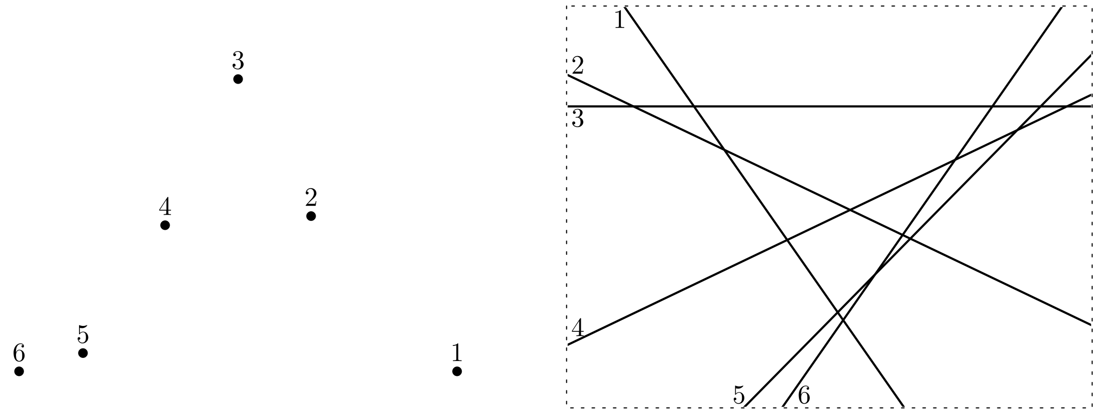


Figure 5.3: A configuration of points in the primal and its dual arrangement of lines.

following way. A point in $\mathbb{P}^2$ is represented by a line through the origin. The dual line of this point is represented by the plane through the center of the sphere such that the plane is normal to the line (see, e.g., [135, pp. 24–26, 316–319]). Goodman and Pollack [80] extended their concept of circular sequences to arrangements of lines, first in the Euclidean and then in the projective plane. A finite set of lines in the plane dissects the plane into a cell complex, called an *arrangement of lines* or *line arrangement*.[6] An arrangement is *simple* if no three lines of the arrangement have a point in common. Consider a simple line arrangement $\mathcal{A}$ in the (coordinatized) Euclidean plane not containing (i) any vertical line, (ii) any two parallel lines, or (iii) any two crossings that have the same x-coordinate. Consider a vertical line $\ell_{-\infty}$ (not part of $\mathcal{A}$) such that all crossings of two lines of $\mathcal{A}$ have a larger x-coordinate than $\ell_{-\infty}$.[7] The intersection points of $\ell_{-\infty}$ with $\mathcal{A}$ ordered along $\ell_{-\infty}$ give a permutation of the lines defining $\mathcal{A}$. If we move a vertical line ℓ starting at $\ell_{-\infty}$ in the positive x-direction and ℓ traverses the crossing of two lines of $\mathcal{A}$, then we obtain a new permutation with exactly these two lines interchanged. Obviously, these two lines were neighbored in the initial permutation. If we continue sweeping $\mathcal{A}$ with the vertical line, we get a half-period of an allowable sequence and afterwards reach the permutation that is the reverse of the initial permutation. Goodman and Pollack [80] identify each point $p = (a, b)$ with a line $p^*\colon y = -ax + b$ (which is one variant of a dual transform, see, e.g., [54, pp. 13–14]). We call p^* the *dual* line of the point p, and p is the *primal* point of the line p^* (the dual point q^* of a line q is defined analogously). See Figure 5.3 for an example. This identification is used to show the following result.

Theorem 5.2 (Goodman, Pollack [80]). *An allowable sequence of permutations can be obtained from an arrangement of lines if and only if it is realizable by a configuration of points.*

As already mentioned at the beginning of this section, the main property of duality is *incidence preservation*. Fixing a vertical direction also allows for stating the property of

---

[6] There has, up to now, not been a proper reason to distinguish between a set of lines and an arrangement of lines. This necessity will become more obvious during the discussion of pseudo-line arrangements.

[7] In the Euclidean plane, we use the convenient terms "above"/"below" and "left"/"right", while the left and right side of a directed line only depends on its direction.

order preservation. In particular, the following two statements are fulfilled by the given dual transform (see, e.g., [54, p. 14]; the signs herein are inverted to conform with [80]).

1. A point p is on a non-vertical line ℓ if and only if the point ℓ^* is on the line p^*.
2. A point p lies above a non-vertical line ℓ if and only if the point ℓ^* lies below the line p^*.

Observe that, under the given dual transform, the half-period of the circular sequence given by an arrangement of lines corresponds to the half-period defined by rotating a line ℓ with starting slope 0 counterclockwise by 180° and orthogonally projecting the corresponding configuration $\mathcal{C}$ of points on it (for this observation, we restrict our discussion to configurations of points without any pair of points sharing the same x-coordinate, analogously to not considering arrangements with parallel lines). In this setting, choosing a different half-period in the circular sequence of $\mathcal{C}$ is equivalent to rotating $\mathcal{C}$ in this setting (or choosing a different starting slope for ℓ). By performing this rotation in the primal, both the slope and the intercept of each line in the dual line arrangement change. Still, the circular sequence of the rotated point set is the same. Hence, we can have "different" line arrangements that give different half-periods of the same circular sequence and are therefore dual to combinatorially equivalent configurations of points. We postpone the discussion about the "difference" of line arrangements to Section 5.1.3.2 until having discussed pseudo-line arrangements and line arrangements in the projective plane.

The definition of circular sequences for arrangements of lines did not use the fact that the lines are straight curves, but only that they intersect exactly once (since they are straight) and that a line in $\mathbb{E}^2$ is an x-monotone plane curve. The concept of straight lines is generalized by pseudo-lines. A *pseudo-line* in the Euclidean plane is an x-monotone plane curve.[8] An *arrangement of pseudo-lines* (or *pseudo-line arrangement*) is a dissection of the Euclidean plane into a cell complex by a set of pseudo-lines such that each pair of pseudo-lines intersects in exactly one point, at which these two pseudo-lines cross.[9] An arrangement of pseudo-lines is *simple* if no three of these pseudo-lines have a point in common. In the same way as an arrangement of lines, an arrangement of pseudo-lines gives a half-period of an allowable sequence. Recall that there are allowable sequences that are not realizable by point sets and therefore also not by an arrangement of straight lines. Goodman [75] provides the following construction that shows that this limitation does not hold for pseudo-lines. Consider any half-period of an allowable sequence of permutations, denoted by $\langle \Pi_1, \ldots, \Pi_m \rangle$, with $m = \binom{n}{2}$. From this half-period, an arrangement of piecewise-linear pseudo-lines, called *wiring diagram*, is constructed in the following way. We place a sequence of $m + 1$ vertical lines (not part of the arrangement), ordered from left to right, with each neighboring pair of distance, say, 1. For each of the n elements of the permutations, we place a point on each vertical line with the y-coordinate of the point

---

[8] We follow the definition of Edelsbrunner [54, p. 37], who uses the Euclidean rather than the projective space. While x-monotonicity of the pseudo-lines in the Euclidean plane will be a convenient property throughout this thesis, this property is sometimes dropped, and pseudo-lines are defined as simple curves approaching a point at infinity in either direction (see, e.g., [66, p. 87]).

[9] Observe that this justifies the separation of the concepts of an arrangement and a set of lines. A set of x-monotone curves might very well contain two distinct elements that intersect in more than one point.

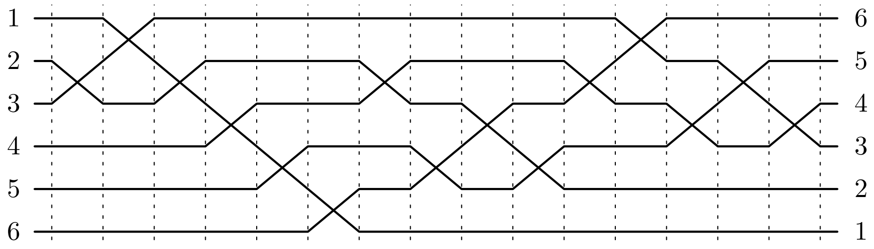


Figure 5.4: A wiring diagram.

representing its position in the corresponding permutation (the rightmost vertical line corresponds to the reverse of the initial permutation). When we connect the points of each element of the permutations and extend the leftmost and rightmost point by a horizontal ray in negative and positive x-direction, respectively, we end up with a simple pseudo-line arrangement realizing the given allowable sequence. For an example, Figure 5.4 provides a wiring diagram of the configuration shown in Figure 5.3.

5.1.3 Semispaces, Order Types, and Stretchability

Recall that two allowable sequences were defined to be combinatorially equivalent if one sequence can be obtained from the other by relabeling its elements, by reflection, or by both. In this section, we examine a coarser classification of point sets in the Euclidean plane. To this end, we will sometimes replace the formulation of relabeling by a bijection between two point sets.

5.1.3.1 From Circular Sequences to Order Types

In Section 5.1.1, we already observed that the circular sequence of a configuration of points encodes, for every pair of points, which points are on one side of the supporting line of the points and which ones are on the other. A *semispace* of a configuration $\mathcal{C}$ of points is a point set consisting of all points of $\mathcal{C}$ lying on one side of a single line [82]. As can be seen from the definition of circular sequences, a set of points is a semispace of $\mathcal{C}$ if and only if the labels of the points occur as the prefix of a permutation in the (infinite) circular sequence of $\mathcal{C}$ (see [82] for a formal proof even for the degenerate case), and therefore two semispaces are also given by the set of all pseudo-lines above a point in the dual pseudo-line arrangement in $\mathbb{E}^2$ and the set of pseudo-lines below it. Two point sets are *semispace-equivalent* if there exists a bijection between them such that the set of semispaces of one point set is equivalent to the set of semispaces of the other point set. While a configuration of points is, due to these observations, semispace-equivalent to its mirrored counterpart, this mirroring changes the orientation of each ordered triple. Still, for any two points u and v of a configuration of points, its circular sequence (in contrast to its combinatorial equivalence class) encodes which points are to the left of the directed line uv and which ones are to the right. Equivalently, for any (ordered) point triple (u, v, w), we know whether it is oriented clockwise or counterclockwise when given the circular sequence of the configuration of points, i.e., the circular sequence can be used to answer all sidedness queries on the point set. However, for two combinatorially equivalent configurations of

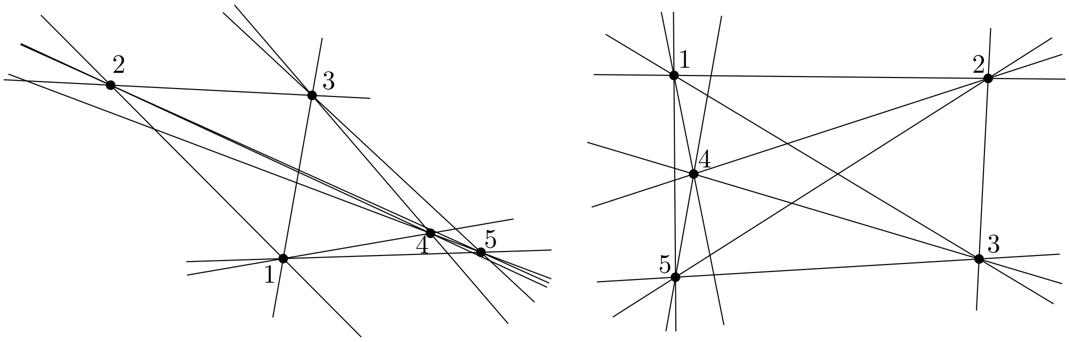


Figure 5.5: Two point sets having the same order type (adapted from [7]). The labeling indicates one possible bijection between the sets indicating that the sets have the same order type.

points, three identified points may not have the same orientation, as one of the circular sequences may have been reversed. This case, however, corresponds to having mirrored one of the two configurations of points, and therefore, the orientation of all point triples have been inverted. The notion of order types is introduced by Goodman and Pollack [81]. Two point sets have the same *order type* if there exists a bijection π between the two point sets such that either each point triple (u, v, w) of one set is oriented in the same direction as the point triple $(\pi(u), \pi(v), \pi(w))$ of the other set, or, for each point triple (u, v, w), the triple $(\pi(u), \pi(v), \pi(w))$ has the opposite orientation.[10] See Figure 5.5 for two point sets exemplifying such a bijection. It is rather easy to see that two point sets having the same order type are also semispace-equivalent; more surprisingly, also the opposite holds [82]. In this connection, Goodman and Pollack [81] provide the basic theorem of geometric sorting: being given, for each directed supporting line of two points, the number of points to the left of this line is equivalent to being given the set of points to the left of this line (which also holds for arbitrary dimensions).

The classification of point sets by their order type has many useful properties. For example, it captures all the properties that are invariant under affine transformations. In particular, given two line segments by their endpoints, the order type determines whether these two segments cross. Also, the convex hull of the point set is given by the order type. This makes the order type useful for working with geometric graphs (as long as we are not concerned with properties like angles, distances, or circles defined by point triples). For example, whether a geometric graph is a triangulation of the point set (or of a simple polygon spanning the point set) is determined by the order type. The order type also determines whether an edge in a triangulation is flippable. See [12] for an extensive list of properties depending on the order type. For other problems, the order type does not define the solution of the problem. One example would be the minimum spanning tree of a point set. While this is also a crossing-free graph, it changes when applying affine transformations to the point set.

---

[10] In [81], the corresponding triples have to have the same orientation. However, the given definition implies that a point set and its mirror image have the same order type, following, e.g., Krasser [107].

Let us analyze how the order type is defined by the circular sequence by a detailed example. Consider any two points u and v of a configuration $\mathcal{C}$ of points. An example can be seen in Figure 5.6 with u being point 1 and v being point 2 in both configurations shown. In the circular sequence of $\mathcal{C}$, consider the two adjacent permutations Π_v and Π_u such that v precedes u in Π_v and v succeeds u in Π_u. Since the rotating line ℓ that defines the circular sequence rotates counterclockwise, we know that any point w that precedes v in the first permutation Π_v is to the left of the directed line uv. In this way, a half-period gives us the orientation of each triple in a configuration of points. Let us observe this fact under the aspect of order preservation. Consider the dual line arrangement $\mathcal{A}$ of $\mathcal{C}$. If the x-coordinate of u is larger than the one of v, then the slope of the line u^* is smaller than the one of v^*. Hence, the line u^* "starts" above v^* when going from left to right (i.e., u precedes v in the half-period defined by $\mathcal{A}$). Since w is to the left of uv, it is below the line uv. Therefore, in the dual, the intersection point of u^* and v^* (i.e., $(uv)^*$) is above the line w^*. In this manner, the lines above and below a crossing in the line arrangement gives us the orientation of each triple in which the two points defining the crossing are involved. However, it is necessary to know which of the two points precedes the other in the first permutation of the half-period given by the arrangement. Again, this reasoning is not limited to arrangements of lines, it works equally well if we are given a dual pseudo-line arrangement of a configuration of points.

Consider the two configurations of points in Figure 5.6. Let the one at the top be $\mathcal{C}_1$ and the one at the bottom be $\mathcal{C}_2$. With this fixed labeling, these two configurations do not have the same circular sequence, and it can be verified that they are not combinatorially equivalent. This can be observed in the dual line arrangements by the dashed vertical line, which passes through the intersection between line 1 and line 6. In $\mathcal{C}_1$, the switch between the lines 2 and 4 happens before the switch between the lines 1 and 6; the inverse happens in $\mathcal{C}_2$. Still, in the two dual arrangements and under the bijection indicated by the labels, the set of lines above and below each crossing are the same. Hence, the two sets have the same order type. Observe that, while the circular sequence captures on which sides the supporting lines of four points in convex position meet, this is not indicated by the order type. Recall that we defined line arrangements in the Euclidean plane as cell complexes. The two arrangements ($\mathcal{A}_1$ and $\mathcal{A}_2$ in Figure 5.6) are isomorphic. This gives evidence that there is a connection between the isomorphism of arrangements and the order type. But if we rotate $\mathcal{C}_2$, the order type naturally stays the same, but this alters $\mathcal{A}_2$ to represent a different half-period of the circular sequence of $\mathcal{C}_2$. On the other hand, if we rotate $\mathcal{A}_2$, the arrangement stays isomorphic but the order type of the primal configuration of points changes. However, as pointed out in [82], a connection between order types and isomorphism of arrangements can be seen in the projective plane.

5.1.3.2 Arrangements in the Projective Plane

In [80], the duality between configurations of points and arrangements of lines is extended to the real projective plane $\mathbb{P}^2$ in the following way. In order to get a circular sequence from a configuration $\mathcal{C}$ of points in $\mathbb{P}^2$, we have to fix a line at infinity (not passing through any point of $\mathcal{C}$ or the crossing of two supporting lines) and an orientation of its complement (as $\mathbb{P}^2$ is non-orientable), which brings us basically back to $\mathbb{E}^2$, where we can obtain the

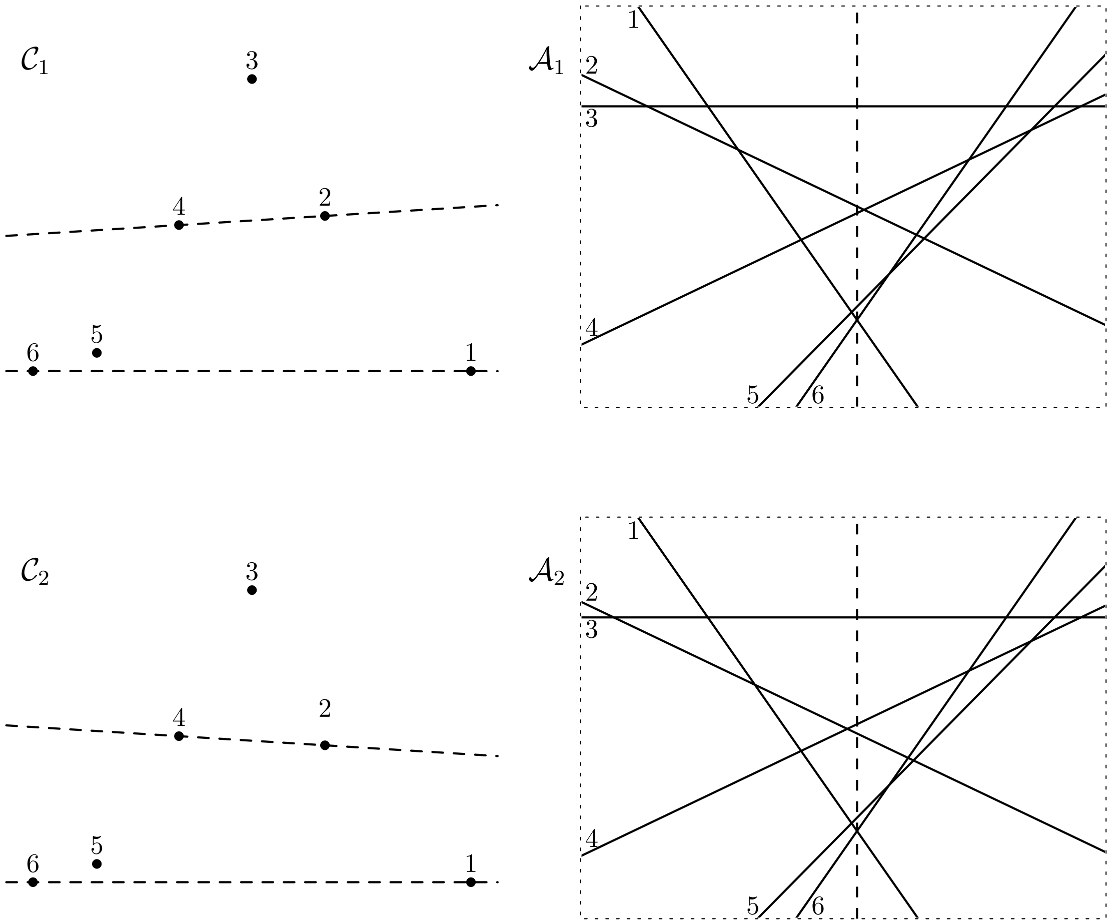


Figure 5.6: Two point sets having the same order type but a different circular sequence.

circular sequence. (To be more precise, removing the line at infinity brings us to the affine plane, and the choice of the line at infinity prevents any two supporting lines of point pairs to become parallel.) For an arrangement $\mathcal{A}$ of lines, we choose a *distinguished point* ψ not on $\mathcal{A}$ and fix a line at infinity passing through ψ but not through any crossing of $\mathcal{A}$. Let ℓ be a rotating line through ψ. With the choice of the line at infinity, we can identify ℓ with a vertical line sweeping the arrangement in the Euclidean plane (the rotation of ℓ gives the direction from left to right, and ψ is the point at "vertical infinity", i.e., the point at infinity where two vertical lines meet). See Figure 5.7 for an illustration. Observe that, for configurations of points in $\mathbb{P}^2$, different circular sequences can be obtained by choosing a different line at infinity. For arrangements of lines in $\mathbb{P}^2$, different circular sequences can be obtained by a different choice of the distinguished point ψ; however, the line at infinity can be chosen arbitrarily among the lines passing through ψ but not through any crossing of the arrangement.

The combinatorial equivalence between the circular sequences of arrangements of lines is more natural in the projective plane. The choice of the line at infinity does not play a role here, but only the choice of the distinguished point defining the vertical direction.

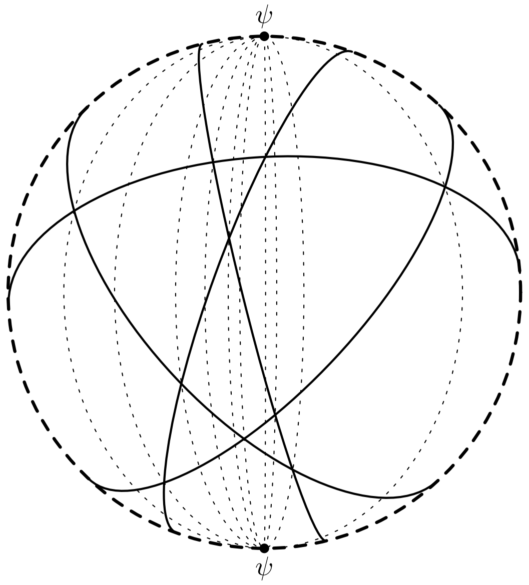


Figure 5.7: Sweeping a line arrangement in $\mathbb{P}^2$ by a line rotating through ψ, shown in the sphere model of $\mathbb{P}^2$.

As can be observed using the correspondences given in [80], the distinguished point ψ in the dual line arrangement is the dual of the line at infinity chosen for the configuration of points. Therefore, the lines defining that cell correspond to the points on the boundary of the convex hull of the configuration.

Let us further analyze this extension to the projective plane (see [82] for a more formal discussion). For configurations of points, the choice of the line at infinity and the orientation basically brings us back to the Euclidean plane (we do not have to choose a vertical direction). Suppose we can move the line at infinity ℓ_∞ in such a way that we sweep over a crossing of two supporting lines of the configuration of points, but not over any point of $\mathcal{C}$. When projecting back to the Euclidean plane, the resulting configuration $\mathcal{C}'$ has the same order type as $\mathcal{C}$, but, similar to the example in Figure 5.6, the circular sequence of $\mathcal{C}'$ will, in general, be different. If we move ℓ_∞ even further until it sweeps over a point of the configuration, then, in general, the order type changes as well (there can, of course, be symmetries, like for a configuration of only three points). Similarly, in the dual, consider the distinguished point $\psi = \ell^*_\infty$ and the cell containing ψ, which is called the *marked cell.* If ψ can be moved around in the marked cell such that it traverses a supporting line of two crossings in $\mathcal{A}$, then, in general, the circular sequence of $\mathcal{A}$ changes (this would correspond to slightly changing the vertical direction, as can be observed the example of Figure 5.6). If we continue moving ψ until it is in a different cell of $\mathcal{A}$, then, in general, also the order type of the primal configuration of points changes. As already discussed, the choice of the line at infinity for $\mathcal{A}$ does not influence the circular sequence (and therefore also not the order type); only the choice of ψ does this. Hence, if we have two arrangements that are isomorphic in the projective plane and if, for both, we place the distinguished point in the same cell (given by the isomorphism), then these arrangements represent the same order type.

5.1.3.3 Abstract Order Types and Pseudo-Line Stretchability

As the order type of a circular sequence is only defined by the set preceding and succeeding a switched pair in the circular sequence, we can also obtain the orientation of each triple out of a non-realizable allowable sequence. Intuitively, we can therefore answer sidedness queries on a point set that may not even exist. Such a mapping of orientations to all triples defined by an allowable sequence defines a so-called *abstract order type*. Given a non-realizable allowable sequence Σ does not mean that there is no configuration of points with a different circular sequence that has the order type defined by Σ. To analyze when an abstract order type is the order type of a configuration of points, we consider arrangements of pseudo-lines in the projective plane (see [82]). A *pseudo-line* in the projective plane is a non-contractible simple closed curve (observe that this means that the curve intersects the line at infinity and that therefore two such curves cross an odd number of times). An *arrangement of pseudo-lines* in the projective plane is a dissection of the projective plane into a cell complex by a set of pseudo-lines that pairwise meet at exactly one point, at which they cross. Intuitively, a projection of a pseudo-line arrangement from the projective plane to the Euclidean plane should result in "something similar" to a pseudo-line arrangement in the Euclidean plane. However, x-monotonicity (or, more general, fixing a "vertical" direction) is crucial for the definition of allowable sequences of arrangements of pseudo-lines. Goodman and Pollack [82] show how to establish a correspondence between the two concepts. For this, they use a result by Levi [110] (which will be used again several times throughout this thesis).

Lemma 5.3 (Levi Enlargement Lemma [110]). *Given a pseudo-line arrangement $\mathcal{A}$ in $\mathbb{P}^2$ and two points that do not both lie on the same pseudo-line of $\mathcal{A}$, there exists a pseudo-line arrangement $\mathcal{A} \cup \{\chi\}$ such that the pseudo-line χ passes through these two points.*

Suppose now we apply the Levi Enlargement Lemma to obtain a pseudo-line χ passing through the distinguished point ψ and some crossing c of $\mathcal{A}$, obtaining an extended, non-simple pseudo-line arrangement $\mathcal{A}'$. If we traverse χ starting at ψ in some direction, we meet the pseudo-lines of $\mathcal{A}$ in a fixed cyclic order. If the arrangement $\mathcal{A}'$ would consist of straight lines, then χ would represent a vertical line and therefore determine the points on each side of the primal line c^*. By continuously adding pseudo-lines through ψ and all crossings of $\mathcal{A}$, Goodman and Pollack [82] obtain a *ψ-augmentation* of $\mathcal{A}$, denoted by $\overline{\mathcal{A}}$.[11] After choosing a direction of rotation around ψ in $\overline{\mathcal{A}}$, we get a sequence of permutations of the pseudo-lines by traversing all the pseudo-lines through ψ in the chosen direction of rotation. This sequence can be shown to be an allowable sequence. Hence, by constructing the wiring diagram of the allowable sequence and projecting it to the projective plane, we get an arrangement of pseudo-lines in the projective plane isomorphic to $\mathcal{A}$.

Clearly, not only the choice of the arrangement in the projective plane, but also the choice of the marked cell containing ψ determines the abstract order type. When project-

---

[11] The original name is "P-augmentation", we use ψ instead of P to make the point more distinguishable throughout this work.

ing the arrangement to the Euclidean plane, the marked cell is split into two vertically unbounded cells. Following Felsner [66, p. 88], we call the upper one the *north face*, and the lower one the *south face*.

Suppose that $\mathcal{A}$ is isomorphic (in the projective plane) to a line arrangement (i.e., it is *stretchable*). Then we can use the straight lines to obtain a point set that realizes the abstract order type defined by $\mathcal{A}$. Using these arguments, one can obtain the following result.

Theorem 5.4 (Goodman, Pollack [82])**.** *An abstract order type defined by a pseudo-line arrangement is realizable by a configuration of points if and only if the pseudo-line arrangement is stretchable.*

As mentioned in Section 5.1.1, Goodman and Pollack [77] gave an example of an allowable sequence Σ with five elements that is not realizable by a configuration of points. However, the abstract order type defined by Σ can be realized by a point set. For the corresponding arrangement of pseudo-lines in the projective plane this means that it can be stretched, but the relative position of the crossings is different from the one in Σ for any choice of the distinguished point ψ. However, this changes for larger sets. Ringel [138] provided a simple arrangement of nine pseudo-lines in the projective plane that is not stretchable, derived using Pappus' Theorem. This is in fact the smallest such configuration [78].

Theorem 5.5 (Goodman, Pollack [78])**.** *Any arrangement of eight pseudo-lines is stretchable.*

Interestingly, it follows from the Schoenflies Theorem that two pseudo-line arrangements in $\mathbb{P}^2$ are isomorphic if and only if they are homeomorphic [82]. The intuition behind the term "stretchable" actually reflects this.

Note that we defined stretchability for pseudo-lines in the projective plane. There is a subtle difference between stretchability of pseudo-lines in the Euclidean plane and in the projective plane. Recall that we defined arrangements as cell complexes. An isomorphism for pseudo-line arrangements in the Euclidean plane does not capture the equivalence we are interested in. Krasser [107, p. 18] provides the following example. Suppose we are given a non-stretchable pseudo-line arrangement of nine lines in $\mathbb{P}^2$. We can alway stretch one pseudo-line ℓ, obtaining an isomorphic arrangement. If we take ℓ as the line at infinity and project to the Euclidean plane, we obtain an arrangement in $\mathbb{E}^2$ of eight curves that pairwise cross exactly once (i.e., a "pseudo-line arrangement" where we drop the requirement of x-monotonicity). If there would be a homeomorphism transforming this arrangement of curves to an arrangement of straight lines, we would obtain an isomorphic line arrangement of the nine lines in $\mathbb{P}^2$, a contradiction. If we remove ℓ from the arrangement in $\mathbb{P}^2$, we can stretch it to an arrangement of eight lines. When projecting these back to the Euclidean plane by again choosing ℓ as the line at infinity, we get a "different" (by the isomorphism in $\mathbb{E}^2$) arrangement of lines in $\mathbb{E}^2$. Intuitively, a continuous transformation of the arrangement of curves to the arrangement of lines in the Euclidean plane includes at least one crossing "jumping" over the line at infinity. Therefore, in contrast to, e.g., Krasser [107], we do not define stretchability in terms of an isomorphism in the Euclidean plane, which is inadequate to directly indicate when two arrangements with a marked cell define the same order type, and leads to a counterintuitive difference between stretchability in the Euclidean and the real projective plane (see [107, p. 18]).

Deciding whether an arrangement of pseudo-lines is stretchable has been shown to be NP-hard by Shor [148]. However, the problem is even more interesting under the aspect of computational complexity. Mnëv [126] showed that the problem of stretchability of simple pseudo-line arrangements is as hard as the existential theory of the reals. Schaefer [143] gave more examples of similar problems and suggested that this class of problems should be handled as a special complexity class. Further, it is known that there exist order types that can only be embedded on a grid of doubly-exponential size [85].

Consider an allowable sequence that is not realizable as a circular sequence of a configuration of points. The wiring diagram obtained from the allowable sequence may still be stretchable. However, the resulting configuration of points then will have a different circular sequence. Deciding whether an allowable sequence is realizable is closely related to the problem of stretchability, but they are not the same, as follows from the existence of a non-realizable allowable sequence with five elements in contrast to the statement of Theorem 5.5. The problem can be reduced to deciding stretchability of a pseudo-line arrangement by deciding whether the corresponding ψ-augmentation is stretchable. In the other direction, it seems that there is no obvious way to add a ψ-augmentation to Shor's construction [148] without breaking it, in order to make the reduction also applicable to show NP-hardness of realizability of allowable sequences. While intractability of the construction problem is implied by the fact that there are order types that can only be embedded on a grid of doubly-exponential size [85], the corresponding decision problem seems to have been overlooked due to its close relation to the pseudo-line stretchability problem. See also [9] for a discussion of this issue in connection with simultaneous monotone embeddings of paths.

Open Problem 1. *What is the complexity of deciding whether an allowable sequence is the circular sequence of some point set?*

For the number of realizable point sets, Goodman and Pollack [83] give an upper bound of n^{6n}. Felsner [65] gives an upper bound of $2^{0.6974 \cdot n^2}$ on the number of abstract order types by encoding each abstract order type in a binary $n \times (n-1)$ matrix. It is a curious fact that there are far more non-realizable abstract order types than realizable ones. Goodman and Pollack [81] provide a lower bound of $2^{n^2/8}$ on the number of abstract order types.

We have seen that, while pseudo-line arrangements in $\mathbb{P}^2$ that represent the same abstract order type are isomorphic, there are, in general, several different pseudo-line arrangements in $\mathbb{E}^2$ that represent the same order type (at least when defining the isomorphism in $\mathbb{E}^2$ in the same way as in $\mathbb{P}^2$). For example, consider again the wiring diagram in Figure 5.4. The crossing between pseudo-line 2 and pseudo-line 3 is the first crossing of both lines from the left. If we "untangle" the crossing and let the two pseudo-lines cross to the right of the rightmost crossing, the resulting wiring diagram represents the same order type. (In the projective plane, this would correspond to moving the pseudo-line at infinity over this crossing.) See Figure 5.8. In this way, we can always obtain a wiring diagram such that for a fixed pseudo-line h on the upper or lower envelope of the arrangement, the first $n-1$ crossings from left to right are with line h (see again Figure 5.8, where this

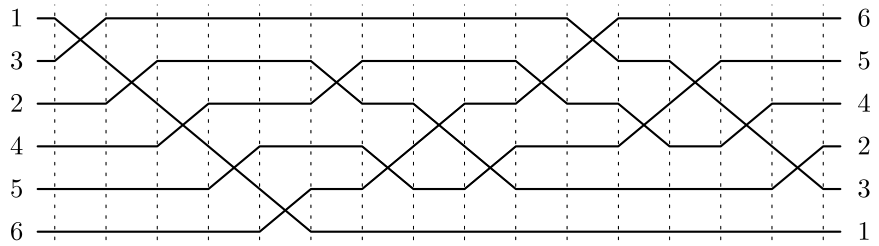


Figure 5.8: A wiring diagram representing the same abstract order type as the one in Figure 5.4 on page 64.

happens for pseudo-line 1). This representation is, e.g., conceptually used by Aichholzer, Aurenhammer, and Krasser [7] when generating all order types of up to eleven points.

5.1.3.4 Axioms for Abstract Order Types

In the (coordinatized) Euclidean plane, a sidedness query for an ordered tripe (u, v, w) can be answered by evaluating

$$\det \begin{pmatrix} u_x & u_y & 1 \\ v_x & v_y & 1 \\ w_x & w_y & 1 \end{pmatrix} > 0 \ ,$$

which indicates whether the triple is oriented counterclockwise (i.e., w is to the left of the directed line uv). The truth value of this inequality therefore gives a predicate $\nabla(u, v, w)$ that is true if and only if the triple is oriented counterclockwise. This is equivalent to the sidedness test on the point triple. Recall that we assume that all point sets are in general position, and therefore either $\nabla(u, v, w)$ or $\nabla(u, w, v)$ is true, i.e., the determinant never becomes 0. Knuth [105] extracts five propositions that are fulfilled by the predicate ∇. He makes these propositions the following five axioms of a ternary predicate P on a finite set S.

Axiom 1 (cyclic symmetry): $P(p, q, r) \Rightarrow P(r, p, q)$.

Axiom 2 (antisymmetry): $P(p, q, r) \Rightarrow \neg P(p, r, q)$.

Axiom 3 (nondegeneracy): $P(p, q, r) \vee P(p, r, q)$.

Axiom 4 (interiority): $P(t, p, q) \wedge P(t, q, r) \wedge P(t, r, p) \Rightarrow P(p, q, r)$.

Axiom 5 (transitivity): $P(p, q, r) \wedge P(p, q, s) \wedge P(p, q, t) \wedge P(p, r, s) \wedge P(p, s, t) \Rightarrow P(p, r, t)$.

Knuth [105] calls a predicate obeying these axioms a *CC system*. Hence, two point sets have the same order type if they represent the same CC system with either $P = \nabla$ or $P = \neg\nabla$. But there are further such sets. As shown by Knuth, the equivalence class defined by P and its negation is exactly the one of abstract order types. While the first three

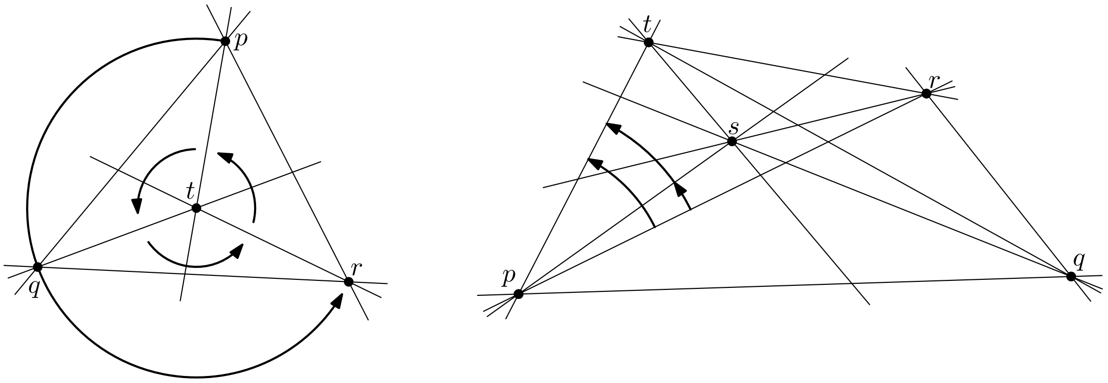


Figure 5.9: Illustrations for Axioms 4 and 5 for CC systems.

axioms capture the properties of three points in general position, Axiom 4 and Axiom 5 give properties fulfilled by four and five points, respectively. Roughly speaking, Axiom 4 states that if the counterclockwise radial order of three points around a fourth point t is circular, then the triangle formed by these three points is oriented counterclockwise. Axiom 5 formalizes that, for all points in the left halfplane defined by the directed line pq, the predicate defines a transitive order on these points around p (Knuth also gives a formal derivation of the symmetric statement for the right halfplane using only the axioms). See Figure 5.9 for an illustration of the properties captured by these two axioms. We will see in Section 5.2.3 that these axioms make it rather straightforward to show that order types appear in the related setting of shortest paths between points inside a simple polygon.

5.1.4 Generalized Configurations of Points

We have seen that there exist allowable sequences that are not realizable, and also that non-realizable abstract order types exist. Nevertheless, we have seen that pseudo-lines allow us to give a representation of both. In the figures used so far, we made use of straight strokes representing the supporting lines of pairs of points in the configuration (see, e.g., Figure 5.5 and Figure 5.9). This helped in visualizing the circular sequence and the order type. Informally speaking, we will see in this section that replacing straight supporting lines by pseudo-lines gives us the flexibility of representing all allowable sequences. Goodman and Pollack [82] give a formal definition of such generalized configurations of points, based on the transform described by Goodman [75]. (This concept was described simultaneously by Cordovil [46] in connection with oriented matroids of rank 3.) In the projective plane, a *generalized configuration of points* is a pair $(\mathcal{C}, \mathcal{A})$ of a configuration $\mathcal{C}$ of n points, a pseudo-line arrangement $\mathcal{A}$ defined by m pseudo-lines and a directed pseudo-line ℓ_∞ such that

1. each pair of points in $\mathcal{C}$ lies on some pseudo-line of $\mathcal{A}$,
2. each pseudo-line of $\mathcal{A}$ except ℓ_∞ contains at least two points of $\mathcal{C}$, and
3. the pseudo-line ℓ_∞ contains no point of $\mathcal{C}$.

The pseudo-line ℓ_∞ is the pseudo-line at infinity, the other pseudo-lines of $\mathcal{A}$ are the *supporting pseudo-lines*.[12] In particular, a configuration of points in $\mathbb{E}^2$ and the (straight) supporting lines of every pair of points defines a generalized configuration of points (with ℓ_∞ being implicitly given by $\mathbb{E}^2$). Analogously to our general position assumption for point sets in $\mathbb{E}^2$, we require, unless otherwise stated, that every supporting pseudo-line contains exactly two points of the configuration (hence, $m = \binom{n}{2}$). Observe that $\mathcal{A}$ cannot be simple for $n > 3$, and if the crossing of two pseudo-lines is on ℓ_∞, the two pseudo-lines correspond to parallel lines.

An example is given in Figure 5.10. The pseudo-line at infinity is given by the dashed Jordan curve. Similar to Perrin's setting (see Section 5.1.1), the order of the intersections of the connecting pseudo-lines with ℓ_∞ give the switches of adjacent labels in an allowable sequence. In particular, if we consider the straight supporting lines of each pair of points in a point set in $\mathbb{E}^2$, the circular sequence of this point set is the same as the allowable sequence determined by traversing the line at infinity in $\mathbb{P}^2$. Goodman and Pollack [82] prove the following result.

Theorem 5.6 (Goodman, Pollack [82])**.** *Every allowable sequence can be realized by a generalized configuration of points.*

Intuitively, generalized configurations of points allow us to draw all possible combinatorially equivalent allowable sequences as point sets by "bending" the supporting lines. Also, the orientation of each point triple is given by the orientation of the triangle spanned by three pseudo-lines. A corresponding wiring diagram is shown at the bottom of Figure 5.10. The point set is the same as the one used in previous examples. However, as can be seen from, e.g., the wiring diagram, the order type of the generalized configuration is the one of six points in convex position. Using simple topological arguments, one can see that the actual position of the points is not of any importance when drawing a generalized configuration of points. We will sometimes denote a generalized configuration of points by $\mathcal{C}$ only.

By considering again Figure 5.10, we can observe that a generalized configuration contains more information than the corresponding allowable sequence. Consider the connecting pseudo-lines of the point pairs 1 and 4, 2 and 5, as well as 3 and 6. There is a cell defined by exactly these three pseudo-lines. If we extend the configuration by a point x inside this cell, then x will be to the left of the directed connecting line of 3 and 6. Consider again the original configuration (without x). We can slightly move the pseudo-line for, say, the point pair 2 and 5, to get a new arrangement that does not contain this cell, but another one. This new configuration realizes the same allowable sequence as the original one. However, if we extend this configuration by placing a point x' in the newly created cell, then x' has to be to the right of the directed connecting line of 3 and 6. The Levi Enlargement Lemma allowed us to extend a pseudo-line arrangement by giving two points. The relative position of the new pseudo-line and the crossings in the arrangement is determined for some but not all of the crossings. Similarly, we can always extend a configuration of points by placing a point. While, algorithmically, when extending a pseudo-line arrangement in

---

[12] The term used by Goodman and Pollack [82] is "connecting pseudo-lines". Our naming is chosen to remain consistent with the remainder of this thesis.

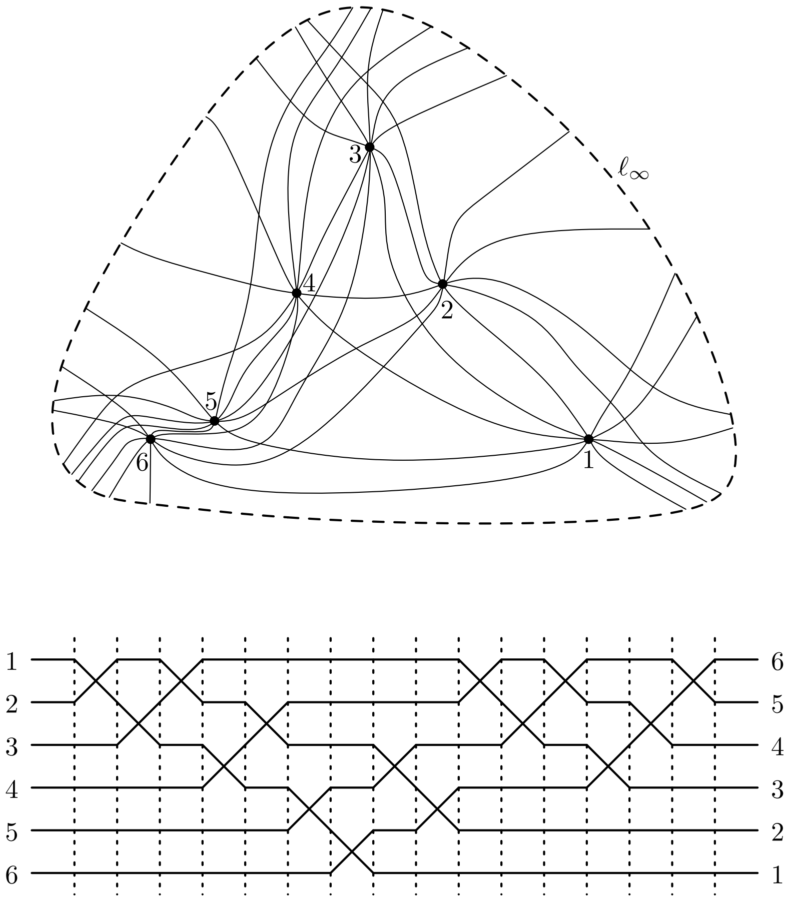


Figure 5.10: A generalized configuration of points and the corresponding wiring diagram.

the projective plane by another pseudo-line, we have some nondeterministic choices, these choices were already made when embedding the pseudo-line arrangement of the generalized configuration of points. It seems that, while these facts are well-known and implicitly used, there is no written examination of these facts and their consequences. Observe that this is also an issue for realizable configurations of points; the order type does not capture all information on the cells in the arrangement of supporting lines.

In our summary, we considered pseudo-line arrangements and generalized configurations of points separately. However, the construction by Goodman [75] works for pseudo-line arrangements and generalized configurations of points at the same time, just like a set of points and a set of lines in the primal can be transformed to a dual set of lines and a dual set of points in the same plane. Goodman's construction is in $\mathbb{P}^2$. Agarwal and Sharir [2] adapt Goodman's work by considering a dual transform of points and pseudo-lines that are given by x-monotone curves with an explicit representation.

There is an interesting aspect under which there is a non-obvious combinatorial separation between non-realizable generalized configurations of points and realizable configurations of points in connection with the crossing number of a graph. The *crossing number* of a graph is the minimum number of crossings required in any drawing of the graph (see [142] for a survey). The *rectilinear crossing number* of a graph is determined by all drawings of the graph as a geometric graph. Hence, an edge consists of the part of a supporting line between two points. This notion has been extended to generalized configurations of points by Pan [128] to the so-called *pseudo-linear crossing number*. We are therefore concerned with points being connected by curve segments in a way that the curve segments pairwise intersect at most once, and, in addition, the curve segments can be elongated to form the arrangement of the generalized configuration of points. This can be seen as an intermediate step between so-called *good drawings* of graphs (in which we do not have to be able to elongate such curve segments, see [142]), and the rectilinear case. It was conjectured that the rectilinear crossing number and the pseudo-linear crossing number of a graph are actually the same [27] (see also [142, p. 53]). Very recently, Hernández-Vélez, Leaños, and Salazar [92] showed that this is not the case; in particular, they showed how to construct, for any number m, a graph such that the rectilinear crossing number and the pseudo-linear crossing number of that graph differ by m. Hence, under this combinatorial aspect, there is a clear distinction between non-realizable generalized configurations of points and realizable configurations of points. It would be interesting to know whether there is also a gap for the complete graph, i.e., whether the rectilinear crossing number of the complete graph differs from the pseudo-linear crossing number.

5.2 Motivation

While combinatorial equivalence and the order type of point sets are definitely interesting from a combinatorial point of view, they are also relevant from an algorithmic point of view. The second part of this thesis, in particular Chapters 6 and 7, will present algorithms that use only sidedness queries, and will also work for abstract order types (i.e., the calculation of the determinant as discussed in Section 5.1.3.4 can be replaced by any predicate fulfilling the axioms). Informally, we are interested in algorithms on point sets that do not (directly) use the coordinates. In this section, we present some examples that motivate the development of algorithms that use only sidedness queries. These examples span the range between application-driven developments and the investigation of purely theoretic interest.

5.2.1 Robust Implementations

Usually, algorithms in computational geometry are described under the assumption that computations can be carried out with infinite precision (like in the real RAM model [135, p. 28]). However, when implementing an algorithm using a certain programming language, it is often overseen that the data types and the operators used by the programming language do not fulfill this. Geometric algorithms often handle entities like line intersections, for which the precision required increases rapidly, and therefore, robustness of computation is a topic that is of interest in particular for algorithms that implement geometric algorithms. Even when given rational variables as input, intersecting, say, a circle with a line may already results in algebraic numbers. (We have already discussed such issues in the hardness reduction of Section 3.3; there, however, we were only concerned with keeping the representation rational and within a polynomial bound.) To circumvent these problems, two major approaches are to provide means to ensure exact computations and to design algorithms that are able to handle inaccuracies. See [144] for further details.

When requiring exact computations, it is a straightforward goal to minimize the precision needed by the algorithm for the exact computations. Boissonnat and Snoeyink [34] take the approach of reducing the algebraic degree of predicates needed to solve a geometric problem. They investigate the following classic problem in computational geometry: given a set of n line segments in the plane (represented by their end points), report each pair in the set that crosses. Naturally, since there are instances of the problem where every pair crosses, the worst-case running time of the algorithm is in $\Theta(n^2)$. Therefore, output-sensitive algorithms have been designed whose running time depends on the number k of pairwise crossing segments. Several output-sensitive algorithms have been developed (see [34, Table 1] for a selection), where the one of Balaban [26] runs in optimal $O(n \log n + k)$ time. To measure the precision requirements of an algorithm, Boissonnat and Snoeyink use the algebraic degree of the basic predicates used by the algorithm. When testing the sign of a homogeneous multivariate polynomial whose arguments are the input variables, the *algebraic degree* of the test is the maximum degree of its polynomial factors that are irreducible over the rationals and have non-constant sign [34]. A sidedness query (see Section 5.1.3.4) has algebraic degree 3. For example, testing the x-order of intersections of line segments has algebraic degree 5. See [34, Table 2] for an extensive listing of operations and their algebraic degrees. While the trivial $\Theta(n^2)$ time algorithm that tests all segment pairs uses only operations of algebraic degree 2, Balaban's algorithm uses predicates of algebraic degree 3, as pointed out by Boissonnat and Snoeyink. In their paper, they provide an $O(n \log^2 n - k \log n)$ time algorithm (that is an adaption of the one by Balaban) using sidedness queries. Whether the additional logarithmic factor can be removed is left as an open problem.

Hence, using only sidedness queries for problems defined on order types may assist in the design of more robust implementations. For integer coordinates, sidedness queries can be performed in a quite reliable way. Avnaim et al. [24] provide a method to evaluate the sign of a 3×3 determinant with b-bit integer entries using only $(b+1)$-bit arithmetic. However, using only sidedness queries is no universal remedy for robustness problems in computational geometry. Kettner et al. [100] give an illustrative account on robustness problems, in particular in connection with testing the sign of the determinant for sidedness queries and floating point arithmetic. Still, the approach by Boissonnat and Snoeyink [34] motivates the development of algorithms using only sidedness queries.

Note that these considerations do not only apply to sidedness queries. Guibas and Stolfi [87] obtain an algorithm for constructing Delaunay triangulations by using a predicate that indicates whether a point lies inside a circle defined by a point triple; this facilitates robust implementation. We will discuss this again at the beginning of Chapter 6, as this work actually motivated Knuth's question for which we present a solution in that chapter.

Sidedness queries are also used as an interface to configurations of points when working with triangulations in any dimension. A detailed account on such algorithms can be found in the book of De Loera, Rambau, and Santos [50, Chapter 8]. Experiences with an actual implementation of these concepts are reported in [131].

5.2.2 Mechanically Proving the Correctness of Algorithms

Geometric algorithms do not only fail because of wrong assumptions on the number representation in computers. In several cases, the description of an algorithm is incorrect per se. A prominent example for a problematic task is computing the convex hull of a simple polygon in linear time. Several algorithms have been developed that later turned out to be incorrect [19]. One method proposed to increase the confidence in algorithms is formal and mechanic verification of the algorithms. Pichardie and Bertot [132] take Knuth's work on axiomatizing abstract order types [105] (see Section 5.1.3.4) as a basis to develop formal descriptions of convex hull algorithms. In their work, they use a theorem proving software to verify the correctness of two convex hull algorithms. Due to an extension of the basic axiomatic system to degenerate cases, they are able to separate the task into two distinct parts; first, it is verified that the axioms hold for implementations of the predicates, and then the main part of the algorithm can be verified based on the axioms. A similar approach, using Hoare logic, is also taken by Meikle and Fleuriot [123].

5.2.3 Geodesic Order Types

Order types occur naturally in point sets and, by duality, in line arrangements. As already mentioned, there are far more non-realizable abstract order types than realizable ones. In addition to generalized configurations of points and pseudo-line arrangements, abstract order types also have a realization in connection with geodesics in simple polygons.

A *pointgon* [17] is a pair (S, P) of a point set S and a simple polygon P with $S \subset P$ and such that S contains all vertices of P (however, S may contain additional points inside the simple polygon). Hence, for any point set S, the pair $(S, \mathrm{CH}(S))$ is also a pointgon. While for such a pointgon, the shortest path between two points is given by the straight line segment between them, the shortest path inside a pointgon with a general simple polygon is a simple polygonal chain, called a *geodesic*. Toussaint [155] defines a generalization of the convex hull of a point set for point sets $S' \subseteq S$ of a pointgon. Let P be a simple polygon and let Q be a subset of P. Then Q is *geodesically convex* if for every two points p and q in Q (not necessarily part of a finite point set), the geodesic path between p and q in P is also in Q. Observe that Q may not be a simple polygon, but may contain 1-dimensional components. Still, the whole boundary of the region can be traversed by starting and ending in the same point of the boundary such that the sum of all the angles turned is equal to 360°. Such a region is called a *weakly simple polygon* (see [155] for a more formal definition). For a set S of points in P, the *geodesic convex hull* of S in P is the intersection of all geodesically convex sets containing S. Hence, for a pointgon $(S, \mathrm{CH}(S))$, the geodesic convex hull of $S' \subseteq S$ is the convex hull of S'. Apart from the convex hull, several other entities defined for point sets have been generalized to the geodesic setting, like linear separators [51] and ham-sandwich cuts [35] (see [8] for further examples).

For points in the plane we can observe that an ordered triple (p, q, r) of points is oriented counterclockwise if and only if there exists a counterclockwise traversal of the triangle $\mathrm{CH}(\{p, q, r\})$ such that the points appear in that order. We can generalize this way of performing a sidedness query by taking the geodesic convex hull of the triple inside a simple polygon P. This leads to the definition of *geodesic order types* [11] analogous to unconstrained point sets in the plane. Observe that encapsulating a triple of points in a simple polygon might change the orientation of the triple (in the geodesic order type). Aichholzer et al. [11] show that the triples of points in a pointgon actually fulfill all five axioms given by Knuth (see Section 5.1.3.4). Therefore, every geodesic order type is an abstract order type. We already discussed in Section 5.1.3.3 that Ringel [138] gave an example of a set of non-realizable abstract order types (in terms of a non-stretchable pseudo-line arrangement) of nine elements. In [11], an example is provided showing that one of these abstract order types can actually be realized as the geodesic order type of nine points inside a simple polygon. Hence, geodesic order types are a proper superset of order types. Up to now, it is not clear whether there is a large class of non-realizable abstract order types that can be realized in the geodesic setting. It is even conceivable that for every abstract order type there is a set of points in a pointgon realizing the abstract order type (given a simple polygon with a sufficiently large number of vertices).

From an algorithmic point of view, the following result on sidedness queries is known.

Theorem 5.7 (Aichholzer et al. [8, Corollary 5]). *A simple polygon P of m vertices out of which $r > 0$ are reflex can be preprocessed in $O(m)$ time and space such that, for any three points $a, b, c \in P$, their orientation can be determined in $O(\log r)$ time.*

Hence, algorithms that work on order types in the plane using only sidedness queries can be adapted to the geodesic setting without large overhead. One example is given at the end of Chapter 7. In particular, this allows for separating the part of the algorithm that is specific to the problem from the one that handles the sidedness queries on points in the simple polygon.

5.2.4 Generation of Order Types and Abstract Extension

There is one important consequence of the combinatorial classification of point sets that we have not discussed so far. Since we have a finite number of equivalence classes, it is, in theory, possible to generate a member for each such class. Of course, from a computational point of view, this problem is highly intractable; the number of order types increases exponentially, checking whether an abstract order type is realizable is an NP-hard problem, and the realization may require an exponential-size representation. Aichholzer, Aurenhammer, and Krasser [7] managed to generate a data base of all distinct order types of up to ten points by giving for each an explicit representation as a point set with integer coordinates. The data base was later extended to eleven points [13]. The data base has successfully been used to investigate a large number of problems on point sets in the plane; see [12] for a list of applications, including, e.g., counting the number of triangulations, determining the rectilinear crossing number and determining the number of empty and non-empty convex k-gons every set of a given size must have. The applications of the order type data base can be grouped into disproving or supporting conjectures, and to provide values for small point sets to use them as a basis for more general bounds for combinatorial properties. Finschi and Fukuda [69] extend this work by considering abstract order types also in higher dimensions and allow collinear triples (or, in general, $d + 1$ points not forming a d-simplex).

For eleven points, the data base requires nearly 100 Gigabyte of storage. The increasing storage requirements as well as the calculation time are reasons for not extending the data base to larger sets. Still, for several problems it is interesting to consider small point sets of more than eleven points. For doing this, the technique of *abstract order type extension* [13] has been developed. We give a short explanation using an example given in [13]. Erdős and Szekeres [61] asked for the smallest number $g(k)$ such that each point set of size at least $g(k)$ contains a convex k-gon. Suppose we want to identify all sets of twelve points in the data base that do not have a 6-gon. We know that no such point set contains a proper subset that contains a convex 6-gon (in general, this is called the *subset property*). Hence, a naive approach would be to take all point sets of size eleven not containing a convex 6-gon and try all combinatorially different ways to add a twelfth point (i.e., by placing the point in a cell of the arrangement of all supporting lines of point pairs). However, as we have seen in Section 5.1.4, the cells in the arrangement of the supporting lines do not offer all possibilities to extend a point set, but the corresponding pseudo-line arrangement does. Having a representation of a pseudo-line arrangement as a graph, we can enumerate all different possibilities to extend it with another pseudo-line using standard algorithmic tools. Even though we may not have a coordinate representation of the resulting abstract order type, the presence of a 6-gon in the abstract order type can be tested. If it contains a 6-gon, it is filtered out, otherwise, the extension process is continued. Abstract extension has in particular been used to gain insight into the properties of the rectilinear crossing number; see [13] for details.

For abstract order type extension the main challenge is again the fast growing number of different sets to consider. Algorithmic aspects of the problem considered are of minor interest, as for relatively small instances, less sophisticated approaches may lead to faster implementations. Nevertheless, abstract order type extension imposes a field where non-realized abstract order types occur naturally that is worth being mentioned in our context.

5.2.5 Algorithmic Properties

Finally, let us emphasize that the development of algorithms that use only sidedness queries is of interest in its own right, with the goal of getting more insight into algorithmic properties of problems on point sets. Edelsbrunner considers arrangements of hyperplanes (generalizing 2-dimensional order types to higher dimensions) to be “at the very heart of computational geometry” [54, p. VII]. Using geometric artifacts in the algorithm description or as actual elements it operates on might hide the combinatorial properties the algorithm is actually using. This is apparently a motivation for Knuth’s work [105] on CC systems. (Computing the convex hull of disks by the most basic predicates possible is also given as a motivation by Habert and Pocchiola [88], who use a setting comparable to that of point set order types.) In this connection, there is of course one major intriguing question. Does there exist a problem for which an algorithm exists that, by using the realization of an order type, is asymptotically faster than any algorithm that uses only sidedness queries?

Of course, the answer to this question depends on the model of computation, or, more generally, on our “rules”. Using the real RAM model, the asymptotic running time of most order-type-related problems can be given in terms of the number of points. Using a more strict model, we would give the running time as a function of the size of the binary representation of the input. How to compare an algorithm getting the answer to each sidedness query in constant (or even polynomial) time from an oracle to an algorithm using the coordinates when, e.g., given an order type that requires a coordinate representation of exponential size (whose existence has been shown in [85]) depends on the model of computation (independent of the actual running time of an implementation on a computer). Also, one can come up with problems that, e.g., involve preprocessing, where the algorithm that is given the realization can make use of the restricted possibilities to add points to the set. Such an example can be easily obtained by slightly modifying algorithms and lower time bound proofs for the *half-space emptiness problem* (see, e.g., [64, pp.92–93] for details) to a setting where half-planes are given by pairs of points.[13]

However, in a more "fair" setting, there seems to be no known algorithmic separation of these two types of geometric algorithms. Similarly, the question for an algorithmic separation of order types and non-realizable abstract order types can be asked when limiting both algorithms to the same set of predicates, e.g., to only sidedness queries. As formulated by Erickson, "there is no known problem that can sensibly be asked about both lines and pseudolines [...], such that an efficient algorithm is known for the straight line version that doesn't also work for the pseudoline version" [64, p. 29]. In this connection, Erickson also mentions a result by Steiger and Streinu [153], that shows that any decision tree for x-sorting the crossings in a pseudo-line arrangement must have depth $\Omega(n^2 \log n)$, but the vertices for line arrangements can be sorted using a decision tree of depth $O(n^2)$. However, the x-order of the crossings is not a property of the corresponding order type.

Summing up, many classic algorithms only use sidedness queries (their authors often do not mention this fact), or can easily be modified to do so, but there are others that make explicit use of the coordinates of the given point set. In the following two chapters, we provide two non-trivial examples where, asymptotically, there is no difference in the running time between the general and the restricted setting.

5.3 Chapter Summary

In this chapter we summarized some of the basic knowledge on the vast topic of combinatorial classification of point sets. We gave definitions and properties of configurations of points, line arrangements, and their generalizations in both the Euclidean and the projective plane. Further, we gave a selection of examples of topics that motivate the research on algorithms that use only sidedness queries. The intention of this chapter was to provide a sound foundation and intuition for the concepts used in the remainder of this thesis, in particular in Chapter 6 and Chapter 7, as well as motivation and work related to the contribution presented there.

---

[13] Personal communication with Wolfgang Mulzer, 2013.

Chapter 6

Extreme Point Search in Abstract Order Types

In this chapter, we will develop our first algorithm that uses only sidedness tests. In computational geometry, constructing the convex hull is a basic building block of many algorithms. In his monograph on that topic, Knuth [105] discusses convex hull algorithms for abstract order types. An even more basic problem is selecting an extreme point. As it turns out, while this problem is trivial to solve when using the coordinate representation, it is apparently not that straightforward when using only sidedness tests.

A preliminary version of this chapter's content has been presented [124] and also a full version has been published [14].

6.1 Introduction

While sidedness queries are sufficient to define many structures on point sets, there are further, more "metric" properties of a point set that are not determined by the order type. This includes the set's Delaunay triangulation; it is straightforward to construct two sets having the same order type that have different Delaunay triangulations. Nevertheless, the problem can still be considered as being discrete. Analogously to the predicate $\nabla(p,q,r)$ that indicates whether a triple of points is oriented counterclockwise (recall Section 5.1.3.4), Guibas and Stolfi [87] separate topological from geometric aspects, using a predicate $\mathrm{InCircle}(p,q,r,s)$ that is true if and only if the triple (p,q,r) is oriented counterclockwise and the point s lies inside the circle defined by the first three points. This predicate is equivalent to

$$\det\begin{pmatrix} p_x & p_y & p_x^2+p_y^2 & 1 \\ q_x & q_y & q_x^2+q_y^2 & 1 \\ r_x & r_y & r_x^2+r_y^2 & 1 \\ s_x & s_y & s_x^2+s_y^2 & 1 \end{pmatrix} > 0.$$

Their Delaunay triangulation algorithm depends almost entirely on this predicate, making it a robust approach, that is intended to be easy to implement and to prove. Knuth's axiomatic approach (see Section 5.1.3.4) was motivated by the work of Guibas and Stolfi [105, p. v].

The concept of the convex hull of a point set generalizes to all CC systems. (This follows from the mentioned correspondence of CC systems with abstract order types [105, pp. 29–31]; Knuth also gives an independent proof [105, p. 45]).[1] Knuth also extends his axiomatic approach to cover Delaunay triangulations. In the axiomatic settings, Knuth provides $O(n \log n)$ time algorithms for constructing the convex hull and the Delaunay triangulation, where the time bound for the latter holds in the expected case. He points out that the algorithm of Guibas and Stolfi uses the coordinate representation to find a line that partitions the point set into two equally-sized subsets (cf. [87, pp. 110–111]). Open Problem 1 in [105, pp. 97–98] therefore asks for an algorithm to find such a partition of a CC system in linear time. The problem is straightforward when given an extreme point of the set (i.e., an element of the convex hull boundary). Proving the existence of a linear-time algorithm for finding a single extreme point is also explicitly part of Open Problem 1.

In [14], both parts of the open problem are answered in the affirmative.

- There exists an $O(n)$ time algorithm that, given a point c of a set S of size n, finds a halving edge through c (more specifically, it finds a second point $c' \in S$ such that not more than $\lceil \frac{n-2}{2} \rceil$ points are on each side of the supporting line of c and c').

- There exists an $O(n)$ time algorithm that, given two points p and q, finds the edge of the convex hull that is crossed by the ray from p through q.

Both algorithms also work for general CC systems (i.e., abstract order types). In this chapter, we present the second algorithm. We first show that the algorithm runs in $O(n)$ time for realizable sets. We then show that the time bound is also correct for non-realizable sets, that is, for all abstract order types.

While an arbitrary halving edge can easily be found by picking a point with median, say, x-coordinate (using the linear-time selection algorithm by Blum et al. [33]), the problem is more sophisticated when the halving line should pass through a predefined point. For example, the linear time ham-sandwich cut algorithm of Lo, Matoušek, and Steiger [112] can be adapted to find a halving line through a point. The straightforward way of finding an extreme point of a set given by coordinates is selecting the one with, say, lowest x-coordinate. Finding a convex hull edge that is traversed by a given line in linear time is a subroutine of the so-called Ultimate Convex Hull Algorithm of Kirkpatrick and Seidel [103]. There, the median of the slopes of an arbitrary matching of the points is used for the prune-and-search approach. An extreme point is actually searched in every iteration using the coordinates.

We first give the algorithm for realizable point sets using some reasoning on the actual realization of the point set in Section 6.2, and then show that the algorithm works, without modifications, for all abstract order types (Section 6.3).

---

[1] Knuth actually defines the convex hull as a cycle.

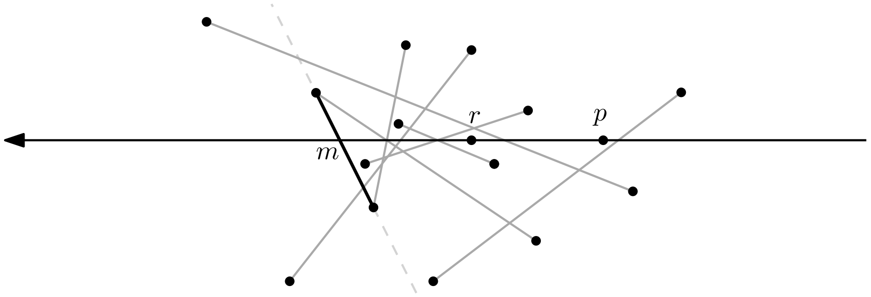


Figure 6.1: An arbitrary matching of edges that partition S, and m, the result of BasicMin.

6.2 Realizable Sets

As a first step we describe an algorithm called BasicMin, which plays a crucial role as a subroutine. Let S be a point set in the plane and suppose we are given two points $p, r \in S$. We assume that pr is a halving edge of S and that $n = |S|$ is even. Without loss of generality, let r be the coordinate origin and let p be on the positive part of the x-axis. Let M be an arbitrary perfect matching between the points above and below the x-axis, i.e., for any edge $s = ab \in M$ we have $\nabla(p, r, a) \neq \nabla(p, r, b)$. See Figure 6.1 for an illustration.

Let $\ltimes$ be the binary operator that accepts two edges $s, s' \in M$ as input and returns the edge on the convex hull boundary of $s \cup s'$ that crosses the x-axis at the smallest x-coordinate, i.e., the pair of endpoints whose upward-directed supporting line has all other points of s and s' to the right. The relevant property of the operator is that the crossing of $\ltimes(s, s')$ with the x-axis is not to the right of the crossings of s and s' with the x-axis.

BasicMin takes a point set S and two points p and r as input, partitions S arbitrarily into the matching $M = \{s_1, \ldots, s_{\left(\frac{n-2}{2}\right)}\}$, and computes a special edge $m = m_{\left(\frac{n-2}{2}\right)}$ iteratively via

$$\begin{aligned} m_1 &= s_1 \\ m_{(i+1)} &= \ltimes(m_i, s_{(i+1)}) \end{aligned}$$

Obviously, the running time of BasicMin is linear in n. Note that m does not need to be on the convex hull of the whole set. Also, m may depend on the (undefined) order in which the elements of M are processed by BasicMin. However, m has the following useful property.

Lemma 6.1. *Let ℓ be a line directed upwards, crossing the x-axis to the left of the crossing of m with the x-axis. Then at least $\frac{n-2}{2}$ points of S are to the right of ℓ.*

Proof. Note that the crossing of ℓ with the x-axis is further to the left than any other such crossing of M. Assume, for the purpose of contradiction, that there is an edge in M for which both points lie to the left of ℓ. Then also the crossing with the x-axis is to the left of the crossing of m, a contradiction. Hence, for each of the $\frac{n-2}{2}$ edges in M, at least one endpoint lies to the right of ℓ. □

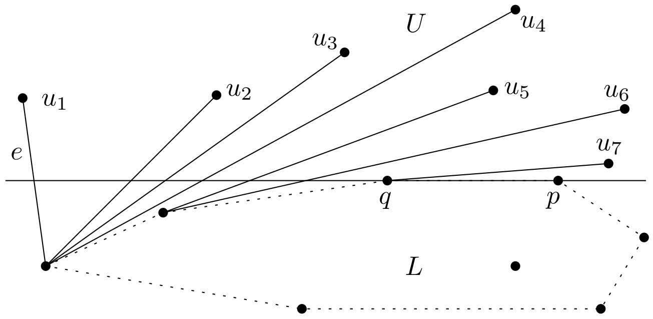


Figure 6.2: The order on U defined by removing vertices of the intersected convex hull edge.

Note that, even though the argumentation of Lemma 6.1 involves relative positions of crossings, the constant-time operation $\ltimes$ can be expressed by using only ∇. Now we are ready to give the main algorithm.

Theorem 6.2. *Given two points p, q of a point set $S \subset \mathbb{E}^2$ of size n in general position, one can find the edge e of the set's convex hull that passes through the ray pq in $O(n)$ time using only sidedness queries.*

Proof. Without loss of generality, let pq be horizontal with q to the left of p. Note first that the case where q is a vertex of the convex hull can be identified in linear time. We therefore concentrate on the setting where one endpoint of e is below pq and the other one is above. Let U be the set strictly above pq, whereas L is the set strictly below pq. Without loss of generality, let $|U| \geq \frac{n-2}{2}$.

Consider the endpoint $u_1 \in U$ of e. If we remove u_1, the ray pq intersects the boundary of the (new) convex hull at a new edge e' with an endpoint $u_2 \in U$. Note that the other endpoint of e' might now be q. In any case, iteratively removing points from U of the intersected convex hull edge induces an order on U (see Figure 6.2). Note that this corresponds to the order in which the points of U are traversed by a line t that supports the boundary of $\mathrm{CH}(L \cup \{p, q\})$ in one connected component and that is rotated clockwise around that hull, starting at e and ending at pq. The main observation is that in the search for u_1, given a point u_i and the line t passing through u_i, we can discard all points of U to the right of u_i with respect to the point l where t touches $\mathrm{CH}(L \cup \{p, q\})$, since none of these points can be u_1. The support $l \in L \cup \{q\}$ of t can be found in linear time, since the radial order of $L \cup \{p, q\}$ around any u_i is linear.

Note that these observations already imply the following randomized approach. Select any element u_i of U at random. The other support l of the line t (recall that this might as well be q) can be found in linear time. We discard the points of U to the right of lu_i and iterate. However, consecutive "bad" choices of u_i result in overall quadratic worst-case behavior. We therefore have to make a "good" choice of u_i in order to discard a linear number of points per iteration.

The points of U are ordered linearly around p. Let u_r be the median of this order, which we select in linear time. Let M be an arbitrary perfect matching between the points of U to the left and to the right of pu_r (maybe omitting one point), see Figure 6.3. Now we

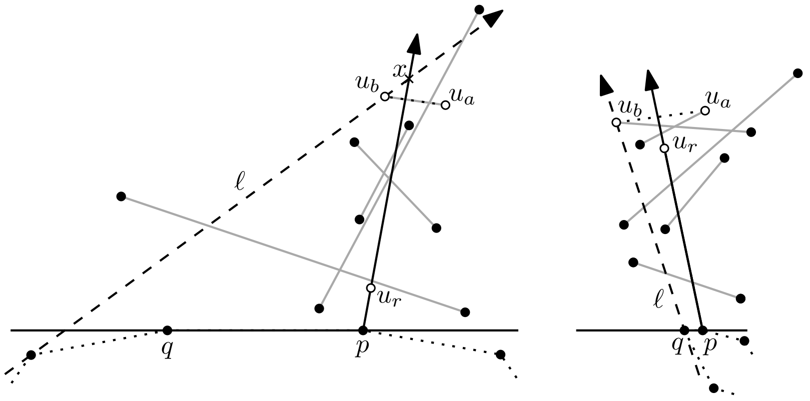


Figure 6.3: The edge $u_a u_b$ allows to prune half of the upper points.

apply BasicMin on M with $r = u_r$, which results in an edge $m = u_a u_b$. By construction, all edges of M as well as $u_a u_b$ cross the ray pu_r. Now, find the lines t_a and t_b that support the boundary of $\mathrm{CH}(L \cup \{p, q\})$ in one connected component and that pass through u_a and u_b, respectively (we consider them being directed upwards). Let $\ell = t_a$ if u_b is to the right of t_a, otherwise let $\ell = t_b$. Note that p is to the right of ℓ since q is included in the set in which we search for the second support of ℓ (see the example to the right in Figure 6.3). There are two cases to consider. If ℓ crosses the ray pu_r at a point x, then the crossing of $u_a u_b$ is on the ray between p and x (by definition of ℓ). Due to Lemma 6.1, at least half of the points of M are to the right of ℓ. Otherwise, if ℓ does not cross the ray pu_r, then all points of U to the right of the ray are also to the right of ℓ. In both cases, we can discard at least half of the points, which is at least a quarter of the overall set S (recall that U was, without loss of generality, larger than L; in each iteration, the process is applied to the larger of the two sets). We can therefore in linear time reduce this problem to constant size such that it then can be solved by a brute-force approach. □

Note that the transitivity of the order on U directly follows from the definition of the convex hull and carries over to general CC systems. Also, the transitivity of $L \cup \{p, q\}$ around any point of U holds for CC systems due to Axiom 5. This already implies that the algorithm is correct for any CC system. However, the linear time bound depends on the number of points that are to the right of ℓ. In order to show that the bound also holds for abstract order types, we need to prove that BasicMin also works as expected on non-realizable sets, and that then also ℓ has at least half of the points of U to the right.

6.3 A General Proof of the Time Bound

To see why care has to be taken when we deal with non-realizable order types, note that in general the order type does not capture the relative position of the supporting lines of point pairs and the crossings of these lines. However, abstract order types capture some of the information that is related to crossings of supporting lines, which allows us to show that the properties needed for BasicMin are also present in the abstract setting.[2] We use the dual representation of abstract order types by arrangements of pseudo-lines in the Euclidean plane (this allows us to use the obvious meaning of terms like "above" and "below" when describing an arrangement $\mathcal{A}$).

Let us recall the problem setting. We are given a set S of n elements (which we call points, even though S might not be a realizable point set), containing two special points p and q. The set S is separated by pq into a set U to the right of pq, and a set L to the left of it (where "left" and "right" are indicated by a predicate P, recall Section 5.1.3.4). We want to obtain the pair $l_1u_1, l_1 \in L, u_1 \in U$ that is consecutive on the convex hull of S, where p is to the right of l_1u_1. This is done by obtaining a pair $\ell = l_ju_i$ such that at least half of the points of U (minus a constant) are to the right of ℓ and no point of L is to the left of ℓ.

Consider the dual pseudo-line arrangement $\mathcal{A}$ representing the abstract order type of S. Keep in mind that p is an extreme point of the set $U \cup \{p\}$, and that $r = u_r$ is the median of the points in U ordered radially around p. We can represent the arrangement such that the crossing of the pseudo-lines p and r (which corresponds to the supporting line of pr in the primal[3]) is the leftmost crossing in $\mathcal{A}$ and p starts above r. The linear order of the points around p in the primal splits the set $U \setminus \{u_r\}$ into *left* and *right* points, separated by pr. In the dual, the right pseudo-lines are above the crossing pr, and the left pseudo-lines are below it. Recall the description of BasicMin. M is an arbitrary perfect matching between the left and the right points. The operator $\ltimes$ accepts two point pairs, each pair consisting of a left and a right point. The output of the operator is a pair z consisting of a left and a right point such that all other points are to the right of the oriented line through these points. This pair z is well-defined in the abstract setting as well (and there is always a geometric representation due to Theorem 5.5). Recall that we compute a special pair $m = m_{\binom{n-2}{2}}$ iteratively via

$$\begin{aligned} m_1 &= s_1 \\ m_{(i+1)} &= \ltimes(m_i, s_{(i+1)}) \end{aligned}$$

The crucial property of the pair $m = u_au_b$ (and later the line ℓ through u_a or u_b) is that at least one endpoint of each pair $s_i \in M$ lies on the same side of m as the pivot p. (We assume that m and the elements of M are directed from the left point to the right point and hence p is to the right of m.) In the dual, each element of M is represented as the crossing of a right and a left pseudo-line in $\mathcal{A}$ (this corresponds to the supporting line of the matched pair of points in the primal). The crucial property for m in the dual therefore is to have at least one pseudo-line of each pair $s_i \in M$ passing below its crossing in $\mathcal{A}$. For realizable point sets, we were able to argue for the correctness of BasicMin using the intersection χ of a matched pair s with the supporting line of p and r. In the dual, this intersection χ corresponds again to a pseudo-line that can be added to $\mathcal{A}$; this pseudo-line

---

[2] This observation is due to an anonymous referee of [14], a previous version proved these properties using a lengthy case distinction.

[3] For simplicity, from now on we use the same name in the primal and the dual, for example we denote the dual a^* of a point a by a as well, i.e., omitting the star indicating the transform. In the following, we will less often change between the primal and the dual, and hence it should be clear from the context which representation is used. Hence, e.g., a line pr becomes the crossing pr in the dual.

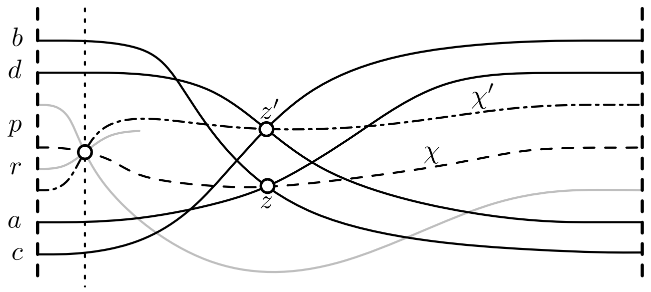


Figure 6.4: Observation 6.3: A pseudo-line χ' (not part of the arrangement) witnesses that no element is to the left of z' in the primal.

passes through the crossing pr and s. Also for non-realizable sets, such a pseudo-line χ exists due to the Levi Enlargement Lemma (stated as Lemma 5.3 herein). In fact, in non-realizable settings the intersections behave in the same transitive manner as in the realizable setting; see Figure 6.4 for an illustration of the following statement.

Observation 6.3. *Let $z = (a, b)$ and $z' = (c, d)$ be two pairs such that the point p is to the right of both pairs (i.e., in the dual the pseudo-line p is below the crossings z and z'). Let χ (χ') be a dual pseudo-line through the crossings pr and z (z'). If none of a and b is to the left of the primal line z', then the dual pseudo-line χ' is above χ in the part of the dual arrangement that is to the right of the dual crossing pr. This also holds if $a = c$ or $b = d$.*

Hence, after applying BasicMin to the matching M, we obtain in the dual a pseudo-line crossing m on a pseudo-line χ_m that passes through the crossing pr, such that no crossing of M is above χ_m. Suppose, for the sake of contradiction, that there is a pair (a, b) such that both dual pseudo-lines a and b pass above the crossing m, but the crossing ab is below χ_m. If the crossing ab is to the left of m in the arrangement (recall that our pseudo-lines are x-monotone curves), then, when traversing a from left to right, one would have to pass below χ_m and then go above it again before m. Otherwise, if the crossing ab is to the right of m, then b would have to intersect χ_m to be above it at m and then has to be below χ_m again to reach the crossing ab. Both cases contradict the fact that a pair of pseudo-lines intersects exactly once in the x-monotone arrangement. Hence, half of the pseudo-lines are below the pseudo-line crossing m in $\mathcal{A}$. This corresponds to at least half of the points being on the same side of m as p.

Recall that in the proof of Theorem 6.2, we chose a directed line ℓ passing through a point $l \in L \cup \{q\}$ and either u_a or u_b such that no elements of $L \cup \{p, q, u_a, u_b\}$ are to the left of it. We now proceed to show that at least one point of each pair (v, w) in M is to the right of the directed line ℓ, as demanded in the proof of Theorem 6.2. The points involved are $\{p, q, r, l, u_a, u_b, v, w\}$, where l and q might be the same point. Since these are at most eight points, we are allowed to use geometric arguments due to Theorem 5.5. However, we must not rely on the positions of the crossing points on the ray pr. Suppose, for the sake of contradiction, that neither of v and w is to the right of ℓ. At least one of v or w has to be to the right of $u_a u_b$. The line ℓ separates v and w from the remaining subset. Further, v and w are separated by pr. These observations imply that vw is an edge of the convex hull of $\{p, u_a, u_b, v, w\}$. However, this means that the crossing of the pseudo-lines v and w is above u_a and u_b in $\mathcal{A}$, which contradicts the fact that $u_a u_b$ is m. Thus, we conclude

Theorem 6.4. *Theorem 6.2 also holds for non-realizable CC systems, i.e., abstract order types.*

6.4 Chapter Summary

We presented an algorithms that only uses sidedness queries on point sets in the plane. A convex hull edge crossing a specified ray can be found in linear time, without being given the coordinate representation. We showed that the algorithm also works for general CC systems (i.e., abstract order types), and thus answer a long-standing open problem of Knuth [105] in the affirmative.

Note that the parts of the so-called Ultimate Convex Hull Algorithm by Kirkpatrick and Seidel [103] that depend on coordinates are essentially the ones in which the convex hull edge on the ray that separates a subproblem into two parts is found. Also, Chan's output-sensitive algorithm [41] can be implemented in our setting using Theorem 6.2. Both allow to improve the time bound given in [105] for realizable point sets[4] regarding output-sensitivity to $O(n \log h)$ for h extreme points.

The proof of Theorem 6.2 is based on the order defined on the set U above and the set L below the given ray pq. While the order on, say, U changes when removing elements from L in the defined way, the first element u_1 remains always the same. The problem solved in the next chapter will turn out to be almost a generalization of this problem; we are essentially looking for the point u_i for a given i. While looking for an extremal element allowed for a relatively simple way to deterministically pick an element with a small index for pruning, we will have to apply a more sophisticated method for this generalization. Interestingly, the algorithm of the generalization will use the one presented here as a subroutine.

---

[4] We omit a detailed analysis of these two algorithms in the abstract setting.

Chapter 7

Pseudo-Verticals and Ham-Sandwich Cuts for Abstract Order Types

Many algorithms in computational geometry are concerned with finding partitions of point sets by halving lines such that the partitions fulfill certain properties. A prominent example is finding the ham-sandwich cut of a bi-chromatic point set. A pattern frequently used by linear-time algorithms of that type is to partition the dual line arrangement of the point set by vertical lines and find the median of the crossings of (a subset of) the lines with each vertical line. This can easily be done by using standard techniques when the point set is given by the coordinates. In this chapter we consider the setting where only the order type of the point set is known, i.e., we are only allowed to use sidedness queries. We give a deterministic linear-time algorithm for the mentioned sub-algorithm and show that this is sufficient to give a linear-time algorithm for constructing a ham-sandwich cut even in our restricted setting. We also show that our methods are applicable to abstract order types. This chapter presents joint work with Stefan Felsner [67].

7.1 Introduction

A considerable fraction of problems in computational geometry deals with partitioning finite sets of points by hyperplanes while imposing constraints on both the subsets of the partition as well as on the hyperplanes. In the plane, this class of problems contains finding, e.g., a ham-sandwich cut of a bi-chromatic point set [112], a four-way partitioning by orthogonal lines, and a six-way partitioning by three concurrent lines [140], as well as finding three concurrent halving lines that pairwise span an angle of 60° [62].

Given a pair (a, b) of points of a bi-chromatic point set S of n points that are either red or blue, the supporting line of a and b is a ham-sandwich cut if not more than half of the red and half of the blue points are on either side of ab. This can be verified by using only sidedness queries (implying a brute-force algorithm running in $\Theta(n^3)$ time). Megiddo [121] presented a linear-time algorithm for the case in which the points of one color are separable from the points of the other color by a line. Edelsbrunner and Waupotitsch [59] gave an $O(n \log(\min\{n_\mathrm{r}, n_\mathrm{b}\}))$ time algorithm for the general case, with n_r red and n_b blue points. Eventually, a linear-time algorithm was provided by Lo, Matoušek, and Steiger [112] for

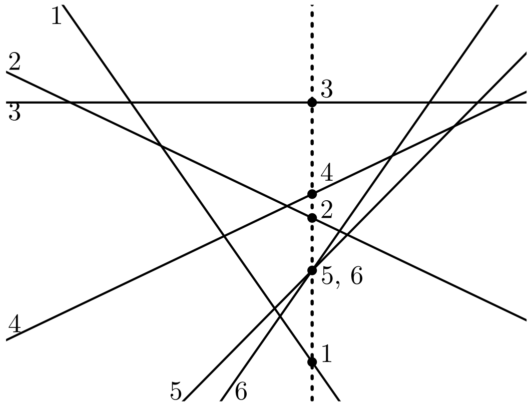


Figure 7.1: Ordering the crossings along a vertical line.

the general setting, which also generalizes to arbitrary dimensions. As already mentioned in Section 5.2.3, Bose et al. [35] generalize ham-sandwich cuts to pointgons, obtaining a randomized $O((n+m)\log r)$ time, where m is the number of vertices of the polygon, of which r are reflex.

The linear-time ham-sandwich cut algorithm works on the dual line arrangement of the point set and has to solve the following sub-problem.

Problem 4. *Given a dual line arrangement $\mathcal{A}$ in the plane and two lines p and q of that arrangement, let v be the vertical line passing through the crossing of p and q. For a subset B of the lines in $\mathcal{A}$ and an integer $k \leq |B|$, find a line $m \in B$ such that the y-coordinate of the point $v \cap m$ is of rank k in the sequence of y-coordinates of the finite point set $v \cap \bigcup_{b \in B} b$.*

This problem can be solved in linear time by directly applying the linear-time selection algorithm [33] to the y-coordinates of the intersections of all lines in B with v. Clearly, the order of the intersections of lines with a vertical line at a crossing is not a property of the order type represented by the arrangement (recall the example given in Figure 5.6 on page 67). The order type only determines the set of lines above and below a crossing. In particular, for abstract order types (or for order types not given by a realization), there are two liabilities of the formulation of Problem 4. First, the vertical direction is not determined by the order type (this is a property of the circular sequence of the point set); even though we can represent the abstract order type by a pseudo-line arrangement in the Euclidean plane (where there is a vertical direction), there is, in general, an exponential number of different ways to draw a wiring diagram representing the abstract order type, each giving a different order of the pseudo-lines along the vertical line through a crossing. Second, even when given such a vertical line, directly applying the linear-time selection algorithm relies on comparing the relative order of any two intersections with the vertical line in constant time.

In this chapter, we show how to overcome these two problems by defining a “vertical” pseudo-line through each crossing in a pseudo-line arrangement and show how a pseudo-line of a given rank in the order defined by such a “vertical” pseudo-line can be selected. We give the definition in Section 7.2, where we also examine important properties of our construction. Our result will be presented in terms of a dual pseudo-line arrangement in the Euclidean plane. However, in our model we are not given an explicit representation but are only allowed sidedness queries. In Section 7.3, we first explain how several queries about a (not explicitly given) pseudo-line arrangement can be mapped algorithmically to sidedness queries, and then give a linear-time algorithm for selecting a pseudo-line with a given rank. We then give an application of our result to replace the vertical lines in the linear-time ham-sandwich cut algorithm by Lo, Matoušek, and Steiger [112] in Section 7.4, showing that the algorithm, with the proposed modification, also works for abstract order types.

Throughout this chapter, let $\mathcal{A}$ be a simple arrangement of n pseudo-lines in the Euclidean plane. The *k-level* of $\mathcal{A}$ is the set of all points that lie on a pseudo-line of $\mathcal{A}$ and have exactly $k-1$ pseudo-lines strictly above them. The level of a crossing pq is denoted by $\mathrm{lv}(pq)$ (i.e., pq is separated from the north face by $\mathrm{lv}(pq)-1$ pseudo-lines). The *upper envelope* of an arrangement is its 1-level, i.e., the union of the segments of pseudo-lines that are incident to the north face.

7.2 Levels at a Crossing

It will be convenient to consider all pseudo-lines being directed towards positive x-direction. Let p and q be two pseudo-lines in $\mathcal{A}$ and let p start above q. We denote the latter by $p \prec q$. In the following, we define a pseudo-line that replaces a vertical line through a crossing in our abstract setting.

For a crossing pq with $p \prec q$ let γ_{pq} be a curve described by the following local properties. Initially, γ_{pq} passes through the crossing pq and enters the cell C directly above pq; see Figure 7.2 (a). Above p and q, γ_{pq} follows p against its direction in C until a pseudo-line r of $\mathcal{A}$ crosses p (in C). If r crosses p from below (i.e., $p \prec r$), γ_{pq} crosses r and continues following p in the new cell C'. Otherwise, if r crosses p from above, γ_{pq} follows r in its opposite direction, remaining inside C. In general, every time the pseudo-line a_i currently followed by γ_{pq} is crossed by a pseudo-line a_j from below, γ_{pq} also crosses a_j, enters the new cell directly above $a_i a_j$ and continues following a_i; see Figure 7.2 (b). If a_i is crossed by a_j from above, γ_{pq} now follows a_j against its direction, but remains in the same cell of the arrangement; see Figure 7.2 (c). This is continued until all crossings of A are to the right of the current position along γ_{pq}. At that point, γ_{pq} continues vertically in positive y-direction to infinity (intuitively, γ_{pq} follows the line at infinity); see Figure 7.3 (a). Below p and q, γ_{pq} continues in a similar manner. It follows p along its direction in the cell directly below pq. If the currently followed pseudo-line a_i is crossed by a pseudo-line a_j from below, then γ_{pq} crosses a_j, entering the cell directly below the crossing $a_i a_j$; see Figure 7.2 (d). Otherwise, γ_{pq} continues along a_j, remaining in the same cell of $\mathcal{A}$; see Figure 7.2 (e). If there are no more crossings to the right of the current position on γ_{pq}, γ_{pq} continues vertically in negative y-direction to infinity; see Figure 7.3 (b).

See Figure 7.4 and Figure 7.5 for two overall examples. We call γ_{pq} a *pseudo-vertical* and, in the following, identify several properties of such a curve. Note that, while we used the (rather informal) notion of "following" a pseudo-line, γ_{pq} is actually defined by the cells

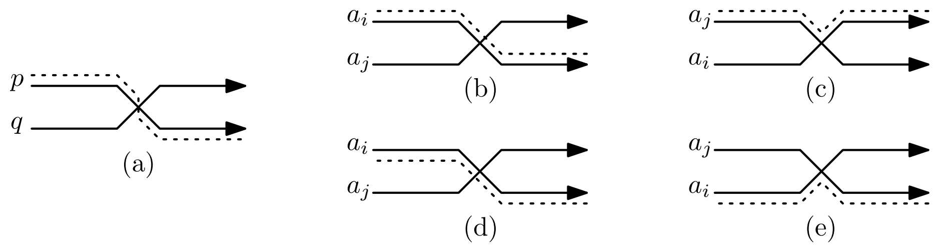


Figure 7.2: Local definition of a pseudo-vertical γ_{pq}. The curve γ_{pq} (dotted) passes through the crossing pq (a). It follows a pseudo-line a_i above (b, c) or below (d, e) the crossing pq. A pseudo-line a_j crossing a_i from below is crossed by γ_{pq} (b, d), if a_j crosses a_i from above, γ_{pq} continues in the same cell along a_j against (c) or along (e) its direction.

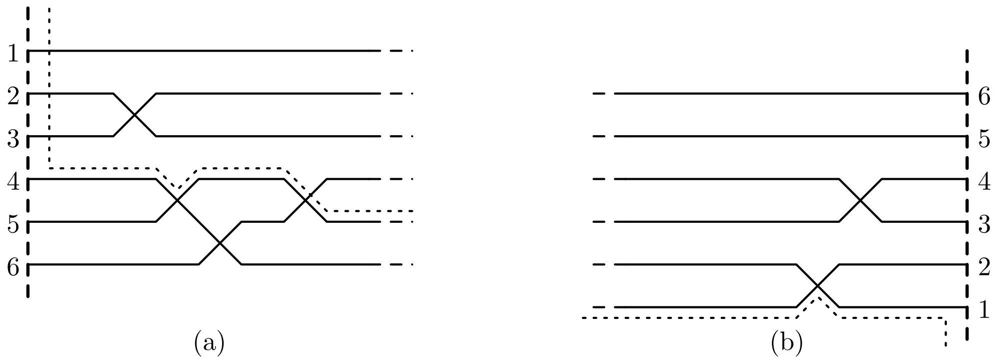


Figure 7.3: The first (a) and the last (b) pseudo-lines of an arrangement defining γ_{pq} (dotted). The conceptual line at infinity is denoted by the bold vertical line.

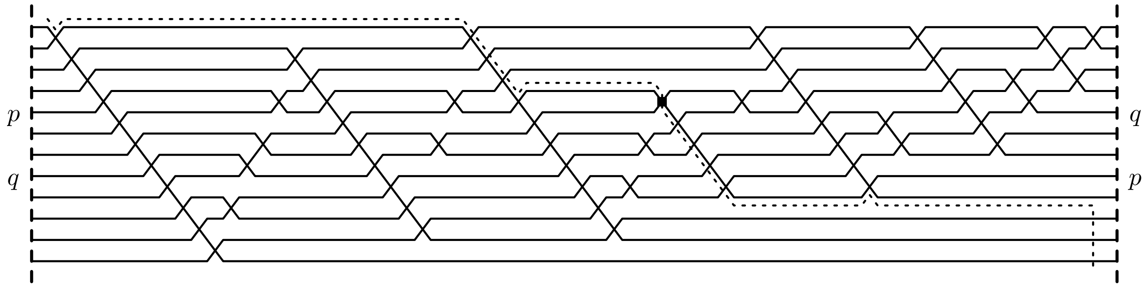


Figure 7.4: A pseudo-vertical γ_{pq} in a pseudo-line arrangement.

it traverses (i.e., two paths in the dual graph of the cell complex starting at the cells above and below pq).

7.2.1 Properties of a Pseudo-Vertical

As γ_{pq} always follows a pseudo-line of $\mathcal{A}$ or continues in a vertical direction, we can observe the following.

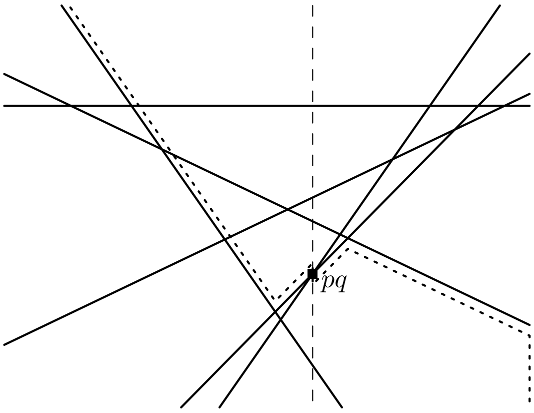


Figure 7.5: A pseudo-vertical γ_{pq} in an arrangement of straight lines.

Observation 7.1. *For any crossing pq in a pseudo-line arrangement $\mathcal{A}$, the curve γ_{pq} is x-monotone.*

The following observation can easily be made by visualizing the arrangement as a wiring diagram.

Observation 7.2. *Consider the number of pseudo-lines above a point moving along γ_{pq} in positive x-direction. At every crossing of γ_{pq} with a pseudo-line of $\mathcal{A}$, this number increases.*

Lemma 7.3. *For any crossing pq in a pseudo-line arrangement $\mathcal{A}$, the curve γ_{pq} is a pseudo-line such that $\mathcal{A}$ can be extended by γ_{pq} to a new pseudo-line arrangement.*

Proof. Let n be the number of pseudo-lines in $\mathcal{A}$. Since γ_{pq} continues to vertical infinity in both positive and negative y-direction, it crosses every pseudo-line of $\mathcal{A}$ at least once. From Observation 7.2, it follows that γ_{pq} crosses at most n pseudo-lines. As γ_{pq} is an x-monotone curve that crosses each pseudo-line of $\mathcal{A}$ exactly once, an extension of $\mathcal{A}$ is again a (non-simple) pseudo-line arrangement. ☐

Our definition of a pseudo-vertical γ_{pq} is symmetric above and below pq. It will be useful to distinguish between the two parts of the pseudo-line. We call the part of γ_{pq} reaching pq through the north face the *northbound ray*, and the part starting at pq towards the south face the *southbound ray*. Nevertheless, γ_{pq} is, like any pseudo-line of the arrangement, considered to be directed in positive x-direction.

Just like for a vertical line in a line arrangement, a pseudo-vertical defines a total order on the pseudo-lines of $\mathcal{A}$ by the order it crosses them. We denote the rank of a pseudo-line $m \in \mathcal{A}$ in this order by $\mathrm{rk}_{pq}(m)$. The following lemma shows how we can identify the rank of an element by the pseudo-lines above crossings in $\mathcal{A}$. Let $L(pq)$ be the subset of pseudo-lines in $\mathcal{A}$ such that each $a \in L(pq)$ is below pq and $a \prec p$.

Lemma 7.4. *Traversing the northbound part of γ_{pq} from the crossing pq in negative x-direction until reaching an unbounded cell for the first time, we visit only cells directly above the upper envelope of the sub-arrangement defined by $L(pq) \cup \{p\}$.*

Proof. The proof is by induction on the sequence $\langle p = a_1, a_2, \ldots\rangle$ of pseudo-lines that we follow. Clearly, the point pq is on the upper envelope of $L(pq) \cup \{p\}$. Suppose we traverse γ_{pq} in negative x-direction, following a pseudo-line $a_i \in L(pq) \cup \{p\}$. If γ_{pq} crosses a pseudo-line r (i.e., r crosses a_i from below), then r cannot be below pq as it would have to cross γ_{pq} again. If a pseudo-line a_{i+1} crosses a_i from above, then a_{i+1} cannot be above pq as it would have to cross a_i again. Further, γ_{pq} continues on a_{i+1}, keeping the invariant that no element of $L(pq)$ is above the point traversing γ_{pq}. ☐

Intuitively, the northbound part of γ_{pq} can be considered as being arbitrarily close to the upper envelope of (the sub-arrangement defined by) $L(pq) \cup \{p\}$, and hence, every pseudo-line that passes through the upper envelope (from below) will cross γ_{pq} immediately after that crossing.

Corollary 7.5. *Let m be a pseudo-line in $\mathcal{A}$ that is above pq and for which there exists a pseudo-line $a \in L(pq) \cup \{p\}$ such that $a \prec m$. Then the rank $\mathrm{rk}_{pq}(m)$ of m along γ_{pq} is given by $\mathrm{lv}(a'm)$ for some $a' \in L(pq) \cup \{p\}$ with $a' \prec m$ and $a'm$ being on the upper envelope of the sub-arrangement defined by $L(pq) \cup \{p\}$.*

If m does not intersect the upper envelope of these pseudo-lines, it crosses q before crossing any of the pseudo-lines of $L(pq)$. Therefore, we observe the following.

Observation 7.6. *If a pseudo-line m starts above every pseudo-line in $L(pq)$, then the rank of m along γ_{pq} is given by the number of pseudo-lines starting above m increased by 1, i.e., $|\{a \in \mathcal{A} : a \prec m\}| + 1$.*

7.2.2 Ordering Pseudo-Verticals

Given two different crossings pq and rs in $\mathcal{A}$, it is easy to see that γ_{pq} and γ_{rs} may follow the same part of a pseudo-line. Nevertheless, one can show that γ_{pq} and γ_{rs} will never intersect when drawn appropriately. See Figure 7.6 for an illustration accompanying the proof of the following lemma.

Lemma 7.7. *The set of pseudo-verticals for all crossings of a pseudo-line arrangement $\mathcal{A}$ can be drawn such that no two pseudo-verticals intersect.*

Proof. Recall that the pseudo-verticals are fully defined by the sequence of cells they traverse. Further, recall that each bounded cell has a unique leftmost and rightmost crossing. For two pseudo-verticals to intersect, they have to enter a common cell C.

Suppose first that C is bounded. Observe that, when traversing, say, γ_{pq} in positive x-direction, then γ_{pq} enters C from above. Let C be between the levels k and $(k+1)$. If the current part of γ_{pq} is northbound, it enters C at the k-level through the pseudo-line defining the leftmost crossing of C (in $\mathcal{A}$). If it is southbound, it leaves C at the $(k+1)$-level through the pseudo-line defining the rightmost crossing of C. For two pseudo-verticals to cross inside C, they would have to enter and leave C through four different pseudo-lines (otherwise, we could draw them without crossing in C, probably changing their relative order in the next cell). But this can only happen when one pseudo-vertical is northbound and the other is southbound in that cell (as otherwise they would either enter or leave C through the same pseudo-line), and in that case, there cannot be a crossing inside C, as the

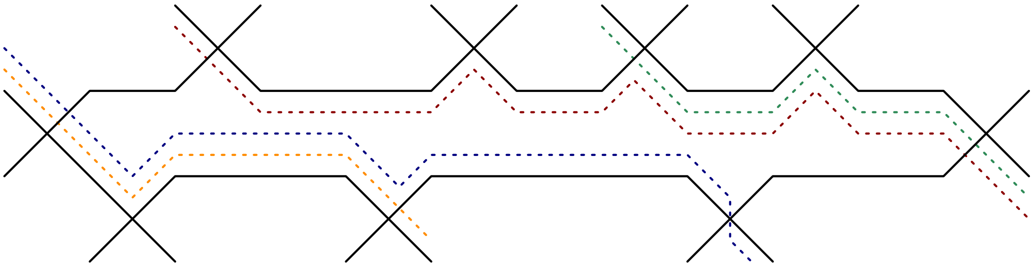


Figure 7.6: Four pseudo-verticals (two northbound and two southbound parts) meeting in a common cell.

pseudo-lines follow the different levels. Once two, say, southbound rays meet in a cell (i.e., they leave a cell through the same pseudo-line of $\mathcal{A}$), they follow the same pseudo-lines until reaching the south face (i.e., they pass through the same sequence of cells), and hence they can be drawn without intersecting each other.

For unbounded cells, the same argument works, with the exception that along the northbound part a pseudo-vertical enters the cell through the leftmost upper pseudo-line (i.e, at level k), and the southbound part leaves the cell through the rightmost pseudo-line at level $(k + 1)$. ☐

Recall the construction of a wiring diagram from a given pseudo-line arrangement in $\mathbb{P}^2$ (see Section 5.1.3.3) using a ψ-augmentation of the initial pseudo-line arrangement. While the abstract order type defined by the arrangement specifies which pseudo-lines in the wiring diagram are above pq, the ψ-augmentation fixes an allowable sequence for the wiring diagram. However, this allowable sequence is not determined, only its existence is shown using the Levi Enlargement Lemma. Lemma 7.7 shows that we can add pseudo-verticals to an arrangement such that they only intersect at vertical infinity. Hence, pseudo-verticals can be considered as a ψ-augmentation of the initial arrangement $\mathcal{A}$, and we can actually draw a wiring diagram where all the pseudo-verticals are represented by vertical lines.

This ψ-augmentation defines an order on the crossings of $\mathcal{A}$ that is also given by the sequence of switches in the corresponding allowable sequence. Let us examine this order. We saw that the circular sequence of an arrangement of lines is given by a vertical line sweeping the arrangement. Given an arrangement of lines, Edelsbrunner and Guibas [55, 56] defined the so-called *topological sweep*, that, informally, corresponds to sweeping an arrangement of lines with a curve that intersects each line exactly once. Just like sweeping the arrangement with a vertical line, a topological sweep defines one allowable sequence for the order type given by the line arrangement. The topological sweep has been generalized to pseudo-line arrangements by Snoeyink and Hershberger [150]. At any point in time during the sweep, the sweeping curve can pass over at least one crossing of the arrangement maintaining the property that it intersects each line exactly once. However, in contrast to a straight vertical line, there can be several crossings that may be swept next. It can be observed that we obtain the order of crossings determined by the pseudo-verticals by always sweeping over the lowest-possible crossing in a topological sweep. In the wiring diagram shown in Figure 7.4, the x-order of the crossings represents this order.

Since we know that there is an order on the pseudo-verticals, let us discuss how this order can be obtained. For two pseudo-verticals γ_{pq} and γ_{rs} (we have $p \prec q$ and $r \prec s$), this means we have to determine whether r crosses s before or after crossing γ_{pq}, i.e., whether rs is to the left or to the right of γ_{pq}. The two pseudo-lines defining a crossing naturally partition the plane into four regions, which we call the upper, lower, left, and right *quadrant* of the crossing. If rs is in the left quadrant of pq (i.e., below p and above q), then rs is definitely to the left of γ_{pq}. Similarly, if rs is in the right quadrant of pq then rs is to the right of γ_{pq}. The analogous holds when exchanging the roles of pq and rs. Therefore, we can assume without loss of generality that rs in the upper quadrant of pq (as the other case is symmetric), and that pq is either in the upper or lower quadrant of rs. Consider first the case where pq is in the upper quadrant of rs. If $r \prec p$, then r is part of $L(pq)$, and rs is, by Lemma 7.4, to the left of γ_{pq}. Analogously, if $p \prec r$, then p is part of $L(rs)$, and therefore rs is to the right of γ_{pq}. We are therefore left with the case where pq is in the lower quadrant of rs. If there exists a pseudo-line $a \in L(pq)$ that is above rs, we again know by Lemma 7.4 that rs is to the left of γ_{pq}. If no such pseudo-line exists, then $\mathrm{rk}_{pq}(r) < \mathrm{rk}_{pq}(s)$, and therefore, the crossing rs is to the right of γ_{pq}.

7.3 Linear-Time Pseudo-Line Selection

We now discuss algorithmic properties of pseudo-verticals. The definition of pseudo-verticals and the rank they define for each pseudo-line of the arrangement used a certain local information on the arrangement. In particular, we assumed to know the relative order in which two pseudo-lines start and whether a pseudo-line is above or below a crossing of an arrangement. In our setting, however, we cannot expect an explicit representation of the arrangement. We therefore describe an oracle that gives an implicit representation of an abstract order type as a pseudo-line arrangement by answering these two types of queries using in turn only a constant number of sidedness queries. We then show how to use the oracle to select a pseudo-line of a certain rank along a pseudo-vertical in linear time.

7.3.1 An Oracle for an Arrangement

Before answering queries on a pseudo-line arrangement, we need to have an internal representation of the arrangement (without explicitly building it) using only sidedness queries for point triples, i.e., a predicate P that indicates whether a triple in the primal point set S is oriented counterclockwise. The two main queries to answer are whether a pseudo-line a starts before a pseudo-line b (formally, $a \prec b$), and whether a pseudo-line r is above a crossing pq.

Recall the discussion of the order-preserving properties of line arrangements in Section 5.1.2. There, in order to relate the orientation of a triple to the above-below relationship in the dual, we needed to know whether one line started above or below the other line. Hence, we will start the description of our representation by defining the order in which the pseudo-lines start.

First, we select an extreme point x of S (see Chapter 6). We then use the internal representation indicated at the end of Section 5.1.3.3; see Figure 5.8 on page 72 and also

Figure 7.4. For all $a \in S\backslash\{x\}$, we define that $x \prec a$. For two points $a, a' \in S\backslash\{x\}$, we define that $a \prec a'$ if and only if a' is to the left of a with respect to x, i.e., we have $P(x, a, a')$. For two points $p, q \in S$ with $p \prec q$, the dual pseudo-line r is below the crossing pq if and only if r is counterclockwise of q with respect to r, i.e., we have $P(p, q, r)$. Hence, for three points $u, v, w \in S \setminus \{x\}$, the dual line r is below the crossing defined by the (unordered) pair (u, v) if and only if $P(u, v, w) = P(u, v, x)$. After selecting the extreme point x once in linear time, every query can be answered in constant time (assuming that the orientation of each triple can be determined in constant time).

Observe that, with this representation, the first unbounded cell we meet when traversing γ_{pq} against its direction is the north face, because every pseudo-line is crossed by the pseudo-line x from above (see again Figure 7.4). However, we will not make use of this fact in the remainder of this chapter, in particular since we use the fact that the problem is symmetric when exchanging the role of the north face and the south face (which corresponds to rotating the arrangement by 180°).

Our final linear-time algorithm will depend on removing a linear fraction of the pseudo-lines in each iteration. However, the oracle must not remove the extreme point x, in order to keep the representation consistent. If x is kept in the internal representation, the relative order in which all remaining pseudo-lines start stays the same.

7.3.2 Selecting a Pseudo-Line

For our abstract version of Problem 4, we want to select the pseudo-line m of a given rank k along γ_{pq}. In particular, m should be an element of a subset B of pseudo-lines, and we want to determine its rank in that order among the elements of B. We denote this restricted rank by $\mathrm{rk}_{pq}(m, B)$.

In the straight-line version, the linear-time selection algorithm is used to find an element of rank k in $O(n)$ time. This relies on the fact that the relative position of two pseudo-lines can be computed in constant time. When discussing the relative position of two pseudo-verticals, we have seen that checking whether a crossing $rs, r \prec s$ is below γ_{pq} (in which case we have $\mathrm{rk}_{pq}(s) < \mathrm{rk}_{pq}(r)$), may require to determine whether rs is below a pseudo-line $a \in L(pq)$, which, in the worst case, results in a linear number of comparisons. If we would directly use this method for comparing the rank of two pseudo-lines on γ_{pq} for the selection algorithm, we would end up with an $\Omega(n^2)$ worst-case behavior. We therefore need a more sophisticated method.

Let m be the (unknown) pseudo-line of rank $k < |B|$. We use a prune-and-search approach to identify m. By counting the elements of B above pq, we determine whether m is above or below pq (using a linear number of queries to our oracle). Without loss of generality, assume m is above pq (the other case is symmetric) and let U be the set of pseudo-lines above pq. Since removing pseudo-lines from U does not change the structure of the northbound part of γ_{pq}, we can (temporarily) remove the elements in $U \setminus B$ from U. We can also remove the pseudo-lines below pq that are not in $L(pq)$, i.e., each pseudo-line l below pq such that $p \prec l$.

As a next step, we can, in linear time, verify whether m starts above all pseudo-lines in $L(pq) \cup \{p\}$. If this is the case, the rank of m is determined by the order in which the pseudo-lines start, and we can apply the standard selection algorithm using this order (recall Observation 7.6).

We are therefore left with the case where m starts below some element $a \in L(pq) \cup \{p\}$. By Corollary 7.5, we know that we have to find the pseudo-line a' where m crosses the upper envelope of $L(pq) \cup \{p\}$ (recall that we have $a' \prec m$).

Basically, the algorithm continues as follows. We alternatingly remove elements in U and $L(pq)$ such that the pseudo-lines a' and m remain in the set until we are left with only a constant number of pseudo-lines in the arrangement. The details of these two pruning steps are given in the next three subsections.

7.3.2.1 Pruning the Pseudo-Lines Below the Crossing

We first show how to remove pseudo-lines from $L(pq)$. Suppose we are given any crossing vw, with $v, w \in L(pq)$ and $v \prec w$, on the (relevant part of) the upper envelope of $L(pq) \cup \{p\}$; see Figure 7.7. Depending on the value of $\mathrm{lv}(vw)$, we remove the pseudo-lines of $L(pq)$ that cannot be on the part of the upper envelope that contains the crossing with m (recall that $\mathrm{lv}(vw)$ is the level of the crossing vw, as defined on page 93). Let us examine how these pseudo-lines can be identified. Consider the crossings vp and wp. The only elements of $L(pq)$ that can contribute to the upper envelope along the northbound ray of γ_{pq} between vw and pq are the ones above wp. Similarly, the only elements of $L(pq)$ that can contribute to the other relevant part (between the north face and vw) of the upper envelope are the ones below vp. Hence, depending on $\mathrm{lv}(vw)$, we can remove the pseudo-lines in $L(pq)$ above vp or below wp. In both cases, the elements that are both above vp and below wp cannot contribute to the upper envelope.

It remains to choose vw appropriately. See Figure 7.7 for an accompanying illustration. Consider the order in which the pseudo-lines of $L(pq)$ intersect p. A pseudo-line a_i precedes a pseudo-line a_j in this order if a_j is above a_ip. Since these are basic queries to our oracle, we can determine the median pseudo-line r in linear time by the standard linear-time selection algorithm. Hence, $L(pq)$ is partitioned into the set L_1 of pseudo-lines below pr and the set $L_2 = L(pq) \setminus L_1$ of approximately equal size. If the pseudo-line of $L(pq)$ that starts above all other pseudo-lines in $L(pq)$ is in L_2 (which be determined by a linear number of queries to the oracle), then no pseudo-line of L_1 can be part of the upper envelope of $L(pq) \cup \{p\}$ and we can remove the elements of L_1. Otherwise, there exists exactly one pair (v, w) of pseudo-lines with $v \in L_1$ and $w \in L_2$ such that the crossing vw is on the upper envelope. Hence, depending on the value of $\mathrm{lv}(vw)$, we can prune at least either L_1 or L_2, and therefore at least approximately half of the pseudo-lines in $L(pq)$. In order to find vw, we use the result presented in Chapter 6. Observe that, in the primal, the upper envelope is a part of the convex hull of the abstract order type. The sets L_1 and L_2 are semispaces defined by the pair rp. Hence, vw is an edge of the convex hull of $L(pq) \cup \{p\}$ connecting a point of L_1 to a point of L_2. We are therefore looking for a bridge edge defined by the pair rp. By applying Theorem 6.4 (see page 90), we get vw after a linear number of queries.

Note that by removing pseudo-lines from $L(pq)$, we obtain a new arrangement $\mathcal{A}'$. In this arrangement, the pseudo-vertical γ_{pq} will, in general, follow different pseudo-lines from $L(pq)$ along its northbound ray. Still, the number of pseudo-lines above the crossing $a'm$

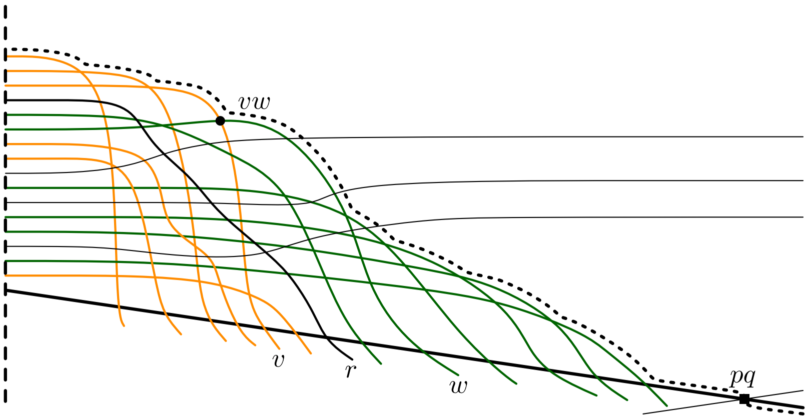


Figure 7.7: Partitioning the pseudo-lines in $L(pq)$ along p by a pseudo-line r.

that we look for remains the same, and m will have the same rank with respect to the new pseudo-vertical γ'_{pq} in the reduced arrangement.

7.3.2.2 Pruning the Pseudo-Lines Above the Crossing

To remove elements of U, we pick a pseudo-line $u_i \in U$. For now, let us suppose that u_i is chosen randomly; we will show later how to make a good selection of u_i in a deterministic way. Recall that we already removed the elements of $U \setminus B$, and hence we have $\mathrm{rk}_{pq}(u_i, B) = \mathrm{rk}_{pq}(u_i)$. We compute the rank $\mathrm{rk}_{pq}(u_i)$ by finding the corresponding pseudo-line $b \in L(pq) \cup \{p\}$ at which u_i passes through the upper envelope of $L(pq) \cup \{p\}$. Clearly, this can be done in linear time using our basic operations. If $u_i = m$, we are done. If $\mathrm{rk}_{pq}(u_i) < k$, then all pseudo-lines in U below bu_i can be removed (we will later see how to choose u_i in order to be able to remove a constant fraction of the pseudo-lines). Otherwise, we remove all pseudo-lines in U above bu_i and update k accordingly.

Note that, while by this operation we obtain a new arrangement $\mathcal{A}'$, the northbound ray of γ_{pq} in $\mathcal{A}'$ is defined by the same pseudo-lines as in $\mathcal{A}$, and we can therefore safely continue with the next iteration.

It remains to show how to pick u_i in a deterministic way such that at least a constant fraction of U can be removed in each iteration. To this end, we use the mighty concept of ε-approximation of range spaces.

7.3.2.3 Using ε-Approximation for Pruning

Our definitions follow [118]. A *range space* is a pair $\Sigma = (X, \mathcal{R})$ where X is a set and $\mathcal{R}$ is a set of subsets of X. The elements of $\mathcal{R}$ are called *ranges*. For X being finite, a subset $A \subseteq X$ is an *ε-approximation* for Σ if, for every range $R \in \mathcal{R}$, we have

$$\left| \frac{|A \cap R|}{|A|} - \frac{|X \cap R|}{|X|} \right| \leq \varepsilon \ .$$

A subset Y of X is *shattered* by $\mathcal{R}$ if every possible subset of Y is a range of Y. The *Vapnik-Chervonenkis dimension* (VC-dimension) of Σ is the maximum size of a shattered subset of X. For sets with finite VC-dimension, Vapnik and Chervonenkis [159] give the following seminal result.

Theorem 7.8 (Vapnik, Chervonenkis [159])**.** *Any range space of VC-dimension d admits an ε-approximation of size $O(d/\varepsilon^2 \log(d/\varepsilon))$.*

For $|X| = n$, the *shatter function* $\pi_{\mathcal{R}}(n)$ of a range space $(X, \mathcal{R})$ is defined by

$$\pi_{\mathcal{R}}(n) = \max\{|\{Y \cap R : R \in \mathcal{R}\}| : Y \subseteq X\} \ .$$

Vapnik and Chervonenkis [159] show that, for a range space $(X, \mathcal{R})$ of VC-dimension d, $\pi_{\mathcal{R}}(n) \in O(n^d)$ holds. Matoušek [117, 119] gives a linear-time algorithm for computing an ε-approximation for range spaces of finite VC-dimension d (simplified by Chazelle and Matoušek [44]), provided there exists an appropriate subspace oracle.

Definition 7.1. *A subspace oracle for a range space $(X, \mathcal{R})$ is an algorithm that returns, for a given subset $Y \subseteq X$, the set of all distinct intersections of Y with the ranges in $\mathcal{R}$, i.e., the set $\{Y \cap R : R \in \mathcal{R}\}$ and runs in time $O(|Y| \cdot h)$, where h is the number of sets returned.*

Theorem 7.9 (Matoušek [117, Theorem 4.1])**.** *Let $\Sigma = (X, \mathcal{R})$ be a range space with the shatter function $\pi_{\mathcal{R}}(n) \in O(n^d)$, for a constant $d \geq 1$. Given a subspace oracle for Σ and a parameter $r \geq 2$, a $(1/r)$-approximation for Σ of size $O(r^2 \log r)$ can be computed in time $O(|X|(r^2 \log r)^d)$.*

Observe that, for such a range space, the running time of the subspace oracle is bounded by $O(|Y|^{d+1})$, as h is at most $\pi_{\mathcal{R}}(|Y|)$.

Suppose that, e.g., X is a point set in the Euclidean plane and $\mathcal{R}$ consists of all possible subsets of X defined by half-planes, defining a range space $\Sigma = (X, \mathcal{R})$. Hence, $\mathcal{R}$ is the set of semispaces defined by the order type of X. It is easy to see that the VC-dimension of Σ is 3. Hence, as pointed out by Lo, Matoušek, and Steiger [112], a constant-size ε-approximation of a point set for $\mathcal{R}$ exists; given this approximation, an approximate ham-sandwich cut can be constructed in constant time, such that on every side of the cut there are no more than $1/2 + \varepsilon$ of the points of each class. The subspace oracle returns, for any subset Y of points, all possible ways a line can separate Y, which can easily be done in time $O(|Y|^3)$.

We apply a very similar approach for derandomizing our algorithm. Note that the above setting works equally well for abstract order types: Since all abstract order types of 8 elements are realizable (see Theorem 5.5), the VC-dimension of 3 for the range space of semispaces holds also for abstract order types. (A more general approach to this setting is taken by Gärtner and Welzl [73], who study the range spaces of arrangements of oriented pseudo-hyperplanes by grouping, for each cell C, the pseudo-hyperplanes having C on their positive side to a range.)

A subspace oracle for semispaces of a given set can easily be implemented using the definition of a semispace by allowable sequences [82]; for each pair $f, g \in Y, f \prec g$, in the dual pseudo-line arrangement, we report the pseudo-lines above the crossing fg and, say, f as a semispace, and the pseudo-lines below fg and g as a second semispace.

Consider again the set U of pseudo-lines above the crossing pq in our arrangement $\mathcal{A}$. Using Theorem 7.9, we obtain an ε-approximation $A \subset U$ for the range space of semispaces, i.e., the pseudo-lines of U above and below a point in $\mathcal{A}$. Recall that A is of constant size for a fixed ε. For each pseudo-line $o \in A$, we obtain, in linear time, the crossing of o with the pseudo-lines in $L(pq)$ that defines the rank $\mathrm{rk}_{pq}(o)$. This gives us the pseudo-line $u_i \in A$ that has the median rank among the elements of A. Hence, we obtained a pseudo-line u_i such that not less than $1/2 - \varepsilon$ pseudo-lines of U are above and below the crossing bu_i on the upper envelope of $L(pq)$. This allows us to prune a constant fraction of the elements in U.

7.3.2.4 Analysis

In each iteration, our problem consists of the remaining pseudo-lines in U and in $L(pq)$, plus a constant number of additional pseudo-lines (i.e., p, q, and the pseudo-lines needed by the oracle, in our case the pseudo-line x). Let n be the number of these pseudo-lines. In each iteration we prune the larger of U and $L(pq)$. In both cases, we remove at least half of the pseudo-lines on one side of pq, and therefore $n/4 - c$ pseudo-lines in each iteration. Since each iteration takes $O(n)$ time, we have overall a linear-time prune-and-search algorithm.

Theorem 7.10. *Given an arrangement $\mathcal{A}$ of pseudo-lines, a subset B of its pseudo-lines, a crossing pq, and a natural number $k \leq |B|$, the pseudo-line $m \in B$ of rank k in B on the pseudo-vertical through pq, i.e., $\mathrm{rk}_{pq}(m, B)$, can be found in linear time using only sidedness queries on the corresponding abstract order type.*

7.4 Revisiting the Ham-Sandwich Cut Algorithm

In this section, we describe an application of pseudo-verticals for a bisection algorithm, namely the linear-time ham-sandwich cut algorithm by Lo, Matoušek, and Steiger [112]. To this end, we revisit the description given in [112]; we adapt some of the terminology (like replacing "line" with "pseudo-line") and argue for the correspondence between entities in the original description and their abstract counterpart.

Lo, Matoušek, and Steiger [112] describe two different variants of the algorithm, one for points in the plane and the other one for points in arbitrary dimension (their work is a generalization of a 2-dimensional version by Lo and Steiger presented in [113]). For the 2-dimensional case, a result by Matoušek [116, Theorem 3.2] is used for appropriately selecting a set of vertical lines. In higher-dimensions, they use a different approach (given in [117]) based on an ε-approximation with the ranges being defined as sets of hyperplanes that are stabbed by segments (we will give a formal definition later). While the higher-dimensional variant appears to be less instructive, it is easier to apply to our setting. We therefore will use this variant for our 2-dimensional setting; here, we do not give the description for arbitrary dimension, but transcribe it to dimension 2 only. Still, our exposition closely follows [112], while merely pointing out the parts where the applicability to our abstract setting might not be obvious.

Let P be a finite set of n points in the Euclidean plane. A line h *bisects* P if no more than $n/2$ points lie in either of the open half-planes defined by h. We call h a *bisector*. If P

is a disjoint union of two point sets P_1, P_2, a *ham-sandwich cut* is a line that simultaneously bisects both P_1 and P_2 (a *red* and a *blue* set). This definition extends to abstract order types in a natural way. It is well-known that a ham-sandwich cut always exists. Let T be an interval on the x-axis, and let $V(T)$ be the vertical slab between the two vertical lines defining T. The interval has the *odd intersection property* with respect to the levels λ_1 and λ_2 if $|(\lambda_1 \cap \lambda_2) \cap V(T)|$ is odd. If $k = \lfloor (n+1)/2 \rfloor$, the k-level is called *median level*. In our case, each slab is defined by two pseudo-verticals.

The algorithm works in a prune-and-search manner. Let us first consider the setting where we are given an actual set of points in $\mathbb{E}^2$. In every iteration, we are given

- an interval T on the x-axis,
- two sets G_1 and G_2 of lines dual to a subset of points in P_1 and P_2, respectively, with $|G_1| = n_1$ and $|G_2| = n_2$, and
- two integers k_1 and k_2, with $1 \leq k_1 \leq n_1$ and $1 \leq k_2 \leq n_2$, denoting the k_1-level λ_1 and the k_2-level λ_2, respectively.

Further, we know that T has the odd intersection property for the k_1-level and the k_2-level. We denote the arrangements corresponding to G_1 and G_2 with $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. Initially, λ_1 and λ_2 are the median levels of the two arrangements (for which the odd intersection property holds). Without loss of generality, suppose $n_1 \geq n_2$. The algorithm consists of the following four steps:

1. Divide T into a constant number of subintervals $T_1, \ldots, T_C$, to limit the number of pseudo-lines that are on λ_1 within each subinterval.
2. Find a subinterval T_j with the odd intersection property.
3. Construct a trapezoid $\tau \subset V(T_j)$ such that
   (a) $\lambda_1 \cap V(T_j) \subset \tau$, and
   (b) at most half of the lines of $\mathcal{A}_1$ intersect τ.
4. Discard the lines of $\mathcal{A}_1$ that do not intersect τ, update k_1 accordingly and continue within the interval T_j.

In our abstract setting, the interval T is given by a pair of pseudo-verticals. Recall that there is a total order on the pseudo-verticals of a pseudo-line arrangement. The trapezoid τ will also be replaced by a corresponding structure that will be described later.

7.4.1 Obtaining Intervals

Step 1 in the algorithm is the one that is technically most involved. The straight-line version can be solved using the following result by Matoušek.[1]

---

[1] Lo, Matoušek, and Steiger [112] refer to [116], where [90, Lemma 4.5] (Lemma 7.12 herein) is used, and also refer to [117] in this context, where a general algorithm for constructing ε-approximations is given.

Lemma 7.11 (Matoušek). *Let H be a collection of n hyperplanes in $\mathbb{E}^d$ and let $\mathcal{R}$ be all subsets of H of the form $\{h \in H : h \cap s \neq \emptyset\}$ where s is a segment in $\mathbb{E}^d$. An ε-approximation for the range space $(H, \mathcal{R})$ of size $O(\varepsilon^{-2}\log(1/\varepsilon))$ can be computed in time $O(f(\varepsilon)n)$, where $f(\varepsilon)$ is a factor depending on ε and d only.*

Let us go into the details why this lemma also holds for arrangements of pseudo-lines. To this end, a general result by Haussler and Welzl [90] is used.

Lemma 7.12 (Haussler, Welzl [90, Lemma 4.5]). *Assume $k \geq 1$ and $(X, \mathcal{R})$ is a range space of VC-dimension $d \geq 2$. Let $\mathcal{R}'$ be the set of all sets of the form $\bigcup_{i=1}^{k} R_i - \bigcap_{i=1}^{k} R_i$, where R_i is a range in $\mathcal{R}$, $1 \leq i \leq k$. Then $(X, \mathcal{R}')$ has VC-dimension less than $2dk\log(dk)$.*

We already discussed that a range space defined by the semispaces of an abstract order type has VC-dimension 3. We can combine this fact with Lemma 7.12 in the following way. Consider two semispaces S_1 and S_2 of an abstract order type, defined by the pseudo-lines above two points p_1 and p_2 in the corresponding pseudo-line arrangement $\mathcal{A}$ (for simplicity, suppose that none of p_1 and p_2 lie on a pseudo-line of $\mathcal{A}$). Let $R = (S_1 \cup S_2) \setminus (S_1 \cap S_2)$. Then R consists exactly of the pseudo-lines that separate p_1 from p_2. By the Levi Enlargement Lemma (see Lemma 5.3), we can obtain a pseudo-line χ for $\mathcal{A}$ containing both p_1 and p_2. Consider the part of χ between p_1 and p_2. We call such a part a *pseudo-segment*. The pseudo-lines crossed by this pseudo-segment are exactly those in R. Applying Lemma 7.12, we can therefore obtain a range space that is defined by the pseudo-lines that can be crossed by pseudo-segments from the range space defined by the semispaces; this new range space has again finite VC-dimension. (Note that, while we explained the application of Lemma 7.12 using points p_1 and p_2, the argument also holds for pseudo-segments defined by crossings of $\mathcal{A}$, as the endpoints of the pseudo-segment can be perturbed to be in one of the four cells adjacent to a crossing.) Using Theorem 7.9, we can state the following counterpart to Lemma 7.11 for abstract order types in the plane.

Corollary 7.13. *Let $\Sigma = (X, \mathcal{R})$ be a range space where X is the set of pseudo-lines in a pseudo-line arrangement $\mathcal{A}$ and $\mathcal{R}$ consists of the sets of pseudo-lines of $\mathcal{A}$ that are crossed by pseudo-segments obtained on $\mathcal{A}$. Then a $(1/r)$-approximation of constant size for Σ can be computed in $O(|X|)$ time for a given $r \geq 2$.*

For future reference, let us call this range space the *pseudo-segment range space* of the arrangement.

For the ham-sandwich cut algorithm, we can now proceed in the following way. Using Corollary 7.13, we obtain an ε-approximation A for the pseudo-segment range space of $\mathcal{A}_1$; we choose $\varepsilon = 1/12$ with foresight. Sort the crossings of A by the order implied by the pseudo-verticals through the crossings on the original arrangement $\mathcal{A}$. Since A is of constant size, this can be done in $O(|\mathcal{A}|)$ time. We use Theorem 7.10 to determine the k_1-level of $\mathcal{A}_1$ at the pseudo-vertical of each crossing in A. Hence, for each pseudo-vertical, we get a crossing in $\mathcal{A}$ that has level k_1 in $\mathcal{A}_1$. Counting the elements of $\mathcal{A}_2$ above each such crossing allows us to find a crossing pq and a crossing $p'q'$ consecutive in A with the odd intersection property. We again use Theorem 7.10 to select the pseudo-lines that are of rank $k_1 - c\varepsilon n_1$ and $k_1 + c\varepsilon n_1$ in $\mathcal{A}_1$ at the pseudo-vertical γ_{pq}, and do the same at $\gamma_{p'q'}$ for a constant c (we fix $c = 3/2$ with foresight). Hence, we have six crossings in $\mathcal{A}$ of which we know the level within $\mathcal{A}_1$. Let g_l and g_r be the crossings at the k_1-level along γ_{pq} and $\gamma_{p'q'}$, respectively. We denote the crossings at the $(k_1 - c\varepsilon n_1)$-level by d_l^- and d_r^-. Their counterparts at the $(k_1 + c\varepsilon n_1)$-level are denoted by d_l^+ and d_r^+.

7.4.2 Properties of a Trapezoid-Like Structure

In the original algorithm [112], the points at the given levels were determined by the intersections of the levels with the vertical lines. These points formed a trapezoid. However, the actual properties used are the ones of the points and not the ones of the trapezoid as a geometric object. In this part, we reproduce the line of arguments used in [112] to show that at least half of the pseudo-lines in $\mathcal{A}_1$ are either above both d_l^- and d_r^- or below both d_l^+ and d_r^+, and that these pseudo-lines are not on the k_1-level between g_l and g_r.

Consider the arrangement $\mathcal{A}_1$. We bound the number of pseudo-lines that separate d_l^- from d_r^-, i.e., the pseudo-lines crossing a pseudo-segment between d_l^- and d_r^-. The levels of d_l^- and d_r^- are the same. Therefore, the numbers of pseudo-lines of the approximation A above these two crossings differ by at most $2\varepsilon\,|A|$. If there would be more than $2\varepsilon\,|A|$ pseudo-lines of A separating d_l^- from d_r^-, then at least one of these pseudo-lines would have to be above d_l^- and below d_r^-, and another one would have to be below d_l^- and above d_r^-. Hence, the crossing between these two pseudo-lines would have to be in the interval between γ_{pq} and $\gamma_{p'q'}$. But this contradicts the choice of γ_{pq} and $\gamma_{p'q'}$, as there is no pseudo-vertical through a crossing of two pseudo-lines of A between them. Hence, any pseudo-segment between d_l^- and d_r^- crosses at most $2\varepsilon\,|A|$ of the pseudo-lines in A. By the ε-approximation property, at most $3\varepsilon n_1$ pseudo-lines of $\mathcal{A}_1$ intersect such a pseudo-segment.

Suppose there is a pseudo-line w of $\mathcal{A}_1$ that is above both d_l^- and d_r^-, but still w is an element of the k_1-level between g_l and g_r. The part of the arrangement where this can happen is bounded by γ_{pq} and $\gamma_{p'q'}$. Then also any pseudo-segment s between d_l^- and d_r^- would have to cross the relevant part of the k_1-level (recall that the pseudo-segment can be considered as a part of a pseudo-line in an extended arrangement). At both d_l^- and d_r^-, the pseudo-segment s is at level $(k_1 - c\varepsilon n_1)$, and therefore has to cross $2(k_1 - (k_1 - c\varepsilon n_1)) = 2c\varepsilon n_1$ pseudo-lines to reach the k_1-level and then return to level $(k_1 - c\varepsilon n_1)$. By the choice of c, this is exactly $3\varepsilon n_1$, the maximum number of crossings the pseudo-segment s can have. Hence, s cannot go below the k_1-level and therefore the pseudo-line w cannot intersect the k_1-level. For our prune-and-search approach, we can therefore remove all pseudo-lines of $\mathcal{A}_1$ that are above both d_l^- and d_r^-, and, by symmetric arguments, can do the same for the ones below both d_l^+ and d_r^+.

It remains to count how many pseudo-lines are removed. There are exactly $2c\varepsilon n_1$ pseudo-lines separating d_l^+ from d_l^-, as well as d_r^+ from d_r^-. Further, we argued that there are at most $3\varepsilon n_1$ pseudo-lines between d_l^- and d_r^-, as well as between d_l^+ and d_r^+. This amounts to $(4c+6)\varepsilon n_1 = 12\varepsilon n_1$, where each pseudo-line is counted twice. Therefore, we have to keep at most $6\varepsilon n_1 = n_1/2$ pseudo-lines of $\mathcal{A}_1$.

7.4.3 A Note on the Intervals

After having pruned a linear fraction of pseudo-lines, the algorithm performs another iteration within a smaller interval for which the odd-intersection property holds. Note that we need to continue within this interval, as the k_1-level (for an updated k_1) equals the median-level only within that region. In the geometric variant, the interval was explicitly given by the vertical lines of the current slab. For pseudo-verticals, we have no such fixed position, and actually the pseudo-verticals will, in general, be different when pseudo-lines are removed from the arrangement (even the relative order of two crossings may change). However, we can safely define the interval for the subproblem by the two crossings g_l and g_r, as the odd intersection property can be seen as a property of two points on one of the two levels (only the number of pseudo-lines of $\mathcal{A}_2$ above each of the two points is relevant here). It is interesting to observe that also γ_{g_l} and γ_{g_r} do not have a different relative position in the new arrangement.

Corollary 7.14. *A ham-sandwich cut of an abstract order type can be found in linear time using only sidedness queries.*

7.5 Connections with Extreme Point Search and the Two-Line Partitioning Problem

Let us have an informal look at the problem of selecting an element of rank k on a given pseudo-vertical γ_{pq} in the primal. For simplicity, suppose there is no pseudo-line starting above the upper envelope of $L(pq) \cup \{p\}$. Consider the upper envelope of the whole arrangement $\mathcal{A}$. The set of pseudo-lines can be partitioned into those above and those below pq. The crossing that witnesses the rank of the first pseudo-line crossed by γ_{pq} is the crossing e at the upper envelope of $\mathcal{A}$ where one pseudo-line from above pq crosses a pseudo-line below pq. In the primal, this crossing corresponds to the supporting line of one of the two bridge edges on the convex hull of the set, which is separated by the line pq. When subsequently removing the pseudo-lines of U that participate in such a crossing e, we get the order as defined in the proof of Theorem 6.2. (See again Figure 6.2; the roles of p and q are interchanged in our description.) So while for Theorem 6.2 we were essentially looking for an element of rank 1 in U (keep in mind that we disregarded the pseudo-lines starting above the upper envelope of $L(pq) \cup \{p\}$), we can find an element of any given rank k by Theorem 7.10. The algorithm BasicMin, which we used for pruning in Theorem 6.2, seems essentially to be only suitable when we want to find a minimal element in that order. This is why we utilized the more sophisticated method of ε-approximation in Section 7.3.2. However, it can be shown that BasicMin can be used to prune the set $L(pq)$, where in the final step we would not look for the first element in the radial order of U around the two endpoints of the last edge m, but for the one of rank k. Conversely, it can also be seen that, by adapting the proof of Theorem 7.10, the ε-approximation for the range space defined by the semispaces of the abstract order type can also be used to obtain Theorem 6.4.

Several years before the linear-time algorithm for ham-sandwich cuts [112] was developed, Megiddo [121] considered the following restricted version of the ham-sandwich cut problem. Given a set of red and a set of blue points with disjoint convex hulls, find a line that bisects both the red and the blue point set. Actually, the resulting line does not have to be a bisector, but the number of red and blue points on one side of the line can be chosen arbitrarily. In the dual representation, we are given m blue lines with positive slope and n red lines with negative slope, and we want to find the intersection point between a k_1-level in the blue lines and the k_2-level in the red lines. If we consider the pseudo-lines in $L(pq) \cup \{p\}$ as red pseudo-lines and the pseudo-lines in U as the blue ones, we are looking for the intersection point between the k-level in U and the 1-level in $L(pq) \cup \{p\}$. However, Megiddo's algorithm also depends on the realization of the line arrangement; the algorithm requires selecting the median of a subset of crossings ordered by their x-coordinate and selecting the intersections of a given rank at vertical lines. These problems also had to be solved when abstracting the general ham-sandwich cut algorithm.

7.6 Chapter Summary

In this chapter, we defined a possible replacement of a vertical line in line arrangements for arrangements of pseudo-lines and showed that it fulfills important algorithmic properties. In particular, we were able to show how to select the kth pseudo-line crossed by this pseudo-vertical line and the crossing where this pseudo-line (locally) enters the k-level in linear time, using only sidedness queries. As an application, we showed how these pseudo-vertical lines replace vertical lines in the linear-time ham-sandwich cut algorithm by Lo, Matoušek, and Steiger [112].

In essence, the order of the pseudo-verticals through all crossings of a pseudo-line arrangement fix one specific allowable sequence for an abstract order type. Theorem 7.10 allows us to select certain elements of a permutation in that allowable sequence in linear time. We have seen that this approach is a generalization of the result presented in Chapter 6. (Note that the oracle also uses the extreme point x, which, in turn, requires applying Theorem 7.10; any internal representation that can answer queries of the form $a \prec b$ in constant time allows us to find an extreme point in linear time.) It would be interesting to see whether the ε-approximation used can be replaced by a less sophisticated method similar to the BasicMin algorithm.

The observation that the approach by Lo and Steiger [113] in principle also works for pseudo-line arrangements in the Euclidean plane is not new. It has been used by Bose et al. [35] for their randomized linear-time algorithm for geodesic ham-sandwich cuts inside a pointgon (see also Section 5.2.3). However, in their setting, the pseudo-lines are given by (weakly) x-monotone polygonal paths with a constant number of edges. Hence, the intersection of such a path with a vertical line can be computed in constant time, like in the straight-line setting. Their algorithm is randomized and runs in $O((n+m)\log r)$ time, where n is the number of red and blue points, m is the number of vertices of the polygon, of which r are reflex. They show that the algorithm is optimal in the algebraic computation tree model when the running time is parameterized by $(n + m)$ and r.

As discussed in Section 5.2.3, geodesic order types are a subset of abstract order types. When we apply Theorem 5.7 to get, after $O(m)$ preprocessing time, the orientation of each triple of points in a pointgon in $O(\log r)$ time in combination with the ham-sandwich cut algorithm for abstract order types, we obtain a deterministic $O(n \log r + m)$ time algorithm for geodesic ham-sandwich cuts “for free”. Note that this does not contradict the lower bound on the worst-case behavior shown by Bose et al. [35], as their analysis is parameterized by the sum of the number of points and the number of vertices of the polygon. We emphasize that a detailed analysis of their approach may give a more fine-grained runtime analysis, and may allow for directly applying common derandomization techniques. But nevertheless, our technique results in a complete separation of the part that is specific to the geodesic setting, implementing a general subroutine, and the ham-sandwich cut algorithm.

Chapter 8

Algorithmic Aspects of *k*-Convex Point Sets

8.1 Introduction

The concept of convexity in Euclidean spaces (or, more general, in vector spaces of ordered fields) is ubiquitous in geometry. As we already saw in the previous chapters, the notion of convexity for 2-dimensional domains (such as convex polygons) has been transcribed to point sets in convex position in the field of combinatorial geometry. Various generalizations and relaxations of convexity have been considered in the literature [5]. For example, in the concept of restricted-orientation convexity (or D-convexity), a set is considered D-convex if the intersection with any line having a direction contained in a set D of vectors is either empty or connected [68, 120].

This chapter is concerned with another generalization of convexity. The notion of k-convexity was recently introduced in [5] for 2-dimensional subsets of the (Euclidean) plane. In particular, this comprises simple polygons. While the definition of k-convexity is rather involved for 2-dimensional subsets of the Euclidean plane, we will use the definition restricted to simple polygons. In this case we say that a simple polygon is *k-convex* if the intersection of any straight line with the polygon consists of no more than k disjoint intervals. This implies that 1-convexity refers to the classic concept of convexity of a simple polygon. Among other results, several properties of k-convex planar polygons are presented in [5]. This work has been generalized by extending k-convexity to finite point sets in the plane [6]. A point set S is said to be k-convex if there exists a k-convex polygon whose vertex set is S. In other words, we are interested in point sets which admit a k-convex polygonization. This chapter presents algorithmic aspects of k-convex point sets that have already been published in [6]. That paper also covers combinatorial aspects of k-convexity of point sets, which are not presented in this thesis. In addition to the results presented in [6], we give a short account on k-convexity for abstract order types in Section 8.5.

With the previous chapters in mind, the reader might already have noted that k-convexity for simple polygons and point sets is a combinatorial rather than an algebraic concept (as opposed to D-convexity). Also, it directly migrates to the domain of abstract order types. We will consider this abstract setting at the end of this chapter (Section 8.5).

The introduction of k-convexity of 2-dimensional domains [5] follows the common approach to generalize the notion of convexity under various aspects. As for point sets in convex position, we can make the step from k-convex polygons to k-convexity of point sets, motivated by the importance of Erdős-Szekeres-type results in combinatorial and computational geometry. A comparable line of research is taken by Arkin et al. [21], who consider the minimum number of reflex vertices among all simple polygonizations of a point set.

We devote Section 8.2 to a precise definition of our generalization. The closely related concepts of stabbing number and j-stabber (of a polygon or a geometric graph) are also discussed. Further, we give some basic combinatorial properties of point sets regarding k-convexity, mainly to give some intuition for the algorithmic results presented. In Section 8.3, a polynomial-time algorithm is given for deciding whether a point set is 2-convex. In contrast, in Section 8.4 the problem of deciding k-convexity is proved to be NP-complete, for all $k \geq 3$. Finally, we address the concept of k-convexity for abstract order types in Section 8.5.

8.2 Preliminaries and Basic Properties

As usual, let S be a finite set of points in the plane. We again assume general position for all point sets, unless otherwise stated. We follow the definitions in [5]. Suppose a line ℓ has non-empty intersection with ∂P. At each component of the intersection, ℓ either crosses ∂P, or locally supports P along the component (which is either an edge or a single vertex of P). A line ℓ is a *j-stabber of P* if it crosses ∂P at least j times. The *stabbing number* of P is the largest number of crossings between ∂P and a line. Let ℓ be a line that intersects the interior of P in exactly k connected components. Since all vertices are in general position, there exists a perturbation of ℓ that is a $2k$-stabber of P. Therefore a polygon is k-convex but not $(k-1)$-convex if and only if its stabbing number is $2k$.

Given a point set S, let $\mathcal{P}$ be the set of all *polygonizations* of S, i.e., the set of all simple polygons whose vertex set is exactly S. If $\mathcal{P}$ contains at least one polygon that is k-convex, then we call S a *k-convex point set*.

Let G be a (possibly self-intersecting) geometric graph. A line ℓ is a *j-stabber of G* if it crosses at least j edges of G. Note that the degenerate cases where a line passes through a vertex of G can be disregarded due to the same perturbation arguments as for the stabbing number of a simple polygon, resulting in a consistent definition of the stabbing number of geometric graphs. We can therefore define that if G has stabbing number at most $2k$, then G is a *k-convex graph*. Now any simple k-convex polygonal cycle C is the boundary of a k-convex polygon, and for every k-convex polygon P, ∂P is a k-convex polygonal cycle. For the sake of brevity, we will sometimes refer to a line ℓ as a *local* j-stabber if ℓ is a j-stabber of a subgraph, where the subgraph will be clear from the context.

The following result formalizes that also a *non-simple* polygonal cycle can be taken to determine the degree of convexity of a set of points.

Lemma 8.1 (Aichholzer et al. [6]). *Let C be a spanning (non-simple) k-convex polygonal cycle of a set S of points. Then S is k-convex.*

In order to reason about the degree of convexity of a point set, it is useful to know its bounds, which are given by the following theorem.

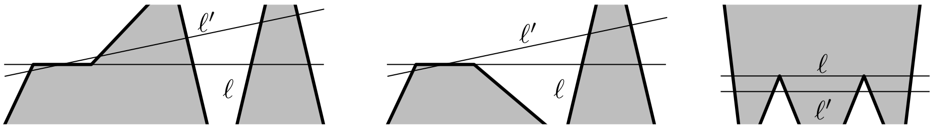


Figure 8.1: Lines stabbing a polygon. An inflection edge (left) is contained in ℓ; its perturbation ℓ' is a 6-stabber. An edge with two convex vertices (middle) is contained in ℓ; ℓ' is a 4-stabber. An inner tangent and its perturbation is shown to the right.

Theorem 8.2 (Aichholzer et al. [6]). *Any set S of n points is $O(\sqrt{n})$-convex, and this bound is tight.*

Further, it will be useful to keep in mind, in particular throughout the NP-hardness construction, that adding points to a point set that is not k-convex cannot make it k-convex.

Lemma 8.3 (Aichholzer et al. [6]). *Every subset of a k-convex point set is k-convex.*

8.3 Deciding 2-Convexity of Point Sets

In this section we turn our attention to algorithmic aspects of 2-convexity. We study the problem of deciding whether a point set is 2-convex and show that if a 2-convex polygonization of a point set exists, it can be constructed in polynomial time.

8.3.1 The Structure of 2-Convex Polygons

Let us first recall basic definitions and facts about 2-convex polygons given in [5]. An edge of a simple polygon P is called an *inflection edge* if it joins a convex and a reflex vertex of P. An *inflection line* is the supporting line of an inflection edge. A line ℓ is an *inner tangent* if it is the supporting line of two nonconsecutive reflex vertices such that there are points interior to the polygon in each of the three intervals in which these two vertices split the line. See Figure 8.1. The following result shows the interrelation between 2-convexity, inner tangents, and stabbers.

Lemma 8.4 (Aichholzer et al. [5, Lemma 10]). *A simple polygon P is 2-convex if and only if P has no inner tangent, and no inflection line that can be infinitesimally perturbed to a 6-stabber.*

We give some further preliminary definitions; see Figure 8.2 for an accompanying illustration. Unless stated otherwise, the edges of a polygon are considered to be directed counterclockwise around the polygon, and all polygonal chains are simple.

Definition 8.1. *A lid of a polygonization of S is an edge of* CH(S) *(not necessarily part of the polygonization).*

Definition 8.2. *A pocket of a polygon is the polygonal chain between the first and second end-vertex of a lid. A pocket consisting solely of the lid is called a trivial pocket.*

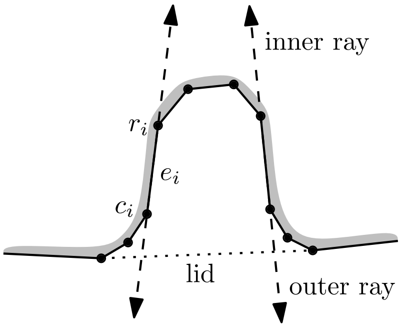


Figure 8.2: A pocket of a 2-convex polygon. The interior of the polygon is gray. The dotted edge illustrates the lid. The two dashed lines illustrate the two inflection lines of the pocket, consisting of the inner ray, the inflection edge, and the outer ray.

Definition 8.3. *For an inflection edge e_i, let c_i and r_i denote the convex and reflex vertex of e_i respectively. We partition an inflection line into the inflection edge and two rays; the inner ray, starting at r_i and the outer ray, starting at c_i.*

Lemma 8.5 (Aichholzer et al. [5, Lemma 12]). *Given a 2-convex polygonization P of S, let $C = \langle p_1, p_2, \ldots, p_t \rangle$ be the chain of vertices that connects (counterclockwise) two consecutive vertices p_1, p_t on $\mathrm{CH}(S)$ (i.e., C defines a pocket). Then the vertices of the chain can be partitioned into three chains $C_1 = \langle p_1, \ldots, p_u \rangle$, $C_2 = \langle p_{u+1}, \ldots, p_s \rangle$, $C_3 = \langle p_{s+1}, \ldots, p_t \rangle$, such that all the elements in C_1 and C_3 are convex vertices of P, while all the elements in C_2 are reflex.*

Hence, each non-trivial pocket in a 2-convex polygonization has exactly one pair of inflection edges. The chain of reflex vertices in a pocket of a 2-convex polygon is called the *reflex chain*.

8.3.2 Deciding 2-Convexity of Polygons

If we are given a polygonization, we can determine in $O(n \log n)$ time whether this simple polygon is 2-convex [5, Theorem 11]. We will use this as a sub-procedure for our algorithm in Section 8.3.4.2. Here, we give a sketch of the proof presented in [5] to provide further intuition for the problem setting, and for our reasoning in Section 8.5.

Lemma 8.4 implies that the algorithm has to check for the existence of 4-stabbing inflection lines and of inner tangents.

For testing the existence of 4-stabbing inflection lines, the rays emanating from edges at reflex vertices are checked using a result of [42], which shows that ray shooting queries in a simple polygon can be done in $O(\log n)$ time after a linear-time preprocessing step. (Observe that, after the edge where a ray leaves the polygon has been determined, it can be checked in $O(\log n)$ time whether it re-enters the polygon at that pocket.)

For determining whether the polygon has an inner tangent, one uses the geometric dual in the following way. For each reflex vertex, the set of lines supporting it gives a double wedge, whose dual is a line segment. The polygon has an inner tangent if and only if two such segments intersect. This can be tested in $O(n \log n)$ time [135, Theorem 7.9].

8.3.3 Outline of the Algorithm

Recognizing 2-convexity of a point set S can be done in polynomial time if it has a star-shaped 2-convex polygonization. A brute-force approach would be to consider all $\Theta(n^4)$ cells of the arrangement of lines spanned by two points of S as part of the potential kernel. For each choice, the resulting star-shaped polygon can be constructed and checked for 2-convexity in $O(n \log n)$ time [5], resulting in an $O(n^5 \log n)$ algorithm. Hence, for the remainder of this section, we assume that the point set S does not have a 2-convex star-shaped polygonization.

Suppose we have fixed a non-trivial pocket that is part of the 2-convex polygonization. Consider any line ℓ that crosses the pocket exactly twice in such a way that ℓ intersects the pocket in exactly two points (i.e., ℓ does not contain an edge of the pocket). The two crossing points partition ℓ into a segment and two rays. Each of these rays crosses ∂P exactly once, since otherwise ℓ would be a 6-stabber. The key observation for the algorithm, which will be proven formally in Lemma 8.8, is that if we rotate ℓ in such a way that it always crosses the pocket twice, the order in which ℓ traverses points not in the pocket is the same as the order of these points along ∂P. We look for a triple of pockets that give us the order for all points and show that if the point set has a 2-convex polygonization, but no star-shaped polygonization, such a triple must exist. The polygonization is found by iterating over all pocket triples. Instead of choosing a polygonization of a pocket, we only consider the $O(n^4)$ possible pairs of inflection edges for each lid. We show that the choice of the inflection edges suffices to find a 2-convex polygonization, if one exists.

8.3.4 Observations and Lemmas

Since no inflection line can be a 4-stabber (see Lemma 8.4), we make the following observations for 2-convex polygons.

Observation 8.6. *Consider the pair of inflection lines of a pocket. The lines must not cross any other part of this pocket.*

This immediately implies the next observation.

Observation 8.7. *For the pair of inflection lines of any pocket, an intersection between them occurs either at both the inner or both the outer rays.*

Consider the 2-convex polygon drawn in Figure 8.3. Any line that passes through a pocket twice can only pass through ∂P two more times. In particular, if such a line also passes through a point of the point set not in the pocket, it separates the neighbors of that point along ∂P. Therefore, the order in which the points appear along the polygonization is constrained by the pocket. We formalize this in the following lemma; see Figure 8.3 for an accompanying illustration.

Lemma 8.8. *Let P be a 2-convex polygon and let e_1 and e_2 be the inflection edges of a pocket K directed from the convex to the reflex vertex. Without loss of generality, c_1 is left*

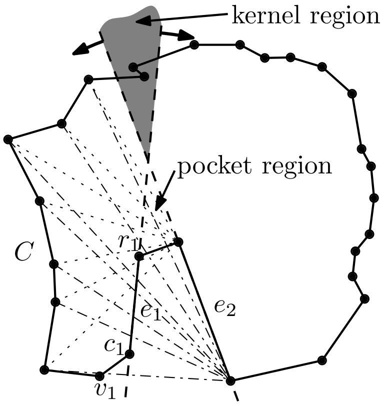


Figure 8.3: The order of the vertices defined by the inflection edges of a pocket. The gray wedge is the kernel region.

of e_2. Let C be the part of ∂P defined by the vertices that are to the left of e_2 and not part of the pocket (starting at v_1, the left endpoint of the lid of K). Then the order of the points in C is the same as the radial order around any point p on e_2. This also holds for any point on e_1 and the points of ∂P to the right of e_1.

Proof. We claim that a ray r starting at $p \in e_2$ and contained in the left halfplane of e_2 cannot cross C more than once. Otherwise, consider the supporting line ℓ of r. If p is an extreme point, slightly perturb ℓ such that it crosses ∂P twice in a neighborhood of p. In any case, ℓ (or its perturbation) crosses the pocket twice, once through e_2 and once to the left of e_2. There is another crossing with ∂P to the right of e_2. Crossing C more than once would make ℓ a 6-stabber, contradicting 2-convexity. Therefore the order around p is the same as the order in C. □

Again let $e_1 = c_1r_1$ and $e_2 = c_2r_2$ be the two inflection edges of a pocket in counterclockwise order. Further, let $\mathcal{H}^-(ab)$ and $\mathcal{H}^+(ab)$ be the closed half-planes to the left and to the right, respectively, of the directed line through the points a and b. We associate two regions to each pocket; see again Figure 8.3. The *kernel region* of the pocket is the intersection of $\mathcal{H}^-(c_1r_1)$, $\mathcal{H}^+(c_2r_2)$, and, if $r_1 \neq r_2$, $\mathcal{H}^-(r_1r_2)$. For a trivial pocket (i.e., only a convex hull edge), the kernel region is the closed half-plane to the left of it. Analogously, the *pocket region* is the intersection of the half-planes $\mathcal{H}^+(c_1r_1)$, $\mathcal{H}^-(c_2r_2)$, and $\mathcal{H}^-(r_1r_2)$ if $r_1 \neq r_2$; if $r_1 = r_2$, then the pocket region is the empty set. Lemma 8.8 tells us that once we know one pocket, the remaining polygonization is fixed except for the points in the kernel region. The most sophisticated part of our proof will be concerned with determining the pocket (which is, as we will see, relevant when there are no points in the kernel region but in the interior of the pocket region).

8.3.4.1 Pocket Triples

For the next lemma, we need a strong result by Helly.

Theorem 8.9 (Helly's Theorem [91], [158, p. 70])**.** *Let F be a finite family of convex sets in $\mathbb{E}^n$ containing at least $n+1$ members. A necessary and sufficient condition that all the members of F have a point in common is that every $n+1$ members of F have a point in common.*

Lemma 8.10. *If a point set S admits a 2-convex polygonization P that is not star-shaped, then there exist three pockets of P that completely determine P.*

Proof. The kernel of a polygon is determined by the intersection of all half-planes to the left of the edges. For each pocket, the kernel region defines this intersection for all the edges of that pocket. Therefore the kernel of P is determined by the intersection of all the kernel regions. Since P is not star-shaped, its kernel is empty. Thus, due to Helly's Theorem, there must exist a triple of pockets such that the intersection of their kernel regions is empty. Since the order in the polygonization is now determined for all vertices due to Lemma 8.8, the result follows. $\square$

Checking all triples of possible pockets and the consistency of the implied orders clearly gives us a 2-convex polygonization if one exists. There may, however, exist an exponential number of pocket candidates for any lid. But there are only $O(n^4)$ possible pairs of inflection edges per lid. For every pair of inflection edges we distinguish two cases:

- If the kernel region contains points of S, we show that the inflection edges completely determine the pocket; we can then check every pocket triple according to Lemma 8.10.
- If the kernel region does not contain any point of S, the pocket is not defined for the part in the pocket region; however, any valid pocket with these two inflection edges determines the whole remaining polygonization, and we show how to find such a pocket in polynomial time, if one exists.

We first prove the case with a non-empty kernel. The second case is more involved and is handled in Section 8.3.4.2.

Lemma 8.11. *Given only the lid and the inflection edges of an unknown pocket in a 2-convex polygonization, the convex vertices of that pocket are determined.*

Proof. Let v_1, e_1, c_1, and r_1 be defined as in the proof of Lemma 8.8. If $v_1 \neq c_1$, then there is a triangular region t defined by $\mathcal{H}^-(c_1 r_1)$, $\mathcal{H}^+(v_1 c_1)$, and the closed half-plane to the left of the lid. Due to the characterization of 2-convex polygons in Lemma 8.5, the convex chain between v_1 and c_1 is defined by the convex hull of the points in t (after removing the edge $v_1 c_1$). The second convex chain can be determined symmetrically. $\square$

Lemma 8.12. *Suppose a pair of inflection lines of a polygonization P of a point set S defines a kernel region containing points of S. Then the corresponding pocket is determined by the inflection edges.*

Proof. What is left after Lemma 8.11 is to determine the vertices of the reflex chain. Obviously, all vertices of the reflex chain must be in the pocket region. We claim that all points in the pocket region are in the reflex chain. Suppose there is a point p inside the pocket region that does not belong to the reflex chain. Then the part of the polygonization

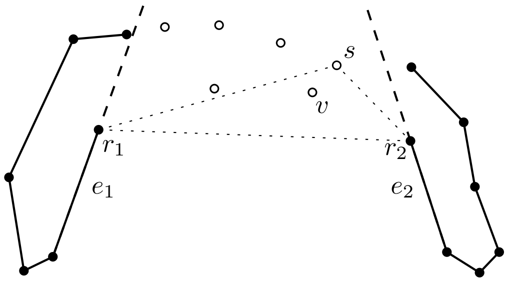


Figure 8.4: A mighty pocket. The empty dots depict points of T, of which s dominates v.

other than the pocket must pass through p. If it enters and leaves the pocket region through the same inflection line, the inflection line is a local 3-stabber, which means that there would exist a 6-stabbing perturbation of the inflection line. Otherwise, each inflection line is traversed once. Since there are still points of S in the kernel region, the polygonization crosses at least one inflection line again, and thus one of the inflection lines could be perturbed to a 6-stabber. Therefore, all points in the pocket region belong to the reflex chain. □

8.3.4.2 Mighty Pockets

The more complicated case arises if there are no points of S in the kernel region, but some points in the pocket region. Note that this case may occur when either the inner or outer rays cross, but we do not need to distinguish between these two possibilities. Let $T \subset S$ be the subset of points in the pocket region. Recall that the points in T are the ones for which we do not know the position in a 2-convex polygonization of S. We now have to split T into the vertices of the reflex chain of the pocket and the rest, which then define the part of the 2-convex polygonization that passes through the pocket region but is not part of the pocket. We call the latter the *opposite chain*. It follows from Lemma 8.8 that after we have correctly split T, the whole 2-convex polygonization is determined. We call such a pocket *mighty*.

Consider two points $s \in S, v \in T \subset S$ and the inflection edges $e_1 = c_1r_1$ and $e_2 = c_2r_2$. Suppose the triangle r_1r_2s contains v. We then say that s *dominates* v. See Figure 8.4 for an illustration. Note that s might not be an element of T. Nevertheless, we have to check the dominance in order to get a subset of T that contains only non-dominated vertices; as soon as we have decided which of these points should be part of the opposite chain, we know the order in which all points appear along ∂P.

Lemma 8.13. *If s dominates v, then v has to belong to the reflex chain, and s cannot be part of the reflex chain.*

Proof. Having s in the reflex chain would contradict the chain's reflexivity. The points s and v have a different radial order around r_1 and r_2; if none of them were in the reflex chain, these different orders would contradict Lemma 8.8. □

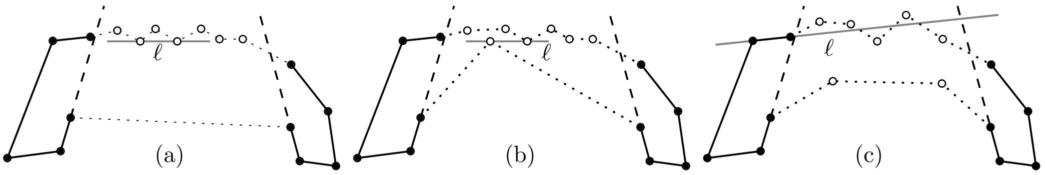


Figure 8.5: Possible conflicts: an inner tangent ℓ (a) between two vertices on the opposite chain; (b) between a vertex on the reflex chain and a vertex of the opposite chain; and (c) a 4-stabbing inflection line ℓ.

Note that the polygonization is already determined for all points not in T. From an algorithmic point of view, this polygonization needs to be checked for 2-convexity, and its pockets may also determine some points that have to belong to the reflex chain and that must go to the opposite chain. A conflict implies that such a polygonization does not exist.

So far, we might not have decided the position of all points of T in the polygonization. There exist configurations with $|T| \in \Theta(n)$ in which any point can be put either to the opposite or the reflex chain, resulting in an exponential number of 2-convex polygonizations. On the other hand, there exist configurations that do not allow a 2-convex polygonization at all. Also note that the two chains might not be linearly separable. In the following lemmas we develop a constructive approach for finding a 2-convex polygonization, if one exists, in polynomial time. More precisely, we try to find a polygonization with the given inflection edges having the smallest possible number of vertices on the reflex chain. See Figure 8.5 for some illustrations of possible conflicts.

Let an *intermediate* polygonization be a polygonization that fulfills the following two properties:

- The radial order of the points not on the mighty pocket around any point on an inflection edge of the mighty pocket is the same as on the polygonization (in conformance with Lemma 8.8).
- All points contained in the reflex chain of the mighty pocket have to be in the reflex chain in every 2-convex polygonization of the underlying point set with the chosen inflection edges of the mighty pocket.

In particular, the first property implies that the sub-chains consisting of points not in T are the same in any intermediate polygonization.

The basic idea of the algorithm is to apply a greedy approach. We build an intermediate polygonization with as few points on the reflex chain of the mighty pocket as possible. If it is not a 2-convex polygonization, we find further points that have to be on the reflex chain of the mighty pocket in every 2-convex polygonization of the underlying point set with the chosen inflection edges of the mighty pocket. Then we iterate on the new intermediate polygonization until a 2-convex polygonization is found or there is an unresolvable conflict. We start by showing some properties of intermediate polygonizations.

Let us first consider possible inner tangents in an intermediate polygonization. In the following, let ℓ be an inner tangent defined by the vertices t and t' in an intermediate

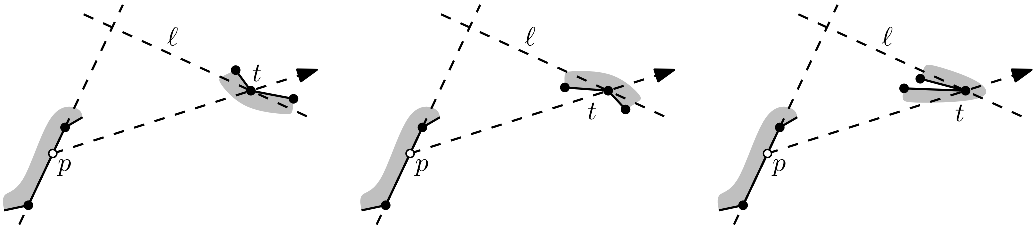


Figure 8.6: The inflection edges of the mighty pocket are on the other side of ℓ from the neighbors of the tangent points (left). The contrary case would disrespect the order induced by the inflection edges (middle and right). The gray regions depict parts of the polygon's interior.

polygonization (we call t and t' the *contact points* of ℓ). We will see later that the point set only allows a 2-convex polygonization using the two inflection edges of the mighty pocket if both t and t' are points in T. However, to obtain this result, we make no assumptions on t and t'. Lemma 8.13 gives us the following property.

Corollary 8.14. *In an intermediate polygonization, the two inflection edges of the mighty pocket are on the same side of an inner tangent* ℓ.

Obviously, we have two different types of inner tangents, one where both t and t' are not part of the mighty pocket in the intermediate polygonization, and one where one point is on the mighty pocket and the other is not. For both types, the following result holds.

Lemma 8.15. *Let* p *be any point on an inflection edge of the mighty pocket and, without loss of generality, let the inflection edges of the mighty pocket be below the inner tangent* ℓ. *Suppose, without loss of generality, that* t *is not part of the mighty pocket. Then the two neighbors of* t *are above* ℓ *and the line through* p *and* t *crosses* ∂P *at* t.

Proof. Recall that there are no points of S in the kernel region of the mighty pocket. The result immediately follows from the fact that the order of all points with respect to any point p on the inflection edges is determined (as stated in Lemma 8.8). If the neighbors of t (a contact point not at the mighty pocket) would also be below ℓ, the ray starting at p passing through t would not leave the polygon at t, which contradicts the order determined by the mighty pocket (see Figure 8.6). As shown in Lemma 8.8, the supporting line of the ray would be at least a 6-stabber, as it would have to leave the polygon at another point. Similarly, if the neighbors of t are above ℓ but the ray does not leave the polygon at t, there is as well a contradiction with the order of the polygon. □

We will use the previous lemma to show that both t and t' have to be in the mighty pocket in the final 2-convex polygonization, if there is one. If we do not have an inner tangent but the intermediate polygonization is not 2-convex, then there has to be a 4-stabbing inflection line. The next lemma will allow us to assume a certain structure of the pockets when there is no inner tangent.

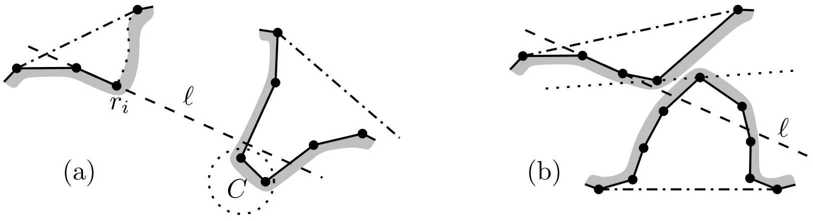


Figure 8.7: A conflict induced by the inflection line ℓ. The vertices in C are separated from the rest of their pocket by ℓ. If the mighty pocket is involved in the conflict (b), there also exists an inner tangent (dotted).

Lemma 8.16. *Consider an intermediate polygonization that contains a pocket with more than two inflection edges. Let* e_1 *and* e_2 *be the first and the last inflection edge, respectively, when traversing the pocket. Then either it is the case that one of the two corresponding inflection lines crosses that pocket at least two more times, or there also exists an inner tangent with both tangency points contained in that pocket.*

Proof. The proof is similar to that of Lemma 12 in [5]. Without loss of generality, we assume that the lid of the pocket is horizontal, and the polygon is below it. Let e_1 be the first inflection edge encountered when traversing the pocket counterclockwise. Let p_k be the point of the pocket with lowest y-coordinate; p_k is obviously reflex. Let the line ℓ be the supporting line of e_1. If ℓ crosses the pocket another time we are done. Otherwise, rotate ℓ clockwise keeping it supporting the chain between e_1 and p_k. If there is a convex vertex between e_1 and p_k, then we will find an inner tangent having a contact point at the pocket. The same argument holds for the other side with e_2. □

We consider now the situation where there is no inner tangent. Since the radial order of all points not in the mighty pocket is fixed for any intermediate polygonization, the relative position of an inflection edge $e_i = c_i r_i$ and the mighty pocket is determined. We formalize this in the following two lemmas.

Lemma 8.17. *Consider an intermediate polygonization without inner tangents but containing an inflection edge* e_i *supported by a 4-stabbing inflection line* ℓ*. Then* ℓ *cannot intersect the mighty pocket.*

Proof. Let $x_1, \ldots, x_k$ be the sequence of points where the inner ray of e_i crosses ∂P. Now suppose x_j and x_{j+1} are the two crossing points of ℓ with the mighty pocket (see Figure 8.7 (b)). Again, any ray starting at a point p on an inflection edge of the mighty pocket crossing e_i has to leave the polygon through e_i, which follows from the order induced by the inflection edges of the mighty pocket, as already handled in the proof of Lemma 8.15. Consider the shortest path inside the intermediate polygon from the convex vertex of e_i to x_{j+1}, the second intersection point of the inner ray with the mighty pocket. This path has at least one left turn and one right turn. Therefore one of the edges of the path would define an inner tangent; a contradiction. □

Intuitively, if we want to "repair" a situation in which a 4-stabbing inflection line occurs, we move the points that are "cut off" by the inflection line to the mighty pocket (if possible). We argue about the position of such points in the following lemma.

Lemma 8.18. *Consider an intermediate polygonization without inner tangents but containing an inflection edge e_i supported by a 4-stabbing inflection line ℓ. The first two crossings of the inner ray of e_i with ∂P partition it into two sub-chains. Among these two chains, let C be the one that does not contain e_i (see Figure 8.7). Then C is on the same side of ℓ as the inflection edges of the mighty pocket.*

Proof. This again follows directly from Lemma 8.8, with similar arguments as in the proof of Lemma 8.17. Without loss of generality, let the mighty pocket be to the right of $e_i = c_i r_i$. Let x_1 and x_2 be the first two crossing points of the inner ray of e_i with ∂P in the order as they occur along the ray. The inner ray leaves and then enters the polygon at these points. (Intuitively, ℓ "cuts off" C at x_1 and x_2.) Suppose the points of C are to the left of e_i. From Lemma 8.8 we know that ∂P turns left at r_i. However, the shortest path from c_i to x_2 inside P has to turn right again before reaching x_2 (at some point of C). Hence, there is an inner tangent, a contradiction. ☐

We have now obtained enough insight into the structure of the intermediate polygonization to state the main lemmas for assigning the points in T to a chain. For both of the following lemmas, recall the invariant that, for a given choice of inflection edges of the mighty pocket, all points that are in the reflex chain of the mighty pocket in an intermediate polygonization have to be there in any 2-convex polygonization of the underlying point set.

Lemma 8.19. *Let t and t' be two tangency points of an inner tangent ℓ in an intermediate polygonization. Then both t and t' have to be in the reflex chain in any 2-convex polygonization of the underlying point set (with the given inflection edges of the mighty pocket).*

Proof. See Figure 8.8. We know that any point that is not fixed is either at the opposite chain or at the reflex chain of the mighty pocket in any 2-convex polygonization, if one exists.

First, suppose that neither t nor t' is part of the reflex chain. Then we know due to Lemma 8.15 that their neighbors are on the other side of the inner tangent ℓ, say above it, and that any point p on one of the inflection edges of the mighty pocket t defines a line that separates the two neighbors of t. The same holds for p and t'. Let ℓ' be a perturbation of ℓ that is a 6-stabber. To prevent ℓ from being an inner tangent and moving neither t nor t' to the reflex chain, one would have to get rid of some of the edges adjacent to them in some way. Suppose we repolygonize the point set with a neighbor of t now being part of the reflex chain (which is the only way of getting rid of an edge). Both ℓ and ℓ' then would have to cross the reflex chain twice. However, this can only happen either to the left of pt, between pt and pt', or to the right of pt', and the reflex chain cannot pass through the rays starting at p more than once. Hence, at most two edges adjacent to t and t' can be removed. However, ℓ' now crosses the reflex chain twice and therefore remains a 6-stabber. Thus, at least one of t or t' must be part of the reflex chain.

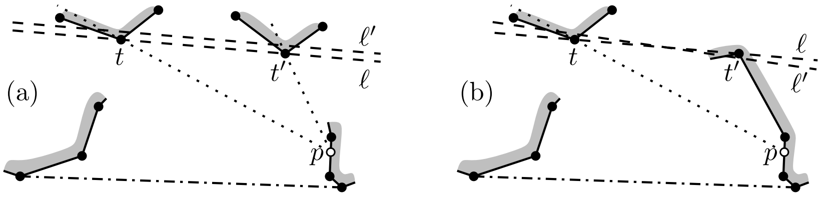


Figure 8.8: Moving a neighbor of t to the reflex chain either preserves the conflict or results in a non-simple polygonization.

Suppose now, without loss of generality, that t' is part of the reflex chain. Again, let the two inflection edges of the mighty pocket be below ℓ and t' be to the right of t. For any point p on the inflection edges, the supporting line of p and t separates the neighbors of t. Therefore only the right edge adjacent to t can be removed by changing the reflex chain while keeping t on the opposite chain, but since the number of times ℓ' crosses the boundary of the polygon is even, ℓ' remains a 6-stabber. Hence, also t has to be part of the reflex chain. The lemma follows. □

Lemma 8.20. *Consider an intermediate polygonization without inner tangents but containing an inflection edge e_i supported by a 4-stabbing inflection line ℓ. Let C be a part of a pocket that is separated by ℓ from the polygonization (as in Lemma 8.17). Then the points of C must be part of the reflex chain of the mighty pocket in any 2-convex polygonization of the point set (with the given inflection edges of the mighty pocket).*

Proof. See Figure 8.7 (a). Due to Lemma 8.16 we can assume that the inflection edge e_i that causes the conflict is the first or the last inflection edge encountered when traversing its pocket. Due to Lemma 8.17, we know that C is not part of the reflex chain. Suppose we do not want all of the points in C be part of the reflex chain of the mighty pocket. Again, let ℓ' be a perturbation of ℓ that is a 6-stabber. We can now use exactly the same line of argument as in the proof of Lemma 8.19; the reflex chain would have to pass through ℓ', but we can only remove at most two edges crossed by ℓ'. The only difference is that the conflict might be resolved after just adding the points of C to the reflex chain of the mighty pocket, but not when adding only the reflex vertex of e_i (and keeping some points of C on the opposite chain). The lemma follows. □

These lemmas now immediately imply an algorithm for finding a valid reflex chain for a 2-convex polygonization with a given pair of inflection edges for a mighty pocket (i.e., the pair of inflection edges defines a kernel region not containing any point of S). We start with an intermediate polygonization that includes all dominated points on the reflex chain. If we find an inner tangent (in $O(n \log n)$ time), then we add both vertices involved to the reflex chain. If there is no inner tangent but a 4-stabbing inflection line, we add the points of C of Lemma 8.20 to the reflex chain. During any step we know that all points in the reflex chain have to be there. Hence, we either arrive at a 2-convex polygonization after adding $O(n)$ points, or we cannot change the position of a point, which means that there is no 2-convex polygonization of the underlying point set with the given pair of inflection edges of the mighty pocket. This implies the following lemma.

Lemma 8.21. *Whether two inflection edges can be completed to a mighty pocket in a 2-convex polygonization using the points of* T *can be decided in* $O(n^2 \log n)$ *time.*

8.3.5 Putting Things Together

The overall algorithm for checking 2-convexity of a point set is the following.

1. Check whether there is a star-shaped 2-convex polygonization by creating the arrangement of all lines defined by two points of the set S. Radially sort the points around a pivot in each cell inside $\mathrm{CH}(S)$ and check all the resulting polygonizations for 2-convexity.

2. For each convex hull edge (i.e., a lid), iterate over all possible inflection edges that have no points of S in the kernel region. Try to construct a mighty pocket giving a 2-convex polygon (see Section 8.3.4.2).

3. If there is no mighty pocket, check all triples of lid/inflection-edge combinations having points of S in their kernel regions (see Section 8.3.4.1).

Theorem 8.22. *2-convexity of a point set can be decided in time polynomial in the size of the point set.*

While we achieved our goal of showing that the problem is solvable in polynomial time, the approach we propose is far from being efficient. Clearly, checking all triples of pocket candidates is the most time-consuming step. There are $O(n^{12})$ choices for the inflection edge combinations. For each pair of inflection edge candidates, there are at most two possible lids, and these can be stored beforehand for every inflection edge candidate. Since we can also store the radial order of the point set around each point of the set, we only need linear time to check whether the orders induced by the inflection edge candidates are compatible. This approach leads to a running time of $O(n^{13})$.

8.4 Deciding k-Convexity of Point Sets

The algorithm shown in the previous section is quite involved, but has polynomial running time. A natural next step is to consider algorithmic properties when the degree of convexity is increased. This section shows NP-completeness of the problem of deciding whether a point set in the plane allows a 3-convex polygonization. The proof can easily be adapted for any higher degree of convexity.

For ease of presentation we first consider the setting where some edges of the polygonization are fixed and then extend the result to point sets without any fixed edges.

8.4.1 Fixed Edges

Our proof of the following proposition can be seen as purely instructional, as it is not directly used for showing NP-hardness of the problem without fixed edges. The goal is to give the general idea of the construction, and to address the parts we have to alter later when no edges are fixed.

Proposition 8.23. *Let S be a set of points in the plane and let E be a set of edges with $E \subset S \times S$. Suppose there exists a polygonization of S that contains all edges of E. Then it is NP-complete to decide whether there exists such a polygonization that is 3-convex.*

Note that the problem is in NP as k-convexity of a polygon can be decided in quadratic time [5]. Further note that E is required to allow polygonizations of S, as otherwise the problem would be at least as hard as the NP-complete problem of deciding the existence of a polygonization of a set of line segments [137], rendering the result meaningless.

The NP-completeness is shown by reducing 3SAT [72, p. 259] to our problem. We build gadgets using fixed edges that represent the variables, literals, and clauses of a 3SAT formula and show that there exists a 3-convex polygonization if and only if the given formula is satisfiable. We refer to a *literal* as the occurrence of a variable within a single clause (negated or unnegated). Hence, a literal occurs only once in a formula.

For any given 3SAT formula ϕ, let V_ϕ be the set of its variables, L_ϕ the set of its literals, and C_ϕ its set of clauses. Further, let T be a temporary point set in convex position consisting of three disjoint sets T_V, T_L, and T_C (we will later replace them by other points) in which each point corresponds to a variable, literal, or clause of ϕ, respectively. Place the points of T in convex position such that the points of each group are consecutive on the convex hull boundary of T. Further, every triple of points in $T_V \times T_L \times T_C$ should define a triangle that is “roughly equilateral” (this latter informal requirement is intended to ease the presentation of the construction). The literal points are sorted by the variable they represent and unnegated literals of a variable are encountered before the negated literals when traversing the points of T_L on $\partial\,\mathrm{CH}(T)$ counterclockwise. Between each consecutive pair of the same class, place another temporary point. The set of these points is called T_S. Let $T' = T \cup T_S$.

In the final construction each point in T' is replaced by a corresponding gadget. In order to obtain a valid reduction we have to ensure that we can place the points in polynomial time, in particular, the coordinates of all points need to have a representation that is polynomial in the size of ϕ. One way to do this is to select all the points that are on the convex hull of the final construction among the dense rational points on the unit circle (Canny et al. [40] provide appropriate algorithmic tools), and those from inner points that are adjacent to these points in the final construction on a smaller circle (where the difference in the radii of the circle depends on the number of gadgets needed). The reader can observe throughout the description of the gadgets that the remaining points can be chosen with respect to the arrangement of the supporting lines of these points.

8.4.2 Gadgets

The basic idea is that the information in the construction is transported by lines that are potential 8-stabbers, between the variable gadgets and the literal gadgets, and between the literal gadgets and the clause gadgets. We call sets of such lines common to two gadgets a *beam*[1]. More precisely, a beam is defined by the union of potential 8-stabbers through a gadget pair. Hence, the beams "transport" the truth assignment of variables.

We introduce the gadgets by describing their intended behavior. We then show that the gadgets actually have to behave in the intended way. Note that the graphical representations of the gadgets are sketches.

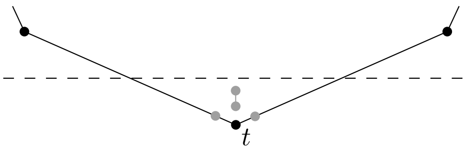


Figure 8.9: Placement of some gadget (gray), replacing a temporary point t, in order to prevent a line passing through three gadgets.

Every gadget replaces a point $t \in T$ and therefore some of its parts are in extreme position. The gadgets need to be "small" enough such that there is a line through the two edges incident to t that separates the gadget from the remaining domain; see Figure 8.9. This, in connection with the construction of the gadgets, will ensure that there exists no 8-stabber through gadgets of the same class.

8.4.2.1 Variables

The variable gadget is shown in Figure 8.10. The dotted lines in Figure 8.10 are part of beams leading to literals of the variable, one to an unnegated literal and the other to a negated literal. Note that several beams pass through a variable in this way, one for each literal of the variable. The intended behavior of the variable gadget with assignment "true" is that no line to the unnegated literals is a local 3-stabber but the lines to the negated ones are local 3-stabbers, and vice-versa (see Figure 8.10 (b) and (c), respectively).

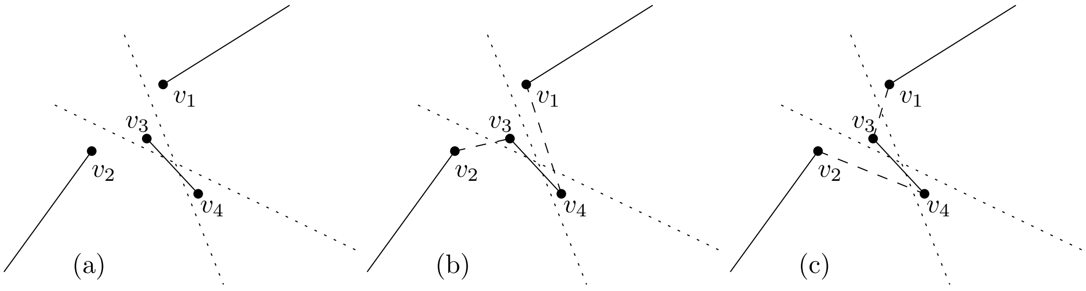


Figure 8.10: A variable gadget: its fixed edges (solid), its intended polygonizations (dashed) for true (b) and false (c), and two potential 8-stabbers in it (dotted).

---

[1] Culberson and Reckhow [48] use a similar terminology.

8.4.2.2 Literals

The literal gadgets relate each clause to the variables contained in the respective clause. The literals of a variable are placed on neighboring points of T, the unnegated literals below the supporting line of v_3v_4, and the negated literals above. Figure 8.11 shows two literal gadgets x_i and $\neg x_j$ and their interaction with their variable x.

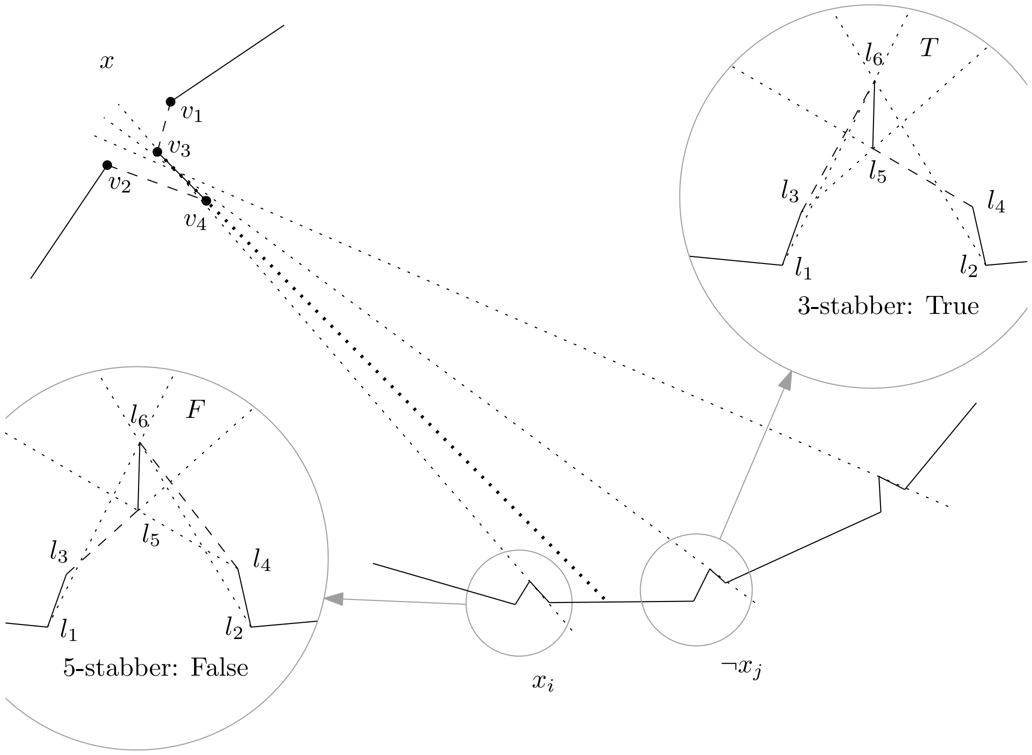


Figure 8.11: The interaction between literals and their variable.

In the example in Figure 8.11, the variable is set to false, as indicated by the dashed edges v_2v_4 and v_1v_3. Literals are defined by the fixed edges l_1l_3, l_2l_4, and l_5l_6. Their beams are defined by the lines through l_1l_6 and l_3l_5, and the lines through l_2l_6 and l_4l_5. Let the former beam be the *clause beam* and the latter the *variable beam*. Note that the edges have to be short enough and need to be placed appropriately such that the variable beam is narrow enough to pass through v_3v_4. (This can, e.g., be done by first choosing l_1, l_2, l_3, and l_4; the arrangement of supporting lines of the points together with v_3, v_4 and points at the clause gadgets defines a convex region in which the edge l_5l_6 can be placed.)

Consider literal x_i. Its variable beam contains local 3-stabbers at the variable, which is therefore assigned to “false”. Hence, the polygonization of x_i is chosen such that the variable beam does not contain any local 5-stabbers at x_i. The variable beam of literal $\neg x_j$, however, contains no 3-stabbers at the variable. Therefore it can be polygonized the opposite way to x_i. This makes the clause beam of $\neg x_j$ contain no local 5-stabbers at the literal, whereas the clause beam of x_i already contains local 5-stabbers. We define a literal to be “false” if its clause beam contains local 5-stabbers, and true otherwise. The intended behavior of the literal construction is that an unnegated and a negated literal of the same variable cannot both be true (they may both be false but that obviously does not influence the satisfiability properties of the formula).

8.4.2.3 Clauses

We defined a clause beam to transport an assignment of "false" if it contains lines that are local 5-stabbers at the literal. This means that its lines cannot be local 3-stabbers at the clause. The clause gadget is constructed in a way that it allows a 3-convex polygonization if at least one of the beams does not contain any lines that are local 5-stabbers at the variable, which therefore can also be local 3-stabbers at the clause. We will later show that a 3-convex polygonization exists only if one of the beams of each clause transports "true".

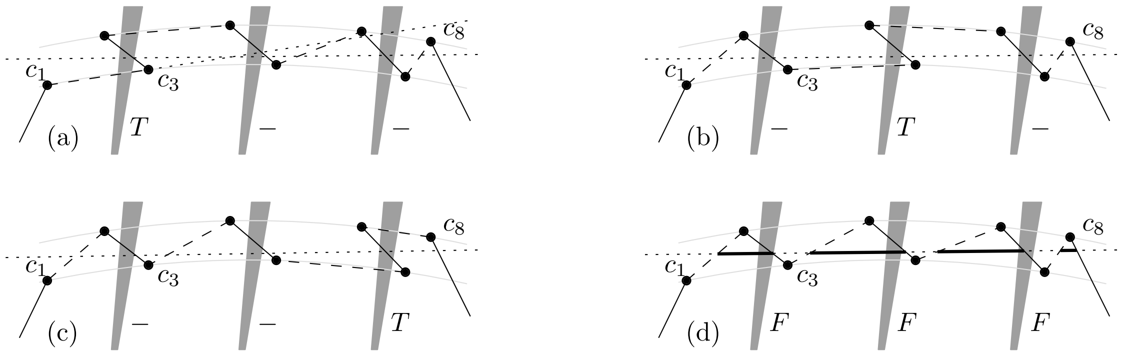


Figure 8.12: The clause gadget and its expected polygonizations (a–c). If no beam contains a local 3-stabber at a clause gadget then there has to be an 8-stabber, indicated by the dotted line and the fat segments (d).

The clause gadget is shown in Figure 8.12. The beams pass through fixed edges whose end vertices are placed along two flat arc segments (light gray). If one of the beams is true, a 3-convex polygonization can be done as shown in (a) to (c). Note that in (a) the point c_8 is below the line through c_1c_3. To have no local 3-stabber in any of the three beams one could sequentially connect the edges, as shown in Figure 8.12 (d). However, this would introduce an 8-stabber, as depicted by the bold-style segments. The intended behavior of a clause gadget is that there is no 3-convex polygonization if all its literals are "false". Observe that lines that are local 5-stabbers at the gadget (e.g., a line passing by close to c_8 in Figure 8.12 (b)) leave any polygonalization in a close neighborhood of the gadget if the two flat arcs on which we place the points are sufficiently close to each other. Hence, no line passing, say, through two clause gadgets can become an 8-stabber. (We give a short account on placing all points such that they have coordinates with a polynomial representation at the end of Section 8.4.6.)

8.4.3 Necessity of Satisfiability

By construction, such a set of edges allows a 3-convex polygonization whenever the formula is satisfiable. What remains to be shown is that a non-satisfiable formula prevents a 3-

convex polygonization; i.e., that the gadgets behave in the intended way. The major difficulty in showing this is that the whole configuration needs to be considered. It is insufficient to inspect the gadgets only locally. We can, however, restrict our attention to the local behavior with the help of a construction we call a separator.

A *separator* gadget is constructed by slightly moving apart the two convex hull edges incident to a temporary point of T_s. The resulting gap is filled by edges as shown in Figure 8.13. A line in the beam is a local 5-stabber at the separator. Together with the antipodal edge through which the beam passes, such a line becomes a 6-stabber and therefore there cannot be any more edges crossing the separator beam. If we place a separator between all neighboring gadgets (see Figure 8.14), we may return to our local view.

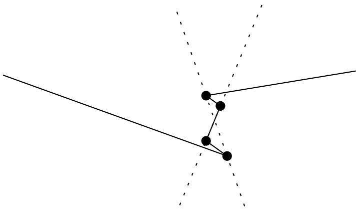


Figure 8.13: A separator produces a beam of 6-stabbers between the two dotted lines.

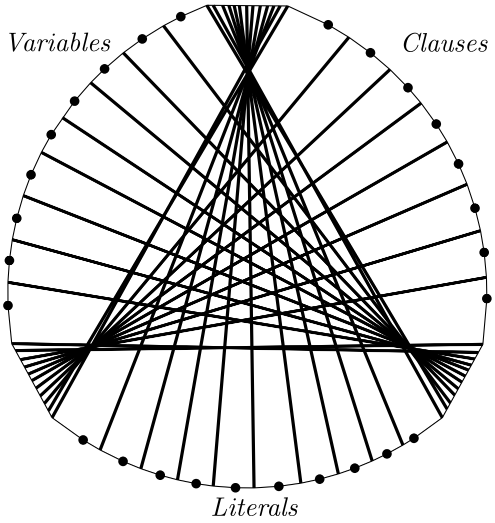


Figure 8.14: The whole configuration representing a formula. The dots denote the position of the gadgets of the formula, and the thick strokes denote the beams of the separators.

The following simple observations are useful when proving the correct behavior of the gadgets.

Observation 8.24. *When walking along the boundary of a polygon, any intersection of half-planes is entered as many times as it is exited.*

Observation 8.25. *A polygonal chain connecting two points separated by a line crosses that line an odd number of times.*

Lemma 8.26. *Given any pair of lines through a variable gadget, one to an unnegated and one to a negated literal, one of these lines must be (at least) a local 3-stabber.*

Proof. Consider again Figure 8.10. As the clause is isolated by two separators, it contains a path from v_1 to v_2, which means that the dotted stabbers are crossed locally an odd number of times. The two lines separate the plane into four regions. Let A be the one that contains v_3. A is already entered (or left) by the edge v_3v_4. This means that there has to be another edge leaving A, crossing one of the lines. As that line is crossed twice, it needs to be crossed at least a third time to result in an odd number of crossings. ☐

Lemma 8.27. *Given two literal gadgets, one representing an unnegated and the other one a negated literal of the same variable, at least one of their clause beams contains lines that are local 5-stabbers at the literal.*

Proof. Take any pair of lines, of which one is contained in the clause beam and the other one is contained in the variable beam of the first literal. Arguing analogously to the proof of Lemma 8.26, it is obvious that one of the lines is a local 5-stabber. If it is in the clause beam, we are done. As it otherwise has to be in the variable beam, we know from Lemma 8.26 that the variable beam of the second literal contains local 3-stabbers at the variable. Hence, the clause beam of the other literal contains local 5-stabbers. ☐

Note that two literals of the same variable might both be set to false, but this obviously does not impose a problem for the overall argumentation. Further note that the proofs of the previous two lemmas are kept quite general, as we will use similar techniques when proving the correctness of gadgets for point sets.

Lemma 8.28. *There is no 3-convex polygonization with a clause gadget having all its literals set to false.*

Proof. Consider again Figure 8.12(d). As the gadgets are divided by separators and all beams contain local 5-stabbers when set to false, the beams define isolated regions. As each region contains only two points, the only choice is to draw an edge between each of these pairs. This, however, yields exactly the polygonal path shown in Figure 8.12(d). As this path creates an 8-stabber, the proof follows. ☐

Proof of Proposition 8.23. As already discussed, the problem is in NP. For any given 3SAT formula ϕ we can construct a set of edges E in polynomial time representing ϕ. By construction, E has a 3-convex polygonization if ϕ is satisfiable. Lemma 8.27 shows that an unnegated and a negated literal of the same variable cannot both be true in a 3-convex polygonization of E, and Lemma 8.28 shows that if all literals of a clause are false, then there cannot be a 3-convex polygonization of E. This establishes that there is a 3-convex polygonization of E only if ϕ is true. ☐

8.4.4 General Point Sets

The proof of Proposition 8.23 relies on fixed edges. In fact, we did not make use of any isolated points. To transfer the previous result to the domain of point sets, we need a way to force edges to a more or less fixed position.

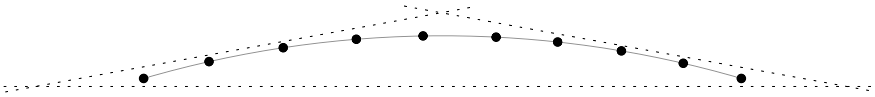


Figure 8.15: A chain of at least ten points inside a triangle defined by any three lines implies an edge between two of the points.

Lemma 8.29. *Let R be any subset of a point set S contained in the triangle defined by three lines. If $|R| > 9$, then there has to be at least one edge between two points in R in any 3-convex polygonization of S.*

Proof. See Figure 8.15 for an accompanying illustration. Suppose that there is no edge between any two points in R. Then every edge incident to a point in R must cross at least one of the lines defining the triangle region. Every line may only be crossed six times for the polygonization to be 3-convex. As every point is incident to two edges, the bound follows by the pigeonhole principle. ☐

Note that this bound may be tightened when considering that there has to be a path between the edges entering and leaving the triangular region. For ease of presentation, at least ten points are chosen.

Let such a subset of ten points along a flat arc segment be called a *bunch*. We will need that the supporting lines of the edges of the bunch lie within a given wedge. For a sufficiently flat arc segment, all the edges spanned by two points of the bunch will fulfill that property. Thus, we can suppose, without loss of generality, that the edge in the statement of Lemma 8.29 is similar in that sense to an edge which connects two successive points on the arc segment.

During the construction of the gadgets using fixed edges, the positions of the beams were fixed. We show that we can still guarantee the existence of the beams using bunches (but not their exact position). We demonstrate a construction that, by cascading bunches, allows us to place in a defined region any number of edges that all cross a common stabber.

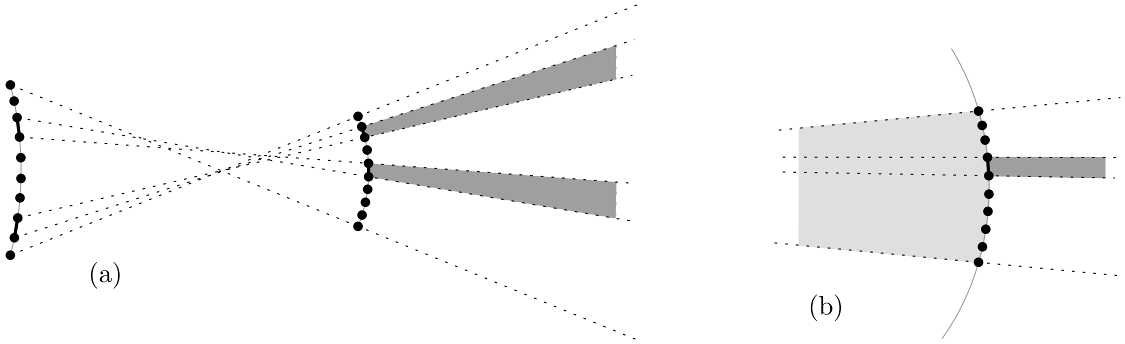


Figure 8.16: The first two bunches define the width of the hyperbeam (a). Bunches at the intersection of each beam with an arc segment increase the stabbing number of the lines in the part of the beam (b).

The main idea is to replace the fixed edges with flat arc segments on which the bunches are placed. Let a *hyperbeam* be the union of potential beams. A hyperbeam is defined by these arc segments and is directed in the same way as the beams in the previous section. Now consider the first two arc segments of a hyperbeam, as depicted in Figure 8.16(a). The length of these two segments defines the width of the hyperbeam, and they should therefore be sufficiently narrow. Place a bunch on each of these two segments. Any pair of edges, one on the first segment and the other one on the second, would define a beam within the hyperbeam defined by the two segments. Each of these beams intersects a part of the third segment. We now place a bunch at each of these intersections, which thus guarantees a local 3-stabber inside the hyperbeam (the intersections might intersect themselves, but it is only necessary that each intersection contains 10 points). This construction can be continued in the same way for all further segments. Hence we may now determine not the exact, but the approximate position of potential 8-stabbers.

8.4.5 Point Set Separators

In the previous section, the use of separators allowed the correctness of the construction to be verified. Using bunches, we can create a similar construction (as in Figure 8.17) and therefore prevent other edges from crossing such a separator. However, paths from one side of the domain to the other could still use the edges of such a separator (as shown in Figure 8.18), since we can no longer say for sure which edges are adjacent. In order to prevent this we apply the following construction, shown in Figure 8.19. We place two separators (instead of one) indicated by bunches between every formula gadget. The separator beams are directed to the antipodal side of the polygon, intersecting each other. We then place a separator on the antipodal side between the beams. Recall that by using bunches we cannot exactly define the position of a separating line, but can assure its existence somewhere inside the hyperbeam. Hence, such a separator array again allows us to consider the formula gadgets only locally.

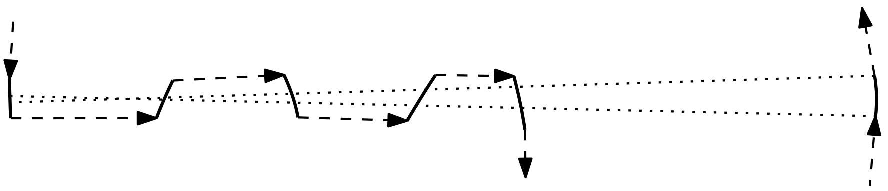


Figure 8.17: A separator constructed using points. The bold-style arc segments sketch the regions where bunches can be placed, and the dashed line sketches a possible sequence in the polygonization.

8.4.6 Adaption of the Gadgets

With the help of the bunches, the gadgets can be constructed, up to a certain extent, in a manner similar to the fixed-edges case. When placing the arc segments for the bunches like the fixed edges, a 3-convex polygonization is possible if the formula is satisfiable.

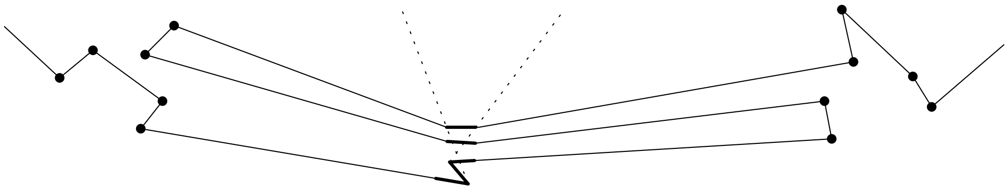


Figure 8.18: An example of an unintended behavior of a separator gadget.

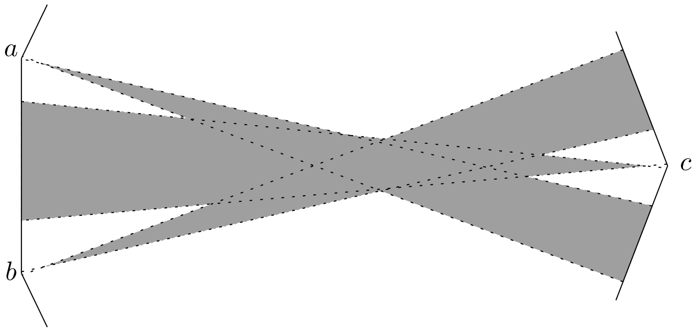


Figure 8.19: A separator array preventing unintended paths from passing through it.

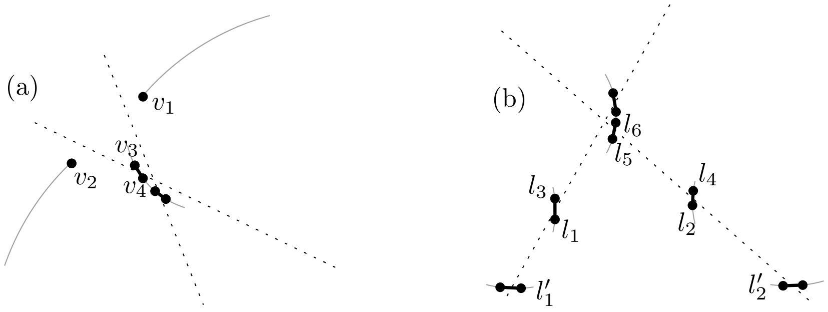


Figure 8.20: A point set variable gadget (a) and a literal gadget (b). The gray arc segments denote the position of the bunches, the dotted lines denote a pair of potential stabbers, and the solid segments denote edges that have to exist somewhere within the bunches.

A variable and a literal gadget are shown in Figure 8.20. Instead of fixing edges, we cascade the bunches as described above. The proof of Lemmas 8.26 and 8.27 can be applied directly to these gadgets, since one of the (many) edges on the innermost arc segment of the variable or the literal takes the role of v_3v_4 or l_5l_6, respectively.

Showing the correctness of the adapted clause gadget is more involved. The sketch in Figure 8.21 accompanies the description. There, the gray regions depict the hyperbeams carrying the literal assignment. When points are placed along the dark arc segments, there obviously exists a 3-convex polygonization if at least one literal is true. Suppose, without loss of generality, that h is a horizontal line. Recall that a hyperbeam is a union of beams. Place at least one point in the interior of each beam on the arc segments between c_2 and c_3, c_4 and c_5, and c_6 and c_7, respectively. Observe that all of these points are above h.

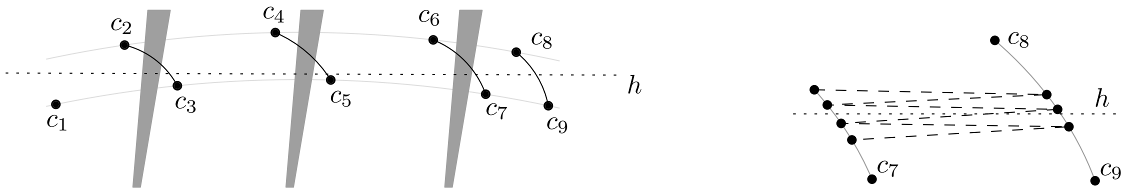


Figure 8.21: A clause gadget. The points to the right ensure the existence of an 8-stabber if all literals are assigned false.

Then place four points in the vicinity of the line h on the arc segment between c_6 and c_7, and three further points in a similar manner between c_8 and c_9, such that when sorted along the y-axis no point has its successor on the same arc segment.

Lemma 8.30. *There is no 3-convex polygonization of the point set with all literals of a point clause gadget set to false.*

Proof. The interesting case is the one where the hyperbeams contain lines that are local 5-stabbers at the corresponding literals. Since the beams may now only be crossed once, every region between two such 5-stabbers is entered and left once. Consider the dashed line h, which should be an 8-stabber if the clause evaluates to false (as in Figure 8.21). It is essential for the behavior of the gadget that in this case h is crossed twice between two beams (Property 1). Further, it has to be crossed once in the region of c_1 (Property 2) and three times in the region of c_7 (Property 3).

Property 1: Observe that if all literals for the clause are set to false, there exists a beam that contains an infinite number of 5-stabbers at each of the dark arc segments. Since there is a point placed inside the region of each such beam, there is an infinite number of 5-stabbers through the neighborhood of such a point. This means that the path consisting of the two edges incident to this point has to cross each of these 5-stabbers in that neighborhood. Recall that all these points are placed above h. Therefore, the region between such 5-stabbers (containing the region between two hyperbeams) is entered and left above h by the path defining the overall polygonization. Since the path has to “fetch” the point below h, it has to cross h twice within that region.

Property 2: The line h is obviously crossed within the region of c_1 and c_2, since the gadgets are isolated by separator arrays, and c_1 and c_2 are on different sides of h.

Property 3: The path enters the region above h (by the same arguments as used for Property 1) and leaves it below h, hence h is crossed an odd number of times. Suppose that the path is y-monotone through the points on the arcs. Then the path zig-zags through these seven points, provoking an 8-stabber. If the path is not monotone and crosses h only once, there is a vertex m with both edges leaving it in the same y-direction. Translate h to another horizontal line h' past m. The new line h' crosses the path twice in the vicinity of m, which means that it crosses the path three times. □

Finally, observe that we can choose the points in such a way that all coordinates are rational and have numerators and denominators that are polynomial in the size of the input. The points on the convex hull of the construction can be selected from the (dense) set of rational points on the unit circle. The additional interior points for the gadgets can be placed inside convex regions defined by the supporting lines of a constant number of pairs of initial points. That is, they can be chosen inside the solution space of linear programs with a constant number of polynomial-sized constraints. Note that the bunches for the hyperbeams do not necessarily have to lie on an arc as in the sketches; they simply have to lie inside a triangle, as demanded by Lemma 8.29, without increasing the stabbing number in an unintended way.

From these arguments our final theorem follows.

Theorem 8.31. *It is NP-complete to decide whether a point set is 3-convex.*

8.5 Abstract Order Types and 2-Convexity

Clearly, k-convexity of a point set S in the Euclidean plane does only depend on the order type of the point set. For any j-stabber ℓ, we can rotate ℓ until we obtain a line ℓ' that is the supporting line of two points in S. Depending on the relative position of the vertices before and after the points on ℓ' in the polygonization, we can determine j. While the definition of k-convexity carries over to abstract order types, our algorithm for deciding 2-convexity of S used properties of the point set that required more information than the order type. In this section, we revisit the main part that relied on the embedding and argue why 2-convexity can also be determined in polynomial time for abstract order types.

8.5.1 Star-Shaped 2-Convex Polygonizations

For deciding whether a point set S admits a star-shaped 2-convex polygonization (see Section 8.3.3), we considered the arrangement of all supporting lines of two points of S. As already discussed, this information is not encoded in the order type for all pairs of lines (and also not in the circular sequence). Being allowed only sidedness queries on the vertices of a simple polygon, we can, in general, not decide whether it is star-shaped or not, see Figure 8.22 for a counterexample. More generally, we cannot obtain the radial order of the points around the crossing of two supporting lines.

However, if the kernel of the polygonization contains a point of S, then the radial order around this point gives the polygonization. In the opposite extreme, if the pocket region of a pocket in the polygonization contains points not part of the pocket (which means that the pocket is a mighty pocket), we know that the polygonization cannot be star-shaped.

But what about all the other configurations? Consider the point set S and its arrangement of supporting lines as a generalized configuration $\mathcal{C}$ of points. Consider a polygonization of (the points of) $\mathcal{C}$ that is not star-shaped, like the example to the left of Figure 8.22. When changing the arrangement of the supporting pseudo-lines accordingly (without moving a pseudo-line over a point), we get a generalized configuration of points that can be extended by a point that then is in the kernel of the polygon. The resulting arrangement of supporting lines may even be stretchable again, like in the example to the right in Figure 8.22. It is intriguing that for one embedding we are allowed to apply Helly's Theorem to see that we have a non-empty triple of kernel regions, while for the other we

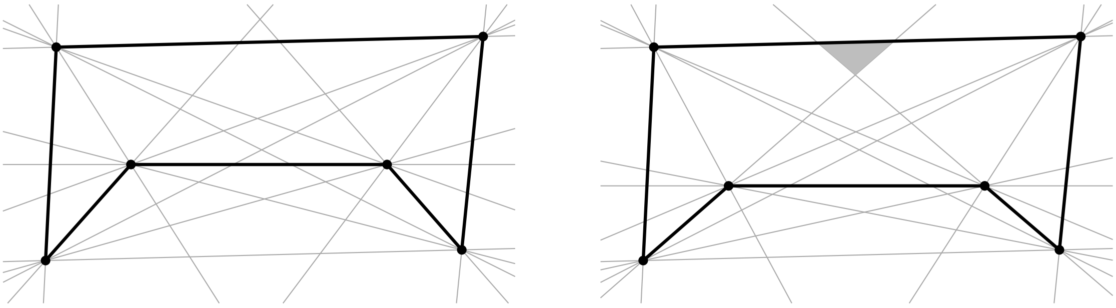


Figure 8.22: The order type of the vertices does not encode whether a polygon is star-shaped. The polygon to the left is not star-shaped, the one to the right is star-shaped, with the kernel marked gray. Both vertex sets have the same order type.

are not, even though we have two instances with the same combinatorial structure and we want to solve a combinatorial problem on them. (Streinu [154] considers the realizability of so-called star-like pseudo-polygons on generalized configurations of points; however, this class is restricted to a convex region with triangular spikes attached to it.) In order to get closer to the core of the problem (or at least to better understand our solution), we take a look at a variant of Helly's Theorem.

8.5.2 Helly's Theorem for Pseudo-Lines

Among their several seminal papers on order types and arrangements of pseudo-lines, Goodman and Pollack [79] have proven a dual of Helly's Theorem for (non-simple) arrangements of pseudo-lines.

Theorem 8.32 (Goodman, Pollack [79]). *Let $\mathcal{A}$ be an arrangement of pseudo-lines in $\mathbb{P}^2$, not all meeting in one point. If $\mathcal{A}_1, \ldots, \mathcal{A}_m$ are subsets of $\mathcal{A}$, and ψ is a point not on any pseudo-line of any $\mathcal{A}_i$, such that, for any i, j, k, $\mathcal{A}$ contains a pseudo-line $\ell_{i,j,k}$ such that ψ cannot be moved continuously to $\ell_{i,j,k}$ without meeting each of $\mathcal{A}_i, \mathcal{A}_j, \mathcal{A}_k$, then there is an extension $\mathcal{A}'$ of $\mathcal{A}$ containing a pseudo-line ℓ such that ψ cannot be moved continuously to ℓ without meeting each of $\mathcal{A}_1, \ldots, \mathcal{A}_m$.*

In the primal, we get a generalized configuration of points corresponding to $\mathcal{A}$ with ψ as the line at infinity. Not being able to continuously move ψ to $\ell_{i,j,k}$ without meeting each of $\mathcal{A}_i, \mathcal{A}_j, \mathcal{A}_k$ corresponds to the point ℓ being inside the convex hull of each of $\mathcal{A}_i, \mathcal{A}_j, \mathcal{A}_k$ in the primal.

The formulation of Theorem 8.32 suggests that the convex sets we considered for applying Helly's Theorem in Lemma 8.10 can be replaced by a combinatorial counterpart. If, for each pocket, we look at the convex hull of the (possibly empty) set of points inside its kernel region, we have a more combinatorial definition of the convex sets that we are interested in. We call this the *kernel hull* of the pocket. Theorem 8.32 tells us that if we cannot find an extension point x to the abstract order type defined by S such that x is in the kernel region of every pocket, then there exists a triple of pockets such that their kernel hulls do not share any point of S.

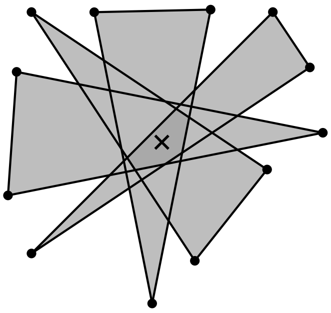


Figure 8.23: An extension of a configuration for which Theorem 8.32 does not apply.

If x could always be chosen from S, then we would be done, as we could check for each point in S whether it is inside the kernel of a 2-convex polygonization of S and otherwise we would know that there is a pocket triple whose intersection of points in the kernel region is empty. However, one can construct examples where there is an extension point x and x cannot be chosen from S (and hence, it is hopeless to try obtaining a stronger Helly-type theorem for our needs).

Interestingly, Theorem 8.32 does not allow $\ell_{i,j,k}$ to be an extension, it has to be an element of $\mathcal{A}$; this assumption somehow weakens the statement in the way it is stated compared to the setting where the convex hull is a convex subset of $\mathbb{E}^2$. Assume there is a triple that does not contain a common point of S inside its convex hull. There could still be an extension of the configuration of points by a point x such that x is inside the convex hull of each triple. However, we cannot extend the triples beforehand, since, if the extension of one triple is chosen in the wrong way, there could be triples that would have had such an extension in the original configuration, but not in the extended one. See Figure 8.23 for an example; there, the convex hulls of the subsets are shown in gray. Theorem 8.32 does not apply in that case, as no point of the initial configuration is part of the intersections of the convex hulls of any triple of subsets. Still, there is an extension (depicted by the cross). Since we need the inversion of the theorem for our purposes, this difference does not affect us.

Suppose there is an extension point x for the abstract order type of S such that x is in each kernel hull of the pockets, but there is no point in S that could take the role of x (i.e., the intersection of the kernel hulls does not contain a point of S). Observe that x can be chosen to be on the intersection of the boundaries of two kernel hulls, therefore being a crossing point of two line segments defined by four points that are contained in only two of the subsets of S. Even with this extra information, the order type does not always allow us to answer sidedness queries involving the point x. What happens if we fix the arrangement of the supporting pseudo-lines in the generalized configuration of points arbitrarily, and then proceed in checking for a star-shaped 2-convex polygonization around every node of the arrangement?

It turns out that a result in this setting can (conceptually, not algorithmically) easily be obtained. Recall that Theorem 8.32 holds also for non-simple arrangements of pseudo-lines. Consider the arrangement of supporting lines of a generalized configuration of points $\mathcal{C}$ (representing the order type of S or some abstract order type) in $\mathbb{P}^2$. We extend $\mathcal{C}$ by adding

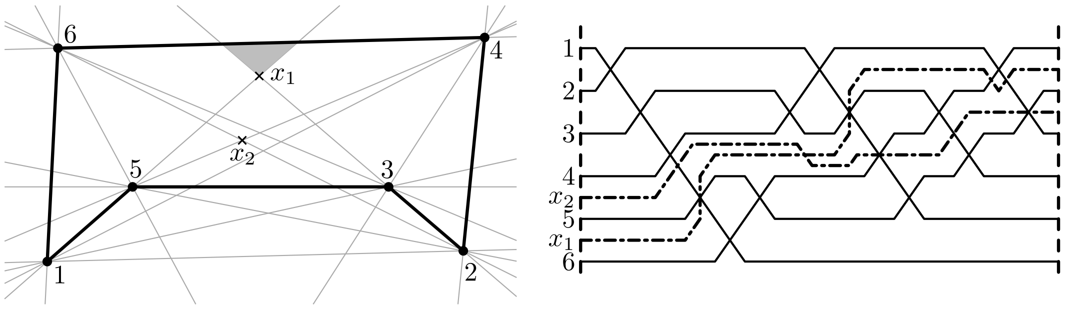


Figure 8.24: An extension of a configuration of points by two points and the corresponding wiring diagram.

a point at every crossing of two supporting pseudo-lines in CH(S) to a new generalized configuration of points $\overline{\mathcal{C}}$. See Figure 8.24 for an intermediate step in such an extension. Let X be the set consisting of these points. Note that this extension depends on the arrangement of supporting lines and not only on the order type. Now suppose there exists a 2-convex polygonization P of $\mathcal{C}$. This does not depend on the arbitrary extension of C. The kernel hull of a pocket in $\overline{\mathcal{C}}$ now contains the points of $S \cup X$ in the kernel region of the pocket (which includes the points on the boundary of the kernel region). Hence, the convex hull of the kernel extension points is identical to the intersection of the kernel region with CH(S). We can now apply Theorem 8.32 to these kernel hulls in the following way (as the theorem is applicable to non-simple arrangements). If there is an extension of $\overline{\mathcal{C}}$ by a point x such that x is inside the convex hull of each kernel hull, then there is a point $p \in S \cup X$ such that the radial order of S around p is the same as the one around x (since x is contained in a cell of the arrangement of supporting lines of S). Otherwise, there is a triple of pockets such that the union of the points in their kernel hulls in $\overline{\mathcal{C}}$ is empty, and hence, the inflection edges of the three pockets define the order of all points of S around P. Note that the radial order of X around x might differ from the one around p, but we are interested only in the order of S.

This approach is analogous to the one when we are given a realization of the point set; we do not consider the kernel hulls, but rather the kernel region itself. The choice of the kernel regions by fixing the supporting pseudo-lines of $\mathcal{C}$ is, within the constraints implied by the order type, arbitrary, but has to be consistent (recall Figure 8.22). It remains to show how to obtain the elements of X algorithmically.

8.5.3 A Conceptually Simple Extension Algorithm

Suppose we are given the dual pseudo-line arrangement $\mathcal{A}$ of S represented as a planar graph. Such a representation can be constructed in $O(n^2)$ time: Edelsbrunner, O'Rourke, and Seidel [58] gave an algorithm to construct an incidence graph of a line arrangement in $O(n^2)$ time (see also [54, Chapter 7]); this has also been shown independently by Chazelle, Guibas, and Lee [43] (in [58], arbitrary dimensions are handled as well). In both papers, the so-called *zone theorem* is used, stating that the number of edges in the faces of the graph intersected by an added line is linear. The proof given, e.g., in [43] directly extends to pseudo-line arrangements (see also [86] where arrangements of more general curves are discussed). The zone theorem allows adding pseudo-lines of the arrangement incrementally to the graph in linear time using straight-forward graph manipulation techniques.

We observe that we can also extend the arrangement by a pseudo-line through two crossings within the same time bounds due to the zone theorem. Given two crossings a and b, we can, in constant time, determine how such a pseudo-line has to pass through each of the crossings. To create a pseudo-segment between a and b, let B be the set of pseudo-lines that separates the two crossings. We walk from a to b, greedily passing only over pseudo-lines that are in B and that we did not cross so far. The Levi Enlargement Lemma (Lemma 5.3 on page 69 herein) shows that we can never "get stuck": the pseudo-lines in B are exactly those on a pseudo-segment from a to b (recall that we defined a pseudo-segment as part of a pseudo-line compatible with the arrangement); after a walk from a to some point p in the described way, the pseudo-segment between p and b crosses a subset of B. (This can actually be seen as an algorithmic interpretation of the constructive proof given by Levi [110].) The same procedure then can be done for the pseudo-lines not in B.

If we do this for all pairs of crossings in $\mathcal{A}$ we end up with an arrangement $\mathcal{A}'$ of $O(n^4)$ pseudo-lines. This takes $O(n^8)$ time, as the number of pseudo-lines of the intermediate arrangements is in $O(n^4)$ and we extend it by $O(n^4)$ pseudo-lines. The added pseudo-lines correspond to the extension points in X. When traversing such a pseudo-line, we get the radial order of the points around the primal extension points (there are actually four different possible orders, depending on which relative order we choose for the pseudo-line pair defining a crossing through which we extend the arrangement). If one such radial order defines a 2-convex polygonization, we are done. Otherwise, due to Theorem 8.32 and the above arguments, we know that there is a triple of kernel hulls whose intersection does not contain a point of S.

8.5.4 Discussion of the Extension

It is interesting to observe that this approach has some intrinsic non-determinism. Suppose the point set has a 2-convex polygonization but there is no point of S inside its kernel. The extension of $\mathcal{C}$ can be arbitrary, but depending on this choice, the 2-convex polygonization will be found as a star-shaped one or by applying Helly's Theorem. This behavior suggests that we may actually be missing a property of 2-convex point sets that is at the heart of the problem.

Looking for a star-shaped polygonization is actually an artifact of the initial geometric approach. As already discussed, we actually want to have no points in the intersection of three kernel hulls. When looking for an extension point in the interior of the intersection of all kernel hulls, we observed that this point can be represented by (a perturbation of) the crossing point of two pseudo-segments between two points of the generalized configuration. While it does not make a difference whether we extend only such points instead of all crossing points of the arrangement of supporting pseudo-lines from an asymptotic point of view, it could be interesting for an algorithmic approach that does not explicitly construct the dual arrangement and its extension. In Chapter 7, we defined pseudo-verticals and showed that there exists an extension of the underlying arrangement by the set of these pseudo-lines. However, we did not actually construct this extension, but it was possible to select an element of a given rank in linear time. To reduce the $O(n^8)$ upper bound on fixing an extension, it would be interesting whether there is an implicit representation of all crossing points in the arrangement of supporting lines. In such an implicit representation, the set of corresponding pseudo-lines could be added to the arrangement, but the extended arrangement does not have to be constructed explicitly to get the order of the original points around the extension points.

In any case, by extending the generalized configuration of points in this (rather blunt) way, we can apply the same approach as when given the point set by coordinates. For the remaining part of the proof of the algorithm, the reader may observe that we only used properties that can be evaluated by sidedness queries; the arguments merely remain the same when arguing about generalized configurations of points. Deciding 2-convexity of a given polygon [5] (see Section 8.3.2) makes use of rather involved sub-algorithms in connection with ray shooting queries. However, 2-convexity of a polygon can easily be tested using only sidedness queries in quadratic time.

8.6 Chapter Summary

In this chapter, we considered the algorithmic question whether a point set admits a k-convex polygonization, i.e., a polygonization such that every line intersects the boundary of the polygon at most $2k$ times. While 1-convexity of a point set coincides with the point set being in convex position, deciding 2-convexity of a point set seems to be a more complicated problem. We showed that deciding whether a point set admits a 2-convex polygonization is a problem that can be solved in polynomial time, but still our algorithm is far from being efficient. The problem changes for higher degrees of convexity. We gave a reduction showing that deciding 3-convexity is an NP-complete problem. The reduction can easily be modified to cover higher degrees of convexity. Finally, we gave an account on 2-convexity in connection with abstract order types and showed how to extend the concepts used for the geometric variant to the abstract version.

Note that for 2-convexity, our primary goal has been to show that the problem is solvable in polynomial time at all. In presence of the large upper bound on the running time of the algorithm, it would be interesting to know the running time of an optimal algorithm. In that respect, the most significant part is looking at all triples of inflection edge pairs. Recall that each pair defines a wedge, i.e., a convex region allowing us to apply Helly's Theorem. However, a half-plane would be sufficient, and actually, when there is no star-shaped 2-convex polygonization but a non-star-shaped one, there will be three inflection edges such that the intersection of the three half-planes they define is empty. However, when given only one inflection edge of a pocket, Lemma 8.12 (see page 117), giving us the polygonization of the pocket, is not applicable. Some further insight into the problem may allow us to use only one inflection edge of a pocket, in particular since in that case we have the information that the pocket region of each pocket contains only points of that pocket.

Concerning the application of our algorithm to abstract order types, it would be interesting whether the pseudo-lines we added to the explicit representation of the dual arrangement can be replaced by a well-defined set of pseudo-lines that are not explicitly constructed, similar to pseudo-verticals in Chapter 7. Such an extension would have to allow us to obtain the order of intersections with the other pseudo-lines of the initial arrangement in a consistent way.

Chapter 9

Conclusion

In this thesis, we presented several results on algorithmic problems on finite point sets in the plane and geometric graphs.

In the first part, we considered flips in triangulations. In Chapter 2, we investigated the properties of the double chain as a special point set with respect to the flip distance. In particular, we showed that the desired properties of the double chain also hold when it is used as a subset of a set of points. The double chain was a crucial part of our constructions in the following two chapters, where we considered the complexity of the flip distance problem for triangulations of point sets (Chapter 3) and simple polygons (Chapter 4). We showed that the problem is APX-hard for triangulations of point sets, using a reduction from MINIMUM VERTEX COVER. For this, we provided an extensive account on how to embed the points of the gadgets used in the reduction. A different approach had to be taken in Chapter 4 to show NP-completeness of the flip distance problem for triangulations of simple polygons. Our reduction from a variant of the RECTILINEAR STEINER ARBORESCENCE problem allowed us to show that the problem is NP-complete. However, the reduction does not provide any insight in the complexity of the approximation variant of the problem.

In the second part of this thesis, the focus was on algorithms on point sets, in particular on algorithms that use only sidedness queries and that work also for abstract order types. In Chapter 5, we provided a revision of the topic of combinatorial classification of point sets, describing circular and allowable sequences, order types of point sets and abstract order types, as well as of properties of the dual pseudo-line arrangements in the Euclidean and real projective plane, for use in the following chapters. Further, we provided several examples of related work that motivated the research on geometric algorithms using only a restricted set of predicates. The first algorithm we described for this setting finds an extreme point in linear time (see Chapter 6), thus solving a long-standing open problem posed by Knuth. In particular, given a pair of points from the set, the algorithm returns the edges of the convex hull that are intersected by the supporting line of these two points. In Chapter 7, we defined an abstraction of a vertical line in a pseudo-line arrangement that had desirable algorithmic properties. We were able to provide an algorithm to select a crossing of the pseudo-lines of the arrangements that has a given rank along such a pseudo-vertical, using only a linear number of sidedness queries. Further, we showed that there is a total order on these pseudo-verticals with respect to any pseudo-line in the arrangement, just like for vertical lines in line arrangements. We then gave an application of pseudo-verticals, showing that the classic linear-time ham-sandwich cut algorithm also works for abstract order types when replacing vertical lines by pseudo-verticals. In Chapter 8, we considered a combinatorial generalization of convexity, the so-called k-convexity. In particular, we addressed the problem of deciding whether a point set allows for a k-convex polygonization. We provided a polynomial-time algorithm that determines whether a point set has a 2-convex polygonization. For $k \geq 3$, we showed that deciding k-convexity of a point set is an NP-complete problem. We closed that chapter with a short account on deciding 2-convexity of abstract order types by showing how Helly's Theorem, a key ingredient of our algorithm for deciding 2-convexity, is replaced by its counterpart for arrangements of pseudo-lines. Of course, the results presented in the different chapters provoke further related questions.

For the triangulation flip graph, the question on the complexity of the flip distance problem for convex polygons is probably the most important one. The problem has resisted a large number of attempts to solve it for around 30 years. Therefore, it seems that getting more insight into closely related problems could be a successful approach. This would comprise, e.g., finding a PTAS for the flip distance problem in triangulations of convex polygons, or, in the other direction, showing that the problem is APX-hard for triangulations of simple polygons. For the latter, we already discussed that the reduction may have to differ significantly from the one presented in Chapter 4. Naturally, showing that there is a PTAS for a problem always requires to also bound the optimal solution from below. Analyzing the recent progress in lower-bound constructions from an algorithmic point of view may be helpful with this respect.

For algorithms using sidedness queries and working on abstract order types, we have seen that adding "imaginary" pseudo-lines to the arrangement allowed us to prove the correctness of our algorithm. This has been done for replacing an argument using crossings of supporting lines, and for replacing vertical lines by pseudo-verticals. For applying Helly's Theorem for pseudo-line arrangements, we actually extended the arrangement to have a consistent order defined by the additional pseudo-lines. As already mentioned, it would be interesting to have a more light-weight approach, in which the additional pseudo-lines do not have to be created explicitly, but where the order defined by all of them is consistent. Using similar techniques as in Chapter 7, it seems plausible that there exists such a definition. In general, of course, it would be interesting to know an answer to the long-standing problem on whether there is a computational gap between abstract order types and order types, or between using only sidedness queries and using the coordinates of a point set, within an expressive model of computation.

Bibliography

[1] Z. Abel, B. Ballinger, P. Bose, S. Collette, V. Dujmović, F. Hurtado, S. Kominers, S. Langerman, A. Pór, and D. Wood. Every large point set contains many collinear points or an empty pentagon. *Graphs Combin.*, vol. 27, pp. 47–60, 2011. Cited on page 12.

[2] P. K. Agarwal and M. Sharir. Pseudo-line arrangements: Duality, algorithms, and applications. *SIAM J. Comput.*, vol. 34, no. 3, pp. 526–552, 2005. Cited on page 75.

[3] O. Aichholzer. The path of a triangulation. In *Proc. 15th Symposium on Computational Geometry (SoCG 1999)*, pp. 14–23, 1999. Cited on page 41.

[4] O. Aichholzer, F. Aurenhammer, P. Brass, and H. Krasser. Pseudo-triangulations from surfaces and a novel type of edge flip. *SIAM J. Comput.*, vol. 32, pp. 1621–1653, 2003. Cited on page 12.

[5] O. Aichholzer, F. Aurenhammer, E. D. Demaine, F. Hurtado, P. Ramos, and J. Urrutia. On k-convex polygons. *Comput. Geom.*, vol. 45, no. 3, pp. 73–87, 2012. Cited on pages 3, 4, 111, 112, 113, 114, 115, 121, 125, and 140.

[6] O. Aichholzer, F. Aurenhammer, T. Hackl, F. Hurtado, A. Pilz, P. Ramos, J. Urrutia, P. Valtr, and B. Vogtenhuber. On k-convex point sets, 2013. Accepted for Computational Geometry: Theory and Applications. Cited on pages 111, 112, and 113.

[7] O. Aichholzer, F. Aurenhammer, and H. Krasser. Enumerating order types for small point sets with applications. *Order*, vol. 19, no. 3, pp. 265–281, 2002. Cited on pages 1, 65, 72, and 79.

[8] O. Aichholzer, T. Hackl, M. Korman, A. Pilz, and B. Vogtenhuber. Geodesic-preserving polygon simplification. In L. Cai, S.-W. Cheng, and T. W. Lam (Editors), *Proc. 24th International Symposium on Algorithms and Computation (ISAAC 2013)*, vol. 8283 of *Lecture Notes Comput. Sci.*, pp. 11–21. Springer, 2013. Cited on pages 78 and 79.

[9] O. Aichholzer, T. Hackl, S. Lutteropp, T. Mchedlidze, A. Pilz, and B. Vogtenhuber. Monotone simultaneous embedding of directed paths. In *Proc. 30th European Workshop on Computational Geometry (EuroCG 2014)*, 2014. Cited on page 71.

[10] O. Aichholzer, T. Hackl, D. Orden, A. Pilz, M. Saumell, and B. Vogtenhuber. Flips in combinatorial pointed pseudo-triangulations with face degree at most four. In *Proc. XV Spanish Meeting on Computational Geometry (EGC 2013)*, pp. 131–134, 2013. Cited on page 20.

[11] O. Aichholzer, M. Korman, A. Pilz, and B. Vogtenhuber. Geodesic order types. In J. Gudmundsson, J. Mestre, and T. Viglas (Editors), *Proc. 18th International Computing and Combinatorics Conference (COCOON 2012)*, vol. 7434 of *Lecture Notes Comput. Sci.* Springer, 2012. Cited on pages 78 and 79.

[12] O. Aichholzer and H. Krasser. The point set order type data base: A collection of applications and results. In *Proc. 13th Canadian Conference on Computational Geometry (CCCG 2001)*, pp. 17–20, 2001. Cited on pages 65 and 79.

[13] O. Aichholzer and H. Krasser. Abstract order type extension and new results on the rectilinear crossing number. *Comput. Geom.*, vol. 36, no. 1, pp. 2–15, 2007. Cited on pages 1, 79, and 80.

[14] O. Aichholzer, T. Miltzow, and A. Pilz. Extreme point and halving edge search in abstract order types. *Comput. Geom.*, vol. 46, no. 8, pp. 970–978, 2013. Cited on pages 4, 83, 84, and 88.

[15] O. Aichholzer, W. Mulzer, and A. Pilz. Flip distance between triangulations of a simple polygon is NP-complete. In *Proc. 29th European Workshop on Computational Geometry (EuroCG 2013)*, pp. 115–118, 2013. Cited on page 4.

[16] O. Aichholzer, W. Mulzer, and A. Pilz. Flip distance between triangulations of a simple polygon is NP-complete. In H. L. Bodlaender and G. F. Italiano (Editors), *Proc. 21st European Symposium on Algorithms (ESA 2013)*, vol. 8125 of *Lecture Notes Comput. Sci.*, pp. 13–24. Springer, 2013. Cited on pages 4, 9, 12, 31, and 41.

[17] O. Aichholzer, G. Rote, B. Speckmann, and I. Streinu. The zigzag path of a pseudo-triangulation. In F. K. H. A. Dehne, J.-R. Sack, and M. H. M. Smid (Editors), *Proc. 8th Workshop on Algorithms and Data Structures (WADS 2003)*, vol. 2748 of *Lecture Notes Comput. Sci.*, pp. 377–388. Springer, 2003. Cited on page 78.

[18] P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theoret. Comput. Sci.*, vol. 237, no. 1–2, pp. 123–134, 2000. Cited on page 24.

[19] G. Aloupis. A history of linear-time convex hull algorithms for simple polygons. `http://cgm.cs.mcgill.ca/~athens/cs601/`, 2000. Last retrieved on January 30, 2014. Cited on page 78.

[20] V. Alvarez and R. Seidel. A simple aggregative algorithm for counting triangulations of planar point sets and related problems. In *Proc. 29th Symposium on Computational Geometry (SoCG 2013)*, SoCG '13, pp. 1–8. ACM, 2013. Cited on page 41.

[21] E. M. Arkin, S. P. Fekete, F. Hurtado, J. S. B. Mitchell, M. Noy, V. Sacristán, and S. Sethia. On the reflexivity of point sets. In B. Aronov, S. Basu, J. Pach, and

M. Sharir (Editors), *Discrete Comput. Geom.: The Goodman-Pollack Festschrift*, vol. 25 of *Algorithms Combin.*, pp. 139–156. Springer-Verlag, 2003. Cited on page 112.

[22] S. Arora. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In *Proc. 37th Annual Symposium on Foundations of Computer Science (FOCS 1996)*, pp. 2–11. IEEE Computer Society, 1996. Cited on page 53.

[23] G. Ausiello, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties.* Springer-Verlag New York, Secaucus, NY, USA, 1999. Cited on pages 12, 23, 24, and 29.

[24] F. Avnaim, J.-D. Boissonnat, O. Devillers, F. P. Preparata, and M. Yvinec. Evaluating signs of determinants using single-precision arithmetic. *Algorithmica*, vol. 17, no. 2, pp. 111–132, 1997. Cited on page 77.

[25] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, vol. 41, no. 1, pp. 153–180, 1994. Cited on page 12.

[26] I. J. Balaban. An optimal algorithm for finding segments intersections. In *Proc. 11th Symposium on Computational Geometry (SoCG 1995)*, pp. 211–219, 1995. Cited on page 77.

[27] J. Balogh, S. P. J. Leaños, R. Richter, and G. Salazar. The convex hull of every optimal pseudolinear drawing of K_n is a triangle. *Australas. J. Combin.*, vol. 38, pp. 155–162, 2007. Cited on page 76.

[28] R. Bar-Yehuda and S. Even. On approximating a vertex cover for planar graphs. In H. R. Lewis, B. B. Simons, W. A. Burkhard, and L. H. Landweber (Editors), *Proc. 14th Symposium on the Theory of Computing (STOC 1982)*, pp. 303–309. ACM, 1982. Cited on page 12.

[29] M. K. Bennett. *Affine and Projective Geometry.* John Wiley & Sons, New York, NY, USA, 1995. Cited on page 61.

[30] S. Bereg. Transforming pseudo-triangulations. *Inf. Proc. Lett.*, vol. 90, no. 3, pp. 141–145, 2004. Cited on page 12.

[31] M. W. Bern and D. Eppstein. Mesh generation and optimal triangulation. In D.-Z. Du and F. K.-M. Hwang (Editors), *Computing in Euclidean Geometry*, no. 4 in Lecture Notes Series on Computing, pp. 47–123. World Scientific, second edn., 1995. Cited on pages 2 and 11.

[32] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. Ziegler. *Oriented matroids*, vol. 46 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1993. Cited on page 57.

[33] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan. Time bounds for selection. *J. Comput. System Sci.*, vol. 7, no. 4, pp. 448–461, 1973. Cited on pages 84 and 92.

[34] J.-D. Boissonnat and J. Snoeyink. Efficient algorithms for line and curve segment intersection using restricted predicates. *Comput. Geom.*, vol. 16, no. 1, pp. 35–52, 2000. Cited on page 77.

[35] P. Bose, E. D. Demaine, F. Hurtado, J. Iacono, S. Langerman, and P. Morin. Geodesic ham-sandwich cuts. In *Proc. 20th Symposium on Computational Geometry (SoCG 2004)*, pp. 1–9. ACM, 2004. Cited on pages 78, 92, 108, and 109.

[36] P. Bose and F. Hurtado. Flips in planar graphs. *Comput. Geom.*, vol. 42, no. 1, pp. 60–80, 2009. Cited on pages 10, 11, and 12.

[37] P. Bose, A. Lubiw, V. Pathak, and S. Verdonschot. Flipping Edge-Labelled Triangulations. *ArXiv e-prints*, 2013. ArXiv:1310.1166. Cited on page 11.

[38] C. Burnikel. Rational points on circles. Tech. Rep. MPI-I-98-1-023, Max-Planck-Institut für Informatik, Saarbrücken, Germany, 1998. Cited on page 38.

[39] I. Bárány and G. Rote. Strictly convex drawings of planar graphs. *Doc. Math.*, vol. 11, pp. 369–391, 2006. Cited on page 31.

[40] J. F. Canny, B. R. Donald, and E. K. Ressler. A rational rotation method for robust geometric algorithms. In *Proc. 8th Symposium on Computational Geometry (SoCG 1992)*, pp. 251–260, 1992. Cited on pages 33, 35, 36, 38, and 125.

[41] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete Comput. Geom.*, vol. 16, no. 4, pp. 361–368, 1996. Cited on page 90.

[42] B. Chazelle, H. Edelsbrunner, M. Grigni, L. J. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, vol. 12, no. 1, pp. 54–68, 1994. Cited on page 114.

[43] B. Chazelle, L. J. Guibas, and D. T. Lee. The power of geometric duality. *BIT*, vol. 25, no. 1, pp. 76–90, 1985. Cited on page 138.

[44] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, vol. 21, no. 3, pp. 579–597, 1996. Cited on page 102.

[45] S. Cleary and K. St. John. Rotation distance is fixed-parameter tractable. *Inf. Process. Lett.*, vol. 109, no. 16, pp. 918–922, 2009. Cited on page 11.

[46] R. Cordovil. Sur les matroïdes orientés de rang 3 et les arrangements de pseudodroites dans le plan projectif réel. *European J. Combin.*, vol. 3, no. 4, pp. 307–318, 1982. In French. Cited on page 73.

[47] H. Coxeter. *The Real Projective Plane.* Springer-Verlag, New York, NY, USA, 1993. Cited on page 61.

[48] J. C. Culberson and R. A. Reckhow. Covering polygons is hard. *J. Algorithms*, vol. 17, no. 1, pp. 2–44, 1994. Cited on page 126.

[49] K. Culik II and D. Wood. A note on some tree similarity measures. *Inf. Process. Lett.*, vol. 15, no. 1, pp. 39–42, 1982. Cited on page 11.

[50] J. De Loera, J. Rambau, and F. Santos. *Triangulations*, vol. 25 of *Algorithms and Computation in Mathematics.* Springer-Verlag, Heidelberg, Germany, 2010. Cited on pages 2 and 77.

[51] E. D. Demaine, J. Erickson, F. Hurtado, J. Iacono, S. Langerman, H. Meijer, M. H. Overmars, and S. Whitesides. Separating point sets in polygonal environments. *Int. J. Comput. Geometry Appl.*, vol. 15, no. 4, pp. 403–420, 2005. Cited on page 78.

[52] S. L. Devadoss and J. O'Rourke. *Discrete and Computational Geometry.* Princeton University Press, Princeton, NJ, USA, 2011. Cited on page 12.

[53] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Ann. of Math. (2)*, vol. 162, no. 1, pp. pp. 439–485, 2005. Cited on page 31.

[54] H. Edelsbrunner. *Algorithms in combinatorial geometry.* Springer-Verlag, New York, NY, USA, 1987. Cited on pages 1, 57, 58, 62, 63, 80, and 138.

[55] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, vol. 38, no. 1, pp. 165–194, 1989. Cited on page 97.

[56] H. Edelsbrunner and L. J. Guibas. Corrigendum: Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, vol. 42, no. 2, pp. 249–251, 1991. Cited on page 97.

[57] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction.* American Math. Soc., Providence, RI, USA, 2010. Cited on page 61.

[58] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, vol. 15, no. 2, pp. 341–363, 1986. Cited on page 138.

[59] H. Edelsbrunner and R. Waupotitsch. Computing a ham-sandwich cut in two dimensions. *J. Symb. Comput.*, vol. 2, no. 2, pp. 171–178, 1986. Cited on page 91.

[60] D. Eppstein. Happy endings for flip graphs. *JoCG*, vol. 1, no. 1, pp. 3–28, 2010. Cited on page 11.

[61] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Math.*, vol. 2, pp. 463–470, 1935. Cited on page 80.

[62] J. Erickson, F. Hurtado, and P. Morin. Centerpoint theorems for wedges. *Discrete Math. Theor. Comput. Sci.*, vol. 11, no. 1, pp. 45–54, 2009. Cited on page 91.

[63] J. Erickson and R. Seidel. Better lower bounds on detecting affine and spherical degeneracies. In *Proc. 34th Annual Symposium on Foundations of Computer Science (FOCS 1993)*, pp. 528–536. IEEE Computer Society, 1993. Cited on page 57.

[64] J. G. Erickson. *Lower Bounds for Fundamental Geometric Problems*. Ph.D. thesis, University of California at Berkeley, 1996. Cited on pages 3 and 81.

[65] S. Felsner. On the number of arrangements of pseudolines. *Discrete Comput. Geom.*, vol. 18, no. 3, pp. 257–267, 1997. Cited on page 71.

[66] S. Felsner. *Geometric Graphs and Arrangements*. Advanced Lectures in Mathematics. Vieweg Verlag, Wiesbaden, Germany, 2004. Cited on pages 63 and 70.

[67] S. Felsner and A. Pilz. Ham-sandwich cuts for abstract order types, 2014. In preparation. Cited on pages 4 and 91.

[68] E. Fink and D. Wood. Fundamentals of restricted-orientation convexity. *Inf. Sci.*, vol. 92, no. 1–4, pp. 175–196, 1996. Cited on page 111.

[69] L. Finschi and K. Fukuda. Generation of oriented matroids - a graph theoretical approach. *Discrete Comput. Geom.*, vol. 27, no. 1, pp. 117–136, 2002. Cited on page 79.

[70] S. Fisk. A short proof of Chvátal's Watchman Theorem. *J. Comb. Theory, Ser. B*, vol. 24, no. 3, p. 374, 1978. Cited on page 2.

[71] A. García, M. Noy, and J. Tejel. Lower bounds on the number of crossing-free subgraphs of K_N. *Comput. Geom.*, vol. 16, no. 4, pp. 211–221, 2000. Cited on page 12.

[72] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, USA, 1979. Cited on page 125.

[73] B. Gärtner and E. Welzl. Vapnik-Chervonenkis dimension and (pseudo-)hyperplane arrangements. *Discrete Comput. Geom.*, vol. 12, pp. 399–432, 1994. Cited on page 102.

[74] P. D. Gilbert. New results in planar triangulations. Tech. Rep. R–850, Univ. Illinois Coordinated Science Lab, 1979. Cited on page 41.

[75] J. E. Goodman. Proof of a conjecture of Burr, Grünbaum, and Sloane. *Discrete Math.*, vol. 32, no. 1, pp. 27–35, 1980. Cited on pages 63, 73, and 75.

[76] J. E. Goodman. Pseudoline arrangements. In J. E. Goodman and J. O'Rourke (Editors), *Handbook of discrete and computational geometry*, pp. 83–109. CRC Press, Inc., Boca Raton, FL, USA, 1997. Cited on page 59.

[77] J. E. Goodman and R. Pollack. On the combinatorial classification of nondegenerate configurations in the plane. *J. Comb. Theory, Ser. A*, vol. 29, no. 2, pp. 220–235, 1980. Cited on pages 1, 58, 59, and 70.

[78] J. E. Goodman and R. Pollack. Proof of Grünbaum's conjecture on the stretchability of certain arrangements of pseudolines. *J. Combin. Theory Ser. A*, vol. 29, no. 3, pp. 385–390, 1980. Cited on page 70.

[79] J. E. Goodman and R. Pollack. Helly-type theorems for pseudoline arrangements in P^2. *J. Comb. Theory, Ser. A*, vol. 32, no. 1, pp. 1–19, 1982. Cited on page 136.

[80] J. E. Goodman and R. Pollack. A theorem of ordered duality. *Geom. Dedicata*, vol. 12, pp. 63–74, 1982. Cited on pages 62, 63, 66, and 68.

[81] J. E. Goodman and R. Pollack. Multidimensional sorting. *SIAM J. Comput.*, vol. 12, no. 3, pp. 484–507, 1983. Cited on pages 65 and 71.

[82] J. E. Goodman and R. Pollack. Semispaces of configurations, cell complexes of arrangements. *J. Combin. Theory Ser. A*, vol. 37, no. 3, pp. 257–293, 1984. Cited on pages 57, 58, 59, 64, 65, 66, 68, 69, 70, 73, 74, and 102.

[83] J. E. Goodman and R. Pollack. Upper bounds for configurations and polytopes in R^d. *Discrete Comput. Geom.*, vol. 1, pp. 219–227, 1986. Cited on page 71.

[84] J. E. Goodman and R. Pollack. Allowable sequences and order types in discrete and computational geometry. In J. Pach (Editor), *New Trends in Discrete and Computational Geometry*, vol. 10 of *Algorithms and Combinatorics*, pp. 103–134. Springer Berlin Heidelberg, 1993. Cited on pages 57, 58, and 59.

[85] J. E. Goodman, R. Pollack, and B. Sturmfels. Coordinate representation of order types requires exponential storage. In *Proc. 21st Symposium on Theory of Computing (STOC 1989)*, pp. 405–410. ACM, 1989. Cited on pages 71 and 81.

[86] L. J. Guibas and M. Sharir. Combinatorics and algorithms of arrangements. In J. Pach (Editor), *New Trends in Discrete and Computational Geometry*, vol. 10 of *Algorithms and Combinatorics*, pp. 9–36. Springer Berlin Heidelberg, 1993. Cited on page 139.

[87] L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and computation of Voronoi diagrams. *ACM Trans. Graph.*, vol. 4, no. 2, pp. 74–123, 1985. Cited on pages 77, 83, and 84.

[88] L. Habert and M. Pocchiola. Computing the convex hull of discs using only their chirotope. In *Proc. 20th European Workshop on Computational Geometry (EuroCG 2004)*, pp. 111–114, 2004. Cited on page 80.

[89] S. Hanke, T. Ottmann, and S. Schuierer. The edge-flipping distance of triangulations. *J.UCS*, vol. 2, no. 8, pp. 570–579, 1996. Cited on pages 11, 12, and 40.

[90] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete Comput. Geom.*, vol. 2, pp. 127–151, 1987. Cited on pages 104 and 105.

[91] E. Helly. Über Mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresber. Deutsch. Math.-Verein.*, vol. 32, pp. 175–176, 1923. In German. Cited on page 117.

[92] C. Hernández-Vélez, J. Leaños, and G. Salazar. Pseudolinear crossing number, 2014. Manuscript. Cited on page 76.

[93] Ø. Hjelle and M. Dæhlen. *Triangulations and applications*. Mathematics and visualization. Springer-Verlag, Heidelberg, Germany, 2007. Cited on pages 2, 9, and 11.

[94] D. S. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM J. Comput.*, vol. 11, no. 3, pp. 555–556, 1982. Cited on page 31.

[95] F. Hurtado, M. Noy, and J. Urrutia. Flipping edges in triangulations. *Discrete Comput. Geom.*, vol. 22, pp. 333–346, 1999. Cited on pages 11, 12, 13, and 32.

[96] D. Husemöller. *Elliptic Curves*. Graduate Texts in Mathematics. Springer-Verlag, New York, NY, USA, 2003. Cited on page 38.

[97] F. Hwang, D. Richards, and P. Winter. *The Steiner Tree Problem*. Ann. Discrete Math. North-Holland, Amsterdam, Netherlands, 1992. Cited on page 42.

[98] G. Karakostas. A better approximation ratio for the vertex cover problem. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung (Editors), *Proc. 32nd International Colloquium on Automata, Languages and Programming (ICALP 2005)*, vol. 3580 of *Lecture Notes Comput. Sci.*, pp. 1043–1050. Springer, 2005. Cited on page 31.

[99] L. Kettner, D. Kirkpatrick, A. Mantler, J. Snoeyink, B. Speckmann, and F. Takeuchi. Tight degree bounds for pseudo-triangulations of points. *Comput. Geom.*, vol. 25, no. 1–2, pp. 3–12, 2003. Cited on page 20.

[100] L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, and C.-K. Yap. Classroom examples of robustness problems in geometric computations. *Comput. Geom.*, vol. 40, no. 1, pp. 61–78, 2008. Cited on page 77.

[101] S. Khot. On the power of unique 2-prover 1-round games. In J. H. Reif (Editor), *Proc. 34th Symposium on the Theory of Computing (STOC 2002)*, pp. 767–775. ACM, 2002. Cited on page 24.

[102] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2 - \epsilon$. *J. Comput. Syst. Sci.*, vol. 74, no. 3, pp. 335–349, 2008. Cited on pages 24 and 31.

[103] D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM J. Comput.*, vol. 15, no. 1, pp. 287–299, 1986. Cited on pages 84 and 90.

[104] G. T. Klincsek. Minimal triangulations of polygonal domains. *Ann. Discrete Math.*, vol. 9, pp. 121–123, 1980. Cited on page 41.

[105] D. E. Knuth. *Axioms and Hulls*, vol. 606 of *Lecture Notes Comput. Sci.* Springer-Verlag, Heidelberg, Germany, 1992. Cited on pages 2, 3, 57, 72, 78, 80, 83, 84, and 90.

[106] H. Komuro. The diagonal flip of triangulations on the sphere. *Yokohama Math. J.*, vol. 44, no. 2, pp. 115–122, 1997. Cited on page 10.

[107] H. Krasser. *Order Types of Point Sets in the Plane*. Ph.D. thesis, Graz University of Technology, 2003. Cited on pages 57, 65, 70, and 71.

[108] C. L. Lawson. Transforming triangulations. *Discrete Math.*, vol. 3, no. 4, pp. 365–372, 1972. Cited on pages 11 and 23.

[109] C. L. Lawson. Software for C^1 surface interpolation. In J. R. Rice (Editor), *Mathematical Software III*, pp. 161–194. Academic Press, NY, 1977. Cited on pages 2 and 11.

[110] F. Levi. Die Teilung der projektiven Ebene durch Gerade oder Pseudogerade. *Ber. Math.-Phys. Kl. Sächs. Akad. Wiss. Leipzig*, vol. 78, pp. 256–267, 1926. In German. Cited on pages 1, 69, and 139.

[111] M. Li and L. Zhang. Better approximation of diagonal-flip transformation and rotation transformation. In W.-L. Hsu and M.-Y. Kao (Editors), *Proc. 4th International Computing and Combinatorics Conference (COCOON 1998)*, vol. 1449 of *Lecture Notes Comput. Sci.*, pp. 85–94. Springer, 1998. Cited on page 11.

[112] C.-Y. Lo, J. Matoušek, and W. Steiger. Algorithms for ham-sandwich cuts. *Discrete Comput. Geom.*, vol. 11, pp. 433–452, 1994. Cited on pages 84, 91, 93, 102, 103, 104, 106, 107, and 108.

[113] C.-Y. Lo and W. Steiger. An optimal time algorithm for ham-sandwich cuts in the plane. In *Proc. 2nd Canadian Conference on Computational Geometry (CCCG 1990)*, pp. 5–9, 1990. Cited on pages 103 and 108.

[114] B. Lu and L. Ruan. Polynomial time approximation scheme for the rectilinear Steiner arborescence problem. *J. Comb. Optim.*, vol. 4, no. 3, pp. 357–363, 2000. Cited on page 53.

[115] A. Lubiw and V. Pathak. Flip distance between two triangulations of a point-set is NP-complete. In *Proc. 24th Canadian Conference on Computational Geometry (CCCG 2012)*, pp. 127–132, 2012. Cited on pages 12, 15, 23, and 31.

[116] J. Matoušek. Construction of ε-nets. *Discrete Comput. Geom.*, vol. 5, pp. 427–448, 1990. Cited on pages 103 and 104.

[117] J. Matoušek. Approximations and optimal geometric divide-and-conquer. In C. Koutsougeras and J. S. Vitter (Editors), *Proc. 23rd Symposium on the Theory of Computing (STOC 1991)*, pp. 505–511. ACM, 1991. Cited on pages 102, 103, and 104.

[118] J. Matoušek. Epsilon-nets and computational geometry. In J. Pach (Editor), *New Trends in Discrete and Computational Geometry*, vol. 10 of *Algorithms and Combinatorics*, pp. 69–89. Springer Berlin Heidelberg, 1993. Cited on page 101.

[119] J. Matoušek. Approximations and optimal geometric divide-and-conquer. *J. Comput. System Sci.*, vol. 50, no. 2, pp. 203–208, 1995. Cited on page 102.

[120] J. Matoušek and P. Plechác. On functional separately convex hulls. *Discrete Comput. Geom.*, vol. 19, no. 1, pp. 105–130, 1998. Cited on page 111.

[121] N. Megiddo. Partitioning with two lines in the plane. *J. Algorithms*, vol. 6, no. 3, pp. 430–433, 1985. Cited on pages 91 and 107.

[122] K. Mehlhorn. *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*, vol. 3 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Heidelberg, Germany, 1984. Cited on page 1.

[123] L. I. Meikle and J. D. Fleuriot. Mechanical theorem proving in computational geometry. In H. Hong and D. Wang (Editors), *Proc. 5th International Workshop on Automated Deduction in Geometry (ADG 2004)*, vol. 3763 of *Lecture Notes Comput. Sci.*, pp. 1–18. Springer, 2004. Cited on page 78.

[124] T. Miltzow and A. Pilz. Selection of extreme points and halving edges of a set by its chirotope. In *Proc. 28th European Workshop Comput. Geom. (EuroCG 2012)*, pp. 85–88, 2012. Cited on pages 4 and 83.

[125] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM J. Comput.*, vol. 28, no. 4, pp. 1298–1309, 1999. Cited on page 53.

[126] N. E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytope varieties. In O. Y. Viro (Editor), *Topology and Geometry—Rohlin Seminar*, vol. 1346 of *Lecture Notes Math.*, pp. 527–544. Springer, 1988. Cited on page 71.

[127] W. Mulzer and G. Rote. Minimum-weight triangulation is NP-hard. *J. ACM*, vol. 55, no. 2, pp. 1–29, 2008. Cited on page 41.

[128] S. Pan. *On the crossing numbers of complete graphs.* Master's thesis, University of Waterloo, 2006. Cited on page 76.

[129] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, vol. 43, no. 3, pp. 425–440, 1991. Cited on page 24.

[130] M. Perrin. Sur le problème des aspects. *Bull. Soc. Math. France*, vol. 10, pp. 103–127, 1882. In French. Cited on pages 1, 58, and 59.

[131] J. Pfeifle and J. Rambau. Computing triangulations using oriented matroids. In M. Joswig and N. Takayama (Editors), *Algebra, Geometry, and Software Systems*, pp. 49–75. Springer, 2003. Cited on page 77.

[132] D. Pichardie and Y. Bertot. Formalizing convex hull algorithms. In R. J. Boulton and P. B. Jackson (Editors), *Proc. 14th International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2001)*, vol. 2152 of *Lecture Notes Comput. Sci.*, pp. 346–361. Springer, 2001. Cited on page 78.

[133] A. Pilz. Flip distance between triangulations of a planar point set is APX-hard. *Comput. Geom.*, vol. 47, no. 5, pp. 589–604, 2014. Cited on pages 4, 9, 23, 25, and 31.

[134] L. Pournin. The diameters of associahedra. *ArXiv e-prints*, 2012. ArXiv:1207.6296. Cited on page 11.

[135] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, NY, USA, 1985. Cited on pages 1, 2, 4, 62, 76, and 115.

[136] S. K. Rao, P. Sadayappan, F. K. Hwang, and P. W. Shor. The rectilinear Steiner arborescence problem. *Algorithmica*, vol. 7, pp. 277–288, 1992. Cited on page 42.

[137] D. Rappaport. Computing simple circuits from a set of line segments is NP-complete. *SIAM J. Comput.*, vol. 18, no. 6, pp. 1128–1139, 1989. Cited on page 125.

[138] G. Ringel. Teilungen der Ebene durch Geraden oder topologische Geraden. *Math. Z.*, vol. 64, pp. 79–102, 1956. In German. Cited on pages 70 and 79.

[139] G. Rote, F. Santos, and I. Streinu. Pseudo-triangulations — a survey. In J. E. Goodman, J. Pach, and R. Pollack (Editors), *Surveys on Discrete and Computational Geometry — Twenty Years Later*, vol. 453 of *Contemporary Mathematics*, pp. 343–411. American Math. Soc., Providence, RI, USA, 2008. Cited on page 12.

[140] S. Roy and W. Steiger. Some combinatorial and algorithmic applications of the Borsuk–Ulam theorem. *Graphs Combin.*, vol. 23, no. 1, pp. 331–341, 2007. Cited on page 91.

[141] F. Santos and R. Seidel. A better upper bound on the number of triangulations of a planar point set. *J. Comb. Theory, Ser. A*, vol. 102, no. 1, pp. 186–193, 2003. Cited on page 12.

[142] M. Schaefer. The graph crossing number and its variants: A survey. *Electron. J. Combin.*, vol. 20, no. 2, p. #DS21, 2003. Cited on pages 75 and 76.

[143] M. Schaefer. Complexity of some geometric and topological problems. In D. Eppstein and E. R. Gansner (Editors), *Proc. 17th Symposium on Graph Drawing (GD 2009)*, vol. 5849 of *Lecture Notes Comput. Sci.*, pp. 334–344. Springer, 2009. Cited on page 71.

[144] S. Schirra. Robustness and precision issues in geometric computation. In J.-R. Sack and J. Urrutia (Editors), *Handbook of Computational Geometry*, pp. 597–632. North-Holland, Amsterdam, 2000. Cited on page 77.

[145] M. I. Shamos. *Computational Geometry.* Ph.D. thesis, Department of Computer Science, Yale University, 1978. Cited on page 1.

[146] M. Sharir, A. Sheffer, and E. Welzl. On degrees in random triangulations of point sets. *J. Comb. Theory, Ser. A*, vol. 118, no. 7, pp. 1979–1999, 2011. Cited on page 24.

[147] W. Shi and C. Su. The rectilinear steiner arborescence problem is np-complete. In D. B. Shmoys (Editor), *Proc. 11th Symposium on Discrete Algorithms (SODA 2000)*, pp. 780–787. ACM/SIAM, 2000. Cited on pages 41 and 42.

[148] P. W. Shor. Stretchability of pseudolines is NP-hard. In P. Gritzman and B. Sturmfels (Editors), *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, vol. 4 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pp. 531–554. American Math. Soc., Providence, RI, USA, 1991. Cited on page 71.

[149] D. Sleator, R. Tarjan, and W. Thurston. Rotation distance, triangulations and hyperbolic geometry. *J. Amer. Math. Soc.*, vol. 1, pp. 647–682, 1988. Cited on page 11.

[150] J. Snoeyink and J. Hershberger. Sweeping arrangements of curves. In *Proc. 5th Symposium on Computational Geometry (SoCG 1989)*, pp. 354–363, 1989. Cited on page 97.

[151] R. P. Stanley. On the number of reduced decompositions of elements of Coxeter groups. *European J. Combin.*, vol. 5, pp. 359–372, 1984. Cited on page 60.

[152] R. P. Stanley. *Enumerative Combinatorics; Volume 2.* No. 62 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1999. Cited on page 2.

[153] W. L. Steiger and I. Streinu. A pseudo-algorithmic separation of lines from pseudo-lines. *Inf. Process. Lett.*, vol. 53, no. 5, pp. 295–299, 1995. Cited on page 81.

[154] I. Streinu. Stretchability of star-like pseudo-visibility graphs. In *Proc. 15th Symposium on Computational Geometry (SoCG 1999)*, pp. 274–280, 1999. Cited on page 136.

[155] G. T. Toussaint. Computing geodesic properties inside a simple polygon. *Revue D'Intelligence Artificielle*, vol. 3, no. 2, pp. 9–42, 1989. Cited on page 78.

[156] V. Trubin. Subclass of the Steiner problems on a plane with rectilinear metric. *Cybernetics*, vol. 21, pp. 320–324, 1985. Cited on page 42.

[157] J. Urrutia. Algunos problemas abiertos. In N. Coll and J. Sellares (Editors), *Proc. IX Encuentros de Geometría Computacional (EGC 2001)*, pp. 13–24. Univ. De Girona, 2001. In Spanish. Cited on pages 15 and 32.

[158] F. A. Valentine. *Convex Sets.* McGraw-Hill, Inc., New York, U.S.A., 1964. Cited on page 117.

[159] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, vol. 16, pp. 264–280, 1971. Cited on page 102.

[160] K. Wagner. Bemerkungen zum Vierfarbenproblem. *Jahresber. Deutsch. Math.-Verein.*, vol. 46, pp. 26–32, 1936. In German. Cited on pages 10 and 11.

[161] C. Yap. *Fundamental Problems of Algorithmic Algebra*. Oxford University Press USA, 1999. Cited on page 36.

Index of Definitions

$\mathbb{E}^2$, 4
$\mathbb{P}^2$, 61
ψ-augmentation, 69
ε-approximation, 101
j-stabber, 112
k-convex graph, 112
k-convex point set, 112
k-convex polygon, 111

abstract order type, 69
abstract order type extension, 80
affine plane, 61
algebraic degree, 77
allowable sequence of permutations, 59
AP-reduction, 29
arrangement of lines, 62
arrangement of pseudo-lines in $\mathbb{E}^2$, 63
arrangement of pseudo-lines in $\mathbb{P}^2$, 69

bisector, 103
boundary, 4

CC system, 72
circular sequence, 59
combinatorial type, 59
combinatorially equivalent allowable sequences, 59
combinatorially equivalent configurations of points, 59
configuration of n points, 58
convex hull, 4
convex polygon, 4
convex vertex, 4
crossing number, 75

diagonal, 10
distinguished point, 67
double chain, 13
duality, 62

edge core, 26
edge crossing, 4
edge flip, 10
extreme triangulation, 14

feasible solution, 24
flip distance, 10
flip graph, 10
flip kernel, 14
flip sequence, 10
flippable, 10

general position, 4
generalized configuration of points, 73
geodesic, 78
geodesic convex hull, 78
geodesic order types, 78
geodesically convex, 78
geometric graph, 4

ham-sandwich cut, 104
hourglass, 14

incidence preservation, 62
inflection edge, 113
inflection line, 113
inner tangent, 113
interior, 4

kernel, 4

level, 93
line arrangement, 62
line at infinity, 61
local j-stabber, 112
local triangulation, 16
lower chain, 13

marked cell, 68
median level, 104

non-degenerate configuration of points, 58
north face, 70
northbound ray, 95
NP optimization problem, 23

odd intersection property, 104
order preservation, 63
order type, 65

performance ratio, 24
point at infinity, 61
pointed pseudo-triangulation, 12
pointgon, 78
polygonal chain, 4
polygonal cycle, 4
polygonization, 112
pseudo-line arrangement in $\mathbb{E}^2$, 63
pseudo-line in $\mathbb{E}^2$, 63
pseudo-line in $\mathbb{P}^2$, 69
pseudo-linear crossing number, 76
pseudo-segment, 105
pseudo-segment range space, 105
pseudo-triangle, 12
pseudo-triangulation, 12
pseudo-vertical, 93

range, 101
range space, 101
real projective plane, 61
realizable allowable sequence, 59
rectilinear crossing number, 76
reflection, 59
reflex vertex, 4

semispace, 64
semispace-equivalence, 64
shatter function, 102
shattered subset, 101
sidedness query, 57
simple arrangement, 62
simple polygon, 4
simple polygonal cycle, 4
simple polygonal path, 4
simple pseudo-line arrangement, 63
south face, 70
southbound ray, 95
spanning graph, 4
stabbing number, 112
star-shaped polygon, 4
stretchability, 70
subset property, 80
subspace oracle, 102
supporting line, 4
supporting lines of a point set, 58
supporting pseudo-lines, 74

topological sweep, 97
trace, 49
triangulation, 9
tunnel of an edge, 25

unavoidable edge, 13
upper chain, 13
upper envelope, 93

Vapnik-Chervonenkis dimension, 101
VC-dimension, 101

weakly simple polygon, 78
wide edge, 16
wiring, 26
wiring diagram, 63

zig-zag edges, 26

Curriculum Vitae of Alexander Pilz

Affiliation and Address

Dipl.-Ing. Alexander Pilz Bakk.rer.soc.oec.
Research Assistant
Institute for Software Technology
Graz University of Technology
Inffeldgasse 16b/II
A-8010 Graz, Austria
E-mail: `apilz@ist.tugraz.at`

Personal Information

Born August 14, 1983 in Graz (Austria)
Austrian nationality

Education

Graz University of Technology, Austria
2009 - present
PhD student, advisor: Prof. O. Aichholzer

Graz University of Technology, Austria
2007 - 2009
Dipl.-Ing. (MSc), advisor: Prof. O. Aichholzer
Studies of the Master's Programme Software Development and Business Management, Master's thesis at the Institute for Software Technology (with distinction)

Graz University of Technology, Austria
2003 - 2007
Bakk.rer.soc.oec.
Studies of the Bachelor Programme Software Development and Knowledge Management

Bundesgymnasium u. Bundesrealgymnasium Weiz, Austria
1994 - 2002
Matura (a-levels) Gymnasium, Schulversuch gem. § 7 SchOG (grammar school) (with distinction)

Professional Experience

Graz University of Technology, Austria
2011 - 2014
Recipient of a DOC-fellowship of the Austrian Academy of Sciences at the Institute of Software Technology, Graz University of Technology, Austria

Graz University of Technology, Austria
2009 - 2010
Institute for Software Technology
Part time research assistant, supported by the Austrian FWF National Research Network 'Industrial Geometry' S9205-N12

Verein zur Förderung der Forschung im Bereich der angewandten Softwareentwicklung in Österreich – SoftNet Austria, Graz, Austria
2009 - 2011
Part time software engineer

Graz University of Technology, Austria
2006 - 2008
Institute for Computer Graphics and Vision
Part time teaching assistant

HGS Informationsmanagement GmbH, Weiz, Austria
2000 - 2004
Freelance software engineer

Other Occupation

Stadtgemeinde Weiz, Austria
2002 - 2003
National service (Zivildienst)

Career-Related Activities

- Long-term research stays at Universidad de Alcalá, Spain, and at Freie Universität Berlin, Germany.
- Assistance in the supervision of three Master's theses.

- Reviewing for
    - Journal of Computational Geometry
    - Graphs and Combinatorics
    - European Symposium on Algorithms (ESA)
    - International Symposium on Algorithms and Computation (ISAAC)
    - Canadian Conference on Computational Geometry (CCCG)
    - Spanish Meeting on Computational Geometry (EGC)
    - Latin American Theoretical Informatics Symposium (LATIN)

Research Interests

Algorithms and data structures, in particular computational geometry, and combinatorial geometry.

Publications of Alexander Pilz

The list is in chronological order (sorted alphabetically by authors within each year), and categorized in journal articles and articles in refereed proceedings. The publications marked bold in the following list arose from parts of this doctoral thesis. The publications are indicated at the beginning of each corresponding chapter, and the results presented have been obtained in collaboration with the coauthors of these publications. Parts of these publications without major contribution of the thesis author are not presented in this thesis.

Five articles published in journals:

- **O. Aichholzer, T. Miltzow, and A. Pilz. Extreme point and halving edge search in abstract order types. *Comput. Geom.*, vol. 46, no. 8, pp. 970–978, 2013.**

- O. Aichholzer, R. F. Monroy, T. Hackl, M. J. van Kreveld, A. Pilz, P. Ramos, and B. Vogtenhuber. Blocking Delaunay triangulations. *Comput. Geom.*, vol. 46, no. 2, pp. 154–159, 2013.

- O. Aichholzer, R. Fabila-Monroy, T. Hackl, C. Huemer, A. Pilz, and B. Vogtenhuber. Lower bounds for the number of small convex k-holes. *Comput. Geom.*, vol. 47, no. 5, pp. 605–613, 2014.

- O. Aichholzer, T. Hackl, M. Hoffmann, A. Pilz, G. Rote, B. Speckmann, and B. Vogtenhuber. Plane graphs with parity constraints. *Graphs Combin.*, vol. 30, no. 1, pp. 47–69, 2014.

- **A. Pilz. Flip distance between triangulations of a planar point set is APX-hard. *Comput. Geom.*, vol. 47, no. 5, pp. 589–604, 2014.**

One article accepted for journal publication:

- **O. Aichholzer, F. Aurenhammer, T. Hackl, F. Hurtado, A. Pilz, P. Ramos, J. Urrutia, P. Valtr, and B. Vogtenhuber. On k-convex point sets, 2013. Accepted for Computational Geometry: Theory and Applications.**

Sixteen articles in refereed proceedings:

- O. Aichholzer, T. Hackl, M. Hoffmann, A. Pilz, G. Rote, B. Speckmann, and B. Vogtenhuber. Plane graphs with parity constraints. In F. K. H. A. Dehne, M. L. Gavrilova, J.-R. Sack, and Cs. D. Tóth (Editors), *Proc. 11th Algorithms and Data Structures Symposium (WADS 2009)*, vol. 5664 of *Lecture Notes Comput. Sci.*, pp. 13–24. Springer, 2009.

- O. Aichholzer, F. Aurenhammer, T. Hackl, C. Huemer, A. Pilz, and B. Vogtenhuber. 3-colorability of pseudo-triangulations. In *Proc. 26th European Workshop on Computational Geometry (EuroCG 2010)*, pp. 21–24, 2010.

- O. Aichholzer, R. Fabila-Monroy, T. Hackl, M. van Kreveld, A. Pilz, P. Ramos, and B. Vogtenhuber. Blocking Delaunay triangulations. In *Proc. 22nd Canadian Conference on Computational Geometry (CCCG 2010)*, pp. 21–24, 2010.

- O. Aichholzer, R. Fabila-Monroy, T. Hackl, C. Huemer, A. Pilz, and B. Vogtenhuber. Lower bounds for the number of small convex k-holes. In *Proc. 24th Canadian Conference on Computational Geometry (CCCG 2012)*, pp. 261–266, 2012.

- O. Aichholzer, M. Korman, A. Pilz, and B. Vogtenhuber. Geodesic order types. In J. Gudmundsson, J. Mestre, and T. Viglas (Editors), *Proc. 18th International Computing and Combinatorics Conference (COCOON 2012)*, vol. 7434 of *Lecture Notes Comput. Sci.* Springer, 2012.

- **T. Miltzow and A. Pilz. Selection of extreme points and halving edges of a set by its chirotope. In *Proc. 28th European Workshop Comput. Geom. (EuroCG 2012)*, pp. 85–88, 2012.**

- A. Pilz. Augmentability to cubic graphs. In *Proc. 28th European Workshop on Computational Geometry (EuroCG 2012)*, pp. 29–32, 2012.

- O. Aichholzer, J. Cardinal, T. Hackl, F. Hurtado, M. Korman, A. Pilz, R. I. Silveira, R. Uehara, B. Vogtenhuber, and E. Welzl. Cell-paths in mono- and bichromatic line arrangements in the plane. In *Proc. 25th Canadian Conference on Computational Geometry (CCCG 2013)*, pp. 169–174, 2013.

- O. Aichholzer, T. Hackl, M. Korman, A. Pilz, and B. Vogtenhuber. Geodesic-preserving polygon simplification. In L. Cai, S.-W. Cheng, and T. W. Lam (Editors), *Proc. 24th International Symposium on Algorithms and Computation (ISAAC 2013)*, vol. 8283 of *Lecture Notes Comput. Sci.*, pp. 11–21. Springer, 2013.

- O. Aichholzer, T. Hackl, D. Orden, A. Pilz, M. Saumell, and B. Vogtenhuber. Flips in combinatorial pointed pseudo-triangulations with face degree at most four. In *Proc. XV Spanish Meeting on Computational Geometry (EGC 2013)*, pp. 131–134, 2013.

- O. Aichholzer, T. Hackl, A. Pilz, P. A. Ramos, V. Sacristan, and B. Vogtenhuber. Empty triangles in good drawings of the complete graph. In *Proc. Mexican Conference on Discrete Mathematics and Computational Geometry*, pp. 21–30. Oaxaca, Mexico, 2013.
- **O. Aichholzer, W. Mulzer, and A. Pilz. Flip distance between triangulations of a simple polygon is NP-complete. In H. L. Bodlaender and G. F. Italiano (Editors), *Proc. 21st European Symposium on Algorithms (ESA 2013)*, vol. 8125 of *Lecture Notes Comput. Sci.*, pp. 13–24. Springer, 2013.**
- **O. Aichholzer, W. Mulzer, and A. Pilz. Flip distance between triangulations of a simple polygon is NP-complete. In *Proc. 29th European Workshop on Computational Geometry (EuroCG 2013)*, pp. 115–118, 2013.**
- J. M. Díaz-Báñez, M. Korman, P. Pérez-Lantero, A. Pilz, C. Seara, and R. I. Silveira. New results on stabbing segments with a polygon. In P. G. Spirakis and M. J. Serna (Editors), *Proc. 8th International Conference on Algorithms and Complexity, (CIAC 2013)*, vol. 7878 of *Lecture Notes Comput. Sci.*, pp. 146–157. Springer, 2013.
- O. Aichholzer, T. Hackl, S. Lutteropp, T. Mchedlidze, A. Pilz, and B. Vogtenhuber. Monotone simultaneous embedding of directed paths. In *Proc. 30th European Workshop on Computational Geometry (EuroCG 2014)*, 2014.
- M. Korman, S. Langerman, W. Mulzer, A. Pilz, and B. Vogtenhuber. Minimum dual diameter triangulations. In *Proc. 30th European Workshop on Computational Geometry (EuroCG 2014)*, 2014.

Manuscript related to this thesis

- **S. Felsner and A. Pilz. Ham-sandwich cuts for abstract order types, 2014. In preparation.**

Master's thesis

- A. Pilz. *Parity Properties of Geometric Graphs.* Master's thesis, Graz University of Technology, 2009.