A STUDY OF CONVEX COVERS IN TWO OR MORE DIMENSIONS

by

Patrice Belleville

B.Sc., McGill University, Montréal, Canada, 1989

M.Sc., McGill University, Montréal, Canada, 1991

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Computing Science

© Patrice Belleville 1995

SIMON FRASER UNIVERSITY

October 1995

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.

APPROVAL

Name: Patrice Belleville

Degree: Doctor of Philosophy

Title of thesis: A study of convex covers in two or more dimensions

Examining Committee: Dr. Ramesh Krishnamurti
Chair

Dr. Thomas Shermer, Senior Supervisor

Dr. Binay Bhattacharya, Supervisor

Dr. Arvind Gupta, Supervisor

Dr. Mark Keil, External Examiner

Dr. F. David Fracchia, SFU Examiner

Date Approved: August 23, 1995

SIMON FRASER UNIVERSITY

PARTIAL COPYRIGHT LICENSE

ereby grant to Simon Fraser University the right to lend my thesis, project extended essay (the title of which is shown below) to users of the Simon ıser University Library, and to make partial or single copies only for such ɜrs or in response to a request from the library of any other university, or ıer educational institution, on its own behalf or for one of its users. I further ꞉ee that permission for multiple copying of this work for scholarly purposes ıy be granted by me or the Dean of Graduate Studies. It is understood that ɔying or publication of this work for financial gain shall not be allowed ꞉hout my written permission.

ːle of Thesis/Project/Extended Essay

Study of Convex Covers in Two or More Dimensions.

ıthor: ______________________

(signature)

____

(name)

September 15, 1995

____

(date)

Abstract

The problem of covering polytopes using simple shapes is central to computational geometry. In particular, a lot of attention has been given to the problem of covering a simple polytope by convex pieces. We call a polytope U_k if there is a collection of k convex sets whose union is that polytope, and B_k if there is a collection of k convex subsets of the polytope whose union contains its boundary. This thesis studies several aspects of the recognition problem for B_k and U_k simple polytopes.

We first give linear time algorithms to recognize B_3 and U_3 simple polygons. These algorithms are developed in three stages. In the first stage, we consider the convex subpolygons of a simple polygon P whose intersection with the boundary of P is a subset of a given set of m non-overlapping intervals. We show how to recognize the polygons whose boundary can be covered by k such subpolygons in $O(k^3m^{2k-2} + T_M(km^{k-1}))$ time, where $T_M(n)$ is the time required to multiply two $n \times n$ matrices (currently known to be in $o(n^{2.376})$). In the second stage, we characterize B_3 polygons by proving that the boundary of every B_3 polygon P can always be covered using a restricted class of convex subpolygons of P; this reduces the problem of recognizing B_3 polygons to that solved in the first stage. Finally, we show how to prune almost all of the covers that this algorithm considers to recognize B_3 and U_3 polygons in linear time.

We then study U_2 polytopes in three-dimensional space (they are the same as B_2 polytopes). We prove that they can be recognized in $O(n \log n)$ time using $O(n)$ space. We also show how to extend this algorithm to recognize U_2 polytopes in d-dimensional space in polynomial time, for every fixed value of d. Finally we present a negative result: we prove that the recognition problem for B_k or U_k polytopes in d-dimensional space is NP-hard for each fixed $d \geq 3$ and $k \geq 3$.

Acknowledgements

I would first like to thank my supervisor, Tom Shermer, for the discussions, the suggestions, and even the criticisms he made throughout the time of preparation of this thesis. Each of them has been a positive contribution. I would also like to thank the other two members of my committee: Binay Bhattacharya and Arvind Gupta, for enlightening discussions about these and other problems. Dave Peters and David Bremner also deserve thanks in that respect.

Thanks also go to the administrative staff in the school of computing science for their help with a thousand different matters: photocopying, software and network support, claim expenses, mail, and so on. I would also like to thank Godfried Toussaint for introducing me to computational geometry, and NSERC and Simon Fraser University for their financial support over the last four years.

Sheelagh Carpendale gets all my thanks and her own paragraph for helping me figure out how to design Figure 4.4.

My stay at Simon Fraser has been made more enjoyable by a number of other people who made me feel welcome. To name but a few, they are Diana Cukierman, Andrew Fall, Eric Guévremont, Lorene Gupta, Eli Hagen, Damon Kaller, Micheline Kamber, Glenn Macdonald, David Mitchell, Alicja Pierzynska, Jorg Ueberla, and many more.

J'aimerais aussi remercier mes parents pour leur appui constant. Finally, I wish to express my most heartfelt thanks to my wife, Wei. Without her constant support and love, I might not have been able to survive the two weeks that lead to the submission of this thesis. She also made the last two years the most enjoyable of my life. Thank you.

Dedication

To Wei, and to our unborn child, with love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis studies covers of simple polygons and polytopes by convex subsets, that is, collections of convex subsets of a given polytope whose union is that polytope. Our goal is to gain a better understanding of covering problems, more specifically those in which the cover elements are convex. In general, the convex cover problem, like most covering problems, is NP-hard. We are interested in determining which instances of this problem can be solved efficiently, and which ones probably cannot unless P = NP. In the latter case, we also want to understand the reasons that make a particular instance difficult.

Convex covers have applications in several different fields because they provide a decomposition of an input polytope into pieces that are relatively easy to handle algorithmically. Some problems for which convex covers have been used include:

Recognition of handwritten chinese characters: Each character is drawn using a certain number of strokes, and there are no closed, smooth loops as in the letter "o". The goal is thus to divide the input character into strokes. Once this is done other methods can examine the relative positions and angles of the strokes, and decide which character is being looked at. Feng and Pavlidis argued that the decomposition of the input character that will yield a good description of the strokes is that into convex pieces [48].

The chinese character shown in Figure 1.1a is the character used for the word

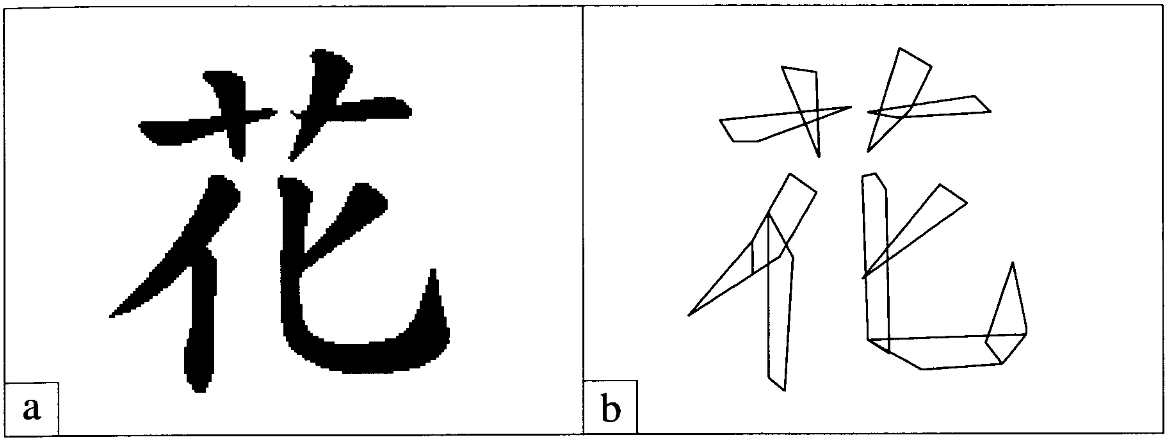


Figure 1.1: Covering the chinese character for "flower" using convex subsets

"flower". In Figure 1.1b, we illustrate a possible decomposition of this character into convex polygons after an edge detection algorithm has transformed some of the smoother curves into line segments.

Filling, shading and hidden surface removal: In order to perform these operations, it is necessary to decompose the shapes to be displayed (which may be very complicated) into simpler shapes. Convex polytopes are natural candidates for this decomposition, since most operations can be performed both efficiently and simply on them. For instance, the simple clipping algorithm of Sutherland and Hodgman [96] clips a polygon against either a convex polygon or polytope.

Robotics and motion planning: In order to compute the subset of space in which a robot is allowed to move, motion planning algorithms usually reduce the robot to a point, and inflate the obstacles by computing the set of points where placing the robot's reference point would cause it to intersect the obstacle. For each obstacle, this can be done by computing the *Minkowski sum* of the obstacle with the set obtained by first placing the robot's reference point at the origin, and then reflecting the translated robot through the origin. Computing Minkowski sums of sets is much simpler when both sets are convex than in the general case. Hence one possible strategy to compute these is to decompose each set into a small number of convex subsets, and then find the union of the Minkowski sums of each pair of subsets. Because of the techniques used to represent the space in which the robot is allowed to move, motion planning is an area in which higher dimensional spaces are the norm rather than the exception, and in which it is sometimes desirable to only consider covers of the boundary of an object.

In the remainder of this chapter, we introduce the concepts that will be used in the thesis. Since we will be dealing with polytopes in d-dimensional Euclidean space, we start with definitions related to planes, vectors, and polytopes. Next, we introduce a concept called *visibility.* We then describe the various kinds of cover by convex subsets that we will be concerned with. In the following two sections, we introduce the logic problem called *3-satisfiability,* and some elements of graph theory that will be needed later. Finally, we outline the organization of the remainder of the thesis.

1.1 Geometric preliminaries

This section introduces basic definitions related to the space in which all polytopes that we will consider are contained: the d-dimensional Euclidean space, denoted by $\boldsymbol{E}^d$. The length of a vector $\overrightarrow{x} = (x_1, \ldots, x_d)$, denoted by $|\overrightarrow{x}|$, is the real number $\sqrt{\sum_{i=1}^d x_i^2}$. The Euclidean distance between two points $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$ in $\boldsymbol{E}^d$ will be denoted by $d(x, y)$ and is the positive real number $|\overrightarrow{x - y}|$. Given a point x of $\boldsymbol{E}^d$, and a positive real number ε, the set of all points y of $\boldsymbol{E}^d$ for which $d(x, y) < \varepsilon$ will be denoted by $N_\varepsilon(x)$. This is sometimes called an ε-*ball* around x, or an ε-*neighborhood* of x.

Let S be a subset of $\boldsymbol{E}^d$. The *interior* of S, denoted by $int(S)$, is the set of all points x of $\boldsymbol{E}^d$ for which there exists some positive real number ε such that $N_\varepsilon(x) \subseteq S$. The *exterior* of S, denoted by $ext(S)$, is the interior of the complement of S. The *boundary* of S, denoted by $bd(S)$, is the set $\boldsymbol{E}^d \setminus (int(S) \cup ext(S))$.

The *inner product* of two vectors $\overrightarrow{n}_x = (x_1, \ldots, x_d)$ and $\overrightarrow{n}_y = (y_1, \ldots, y_d)$, denoted by $\overrightarrow{n}_x \cdot \overrightarrow{n}_y$, is the real number $\sum_{i=1}^d x_i y_i$. The inner product of two vectors is a measure of the angle between these vectors: if θ is the angle between $\overrightarrow{n}_x$ and $\overrightarrow{n}_y$, then $\overrightarrow{n}_x \cdot \overrightarrow{n}_y = |\overrightarrow{n}_x||\overrightarrow{n}_y| \cos\theta$. The *cross product* of two vectors $\overrightarrow{n}_x = (x_1, x_2, x_3)$ and $\overrightarrow{n}_y = (y_1, y_2, y_3)$

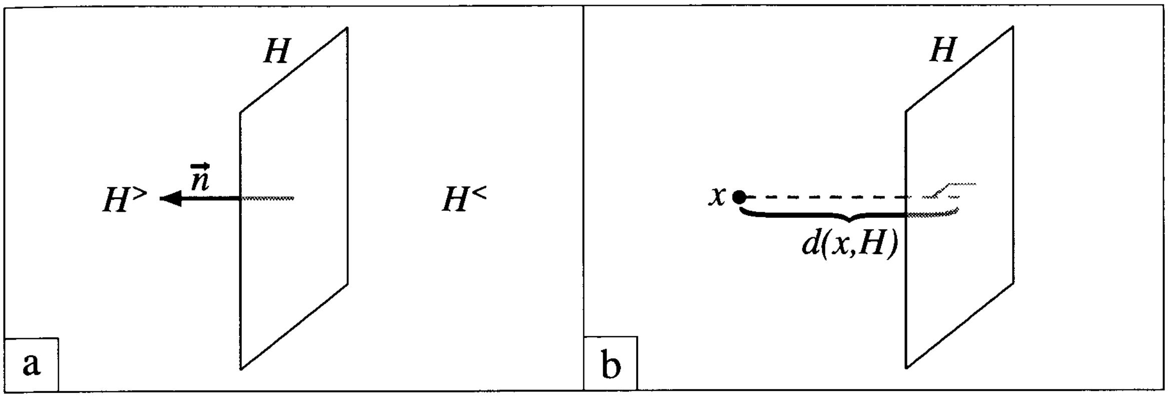


Figure 1.2: Halfspaces determined by a plane, and distance from that plane

in $\boldsymbol{E}^3$, denoted by $\vec{n}_x \times \vec{n}_y$, is the vector $(x_2y_3 - y_2x_3, x_3y_1 - y_3x_1, x_1y_2 - y_1x_2)$. The cross product of $\vec{n}_x$ and $\vec{n}_y$ is perpendicular to both $\vec{n}_x$ and $\vec{n}_y$. Additional properties of the inner and cross products can be found in any linear algebra or introductory calculus textbook [62].

Given a vector $\vec{n}$ in $\boldsymbol{E}^d$, and a point p of $\boldsymbol{E}^d$, the plane H through p with normal vector $\vec{n}$ is the set of all points x of $\boldsymbol{E}^d$ such that $\vec{n} \cdot (\overrightarrow{x - p}) = 0$. For each $op \in \{<, >, \leq, \geq\}$, we will denote by H^{op} the set of all points x of $\boldsymbol{E}^d$ for which $\vec{n} \cdot (\overrightarrow{x - p})\ op\ 0$. The sets H, $H^>$ and $H^<$ are illustrated in Figure 1.2a. The signed distance between a point x of $\boldsymbol{E}^d$ and a plane H with normal vector $\vec{n}$, denoted by $d(x, H)$, is the real number d for which there exists a point y of H such that $x = y + d\vec{n}/|\vec{n}|$ (as shown in Figure 1.2b). We note that the sign of d depends on the choice of $\vec{n}$.

1.2 Polygons and polytopes

The objects that are considered in this thesis are *polygons* in the plane, and their relatives in higher dimensions, *polytopes.* A *polygon* is a set defined by an ordered list of $n \geq 2$ points $v_1, \ldots, v_n$ in the plane called *vertices*, and n line segments $e_1, \ldots, e_n$ called *edges*, where e_i joins v_i to $v_{1+i \bmod n}$ for each i in $\{1, \ldots, n\}$. A polygon is *simple* if no two non-consecutive edges intersect, and if every pair of consecutive edges intersect in a point. The set drawn in Figure 1.3a is a simple polygon, while

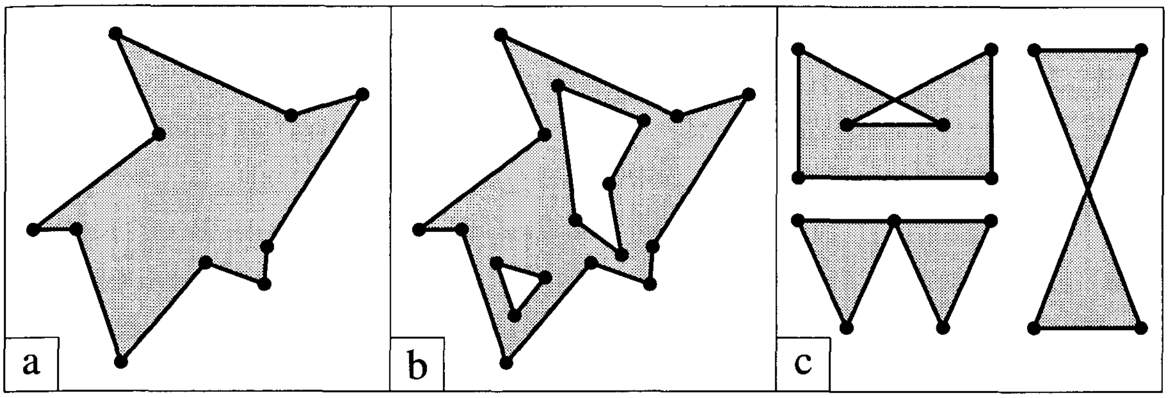


Figure 1.3: A simple polygon, a polygon with holes, and sets that are not either

those shown in Figures 1.3b and 1.3c are not.

A simple polygon is a Jordan curve, and hence its divides the plane into three regions: the polygon itself, the *interior* of the polygon (the bounded region), and the *exterior* of the polygon (the unbounded region). In this thesis, we will use the term *polygon* to refer to the union of a polygon with its interior. Hence the set of points that belong to the edges of a polygon will be called the *boundary* of that polygon. To simplify the notation, all arithmetic on indices of polygon vertices and edges will be performed modulo n.

A *subpolygon* of P is a simple polygon contained in P. A *hole* in a polygon P is a subpolygon of P whose boundary is disjoint from $bd(P)$. A *polygon with holes* is a simple polygon, together with a collection of pairwise disjoint holes in that polygon. The set drawn in Figure 1.3b is a polygon with holes, but those shown in Figure 1.3c are not. A *chord* of P is a line segment that does not intersect the exterior of P, and whose endpoints belong to the boundary of P.

We will call a subset of the boundary of P an *interval* if it is closed, connected, and contains at least two distinct points. Starting at any vertex of a polygon P, we can follow the boundary of P in two opposite directions. A traversal of the boundary of P (and by extension of any of its connected subsets) is called *counterclockwise* if the interior of P lies to the left of the directed line segment $\overrightarrow{v_iv_{i+1}}$ in the neighborhood of the midpoint of $\overrightarrow{v_iv_{i+1}}$. In this thesis, all enumerations of polygon vertices will be done in counterclockwise order.

We can view a polygon as a two-dimensional set bounded by a collection of one-dimensional sets (line segments). Similarly, for $1 \leq k \leq d$, a polytope P in $\boldsymbol{E}^d$ is a k-dimensional set bounded by a collection of $(k-1)$-dimensional sets, called the *facets* of P, that are themselves polytopes. More formally, we define k-dimensional polytopes in $\boldsymbol{E}^d$ (hereafter called k-polytopes) as follows.

A 0-polytope in $\boldsymbol{E}^d$ is a point. By convention, a 0-polytope has exactly one facet, namely the empty set. A 1-polytope (a line segment) is a pair of 0-polytopes. For $k \geq 2$, a k-polytope P is a collection $\mathcal{F}$ of $(k-1)$-polytopes, called the *facets* of P, with the following property:

Every facet of a facet of P (a $(k-2)$-polytope) belongs to exactly two facets of P $(*)$

A *subface* of P (sometimes called a *face* of P) is either P, or recursively a subface of a facet of P. We will call a subface of P of dimension k a k-face of P. We use the terms *vertices* and *edges* to refer to the 0-faces and the 1-faces of P, and denote the closure of a face f of P (the union of f with its boundary) by $cl(f)$. Hereafter, each face of a polytope P will be considered open, that is, it does not contain its boundary.

Two subfaces of P are called *adjacent* if they have a common subface and neither is a subface of the other. We will also call two vertices of P *adjacent* when they are joined by an edge. P is *simply-connected* if no strict subset of its facets possesses property $(*)$. P is *simple* if it is simply-connected, if no pair of non-adjacent subfaces of P intersect, and if the intersection of two adjacent subfaces of P is exactly their common subface. We note that this definition of simplicity differs from the one used in the theory of convex polytopes. In the latter, *simple* refers to a polytope in which each vertex belongs to exactly d facets [100]. For instance, the polytope shown in Figure 1.4 is simple under the first definition (ours), but not under the second since v_1 is incident upon four facets.

Throughout this thesis, we shall assume that a polytope is represented by its *face lattice* [55,100], also called *incidence graph* by Edelsbrunner [41] (graphs are formally introduced in Section 1.6). The face lattice $\mathcal{F}$ of a polytope P contains one vertex for each subface of P. Two vertices v, v' of $\mathcal{F}$ are joined by an edge if the subface of

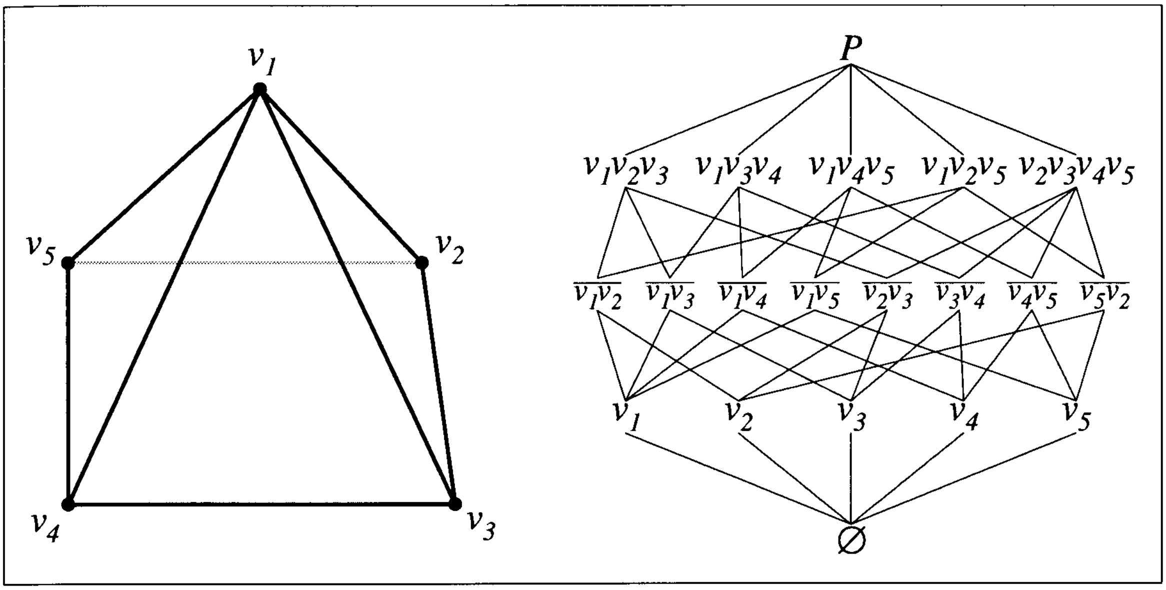


Figure 1.4: A polytope and its face lattice

P corresponding to v is a facet of the subface of P corresponding to v', or vice versa. Figure 1.4 shows a polytope P, and its face lattice.

1.3 Visibility

Every type of subset of $\boldsymbol{E}^d$ that appears in this thesis is defined using a concept called *visibility*. Two points x and y of a subset S of $\boldsymbol{E}^d$ are said to be *visible*, or equivalently x *sees* y, if the line segment $\overline{xy}$ is contained in S. Two subsets S_1, S_2 of S are *completely visible* if each point of S_1 sees every point of S_2. The point x is *link-j visible* from y if there are points $x = x_0, \ldots, x_j = y$ of S such that x_i sees x_{i+1} for each i in $\{0, \ldots, j-1\}$. We note that the usual notion of visibility is the same as link-1 visibility. In Figure 1.5a, x and y are visible, but x and z are not, although they are link-2 visible.

A point of *local nonconvexity* of S is a point x of S such that, for every $\varepsilon > 0$, there are two points of $S \cap N_\varepsilon(x)$ that do not see each other. A subface f of a polytope P will be called *reflex* or *concave* if some point of f is a point of local nonconvexity of P (we note that the points commonly called *saddle points* are reflex using this

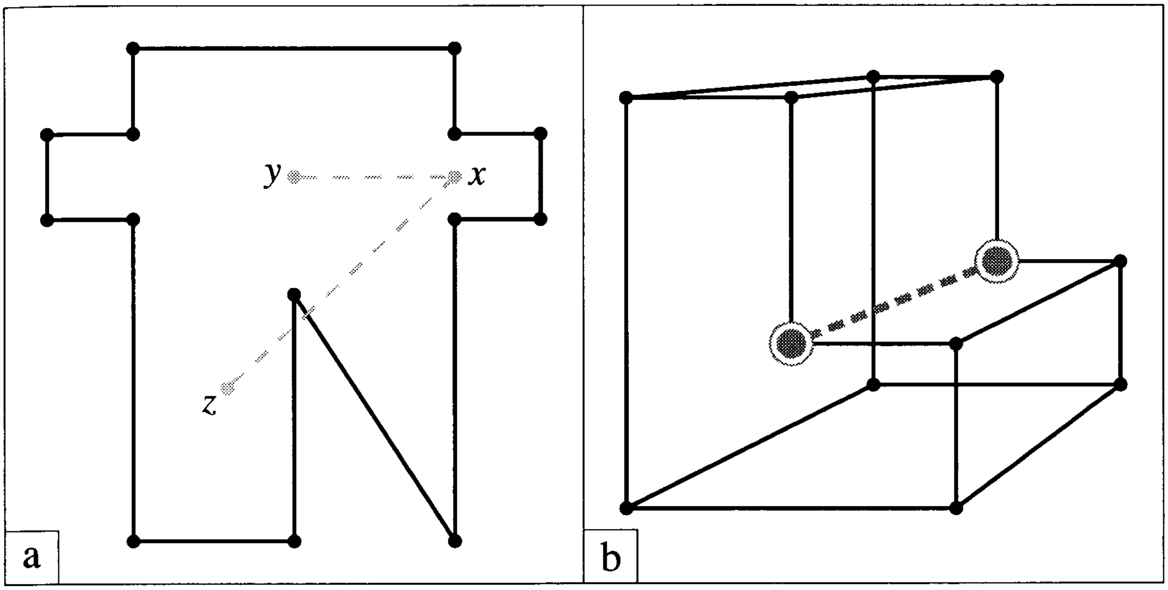


Figure 1.5: Visibility and reflex subfaces

definition). In Figure 1.5b, the reflex subfaces of the polytope are those that are shaded. A subface of P that is not reflex will be called *convex*. In the plane, the angle determined by three points x, y and z in this order, denoted by $\angle xyz$, is called *convex* if z lies to the left or on the line through x and y, and oriented from x to y. If $\angle xyz$ is not convex, then it is called *reflex*.

Consider a facet f of P contained in a plane H. In the neighborhood of any interior point of f, exactly one of the two halfspaces determined by H contains points of P. This halfspace will be called the *interior halfspace* of f; in Figure 1.6a, the interior halfspace of edge e has been shaded.

A set S is *convex* if every two points of S are visible, and *link-j convex* if every two points of S are link-j visible. The *convex hull* of S, denoted by $ch(S)$, is the smallest convex set that contains S. A d-dimensional *simplex* is the convex hull of $d+1$ points in $\boldsymbol{E}^d$. The *visibility region from x in S* is the set of points of S which are visible from a given point x of S. The *link-j visibility region from x* is defined similarly using link-j visibiliy.

S is called *starshaped* if there exists a point x of S that sees every point of S. The *kernel* of a starshaped set S, denoted by $kr(S)$, is the set of all points of S that

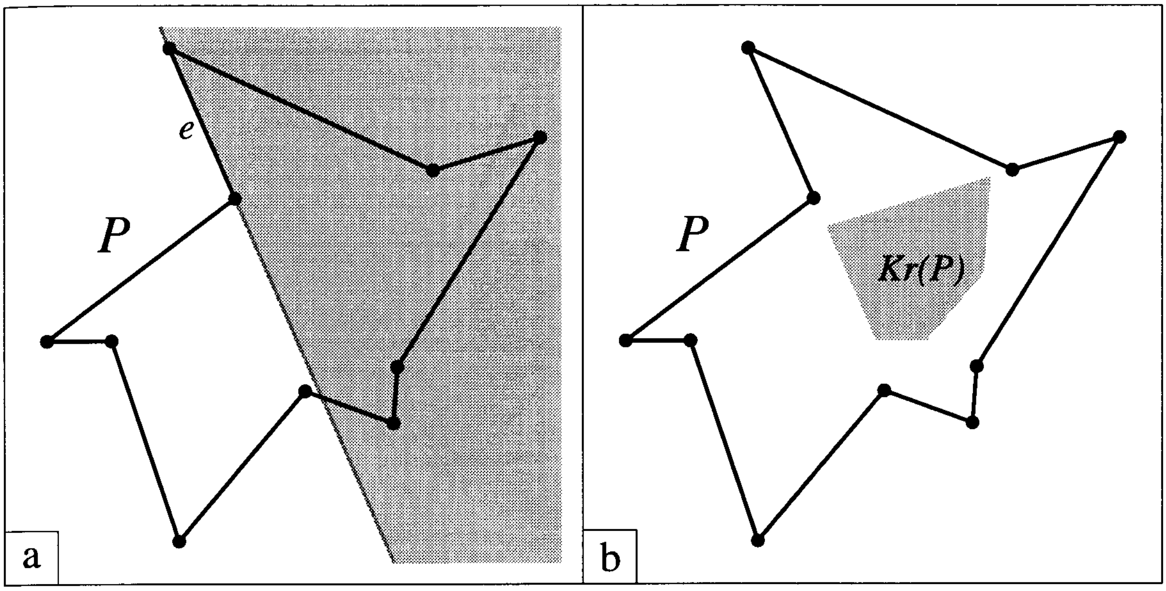


Figure 1.6: Interior halfspaces and starshapedness

see every point of S. Equivalently, the kernel of a polytope P can be defined as the intersection of all interior halfspaces of P; hence the kernel is convex. Figure 1.6b shows a starshaped polygon and its kernel.

The *vertex visibility graph* of a polytope P is the graph that has one node for every vertex of P, and in which two nodes are joined by an edge if the corresponding vertices of P are visible [45]. The *point visibility graph* of a set S is the graph whose vertex set is the set of points of S, and in which two points are joined by an edge if they are visible in S [89,90]. The point visibility graph is a graph with an uncountable number of vertices and edges.

1.4 Covers and partitions

We are now ready to define formally what we mean by a *cover* of a set. Let S be a subset of $\boldsymbol{E}^d$, and S' be contained in S. A collection $\{S_1, \ldots, S_k\}$ of subsets of S *covers* S' if $S' \subseteq \cup_{i=1}^k S_i$. Except for some of the results reviewed in Chapter 2, the sets S and $S_1, \ldots, S_k$ will be simple polytopes throughout this thesis. The set S' will be a subset of the set of subfaces of S.

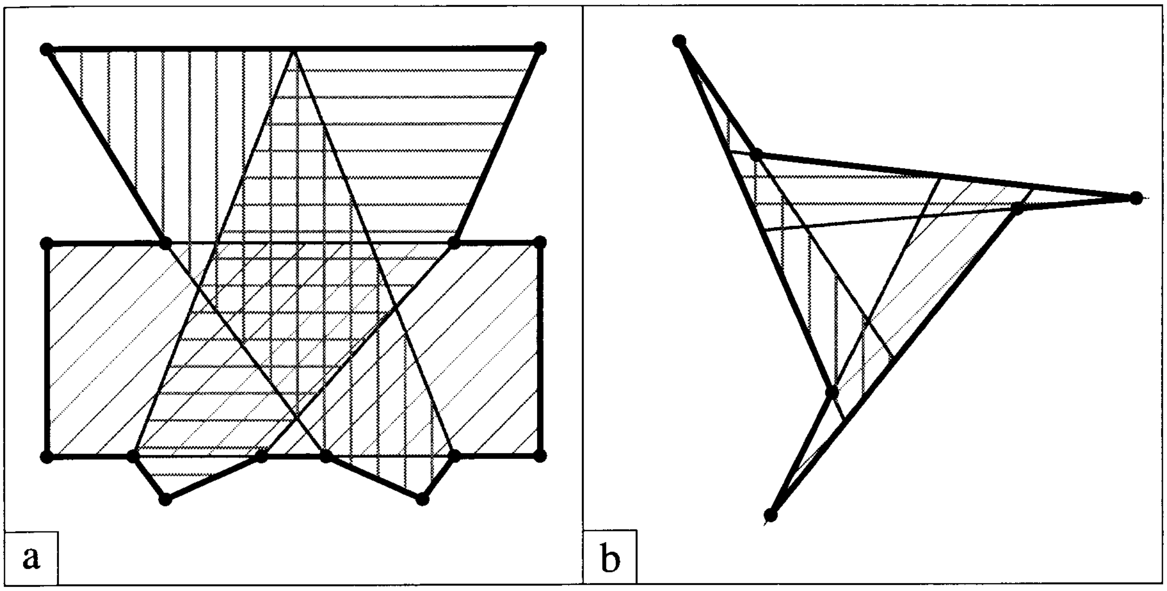


Figure 1.7: (a) A U_3 polygon. (b) A B_3 polygon that is not U_3

A set S is called U_k if there is a collection of k convex subsets of S whose union is S, and B_k if there is a collection of k convex subsets of S whose union contains the boundary of S. The polygon shown in Figure 1.7a is U_3, while that drawn in Figure 1.7b is B_3 but not U_3.

A notion related to covers of S by k convex subsets is that of a *hidden set*: a *hidden set* inside S is a set of points of S, no two of which see each other (such a set is sometimes called *visually independent*). A set is said to have *property* P_k if it does not contain a hidden set of size k. The relation between hidden sets and minimum size covers of sets by convex subsets is the following: if a set S is U_k, then S has property P_{k+1}. However, the converse of this is not necessarily true.

A *partition* of a set S in $\boldsymbol{E}^d$ is a cover of S in which the d-dimensional volume of the intersection of any two cover elements is zero. The type of partition most often considered in computational geometry is the *triangulation*, which subdivides a polytope P into simplices whose vertices are vertices of P. Triangulations of polytopes will be mentioned again in Chapter 2. We will encounter a different type of triangulation in Section 1.6.

In Chapter 3, we shall consider covers of the boundary of a simple polygon P by

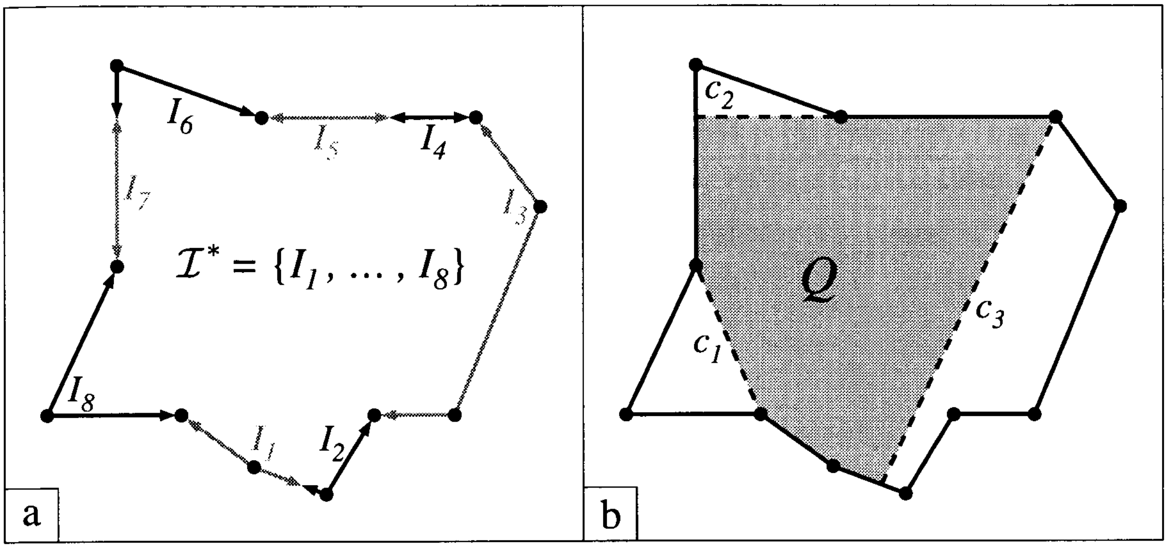


Figure 1.8: A partition $\mathcal{I}^*$ of $bd(P)$ into intervals, and an $\mathcal{I}^*$-elementary convex subpolygon Q of P whose intersection with $bd(P)$ is $\{I_1, I_4, I_5, I_7\}$.

convex subsets of P on which additional constraints have been imposed. Let $\mathcal{I}^*$ be a partition of the boundary of P into intervals, such as that shown in Figure 1.8a, and Q be a subpolygon of P. The intersection of Q with the boundary of P consists of a set $\mathcal{I}$ of zero or more pairwise disjoint intervals, and of a set of zero or more isolated points. Q will be called *$\mathcal{I}^*$-elementary* if it is convex, and if each element of $\mathcal{I}$ is the union of a set of elements of $\mathcal{I}^*$. For instance, the subpolygon Q of P shown in Figure 1.8b is $\mathcal{I}^*$-elementary, since its intersection with the boundary of P consists of the three intervals I_1, $I_4 \cup I_5$, and I_7. We will say that P is B_k with respect to $\mathcal{I}^*$ if there is a collection of k $\mathcal{I}^*$-elementary subpolygons of P whose union contains the boundary of P.

1.5 3-satisfiability

In this section, we define the problem called *3-satisfiability*. This problem is often used in polynomial-time reductions to prove problems NP-hard. The reader unfamiliar with the theory of NP-completeness is referred to the book by Garey and Johnson [50].

Let U be a set of boolean variables. A *truth assignment* is a function $t : U \rightarrow$

{ *true*, *false*}. A *literal* is an element x of U or its complement $\overline{x}$. A literal x is *true* under t if $x \in U$ and $t(x) =$ *true*, or if $\overline{x} \in U$ and $t(\overline{x}) =$ *false*. A literal x is *false* under t if $\overline{x}$ is *true* under t. A *clause* c over U is a set of literals; c is *satisfied* by t if at least one element of c is *true* under t. A set C of clauses admits a *satisfying truth assignment* if there is a truth assignment t that satisfies every clause of C.

Using these notions, the problem called 3-satisfiability (first proved NP-complete by Karp [64]) can be defined as follows.

3-satisfiability (3SAT)

Input: A set C of 3-element clauses over a set U of boolean variables.

Output: *Yes* if C admits a satisfying truth assignment, *no* otherwise.

1.6 Graph theory

A *graph* G is a pair (V, E), where V is a set of elements called *vertices* or *nodes*, and E is an irreflexive, symmetric relation on $V \times V$, that is, a set of two-element subsets of V (see for instance the book by Bondy and Murty [12]). Each element of E is an *edge* of G. An edge of G is *incident* upon a vertex v of G if v belongs to e. A *subgraph* of G is a graph (V', E') such that $V' \subseteq V$ and $E' \subseteq E$. An *induced subgraph* of G is a subgraph of G in which E' contains every element $e = \{v, v'\}$ of E such that $v \in V'$ and $v' \in V'$. Two graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic*, denoted by $G \approx G'$, if there is a bijection $f : V \longrightarrow V'$ such that $\{v, v'\} \in E$ if and only if $\{f(v), f(v')\} \in E'$.

Consider an injection ψ that maps the set V of vertices of G into the plane, and each edge $\{v, v'\}$ of G into a path (called an *edge* of (G, ψ) by abuse of terminology) whose endpoints are $\psi(v)$ and $\psi(v')$. The pair (G, ψ) is a *plane graph* if the following two conditions hold:

- For each vertex v of G, the only edges of (G, ψ) that contain $\psi(v)$ are those incident upon v.

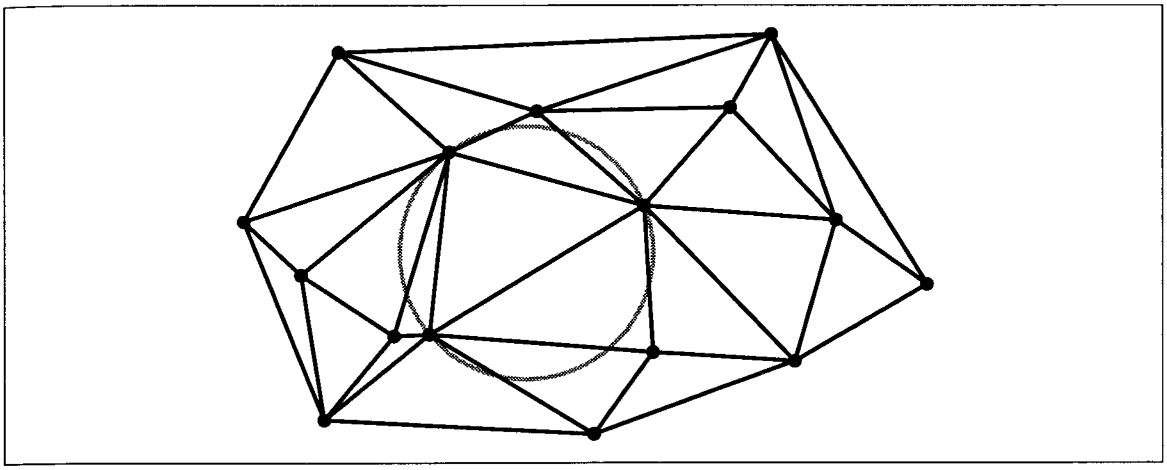


Figure 1.9: The Delaunay triangulation of a point set

- If a point x belongs to two distinct edges of (G, ψ), then x is the image of a vertex of G under ψ.

A graph G is called *planar* if there exists an embedding ψ for which (G, ψ) is a plane graph. By abuse of notation, we will hereafter write about "a plane graph G" when the embedding ψ is known from the context. Throughout this thesis, the paths to which edges of G are mapped by ψ will be line segments. Fary proved that every planar graph can be embedded in this way [47].

A *triangulation* of a set V of points in the plane is a maximal set E of edges for which (V, E) is a plane graph. Each maximal connected subset of $\mathbf{R}^2 \setminus E$ is called a *face* of the triangulation. We note that each face of the triangulation, except possibly the outer face (the face corresponding to the unbounded region), is a triangle.

The *Delaunay triangulation* of V is the triangulation of V in which every triangular face $\triangle vv'v''$ except the outer face has the geometric property that the interior of the circle through v, v' and v'' does not contain any point of V. This property will be used in Section 5.2. Figure 1.9 shows a set of points in the plane, along with its Delaunay triangulation and one of the empty circles circumscribing a face of that triangulation. A plane graph (V, E) will be called a *Delaunay subgraph* if E is a subset of the edges in the Delaunay triangulation of V.

A *k-coloring* of a graph $G = (V, E)$ is a function $\chi : V \to \{1, \ldots, k\}$. A k-coloring

χ of G is called *proper* if $\chi(v) \neq \chi(v')$ for every edge $\{v, v'\}$ of G. If G admits a proper k-coloring, it will be called k-*colorable.* The *chromatic number* of a graph G is the integer k such that G is k-colorable but not $(k-1)$-colorable.

We can decide whether a graph is 2-colorable in linear time using a depth-first search. However, determining whether a graph is 3-colorable is NP-complete, even if the graph is known to be planar [50,52,64]. In Chapter 5, we will consider the following even more restricted version of this problem, and prove that it is also NP-complete.

Delaunay subgraph 3-colorability (DS3C)

Input: A Delaunay subgraph G.

Output: *Yes* if G is 3-colorable, *no* otherwise.

1.7 Organization of this thesis

This thesis studies several aspects of the recognition problem for B_k and U_k simple polytopes. In Chapter 2 we survey the work that has been done on covering problems. In particular, we look at a large number of results that can be found in the mathematical literature. These theorems present various characterizations of sets that admit certain types of cover, or have hidden sets with given properties. We then review some of the reasons why covering problems of the type studied here are difficult. Finally we survey some of the algorithmic results that have been obtained.

In Chapter 3 we consider B_3 and U_3 simple polygons in $\boldsymbol{E}^2$, and give a linear time algorithm to recognize them. One of the tools used to obtain this result is an algorithm to recognize polygons that are B_k with respect to a given partition $\mathcal{I}^*$ of $bd(P)$ into intervals in $O(k^3|\mathcal{I}^*|^{2k-2} + T_M(k\,|\mathcal{I}^*|^{k-1}))$ time, where $T_M(n)$ is the time required to multiply two $n \times n$ matrices (currently known to be in $o(n^{2.376})$ [35]). We observe that this result is interesting in its own right, since it indicates that for every fixed k, the difficulty in recognizing B_k polygons arises from the lack of information about the possible locations of the vertices of the cover elements, rather than from the way in which these points can be used to obtain a cover.

In Chapter 4, we prove that U_2 polytopes in $\boldsymbol{E}^3$ can be recognized in $O(n \log n)$ time using $O(n)$ space. We also show that U_2 polytopes in $\boldsymbol{E}^d$ can be recognized in polynomial time for each fixed value of d.

In Chapter 5, we consider the common generalization of the problems studied in Chapters 3 and 4: recognizing B_3 and U_3 simple polytopes in $\boldsymbol{E}^3$. We prove that this problem is NP-hard using a reduction from the problem *Delaunay subgraph 3-colorability* (DS3C) defined in Section 1.6. The main difficulty lies in proving that the vertices of the polytope obtained from each instance of DS3C can be represented using a small number of bits (or rather in showing how to construct a polytope for which this is the case). DS3C is proved NP-hard using a reduction from 3SAT.

Finally, we present the conclusion and open problems in Chapter 6.

Chapter 2

Literature review

Covers of simple polytopes by convex subsets have received a fair amount of attention in the computational geometry literature. The results obtained can be broadly classified into four categories: complexity results, structural results, algorithmic results and combinatorial results. The complexity results prove that certain problems belong to or are hard for one of the following complexity classes: P, NP, co-NP, PSPACE, etc. The structural results characterize sets that admit a particular type of convex cover. The algorithmic results show how to find a cover of a polytope using a minimum number of cover elements of the given kind. The combinatorial results give upper and lower bounds on the number of cover elements needed, in the worst case, for a polytope with a given number of subfaces. Most of the combinatorial results that establish upper bounds are in fact constructive, and allow covers with a number of elements equal to the upper bound to be found relatively efficiently. Every algorithmic and combinatorial result makes use of one or more structural results, although structural results can exist on their own.

This thesis concentrates on the first, second and third categories. We first review the structural results that have been obtained by the mathematics community on the convex cover problem. These theorems present necessary and/or sufficient conditions for polytopes to be U_k. We then examine results that have been obtained concerning the complexity of covering problems for simple polytopes. In particular, we will see why a large number of these problems are not known to belong to NP. Finally, we survey the algorithmic results that have been obtained. Most of these consider special types of polytope and/or cover; we divide these results further based on the additional constraints that have been imposed on either the input polytope or the cover elements. We review the work done on *orthogonal* polygons (polygons all of whose edges are either horizontal or vertical), on partitions, on covers by a constant number of convex subsets, and finally on covers by convex subsets whose vertices and/or edges have been restricted.

2.1 Structural results

Recent interest for covering problems in the mathematical literature started with a paper by Valentine, in which he considers U_2 sets in the plane [99]. Since then, a large number of results characterizing sets that admit covers of a certain type have appeared. However, we should first mention a classical characterization of U_1 sets due to Tietze: a set is convex (U_1) if and only if it does not have a point of local nonconvexity [97].

2.1.1 U_2 sets and the property P_3

Valentine was the first to consider covers of sets by convex subsets [99], and to relate them to the size of the largest visually independent subset of these sets. He proved several facts about sets with property P_3. For convenience, we group them into a single theorem.

Theorem 2.1 (Valentine)

a. *If S is a closed connected set in $\boldsymbol{E}^d$ that has property P_3, then every point of local nonconvexity of S belongs to its kernel.*

b. *Let S be a closed connected set in $\boldsymbol{E}^2$ that has property P_3, and let r be the number of isolated points of local nonconvexity of S. If $r = 1$ or r is even, then S can be expressed as the union of two closed convex sets having a non-empty* *intersection. If r is odd and greater than 1, then S can be expressed as the union of at most three such sets.*

In the case of a simple polygon, the statement of Theorem 2.1a can be rewritten as: "If S is a closed connected set in $\boldsymbol{E}^d$ that has property P_3, then every reflex vertex of S belongs to its kernel". Following the terminology of Shermer [92], we say that a set S is *KR* if every point of local nonconvexity of S belongs to $kr(S)$. If we know that a set is P_3, then Theorem 2.1b allows us to conclude that it is U_2 whenever $r = 1$ or r is even. The remaining cases (r odd and at least three) were investigated by Stamey and Marr, who proved the following theorem [94].

Theorem 2.2 (Stamey and Marr) *Let S be a closed connected set in $\boldsymbol{E}^2$ that has property P_3, and let r be the number of isolated points of local nonconvexity of S. If r is odd and greater than 1, then S is U_2 if and only if there is a point of $kr(S) \cap bd(S)$ that is not a point of local nonconvexity of S.*

This characterization remained unused by the computational geometry community until recently, when Shermer applied it to obtain linear time algorithms to recognize polygons with property P_3, and U_2 polygons [92]. He observed that if a *KR* polygon has more than three reflex vertices, then it has property P_3. This implies that for simple polygons, having property P_3 and being *KR* are equivalent whenever $r > 3$. Therefore most facts that hold for sets with property P_3 are in fact true for *KR* polygons. Since we can determine in linear time whether a simple polygon is *KR* [72], the proofs of Theorems 2.1b and 2.2 yield an immediate algorithm.

A short time after Stamey and Marr proved Theorem 2.2, McKinney provided a different characterization of U_2 sets in $\boldsymbol{E}^d$ [76], which we shall use in a slightly different form in Chapter 4. He proved that a closed connected set S is U_2 if and only if for every finite subset $x_1, \ldots, x_n$ of S with n odd and such that x_i does not see x_{i+1} for $i = 1, \ldots, n-1$, the points x_1 and x_n are visible.

The proof of this result was greatly simplified by Hare and Kenelly [59], who defined a graph $G(S)$ that is the complement of the point visibility graph of S, and proved that S is U_2 if and only if $G(S)$ is 2-colorable. We observe that this result is not completely obvious, as one needs to show that the convex hull of the set of points of a given color does not contain points of $ext(S)$. McKinney reported using the same technique to prove that a closed connected S is U_3 if and only if $G(S)$ is 3-colorable, and that a closed simply-connected planar set is U_k if and only if $G(S)$ is k-colorable. McKinney also reported that these results fail in higher dimensions when $k \geq 4$ [77].

Finally, Buchman proved that if S is a closed and bounded subset of $\boldsymbol{E}^d$ $(d \geq 3)$ with at least one point interior to its kernel (that is, $kr(S)$ has dimension d), and if the set of points of local nonconvexity of S is interior to the convex hull of S, then S is P_3 if and only if S is U_2 [25]. He noted that this result does not hold in the plane.

2.1.2 U_k sets and the property P_k

The first partial characterization of U_k sets was proved by Guay and Kay. They proved that if the set R of points of local nonconvexity of a closed subset S of $\boldsymbol{E}^d$ has finite cardinality k, and is such that $S \setminus R$ is connected, then S is either convex (that is U_1) or planar, and in the latter case is U_{k+1} [56].

Breen then provided two different partial characterizations of U_k sets, by extending both of Theorems 2.1b and 2.2 to $\boldsymbol{E}^d$ (in different directions). She extended Theorem 2.2 by proving that if S has property P_k, if the convex hull of the set R of points of local nonconvexity of S is contained in S, and if $(bd(S) \cap kr(S)) \setminus R$ is not empty, then S is U_{k-1} [15]. We note that this theorem cannot be tranformed into an algorithm, since this condition is only sufficient and not necessary. Moreover, we observe that the problem of determining whether a polytope is P_k is NP-hard when k is part of the input [89].

The proof of Theorem 2.1b given by Valentine proceeds by constructing a polygon P contained in $kr(S)$ from the set of points of local nonconvexity of S. The bound on the number of convex subsets of S required to cover S arises from the chromatic number of the graph whose nodes correspond to the edges of P, and whose edges join two nodes whose corresponding edges of P have a common endpoint. Breen showed how to extend this construction to d dimensions, proving the following theorem [16].

Theorem 2.3 (Breen) *Let S be a closed subset of $\boldsymbol{E}^d$, and let R be the set of points of local nonconvexity of S. If S is KR and R coincides with the set of points in the union of all $(d-2)$-faces of some d-dimensional polytope P, then S is decomposable into $c(P')$ convex sets, where $c(P')$ is the chromatic number of the graph whose nodes correspond to the facets of P, and whose edges join two nodes whose corresponding facets of P have a common $(d-2)$-face.*

By applying Theorem 2.3 to three-dimensional polytopes, we see that every *KR* polytope in $\boldsymbol{E}^3$ is the union of three or four convex sets. We shall prove in Chapter 5 that the problem of determining which one is the case is NP-hard.

Eggleston extended the theorems of Valentine and of Stamey and Marr in a different direction. He proved that every closed planar set with property P_k is the union of finitely many convex sets [43]. Breen and Kay improved on this result by providing an explicit bound of $(k-1)^3 2^{k-3}$ on the number of convex sets required [21]. Determining whether such a bound exists for closed sets with property P_3 in $\boldsymbol{E}^3$ remains open. However, when $d \geq 4$, not all compact sets in $\boldsymbol{E}^d$ with property P_3 can be expressed as a union of finitely many convex sets [44].

A set S is called (k,t)-*convex* if given any subset S' of S with k elements, at least t of the $\binom{k}{2}$ possible line segments connecting pairs of points of S' are contained in S. A set with property P_k is $(k,1)$-convex. Properties of (k,t)-convex sets were investigated by Breen [13]. She also studied sets that can be expressed as a union of two starshaped subsets in a series of papers [14, 17–20, 22]. We shall not give any details about these results, since this thesis concentrates on covers by convex sets.

2.2 The complexity of covering problems

Unfortunately, in most cases the problem of finding minimum size covers of simple polytopes with subsets that have a given property is NP-hard, even in the plane. The problem of determining whether a simple polygon with holes is U_k was proved NP-hard by O'Rourke and Supowit [84]. The problem of recognizing U_k and B_k simple polygons was shown to be NP-hard by Culberson and Reckhow [37], and independently by

Shermer [90]. This problem is a special case of the following problem:

Link$_{j,m}$-guarding

Input: A polygon P and an integer k.

Output: *Yes* if there is a collection C of at most k link-m convex subpolygons of P such that P is covered by the link-j visibility regions of the elements of C, *no* otherwise.

Shermer proved that Link$_{j,m}$-guarding is NP-hard when at least one of j and m is positive, whether all of P or only its boundary needs to be covered [90]. The convex cover problem is the case $m = 1$, $j = 0$. The case $m = 0$, $j = 1$ (finding a minimum size cover of P by starshaped polygons) had already been proved NP-hard by O'Rourke and Supowit when the polygon to be covered has holes [84], and by Lee and Lin, and Aggarwal when it is simple [1, 71].

Even though these problems are all NP-hard, it is not clear that they belong to NP. In order to place these problems in NP, we would need a method of verifying that a cover using the given number of subsets exists. The natural way to do this would be to guess the cover elements, and then verify that they do cover the polytope. However, no bound is known on the number of bits required to represent the vertices of the cover elements in a minimum size cover. Let us write that a type of cover of a polytope P *requires* a specific kind of point if *every* cover of P of that type has at least one element with a vertex located at that kind of point. It is thus possible that for each large enough n, there is a polygon of n vertices that requires a point whose representation needs $\Omega(2^n)$ bits. We note that this type of configuration is known to exist for order types [54].

O'Rourke was the first to investigate this problem [80–82]. On the positive side, he showed how to express the problem of recognizing U_k polygons as a formula in the existential theory of the reals, whose size is polynomial in the size of the description of the input polygon [81]. It thus follows from a result of Canny that this problem belongs to PSPACE [26].

On the other hand, O'Rourke also gave several examples that seem to show that as the number of vertices of the polygon to cover increases, the number of bits required to represent the vertices of the cover elements increases faster than the number of bits used to represent the vertices of the polygon. In order to do this, he considers points whose representation requires an increasingly larger number of bits. The *order* of a point, denoted by $\omega(x)$, is thus defined as follows:

- $\omega(x) = 0$ if and only if x is a vertex of P.
- $\omega(x) = k \in \mathbf{N}$ if and only if k is the smallest integer for which there are four points p_1, p_2, p_3, p_4 of P such that
  - $k = 1 + \max\{\omega(p_1), \omega(p_2), \omega(p_3), \omega(p_4)\}$;
  - not all four of p_1, p_2, p_3 and p_4 are collinear;
  - x is collinear with p_1 and p_2, and with p_3 and p_4.

We note that a point of P has order zero or one if and only if it is the intersection of two lines, each of which contains two vertices of P. O'Rourke gave an example of a polygon in which *every* convex cover with minimum cardinality requires a point of order two [82]. This polygon is drawn in Figure 2.1a. Every minimum size cover of P looks like the cover shown in Figure 2.1b, and hence the rightmost vertex of the leftmost diamond-shaped cover element must lie on the thick gray line segment. Since this line segment contains no point of order 1, the desired result is obtained.

O'Rourke conjectured that examples requiring points of any finite order can be constructed. However no proof of this fact has been found, for several reasons. First, the polygons in his construction become very complicated very quickly, and hence are difficult to specify. Second, constraining the cover to behave as we want it to becomes problematic. Finally, the number of points of order k is $O(n^{4^k})$, and hence verifying by exhaustive enumeration that none of them occurs in a particular region of the polygon becomes infeasible in practice.

O'Rourke also conjectured that the number of vertices of the polygon that are required in order to force at least one vertex of every minimum convex cover to have order greater than or equal to k grows at least linearly with k. The exact relationship between the number of bits required and the size of the original polygon is thus unclear,

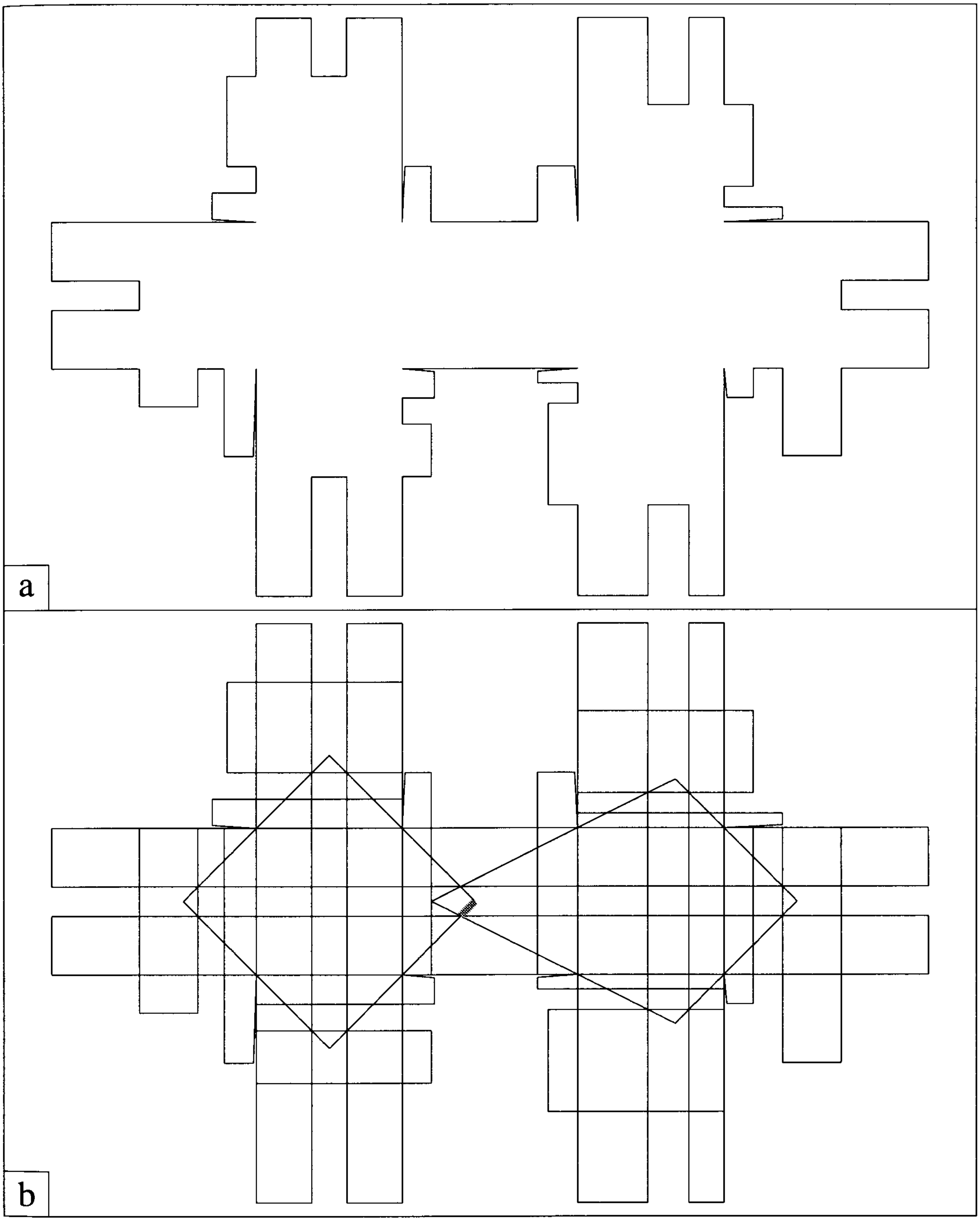


Figure 2.1: A polygon in which every minimum convex cover uses a point of order at least 2

and determining whether U_k and B_k polygons can be recognized in NP remains an outstanding open problem. We will not answer this question, but these considerations will be brought up for B_3 polygons in Chapter 3.

2.3 Algorithmic results

Because the convex cover problem is NP-hard, a large number of variations in which the input polytope or the cover elements are constrained have been studied. These types of problem have many real-world applications. The first important variation is that in which the input polytope is an orthogonal polygon. These polygons have applications in VLSI design, where circuits are build out of rectangular components and wires are either horizontal or vertical, and in graphics, since a large proportion of the objects in the real world are orthogonal. A second variation that is very widely used consists of disallowing overlapping cover elements, that is in considering partitions instead of covers. One may also fix the number of cover elements that can be used; this leads to several interesting results. Finally, we look at covers in which the vertices or the edges of the cover elements have been constrained in some other way.

2.3.1 Orthogonal polygons

Orthogonal polygons possess structural properties that allow some covering problems to be solved more efficiently than for arbitrary simple polygons. Consider the partition of the plane induced by the set of lines containing the edges of an orthogonal polygon P. For all of the problems mentioned in this section, the vertices and edges of the cover elements can be assumed to be vertices and edges of that partition of the plane. Hence every problem discussed here trivially belongs to NP, contrarily to what happens in the case of simple polygons.

Consider for instance the problem consisting of covering an orthogonal polygon with a minimum number of squares. Aupperle et al. proved that this problem is NP-hard for orthogonal polygons with holes, and gave an $O(p^{1.5})$ time algorithm to solve this problem in orthogonal polygons without holes, where p is the number of pixels in the bitmap representation of the input [6]. This was improved to $O(n \log^* n)$ by Bar-Yehuda and Ben-Chanoch, where n is the number of vertices of the polygon [8]. We observe that p (and the number of squares required) may in general be in $\Omega(2^m)$ where m is the number of bits required to represent the vertices of the polygon. For this reason, the algorithm of Bar-Yehuda and Ben-Chanoch requires that a set of squares covering a rectangle be replaced by that rectangle in the output.

The related problem of covering orthogonal polygons with rectangles is however NP-hard. This was proved by Masek for polygons with holes [75]. Conn and O'Rourke extended this result to the case in which only the boundary of the polygon needs to be covered [34], and Berman and DasGupta did the same for the set of corners (convex vertices) [10, 11]. In the case of orthogonal polygons without holes, the problem was proved NP-hard by Culberson and Reckhow when either the interior or the boundary of the polygon needs to be covered [37]. In fact, Berman and DasGupta proved that no polynomial time approximation scheme exists for either unless P = NP [10, 11]. In the same papers, Berman and DasGupta also gave approximation algorithms for various cases of the boundary and corner cover problems.

When dealing with orthogonal polygons, the definition of visibility given in Section 1.3 is not the only one that is natural. One might also want to say that two points x, y of an orthogonal polygon P are visible if there is a path from x to y that intersects every line parallel to one of the coordinate axis in at most one connected component (such a path is often referred to as a *staircase* path). This type of visibility is called *orthogonal visibility*, and is used for instance in VLSI design, where every wire on a chip must be either horizontal or vertical. The points x and y shown in Figure 2.2a are orthogonally visible, but x and z are not.

A *dent* of an orthogonal polygon P is an edge of P whose endpoints are both reflex. We note that each dent of P can be oriented in one of four ways, depending on the direction of the edge and on the position of the two edges of P adjacent to it. Dents play the same role in orthogonal visibility as reflex vertices in the usual type of visibility.

Convexity and starshapedness can be defined for orthogonal visibility in the same

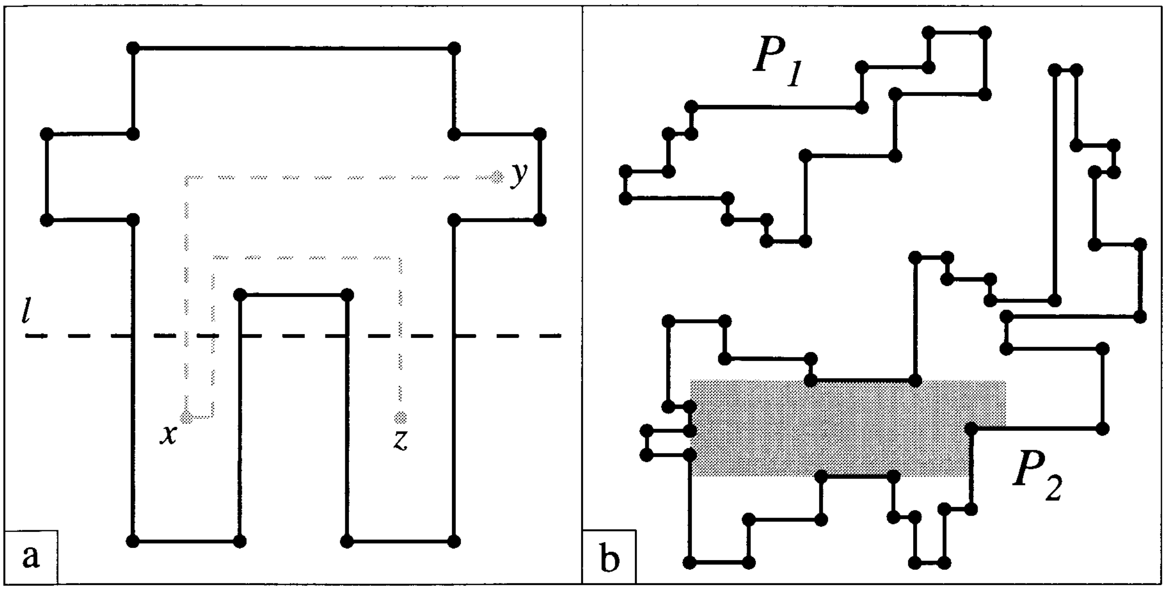


Figure 2.2: Orthogonal visibility, convexity and starshapedness

way as for regular visibility. In Figure 2.2b, P_1 is orthogonally convex and P_2 is an orthogonally convex star (whose kernel is shaded). The covering problems that have been investigated for simple polygons and for orthogonal polygons using the usual type of visibility have also been studied for orthogonal polygons using orthogonal visibility. However, because of the additional freedom granted by orthogonal visibility, some problems that are NP-hard using the usual notion of visibility become tractable. For instance, Motwani et al. showed that a minimum cover of an orthogonal polygon by orthogonally convex stars can be computed in $O(n^8)$ time [79,86].

The complexity of finding a minimum cover of a simple orthogonal polygon using orthogonally convex subsets remains open. However many results have been obtained for restricted classes of orthogonal polygons. If the polygon P is *horizontally convex*, i.e. if the intersection with P of every line parallel to the x-axis is either empty or connected, then Franzblau and Kleitman showed how to find a minimum size cover of P using rectangles in $O(n^2)$ time [49]. Keil gave an $O(n^2)$ time algorithm to cover the same class of polygons using orthogonally convex or starshaped subsets [67]. This corresponds to the case in which the polygon contains dents in only two opposite orientations. Culberson and Reckhow [38], and Motwani et al. [79] showed how to extend these results to cover orthogonal polygons with three dent orientations, and a subset of orthogonal polygons with four dent orientations, using orthogonally convex subsets.

Finally, Bremner and Shermer studied an extension of orthogonal visibility for simple polygons called *$\mathcal{O}$-visibility*, in which two points of the polygon are $\mathcal{O}$-visible if there is a path between them whose intersection with every line in the set $\mathcal{O}$ of orientations is either empty or connected [23, 24]. For instance, orthogonal visibility corresponds to $\mathcal{O} = \{0°, 90°\}$. A polygon P is *$\mathcal{O}$-convex* if every two points of P are $\mathcal{O}$-visible, and *$\mathcal{O}$-starshaped* if there is a point of P from which every other point of P is $\mathcal{O}$-visible. Bremner and Shermer characterized classes of orientations for which minimum covers of a simple polygon by $\mathcal{O}$-convex or $\mathcal{O}$-starshaped subsets can be found in polynomial time.

2.3.2 Polytope partitions

Partitions are usually easier to compute than more general covers, because of the additional constraint that cover elements may not overlap. The most often used type of partition of a polytope is its *triangulation* (it is not a minimum size partition). In the plane, Garey et al. gave a simple $O(n \log n)$ time algorithm to triangulate polygons with or without holes [51], and a triangulation of a simple polygon is now known to be computable in linear time using an algorithm due to Chazelle [30]. More surprisingly, a partition of a simple polygon into convex pieces using a minimum number of such pieces can also be found in polynomial time. Chazelle gave a very complicated algorithm that finds such a partition in $O(n^3)$ time [27,32]. This problem becomes NP-hard for polygons with holes [75, 84].

In three dimensions, not all polytopes can be triangulated [7, 74, 88], and in fact, Ruppert and Seidel proved that the problem of determining whether a simple polytope can be triangulated is NP-hard [87]. There are however some classes of simple polytopes that we know can be triangulated (apart from convex polytopes). In particular, U_3 polytopes can always be triangulated [98]. We shall however prove in Chapter 5 that the problem of recognizing U_3 polytopes is NP-hard.

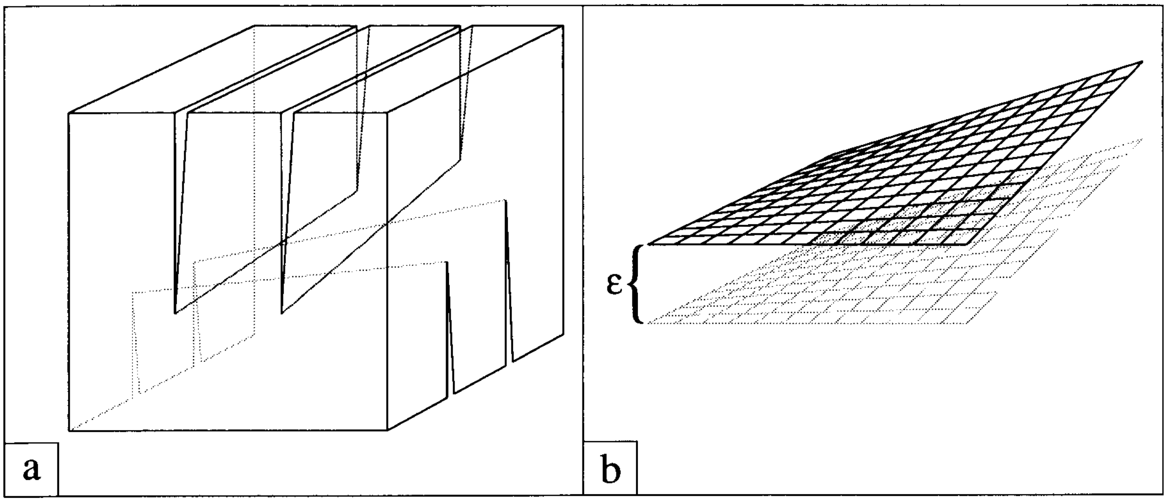


Figure 2.3: A polytope of which every convex partition needs $\Omega(n^2)$ pieces.

Chazelle gave an $O(nr^3)$ time algorithm to find a convex partition (not necessarily of minimum cardinality) of a polytope, where r is the number of reflex edges of the polytope [28]. He also proved the existence of polytopes in $\boldsymbol{E}^3$ with n vertices of which every convex partition has $\Omega(n^2)$ pieces; such a polytope P is shown in Figure 2.3a. The deeper edges of the top and bottom notches belong to the two hyperbolic paraboloids $z = xy + \varepsilon$ and $z = xy$ respectively; these paraboloids are shown in Figure 2.3b. For a suitably small value of ε that depends on n, Chazelle proved that at least $n^2/66$ convex subsets of P are required to cover the region between the notches.

The complexity of the minimum partition problem using starshaped polygons remains unknown. For orthogonal polygons, Asano and Imai solved the minimum rectangular partition problem in $O(n^{1.5} \log n)$ time when the polygon may have holes [5]. This result was extended to polygons that may have degenerate holes (points or line segments) by Soltan and Gorpinevitch [93]. Liou, Lee and Tan showed how to find such a partition in $O(n)$ time when the polygon has no hole [73].

2.3.3 Covers using a constant number of subsets

Trying to find covers by a constant number k of convex subsets can be useful for two reasons. First there are applications in which it is known beforehand that the number of cover elements will be small (for instance the problem of recognizing handwritten chinese characters, discussed in the introdution). Second, studying these types of cover may lead to a better understanding of a the convex cover problem, since the structure of the solution, being more constrained, is also easier to interpret.

The case in which $k = 1$ is very well understood. Simple polytopes that are U_1 or B_1 (convex polytopes) can be recognized using a trivial linear time algorithm exploiting Tietze's theorem [97]. Link-k convex polygons can be recognized in $O(n \log n)$ time using algorithms due to Suri and Ke [65, 95].

Starshaped polytopes can be recognized in $O(n)$ time in the plane by using the fact that the kernel of a simple polytope P is the intersection of the interior halfspaces of the facets of P [72]. Using the same fact, starshaped polytopes can be recognized in $O(n \log n)$ time in $\boldsymbol{E}^3$ [41], and in $O(n^{\lfloor d/2 \rfloor})$ time using Chazelle's optimal convex hull algorithm when $d \geq 4$ [31]. Alternatively, linear programming can be used to recognize starshaped polytopes in polynomial time for any value of d [63, 68], or in $O(n)$ time for each fixed value of d (with a constant factor proportional to 3^{d^2}) [33,40].

The case $k = 2$ is also reasonably well understood, although not as well as the previous, simpler case. Shermer gave a linear time algorithm to recognize U_2 polygons [92], as explained in Section 2.1.1. We observe that a simple polytope is U_2 if and only if it is B_2. Belleville gave an algorithm to recognize simple polygons that are the union of two link-k convex polygons in $O(n^2)$ time [9]. He also found a rather complicated $O(n^4)$ time algorithm to recognize polygons that are the union of two starshaped polygons [9].

The case $k \geq 3$ is not nearly as well understood. We prove in this thesis that B_3 and U_3 polygons can be recognized in linear time, but that determining whether a starshaped polytope in $\boldsymbol{E}^d$ is B_k or U_k is NP-hard for every $k \geq 3$ and $d \geq 3$. However nothing is known about unions of three starshaped or link-k convex polygons, or about unions of more than three convex polygons in the plane.

Finally, Bremner and Shermer proved that for every finite set of orientation and every fixed value of k, minimum covers by $\mathcal{O}$-convex and $\mathcal{O}$-starshaped subsets can be found in time polynomial in the number of vertices of the polygon [23,24]. Their algorithms however require time exponential in both k and the cardinality of $\mathcal{O}$.

2.3.4 Constraining vertices and edges

The last restriction that has been studied consists of constraining the location of the vertices or edges of the cover elements. When every vertex of every cover element must be a vertex of the polygon, Keil showed that a minimum partition of a simple polygon into convex polygons can be found in $O(n^3 \log n)$ using dynamic programming, and that a minimum partition of a simple polygon into starshaped polygons can be found in $O(n^7 \log n)$ time using a similar method [66,67].

Convex covers whose elements are bounded by the extensions of the edges of the polygon through their reflex endpoints were studied by Feng and Pavlidis [48]. Aggarwal et al. gave an $O(n^4 \log n)$ time approximation algorithm that covers a simple polygon with starshaped subpolygons subject to a similar restriction using at most $O(M \log n)$ elements, where M is the size of the minimum such cover of the polygon [2]. Finally, Ghosh gave an $O(n^5 \log n)$ time algorithm that finds a set of starshaped polygons covering P whose kernels contain at least one vertex of P each, with at most $O(\log n)$ time the minimum number of such sets [53].

Chapter 3

Recognizing U_3 polygons

In this chapter, we show how to recognize U_3 and B_3 polygons in linear time using an algorithm that determines whether a simple polygon is B_k with respect to a given partition $\mathcal{I}^*$ of its boundary into intervals. This algorithm is described in Section 3.1. In Section 3.2, we consider the types of point that may be required in a cover of the boundary of a B_3 polygon by three convex subsets of that polygon. We prove that if a simple polygon P is B_3, then there exists a specific partition of $bd(P)$ into a linear number of intervals with respect to which P is B_3. Moreover we give a characterization of the endpoints of these intervals from which they can be computed in linear time. In Section 3.3, we show how the running time of the recognition algorithm for B_3 polygons derived from the results of Section 3.1 and 3.2 can be reduced to $O(n)$. Finally we conclude this chapter by applying this algorithm to recognize U_3 polygons.

3.1 Deciding whether P is B_k with respect to $\mathcal{I}^*$

Let P be a simple polygon, and $\mathcal{I}^* = \{I_1, \ldots, I_m\}$ be a partition of $bd(P)$ into a set of intervals. The description of the algorithm to recognize polygons that are B_k with respect to $\mathcal{I}^*$ proceeds as follows: we first prove that every $\mathcal{I}^*$-elementary subpolygon of P can be replaced by a subset of $\mathcal{I}^*$ with a certain property (these subsets of $\mathcal{I}^*$ will be called *convex*). We thereafter simplify our discussion by using convex subsets of $\mathcal{I}^*$ instead of $\mathcal{I}^*$-elementary subpolygons of P. We then show how union and intersection of these subsets of $\mathcal{I}^*$ can be used to combine covers of subsets of $bd(P)$. Next, we describe the manner in which the subsets of $\mathcal{I}^*$ are subdivided into a polynomial number of equivalence classes; this is required in order to obtain an algorithm that is polynomial in $|\mathcal{I}^*|$, since the number of subsets of $\mathcal{I}^*$ may be in $\Omega(2^{|\mathcal{I}^*|})$. We then show how sets of equivalence classes are merged. Finally we combine these ideas into an algorithm.

3.1.1 Subsets of $\mathcal{I}^*$

We assume, without loss of generality, that every reflex vertex of P is an endpoint of an element of $\mathcal{I}^*$, as otherwise P is not B_k with respect to $\mathcal{I}^*$, and that $k \geq 2$. Instead of covering $bd(P)$ using $\mathcal{I}^*$-elementary subpolygons of P, we will cover $\mathcal{I}^*$ using subsets of $\mathcal{I}^*$ with a given property. We shall prove in Lemma 3.1 that the two problems are equivalent.

If I belongs to $\mathcal{I}^*$, we denote the clockwisemost and counterclockwisemost points of I by $cw(I)$ and $ccw(I)$ respectively. Let x_0 be the the clockwise endpoint of I_1. The choice of x_0 orders the elements of $\mathcal{I}^*$ as follows: I_i precedes I_j if $cw(I_i)$ is met before $cw(I_j)$ during a counterclockwise traversal of $bd(P)$ starting and ending at x_0. We will assume without loss of generality that the elements of $\mathcal{I}^*$ are numbered so that I_i precedes I_j if and only if $i < j$.

The first and last elements of a subset $\mathcal{I}$ of $\mathcal{I}^*$, with respect to the order thus defined on $\mathcal{I}^*$, will be denoted by $\mathit{first}(\mathcal{I})$ and $\mathit{last}(\mathcal{I})$ respectively, and the element of $\mathcal{I}$ immediately following an element I of $\mathcal{I}^*$ by $succ_{\mathcal{I}}(I)$. We define $succ_{\mathcal{I}}(\mathit{last}(I)) = \mathit{first}(I)$. If $\mathcal{I} = \emptyset$, then we write $\mathit{first}(\mathcal{I}) = \mathit{last}(\mathcal{I}) = \mathit{nil}$, where nil represents an unspecified interval.

We can construct a polygon Q from $\mathcal{I}$ by connecting the counterclockwise endpoint of each element of $\mathcal{I}$ to the clockwise endpoint of the following one (as in Figure 1.8b). In other words, Q is the polygon whose boundary consists of the elements of $\mathcal{I}$ in sorted order, along with the line segment $\overline{ccw(I)cw(succ_{\mathcal{I}}(I))}$ for each element I of $\mathcal{I}$. We call Q the *covering polygon* for $\mathcal{I}$, and call each directed line segment $\overline{ccw(I)cw(succ_{\mathcal{I}}(I))}$, such that $ccw(I) \neq cw(succ_{\mathcal{I}}(I))$, a *chord* of $\mathcal{I}$. In Figure 1.8b, the chords of Q are

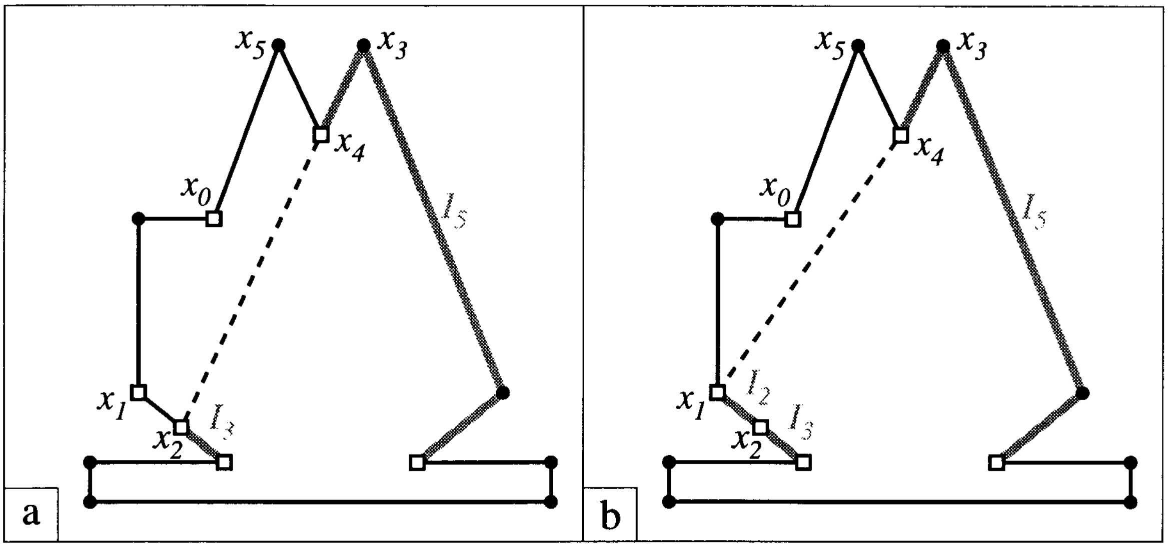


Figure 3.1: A convex subset $\{I_3, I_5\}$ and a subset $\{I_2, I_3, I_5\}$ that is not convex.

drawn using dashed lines and labeled c_1, c_2 and c_3.

To decide whether the polygon Q obtained from $\mathcal{I}$ is $\mathcal{I}^*$-elementary, we use a type of visibility that we call *convex visibility*. Given two points x, y on the boundary of P, we want to say that x *sees* y *convexly* if the chord $\overline{xy}$ can be used as an edge, in a convex subpolygon of P, to join an interval of $bd(P)$ ending at x to an interval of $bd(P)$ starting at y. More formally, let e_i, e_j be edges of P, and x, y be points of $e_i \setminus \{v_i\}$ and $e_j \setminus \{v_{j+1}\}$ respectively. We say that x *sees* y *convexly*, denoted by $x \simeq y$, if

- $x = y$ and x is not a reflex vertex of P, or
- $x \neq y$, x sees y, and the angles $\angle v_i x y$ and $\angle x y v_{j+1}$ are both convex.

In Figures 3.1a and 3.1b, x_2 sees x_4 convexly (because both of the angles $\angle x_1 x_2 x_4$ and $\angle x_2 x_4 x_5$ are convex), but x_4 does see itself convexly (since it is reflex), and x_4 does not see x_1 convexly (since the angle $\angle x_3 x_4 x_1$ is reflex).

A subset $\mathcal{I}$ of $\mathcal{I}^*$ will be called *semi-convex* if $ccw(I)$ sees $cw(succ_{\mathcal{I}}(I))$ convexly for each element I of $\mathcal{I}$ distinct from $last(\mathcal{I})$. The set $\mathcal{I}$ will be called *convex* if it is semi-convex and $ccw(last(\mathcal{I}))$ sees $cw(first(\mathcal{I}))$ convexly. The subset $\{I_3, I_5\}$ shown in

Figure 3.1a is convex, while the subset $\{I_2, I_3, I_5\}$ shown in Figure 3.1b is semi-convex, but not convex because $x_4 \not\preceq x_1$.

We now have all the definitions required to prove the equivalence between $\mathcal{I}^*$-elementary subpolygons of P and convex subsets of $\mathcal{I}^*$.

Lemma 3.1 *Let P be a simple polygon and S be a subset of $\mathcal{I}^*$. There exists a convex subset of $\mathcal{I}^*$ that covers S if and only if there is an $\mathcal{I}^*$-elementary subpolygon of P that covers every element of S.*

Proof: If Q is an $\mathcal{I}^*$-elementary subpolygon of P that contains every element of S, then the set of intervals of $Q \cap bd(P)$ containing more than one point is a subset of $\mathcal{I}^*$ that covers S. Conversely, let $\mathcal{I}$ be a convex subset of $\mathcal{I}^*$ containing every element of S, and let Q be the covering polygon for $\mathcal{I}$. Since $\mathcal{I} \subseteq \mathcal{I}^*$ and every reflex vertex of P is an endpoint of an element of $\mathcal{I}^*$, Q is $\mathcal{I}^*$-elementary. Moreover $\mathcal{I} \subseteq Q$, and hence $S \subseteq Q$. Finally Q is a convex subpolygon of P since $\mathcal{I}$ is convex. □

3.1.2 Restriction and composition

We now define two operations on covers of a subset $\mathcal{I} = \{I_a, \ldots, I_c\}$ of $\mathcal{I}^*$. The first operation is called a *restriction* and allows covers of subsets of $\mathcal{I}$ to be constructed from covers of $\mathcal{I}$. The second operation is called a *composition* and allows covers of $\mathcal{I}$ to be built from covers of its subsets. For the remainder of this section, we shall consider a cover to be an *ordered list* of subsets of $\mathcal{I}^*$, rather than a set. A cover C of $\mathcal{I}$ will be called *semi-convex* if each of its elements is a semi-convex subset of $\mathcal{I}$, and *convex* if each of its elements is a convex subset of $\mathcal{I}$.

The *restriction* of a list C of subsets of $\mathcal{I}^*$ to $\mathcal{I}$, denoted by $rest(C, \mathcal{I})$, is the list obtained by replacing each element of C by its intersection with $\mathcal{I}$. We use restrictions to obtain covers of subsets of $\mathcal{I}$ from covers of $\mathcal{I}$.

Lemma 3.2 *Let $\mathcal{I} = \{I_a, \ldots, I_c\}$ be a subset of $\mathcal{I}^*$, and let b satisfy $a \leq b < c$. If $C = (\mathcal{I}_1, \ldots, \mathcal{I}_k)$ is a semi-convex cover of $\mathcal{I}$, then $rest(C, \{I_a, \ldots, I_b\})$ and $rest(C, \{I_{b+1}, \ldots, I_c\})$ are semi-convex covers of $\{I_a, \ldots, I_b\}$ and $\{I_{b+1}, \ldots, I_c\}$ respectively.*

Proof: It suffices to consider $\mathcal{I}' = \{I_a, \ldots, I_b\}$, since the proof for $\{I_{b+1}, \ldots, I_c\}$ is identical. For each i in $\{1, \ldots, k\}$, let $\mathcal{I}'_i = \mathcal{I}_i \cap \mathcal{I}'$. Clearly $rest(C, \mathcal{I}') = (\mathcal{I}'_1, \ldots, \mathcal{I}'_k)$ covers $\mathcal{I}'$. It remains to prove that $\mathcal{I}'_i$ is semi-convex. Consider an element I_x of $\mathcal{I}'_i$. If $I_x \neq last(\mathcal{I}'_i)$, then $succ_{\mathcal{I}'_i}(I_x) \in \{I_a, \ldots, I_b\}$. Hence $succ_{\mathcal{I}'_i}(I_x) = succ_{\mathcal{I}_i}(I_x)$, which implies that $ccw(I_x) \simeq cw(succ_{\mathcal{I}'_i}(I_x))$ since $\mathcal{I}$ is semi-convex. This holds for every element of $\mathcal{I}'_i$ distinct from $last(\mathcal{I}'_i)$, and hence each $\mathcal{I}'_i$ is semi-convex. □

Let a, b and c be integers such that $1 \leq a \leq b < c \leq m$. If $C = (\mathcal{I}_1, \ldots, \mathcal{I}_k)$ is a semi-convex cover of $\{I_a, \ldots, I_b\}$, and $C' = (\mathcal{I}'_1, \ldots, \mathcal{I}'_k)$ is a semi-convex cover of $\{I_{b+1}, \ldots, I_c\}$, then the *composition* of C and C', denoted by $C \circ C'$, is the list $(\mathcal{I}_1 \cup \mathcal{I}'_1, \ldots, \mathcal{I}_k \cup \mathcal{I}'_k)$. We conclude this subsection with a result that implies the converse of Lemma 3.2; the proof of this lemma follows from the definition of semi-convexity.

Lemma 3.3 *Let a, b and c be integers such that $1 \leq a \leq b < c \leq m$, and let $\mathcal{I}$, $\mathcal{I}'$ be semi-convex subsets of $\mathcal{I}^*$ whose elements belong to $\{I_a, \ldots, I_b\}$ and $\{I_{b+1}, \ldots, I_c\}$ respectively. If $last(\mathcal{I}) = nil$, $first(\mathcal{I}') = nil$ or $ccw(last(\mathcal{I})) \simeq cw(first(\mathcal{I}'))$, then $\mathcal{I} \cup \mathcal{I}'$ is a semi-convex subset of $\mathcal{I}^*$.*

Hence we can determine whether $C \circ C'$ is semi-convex by considering only the following $2k$ points: $ccw(last(\mathcal{I}_1)), \ldots, ccw(last(\mathcal{I}_k)), cw(first(\mathcal{I}'_1)), \ldots, cw(first(\mathcal{I}'_k))$. We observe that $2k$ is independent of the cardinality of $\{I_a, \ldots, I_c\}$.

3.1.3 Dividing semi-convex covers into equivalence classes

We want to compute a convex cover of $\{I_a, \ldots, I_c\}$ by subsets of $\{I_a, \ldots, I_c\}$. This cover will be obtained by composition from a semi-convex cover of $\{I_a, \ldots, I_b\}$ and a semi-convex cover of $\{I_{b+1}, \ldots, I_c\}$, for some $a \leq b < c$. However, it is not clear how to determine a priori which semi-convex covers of $\{I_a, \ldots, I_b\}$ and $\{I_{b+1}, \ldots, I_c\}$ will be used. We thus need to compute *all* of them; we will do the same for semi-convex covers of $\{I_a, \ldots, I_c\}$. We note that we can determine whether a semi-convex cover of $\{I_a, \ldots, I_c\}$ is convex using conditions symmetric to those of Lemma 3.3.

Since the number of semi-convex covers of $\{I_a, \dots, I_c\}$ can be in $\Omega(2^{c-a})$, we cannot obtain an efficient algorithm if we compute them explicitly. We therefore subdivide the set of semi-convex covers of $\{I_a, \dots, I_c\}$ into equivalence classes. We call two covers C, C' *equivalent* if for each pair $\mathcal{I}_i$, $\mathcal{I}'_i$ of corresponding elements of C and C', we have $\mathit{first}(\mathcal{I}_i) = \mathit{first}(\mathcal{I}'_i)$ and $\mathit{last}(\mathcal{I}_i) = \mathit{last}(\mathcal{I}'_i)$.

If C and C' are two equivalent semi-convex covers of $\{I_a, \dots, I_b\}$, and C^* is a semi-convex cover of $\{I_{b+1}, \dots, I_c\}$, then $C \circ C^*$ is semi-convex if and only if $C' \circ C^*$ is semi-convex, and moreover $C \circ C^*$ and $C' \circ C^*$ are equivalent. Hence, instead of computing sets of semi-convex covers, we will compute sets of equivalence classes. The equivalence class $\mathcal{C}$ containing a cover $C = (\mathcal{I}_1, \dots, \mathcal{I}_k)$ is uniquely identified by a pair of k-tuples:

- $\mathit{first}(\mathcal{C}) = (\mathit{first}(\mathcal{I}_1), \dots, \mathit{first}(\mathcal{I}_k))$;
- $\mathit{last}(\mathcal{C}) = (\mathit{last}(\mathcal{I}_1), \dots, \mathit{last}(\mathcal{I}_k))$.

We will denote the set of all equivalence classes thus defined from the set of semi-convex covers of $\{I_a, \dots, I_c\}$ by $\mathcal{C}_{a,c}$. When convenient, we will also treat $\mathcal{C}_{a,c}$ as a set of pairs of k-tuples. The i^{th} elements of $\mathit{first}(\mathcal{C})$ and $\mathit{last}(\mathcal{C})$ will be denoted by $\mathit{first}_i(\mathcal{C})$ and $\mathit{last}_i(\mathcal{C})$ respectively.

Consider two semi-convex covers C, C' whose equivalence classes $\mathcal{C}$, $\mathcal{C}'$ belong to $\mathcal{C}_{a,b}$, $\mathcal{C}_{b+1,c}$ respectively. If $C \circ C'$ is semi-convex, the pair of k-tuples that identifies its equivalence class $\mathcal{C}^*$ is defined by:

$$\mathit{first}_i(\mathcal{C}^*) = \begin{cases} \mathit{first}_i(\mathcal{C}) & \text{if } \mathit{first}_i(\mathcal{C}) \neq \mathit{nil}; \\ \mathit{first}_i(\mathcal{C}') & \text{if } \mathit{first}_i(\mathcal{C}) = \mathit{nil}; \end{cases} \qquad \mathit{last}_i(\mathcal{C}^*) = \begin{cases} \mathit{last}_i(\mathcal{C}') & \text{if } \mathit{last}_i(\mathcal{C}') \neq \mathit{nil}; \\ \mathit{last}_i(\mathcal{C}) & \text{if } \mathit{last}_i(\mathcal{C}') = \mathit{nil}. \end{cases}$$

By analogy, we will call $\mathcal{C}^*$ the *composition* of $\mathcal{C}$ and $\mathcal{C}'$, and write $\mathcal{C}^* = \mathcal{C} \circ \mathcal{C}'$. Also, given two k-tuples $\boldsymbol{x}$ and $\boldsymbol{y}$ whose elements belong to $\mathcal{I}^* \cup \{\mathit{nil}\}$, we will write $\boldsymbol{x} \simeq \boldsymbol{y}$ when for each i in $\{1, \dots, k\}$, either the i^{th} element $\boldsymbol{x}_i$ of $\boldsymbol{x}$ is *nil*, or the i^{th} element $\boldsymbol{y}_i$ of $\boldsymbol{y}$ is *nil*, or $\mathit{ccw}(\boldsymbol{x}_i) \simeq \mathit{cw}(\boldsymbol{y}_i)$. The following result is an immediate corollary of Lemmas 3.2 and 3.3.

Lemma 3.4 *Let a, b and c be integers such that $1 \leq a \leq b < c \leq m$. An equivalence class $\mathcal{C}^*$ of subsets of $\{I_a, \ldots, I_c\}$ belongs to $\mathcal{C}_{a,c}$ if and only if there is an element $\mathcal{C}$ of $\mathcal{C}_{a,b}$, and an element $\mathcal{C}'$ of $\mathcal{C}_{b+1,c}$ such that $last(\mathcal{C}) \simeq first(\mathcal{C}')$ and $\mathcal{C}^* = \mathcal{C} \circ \mathcal{C}'$.*

For the remainder of Section 3.1, we will assume that $\mathcal{C}_{a,c}$ is stored in a data structure with the following fields:

- An ordered list $first_{a,c}$ of all k-tuples $\boldsymbol{x}$ for which there exists an element $\mathcal{C}$ of $\mathcal{C}_{a,c}$ such that $\boldsymbol{x} = first(\mathcal{C})$; we will denote the i^{th} element of $first_{a,c}$ by $first_{a,c}(i)$.
- An ordered list $last_{a,c}$ of all k-tuples $\boldsymbol{x}$ for which there exists an element $\mathcal{C}$ of $\mathcal{C}_{a,c}$ such that $\boldsymbol{x} = last(\mathcal{C})$; we will denote the i^{th} element of $last_{a,c}$ by $last_{a,c}(i)$.
- A boolean matrix $M_{a,c}$ of size $|first_{a,c}| \times |last_{a,c}|$ in which $M_{a,c}(i,j) = 1$ if and only if there exists an element $\mathcal{C}$ of $\mathcal{C}_{a,c}$ such that $first(\mathcal{C}) = first_{a,c}(i)$ and $last(\mathcal{C}) = last_{a,c}(j)$.

We conclude this subsection with a lemma that provides an upper bound on the sizes of $first_{a,c}$ and $last_{a,c}$. This bound will be used heavily in the analysis of the running time of the algorithms presented in the next two sections.

Lemma 3.5 *There are at most $O(k(c-a)^{k-1})$ k-tuples of elements of $\{I_a, \ldots, I_c, nil\}$ that contain either I_a or I_c at least once.*

3.1.4 Merging sets of equivalence classes using matrix multiplication

We now show how to compute $\mathcal{C}_{a,c}$ from $\mathcal{C}_{a,b}$ and $\mathcal{C}_{b+1,c}$ using matrix multiplication. We first describe how this is done in a simple case: that in which no pair of k-tuples corresponding to an element of $\mathcal{C}_{a,b}$ or $\mathcal{C}_{b+1,c}$ contains the element *nil*. We then explain how the general case is handled.

A simple case

Assume that no pair of k-tuples corresponding to an element of $\mathcal{C}_{a,b}$ or $\mathcal{C}_{b+1,c}$ contains the element *nil*. This implies that if $C \in \mathcal{C}_{a,b}$ and $C' \in \mathcal{C}_{b+1,c}$, then $\mathit{first}(C \circ C') = \mathit{first}(C)$ and $\mathit{last}(C \circ C') = \mathit{last}(C')$. Hence $\mathit{first}_{a,c} \subseteq \mathit{first}_{a,b}$ and $\mathit{last}_{a,c} \subseteq \mathit{last}_{b+1,c}$. Consider an element $\boldsymbol{x}$ of $\mathit{first}_{a,b}$, and an element $\boldsymbol{y}$ of $\mathit{last}_{b+1,c}$. We need to determine whether $(\boldsymbol{x}, \boldsymbol{y})$ belongs to $\mathcal{C}_{a,c}$. By Lemma 3.4, this is the same as deciding whether there are k-tuples $\boldsymbol{x}'$, $\boldsymbol{y}'$ such that

1. $(\boldsymbol{x}, \boldsymbol{x}')$ belong to $\mathcal{C}_{a,b}$;

2. $\boldsymbol{x}' \simeq \boldsymbol{y}'$;

3. $(\boldsymbol{y}', \boldsymbol{y})$ belong to $\mathcal{C}_{b+1,c}$.

The appearance of x, x', y' and y in this order suggests the use of matrix multiplication. Let M_b be the boolean matrix of size $|\mathit{last}_{a,b}| \times |\mathit{first}_{b+1,c}|$ in which $M_b(i,j) = 1$ if and only if $\mathit{last}_{a,b}(i) \simeq \mathit{first}_{b+1,c}(j)$. Finally let $M' = M_{a,b}M_b$, and $M = M'M_{b+1,c}$. We claim that $M(i,j) = 1$ if and only if there are k-tuples $\boldsymbol{x}'$, $\boldsymbol{y}'$ for which conditions 1 to 3 hold.

If $M(i,j) = 1$, then there is an integer t such that $M'(i,t) = M_{b+1,c}(t,j) = 1$. Moreover, since $M'(i,t) = 1$, there is an integer s such that $M_{a,b}(i,s) = M_b(s,t) = 1$. Hence the k-tuples $\boldsymbol{x}' = \mathit{last}_{a,b}(s)$ and $\boldsymbol{y}' = \mathit{first}_{b+1,c}(t)$ satisfy conditions 1 to 3. Conversely, if there are elements $\boldsymbol{x}'$ of $\mathit{last}_{a,b}$ and $\boldsymbol{y}'$ of $\mathit{first}_{b+1,c}$ that satisfy conditions 1 to 3, then $M(i,j) = 1$, thereby proving our claim. Consequently we can compute $\mathcal{C}_{a,c}$ by multiplying the three matrices $M_{a,b}$, M_b and $M_{b+1,c}$.

Note that, given integers i and j such that $M(i,j) = 1$, we can find t in time proportional to the size of $\mathit{first}_{b+1,c}$ (using brute force). Similarly, given i and t such that $M'(i,t) = 1$, we can determine s in time proportional to the size of $\mathit{last}_{a,b}$. Hence given $(\boldsymbol{x}, \boldsymbol{y})$, we can find elements $\boldsymbol{x}'$, $\boldsymbol{y}'$ of $\mathit{last}_{a,b}$, $\mathit{first}_{b+1,c}$ respectively for which conditions 1 to 3 hold in time $O(|\mathit{last}_{a,b}| + |\mathit{first}_{b+1,c}|)$.

The general case

The situation becomes more complicated when the pairs of k-tuples corresponding to the elements of $\mathcal{C}_{a,b}$ and $\mathcal{C}_{b+1,c}$ may contain the element *nil*. Several problems arise, and they must be resolved in order to compute $\mathcal{C}_{a,c}$ correctly. Consider an element $\mathcal{C}$ of $\mathcal{C}_{a,c}$.

Problem 1: The k-tuple $first(\mathcal{C})$ may not belong to $first_{a,b}$.

It follows from the definition of composition of equivalence classes that $first(\mathcal{C})$ can be obtained from an element $\boldsymbol{x}$ of $first_{a,b}$ by replacing each element of $\boldsymbol{x}$ equal to *nil* by an element of $\{I_{b+1}, \ldots, I_c, nil\}$ (we will call those elements *replaced* elements). Let $first'_{a,b}$ be the set of all k-tuples that can be obtained from an element of $first_{a,b}$ in this manner. We note that each element of $first'_{a,b}$ is obtained from exactly one element of $first_{a,b}$, and that $first(\mathcal{C})$ belongs to $first'_{a,b}$ for each element $\mathcal{C}$ of $\mathcal{C}_{a,c}$.

We thus need to replace $M_{a,b}$ by a matrix $M'_{a,b}$ whose rows are labeled with the elements of $first'_{a,b}$. For reasons that will become clear later in this section, the correct manner to label the columns of $M'_{a,b}$ is using the elements of the set $last'_{a,b}$ obtained from $last_{a,b}$ in the same way as $first'_{a,b}$ was obtained from $first_{a,b}$.

Problem 2: The k-tuple $last(\mathcal{C})$ may not belong to $last_{b+1,c}$.

The solution to problem 2 is a symmetric version of that used for problem 1. Let $last'_{b+1,c}$ be the set of all k-tuples that can be obtained from an element $\boldsymbol{y}$ of $last_{b+1,c}$ by replacing each element of $\boldsymbol{y}$ equal to *nil* by an element of $\{I_a, \ldots, I_b, nil\}$. Each element of $last'_{b+1,c}$ is obtained from exactly one element of $last_{b+1,c}$, and $last(\mathcal{C})$ belongs to $last'_{b+1,c}$ for each element $\mathcal{C}$ of $\mathcal{C}_{a,c}$.

We also replace $M_{b+1,c}$ by a matrix $M'_{b+1,c}$ whose columns are labeled with the elements of $last'_{b+1,c}$. The labels of the rows of $M_{b+1,c}$ will be the elements of the set $first'_{b+1,c}$ obtained from $first_{b+1,c}$ in the same way as $last'_{b+1,c}$ was obtained from $last_{b+1,c}$. We observe that in order to use matrix multiplication, we now need to replace M_b by a matrix M'_b whose rows and columns are labeled using the elements of $last'_{a,b}$ and $first'_{b+1,c}$ respectively.

Problem 3a: What meaning do we give to a '1' entry of $M'_{a,b}$, M'_b and $M'_{b+1,c}$ respectively?

We will come back to problem 3a after considering another problem: the *consistency* problem. The *type 1 consistency condition* can be described as follows: if a k-tuple $\boldsymbol{x}$ belongs to $first_{a,c}$, then *every element* of $\boldsymbol{x}$ that belongs to $\{I_{b+1}, \ldots, I_c, nil\}$ must come from $first(C')$ for *the same element* C' of $C_{b+1,c}$. The *type 2 consistency condition* can be defined symmetrically. The *consistency problem* is the following:

Problem 3b: How do we enforce both types of consistency condition?

We observe that the elements of a k-tuple for which the consistency problem occurs are precisely those that are *nil*, that is, those for which we do not need to check convex visibility. This suggests the possibility of using M'_b to enforce consistency.

There is one additional complication: the rows and the columns of M'_b are labeled using elements of $last'_{a,b}$ and $first'_{b+1,c}$ respectively. However consistency must be enforced between $first'_{a,b}$ and $first'_{b+1,c}$ on one hand, and between $last'_{a,b}$ and $last'_{b+1,c}$ on the other hand. We therefore need to use both $M'_{a,b}$ and M'_b to enforce the type 1 consistency condition, and both of M'_b and $M'_{b+1,c}$ have to cooperate to enforce the type 2 consistency condition.

These requirements lead us to the solution of problems 3a and 3b. We determine each entry of $M'_{a,b}$ as follows: $M'_{a,b}(i,j) = 1$ if two conditions hold:

- every replaced element of $first'_{a,b}(i)$ equals the corresponding replaced element of $last'_{a,b}(j)$;
- $M_{a,b}(i', j') = 1$ where $first_{a,b}(i')$ and $last_{a,b}(j')$ are the two k-tuples from which $first'_{a,b}(i)$ and $last'_{a,b}(j)$ were obtained respectively.

Entries of $M'_{b+1,c}$ are determined symmetrically. Finally, $M'_b(i,j) = 1$ if the following three conditions are satisfied:

- every replaced element of $last'_{a,b}(i)$ is equal to the corresponding element of $first'_{b+1,c}(j)$;

- every replaced element of $\mathit{first}'_{b+1,c}(j)$ is equal to the corresponding element of $\mathit{last}'_{a,b}(i)$;
- $M_b(i', j') = 1$ where $\mathit{last}_{a,b}(i')$ and $\mathit{first}_{b+1,c}(j')$ are the k-tuples from which we obtained $\mathit{last}'_{a,b}(i)$ and $\mathit{first}'_{b+1,c}(j)$ respectively.

An argument identical to that given for the simple case now shows that we can compute $\mathcal{C}_{a,c}$ by calculating $M = M'_{a,b}M'_bM'_{b+1,c}$. Let $\boldsymbol{x} = \mathit{first}'_{a,b}(i)$ and $\boldsymbol{y} = \mathit{last}'_{b+1,c}(j)$. The equivalence class that corresponds to a pair $(\boldsymbol{x}, \boldsymbol{y})$ of k-tuples belongs to $\mathcal{C}_{a,c}$ if and only if $M(i,j) = 1$. We observe that the elements C of $\mathcal{C}_{a,b}$ and C' of $\mathcal{C}_{b+1,c}$ satisfying the conditions of Lemma 3.4 can be found in time proportional to the sum of the sizes of $\mathit{last}'_{a,b}$ and $\mathit{first}'_{b+1,c}$, using the same procedure that was used in the simple case to retrieve $\boldsymbol{x}'$ and $\boldsymbol{y}'$.

3.1.5 The algorithm

We can finally describe the algorithm that finds a convex cover of $\{I_a, \ldots, I_c\}$. We first compute $\mathcal{C}_{a,c}$ as follows by divide and conquer on $\{I_a, \ldots, I_c\}$. Let $n_0 = c - a + 1$. When $n_0 = 1$ (that is $a = c$), a semi-convex cover of $\{I_a\}$ is a list whose elements are either $\emptyset$ or $\{I_a\}$, but not all of whose elements are $\emptyset$. No two of these covers are equivalent, and hence $\mathcal{C}_{a,c}$ contains $2^k - 1$ elements, and $M_{a,c}$ is a $2^k - 1$ by $2^k - 1$ matrix that can be computed in $\Theta(k\,4^k)$ time.

When $n_0 > 1$ (that is $a \neq c$), we first compute $\mathcal{C}_{a,b}$ and $\mathcal{C}_{b+1,c}$ for $b = \lfloor (a+c)/2 \rfloor$. Next we compute the set $\mathit{first}'_{a,b}$, as explained in Section 3.1.4. This can be done in time proportional to the size of $\mathit{first}'_{a,b}$, which is $O(kn_0^{k-1})$ by Lemma 3.5. We then compute the sets $\mathit{last}'_{a,b}$, $\mathit{first}'_{b+1,c}$ and $\mathit{last}'_{b+1,c}$ similarly. The value of each element of $M'_{a,b}$ can be determined in $O(k)$ time, and hence $M'_{a,b}$ can be computed in $O(k^3n_0^{2k-2})$ time. $M'_{b+1,c}$ can be computed in the same way. Suppose that we have determined whether x_i sees x_j convexly for each pair x_i, x_j of endpoints of elements of $\mathcal{I}^*$, which can be done in $O(|\mathcal{I}^*|^2)$ time [46,61]. Each element of M'_b can then also be computed in $O(k)$ time, and so M'_b can be determined in $O(k^3n_0^{2k-2})$ time.

Finally, we multiply $M'_{a,b}$, M'_b and $M'_{b+1,c}$, which requires $O(T_M(kn_0^{k-1}))$ time. The running time of our algorithm thus satisfies the following recurrence relation

$$T(n_0) = \begin{cases} k4^k & \text{if } n_0 = 1 \\ 2T(n_0/2) + O(k^3 n_0^{2k-2} + T_M(kn_0^{k-1})) & \text{if } n_0 \geq 2 \end{cases},$$

and so $T(n_0) \in O(k^3 n_0^{2k-2} + T_M(kn_0^{k-1}))$.

To find the set of all equivalence classes that contain convex covers of $\{I_a, \ldots, I_c\}$, it suffices to consider each element $\mathcal{C}$ of $\mathcal{C}_{a,c}$ in turn. $\mathcal{C}$ contains a convex cover if and only if, for each $i \in \{1, \ldots, k\}$, either $\mathit{first}_i(\mathcal{C}) = \mathit{nil}$ and $\mathit{last}_i(\mathcal{C}) = \mathit{nil}$, or $\mathit{last}_i(\mathcal{C}) \simeq \mathit{first}_i(\mathcal{C})$. Since every convex cover is semi-convex, this procedure is guaranteed to find an equivalence class containing a convex cover if one exists.

We complete this section by showing how to retrieve one of the convex covers that belong to an equivalence class $\mathcal{C}^*$ of $\mathcal{C}_{a,c}$. When $a = c$, the values of $\mathit{first}_i(\mathcal{C}^*)$ and $\mathit{last}_i(\mathcal{C}^*)$ either indicate that the i^{th} element of the cover is $\emptyset$, or that it is $\{I_a\}$. Hence a convex cover of $\{I_a\}$ can be retrieved from $\mathcal{C}^*$ in $O(k)$ time.

Suppose now that $a \neq c$. We showed in Section 3.1.4 how to compute two elements $\mathcal{C}$, $\mathcal{C}'$ that belong to $\mathcal{C}_{a,b}$, $\mathcal{C}_{b+1,c}$ respectively and satisfy the conditions of Lemma 3.4. Since every subset of a convex cover is convex, if $\mathcal{C}^*$ contains a convex cover then so do $\mathcal{C}$ and $\mathcal{C}'$. We thus find recursively two convex covers of $\{I_a, \ldots, I_b\}$, $\{I_{b+1}, \ldots, I_c\}$ that correspond to $\mathcal{C}$ and $\mathcal{C}'$ respectively. The composition of these two covers will be a convex cover of $\{I_a, \ldots, I_c\}$. Since the total size of the matrices generated by the algorithm is $O(k^2|\mathcal{I}^*|^{2k-2})$, we have obtained the following theorem:

Theorem 3.1 *We can determine whether P is B_k with respect to $\mathcal{I}^*$, and return a cover of $bd(P)$ by k convex subsets of $\mathcal{I}^*$ if one exists, in $O(k^3|\mathcal{I}^*|^{2k-2} + T_M(k|\mathcal{I}^*|^{k-1}))$ time using $O(k^2|\mathcal{I}^*|^{2k-2})$ space.*

3.2 Characterizing B_3 polygons

In this section, we define a type of cover of $bd(P)$ called a *potential* cover, and prove that if P is B_3, then there exists a potential cover of $bd(P)$. Potential covers are interesting because they only contain chords taken from a finite, easily computed set of chords of P. The endpoints of these chords belong to a subset X of $bd(P)$ whose cardinality is linear in the number of reflex vertices of P. We will show in Section 3.3.2 how to compute X in linear time. As we shall see in Section 3.2.2, it then follows from Theorem 3.1 that we can decide whether P is B_3 in polynomial time.

We start by describing X and defining the three types of chord that will be allowed in a potential cover of $bd(P)$. Next, we define a special kind of cover of $bd(P)$ that we call *constrained*; it follows immediately from this definition that every B_3 polygon admits at least one constrained cover. We then characterize constrained covers, and finally use this characterization to show how to transform each constrained cover into a potential cover. We shall assume without loss of generality that P has at least two reflex vertices (since otherwise P is trivially B_2).

3.2.1 Definitions

Let x and y be two distinct points of $bd(P)$. If $z \in bd(P)$, we will write $x \prec z \prec y$ to indicate that a counterclockwise traversal of $bd(P)$ starting at x would meet x, z and y in this order. We will denote by $chain[x, y]$ the set of all points z of $bd(P)$ such that $x \preceq z \preceq y$. Similarly, we will denote by $chain(x, y)$, $chain(x, y]$ and $chain[x, y)$ the sets of all points z of $bd(P)$ for which $x \prec z \prec y$, $x \prec z \preceq y$ and $x \preceq z \prec y$ respectively. If v_i and v_j are two reflex vertices of P, and no reflex vertex of P belongs to $chain(v_i, v_j)$, then we will call $chain[v_i, v_j]$ a *tip* of P.

The two kinds of point of $bd(P)$ that will be used as endpoints of chords of potential covers are defined using the following ray-shooting operation. Suppose that we shoot a ray ρ starting from a point y of P, and oriented in the direction from x to y for some point x of P. This ray will be absorbed as soon as it leaves P, or if it starts overlapping an edge of P (it can go through a vertex without being absorbed). We will denote the point at which ρ is absorbed by $hit(x, y)$. More formally, $hit(x, y)$ is the point z of ρ that maximizes the length of $\overline{yz}$, subject to the conditions that $\overline{yz} \subseteq P$ and $\overline{yz} \cap bd(P)$ is finite.

We call a point x of $bd(P)$ an *extension point* of P if there is a reflex vertex v_i of P and a vertex w of P adjacent to v_i such that $x = hit(w, v_i)$, as shown in

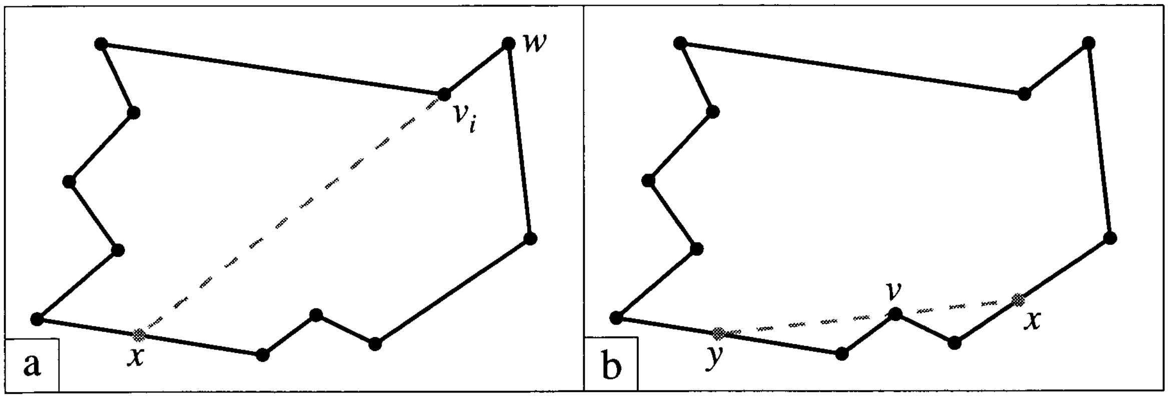


Figure 3.2: Illustrating extension points and bounce points of P.

Figure 3.2a. We will call x the *forward extension point of* v_i if $w = v_{i-1}$, and the *backward extension point of* v_i if $w = v_{i+1}$. A point x of $bd(P)$ is a *forward bounce point* of P (see Figure 3.2b) if $x = hit(y, v)$, where y and v satisfy the following three conditions:

1. y is a reflex vertex or an extension point of P;
2. v is a reflex vertex of P that belongs to $kr(P)$;
3. v is the only reflex vertex of P that belongs to $chain(y, x)$.

Backward bounce points are defined symmetrically; it suffices to replace $chain(y, x)$ by $chain(x, y)$ in condition 3. We will call *potential points* the reflex vertices of P, the extension points of P, and the bounce points of P. Every endpoint of a chord of a potential cover of $bd(P)$ will be a potential point. We observe that in O'Rourke's terminology (see Section 2.2), potential point have order 0, 1 or 2.

We call a line segment l *reflex* if both endpoints of l are reflex vertices of P. We say that l is *extensional* if it joins an extension point x of P to the reflex vertex v_i of P for which $x = hit(w, v_i)$. Finally, we call l *neighborly* if it is not reflex or extensional, and its endpoints belong to two adjacent tips of P. In Figure 1.8b, c_1 is a reflex chord, c_2 is an extensional chord, and c_3 is a neighborly chord. A line segment will be *potential* if it is reflex or extensional, or if it is neighborly and its endpoints are potential points. A cover of $bd(P)$ is *potential* if all of its chords are potential.

3.2.2 Constrained covers

Consider a set C of subpolygons of P, or equivalently a set C of subsets of $bd(P)$. Let X be the set that contains every reflex vertex and extension point of P, as well as every endpoint of a connected component of $\mathcal{I} \cap bd(P)$ for every $\mathcal{I}$ that belongs to C. The set X induces a partition $\mathcal{I}^P$ of $bd(P)$ into intervals: each interval joins two elements of X that are met consecutively in a counterclockwise traversal of $bd(P)$. We observe that every element of C is $\mathcal{I}^P$-elementary. Hence for the remainder of Section 3.2 we will implicitly consider elements of a collection of convex subpolygons of P or subsets of $bd(P)$ as subsets of the corresponding partition of $bd(P)$.

To be able to characterize a type of cover of $bd(P)$, it is essential to restrict the subsets of $bd(P)$ used in a cover of this type. We will thus only consider covers by *maximal* subsets of $bd(P)$. Consider a subset $\mathcal{I}$ of $bd(P)$, and the partition $\mathcal{I}^P$ of $bd(P)$ induced by $\{\mathcal{I}\}$. We will call $\mathcal{I}$ *maximal* if

1. no $\mathcal{I}^P$-elementary superset of $\mathcal{I}$ has fewer chords;

2. no $\mathcal{I}^P$-elementary superset of $\mathcal{I}$ with the same number of chords has more extensional chords;

3. no $\mathcal{I}^P$-elementary superset of $\mathcal{I}$ with the same number of chords covers more reflex vertices of P.

Intuitively, a maximal subset of $bd(P)$ is one that cannot be extended to cover more of $bd(P)$ without using more chords. We observe that this does not mean that the covering polygon for this subset is maximal in the usual sense of the term [82]. Moreover condition 1 implies that the intersection of a maximal subset of $bd(P)$ with any open tip $chain(v_i, v_j)$ of P is connected. Finally every convex subset of $bd(P)$ is contained in a maximal subset of $bd(P)$.

The subset $\mathcal{I}$ of $bd(P)$ drawn with gray edges in Figure 3.3a is not maximal because the convex superset $\mathcal{I}'$ of $\mathcal{I}$ shown in Figure 3.3b has fewer chords. The subset $\mathcal{I}$ of $bd(P)$ shown in Figure 3.3c is not maximal either because the convex superset $\mathcal{I}'$ of $\mathcal{I}$ shown in Figure 3.3d contains one more extensional chord. Finally the subset $\mathcal{I}$ of

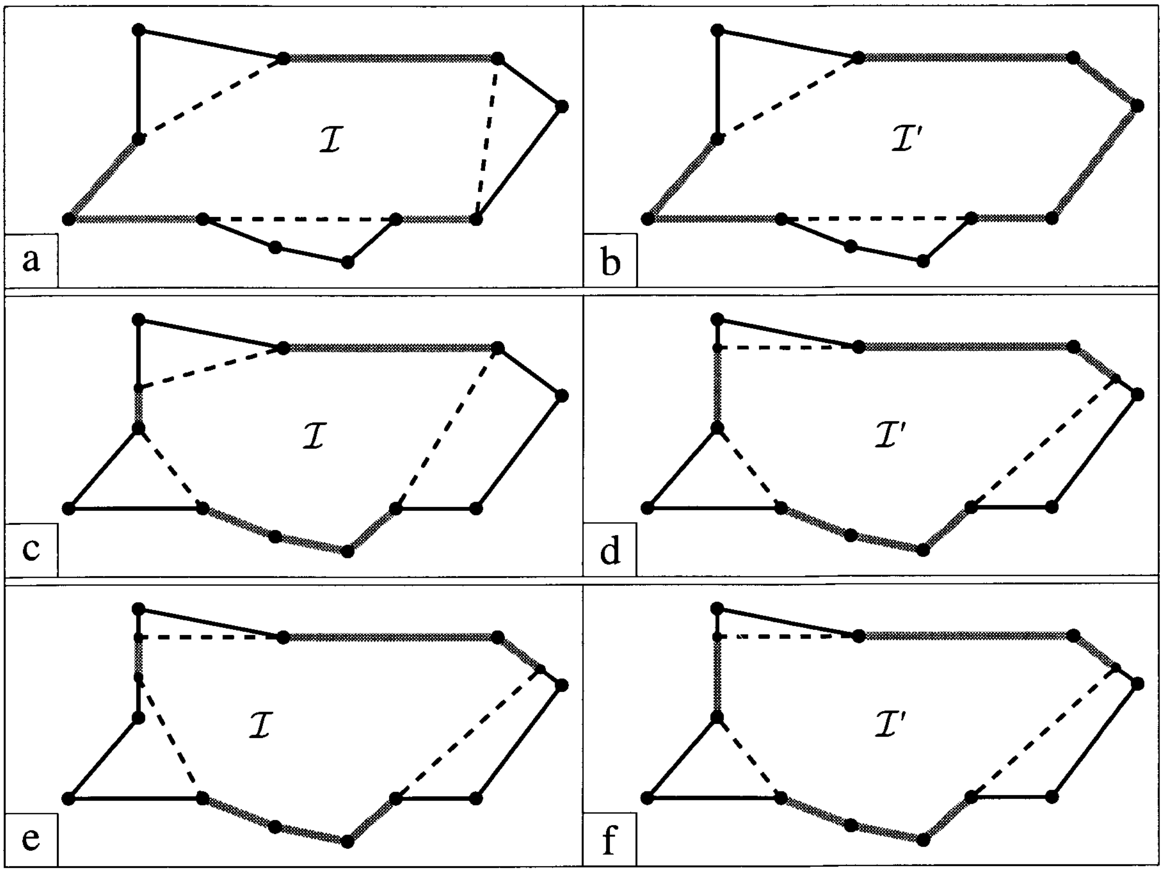


Figure 3.3: The gray subsets of $bd(P)$ shown on the left are not maximal; the corresponding maximal supersets are shown on the right.

$bd(P)$ shown in Figure 3.3e is not maximal because the convex superset $\mathcal{I}'$ of $\mathcal{I}$ shown in Figure 3.3f covers one more reflex vertex of P.

Let C be a cover of $bd(P)$. The cardinality of the corresponding partition of $bd(P)$ will be called the *weight* of C. We will call C *constrained* if it contains three maximal elements and has minimum weight over all covers of $bd(P)$ by three maximal subsets of $bd(P)$. We observe that if $\mathcal{I}$ and $\mathcal{I}'$ are two elements of a constrained cover, then no connected subset of $\mathcal{I} \cap bd(P)$ is contained in a connected subset of $\mathcal{I}' \cap bd(P)$.

Lemma 3.6 *Let $\mathcal{I}$ belong to a constrained cover C of $bd(P)$. If $c = \overline{xy}$ is a chord of $\mathcal{I}$ that is neither a reflex chord nor an extensional chord, then there is a unique reflex vertex v_i of P such that $x \prec v_i \prec y$, and moreover $v_i \in kr(P)$.*

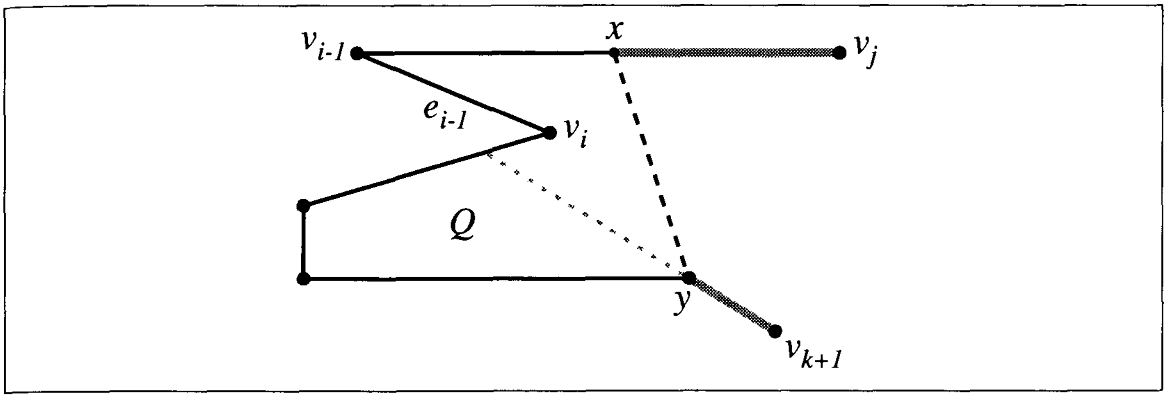


Figure 3.4: Illustrating the proof of Lemma 3.6.

Proof: By symmetry we assume that x is not a reflex vertex of P. Suppose that x and y belong to $e_j \backslash \{v_j\}$ and $e_k \backslash \{v_{k+1}\}$ respectively. Since $\mathcal{I}$ is maximal (condition 1), at least one reflex vertex of P belongs to $chain[x, y]$. Let v_i be the reflex vertex closest to x on $chain[x, y]$, as shown in Figure 3.4. Since $\mathcal{I}$ is maximal (condition 3), $chain[x, v_i] \cup \mathcal{I}$ is not convex, and hence v_i does not see y convexly.

However, the subpolygon Q of P determined by $chain[x, y]$ and $\overline{yx}$ is B_2, which implies that v_i sees every point of $chain[x, y]$, including x and y [92]. Moreover the angle $\angle v_i y v_{k+1}$ is convex, since if y is reflex then $hit(v_{k+1}, y)$ does not belong to $chain[x, v_i]$ ($\mathcal{I}$ is maximal, condition 2). Hence $\angle v_{i-1} v_i y$ is reflex, and so v_i is the only reflex vertex of $chain(x, y)$ seen by the midpoint of e_{i-1}. But since Q is B_2, this implies that no other reflex vertex of P belongs to $chain(x, y)$.

Finally, v_i sees every point of $\mathcal{I}$ since $\angle v_j x v_i$ and $\angle v_i y v_{k+1}$ are convex, and moreover it belongs to the other two elements of C. Hence v_i sees every point of $bd(P)$, and so $v_i \in kr(P)$. □

We note that Lemma 3.6 implies that every chord of a maximal subset of $bd(P)$ that is not a reflex chord or an extensional chord is a neighborly chord. It only remains to show how to restrict the endpoints of neighborly chords to potential points of P. We first characterize the way in which constrained covers of $bd(P)$ cover an open tip of P that contains an endpoint of a neighborly chord. This is the situation illustrated in Figure 3.5d.

Lemma 3.7 *Let $C = (\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3)$ be a constrained cover of $bd(P)$, and $chain[v_i, v_j]$ be a tip of P. If a chord c of C has an endpoint x in $chain(v_i, v_j)$ that is not an extension point, then for a suitable permutation of C,*

- $chain[v_{i-1}, v_{j+1}] \cap \mathcal{I}_1 = chain[v_i, x]$;
- $chain[v_{i-1}, v_{j+1}] \cap \mathcal{I}_2 = chain[x, v_j]$;
- $chain[v_{i-1}, v_{j+1}] \cap \mathcal{I}_3 = e_{i-1} \cup e_j$.

Proof: We can assume by symmetry that $c = \overline{xy}$. Moreover we can assume that c is a chord of $\mathcal{I}_1$, and that $\mathcal{I}_2$ contains the subset $chain[x, z]$ of $bd(P)$ if z is close enough to x. Since $\mathcal{I}_1$ is maximal, it cannot cover $chain[v_j, y]$, and hence $chain[x, v_j]$ does not intersect the interior halfplane of e_j. Since $x \in \mathcal{I}_2$, this implies that $\mathcal{I}_2$ does not cover any point of e_j other than v_j. Hence $\mathcal{I}_3$ covers e_j, and so $\mathcal{I}_2$ covers $chain[x, v_j]$. This is illustrated in Figure 3.5a.

Let $I_1 = chain(v_i, v_j) \cap \mathcal{I}_1$ and $I_2 = chain(v_i, v_j) \cap \mathcal{I}_2$. Since $\mathcal{I}_1$ and $\mathcal{I}_2$ are maximal (condition 1), I_1 and I_2 are connected. Let z_1 and z_2 be the clockwise endpoints of I_1 and I_2 respectively. Since C has minimum weight, I_2 does not contain I_1, and hence $v_i \preceq z_1 \prec z_2$. We now prove that $z_2 = x$ by showing that $z_2 \neq x$ leads to a contradiction. Two cases need to be considered. Let $w = hit(v_{j+1}, v_j)$.

> Case 1 $(w \prec z_2)$: Let $\mathcal{I} = (\mathcal{I}_1 \setminus chain[z_2, x])$. If we substitute $\mathcal{I}$ for $\mathcal{I}_1$ in C, then we obtain a cover of $bd(P)$ by maximal subsets of $bd(P)$, with a weight smaller than that of C (illustrated in Figure 3.5b). This cannot happen since C is constrained.
>
> Case 2 $(z_2 \preceq w)$: Let $\mathcal{I} = (\mathcal{I}_1 \setminus chain[w, x]) \cup chain[v_j, y]$. If we substitute $\mathcal{I}$ for $\mathcal{I}_1$ in C, then we obtain a cover of $bd(P)$ by maximal subsets of $bd(P)$, with a weight smaller than that of C (this cover is shown in Figure 3.5c). This is impossible since C is constrained.

Hence $z_2 = x$. Since x is not an extension point, an argument symmetric to that presented in the first paragraph of this proof completes the proof of the lemma. The manner in which C covers $chain[v_{i-1}, v_{j+1}]$ is shown in Figure 3.5d. □

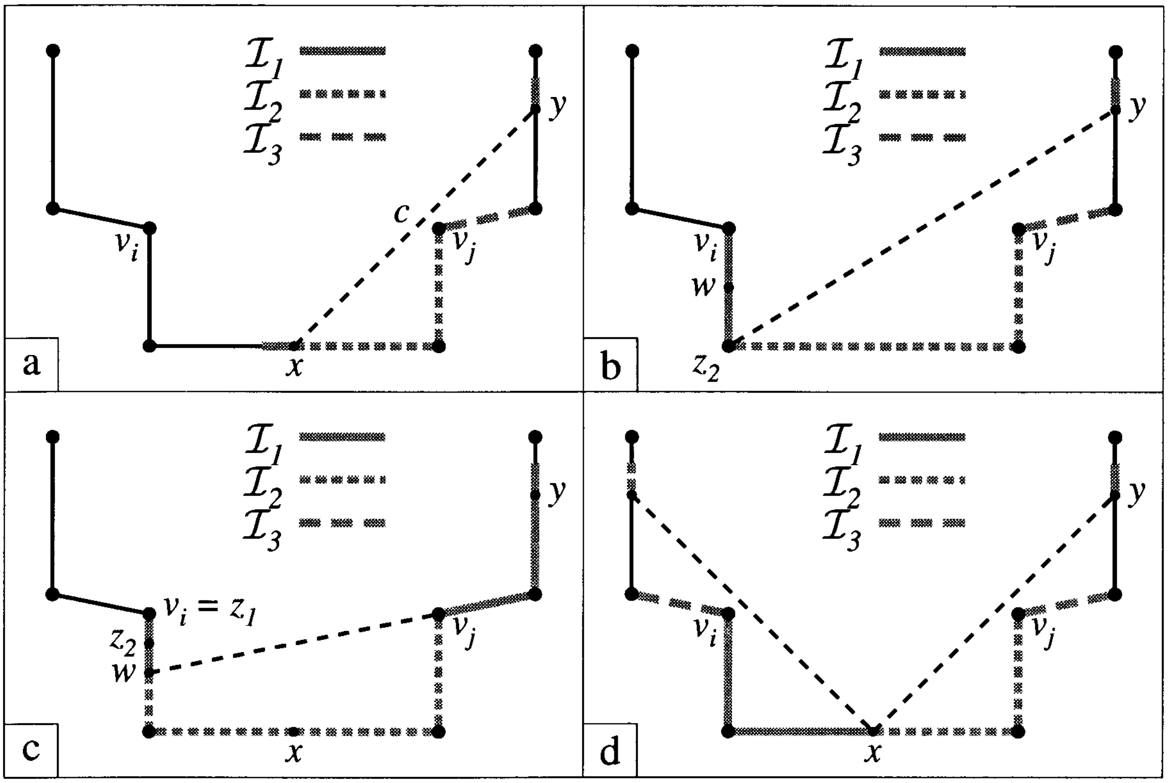


Figure 3.5: Illustrating the proof of Lemma 3.7.

Lemma 3.7 allows us to prove that, in a constrained cover of $bd(P)$, there does not exist a long sequence of chords "bouncing" from one tip of P to the next one.

Lemma 3.8 *If $C = (\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3)$ is a constrained cover of $bd(P)$, then at most two consecutive open tips of P contain an endpoint of a neighborly chord of C that is not an extension point.*

Proof: We prove this by contradiction. Suppose that three consecutive open tips $chain(v_{i_1}, v_{i_2})$, $chain(v_{i_2}, v_{i_3})$ and $chain(v_{i_3}, v_{i_4})$ of P contain such points x_1, x_2 and x_3 respectively. It follows from Lemma 3.7 that, for a suitable permutation of C:

- $\mathcal{I}_1$ covers e_{i_1-1}, $chain[v_{i_2}, x_2]$ and $chain[x_3, v_{i_4}]$;
- $\mathcal{I}_2$ covers $chain[v_{i_1}, x_1]$, $chain[x_2, v_{i_3}]$ and e_{i_4+1};
- $\mathcal{I}_3$ covers $chain[x_1, v_{i_2}]$ and $chain[v_{i_3}, x_3]$.

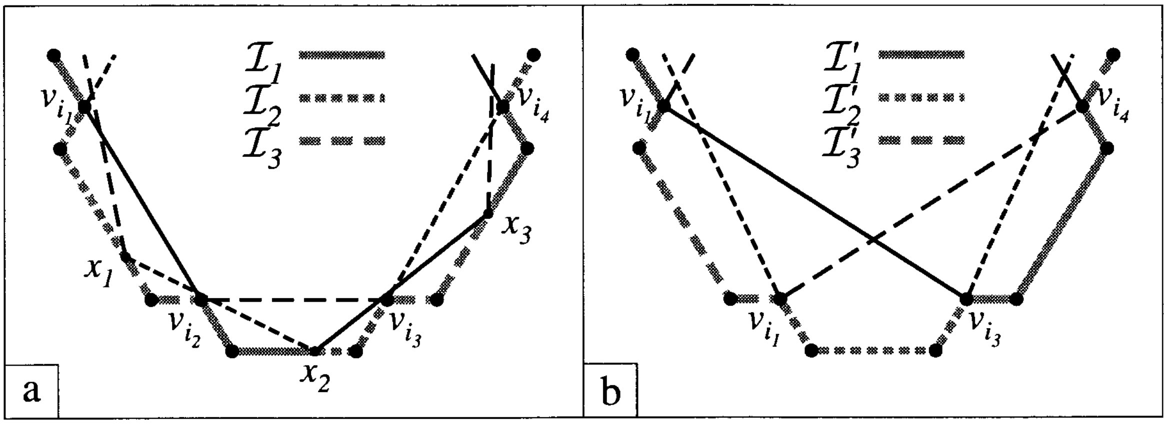


Figure 3.6: Illustrating the proof of Lemma 3.8

as illustrated in Figure 3.6a. Let $\mathcal{I}'_1 = (\mathcal{I}_1 \cap chain[v_{i_4}, v_{i_1}]) \cup chain[v_{i_3}, v_{i_4}]$, let $\mathcal{I}'_2 = (\mathcal{I}_2 \cap chain[v_{i_4}, v_{i_1}]) \cup chain[v_{i_2}, v_{i_3}]$, and let $\mathcal{I}'_3 = (\mathcal{I}_3 \cap chain[v_{i_4}, v_{i_1}]) \cup chain[v_{i_1}, v_{i_2}]$, as shown in Figure 3.6b. The list $(\mathcal{I}'_1, \mathcal{I}'_2, \mathcal{I}'_3)$ is a cover of $bd(P)$ by three maximal subsets of $bd(P)$, and has weight smaller than that of C, which is a contradiction. □

3.2.3 Potential covers

We can at last show how to construct a potential cover from a constrained cover $C = (\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3)$ of $bd(P)$. This construction considers each tip of P, in an arbitrary order, and modifies C to ensure that every endpoint of a neighborly chord contained in this tip is a potential point. Since (as we will see) the changes made to C are independent from each other, it suffices to show how this is done for the tip $chain[v_i, v_j]$. We assume that it contains an endpoint x of a chord of C that is not a potential point, as otherwise nothing needs to be done.

Since C is constrained and the changes made to one tip only affect the endpoints of chords of C contained in that tip, Lemma 3.7 implies that, for a suitable permutation of C,

- $chain[v_{i-1}, v_{j+1}] \cap \mathcal{I}_1 = chain[v_i, x]$;
- $chain[v_{i-1}, v_{j+1}] \cap \mathcal{I}_2 = chain[x, v_j]$;

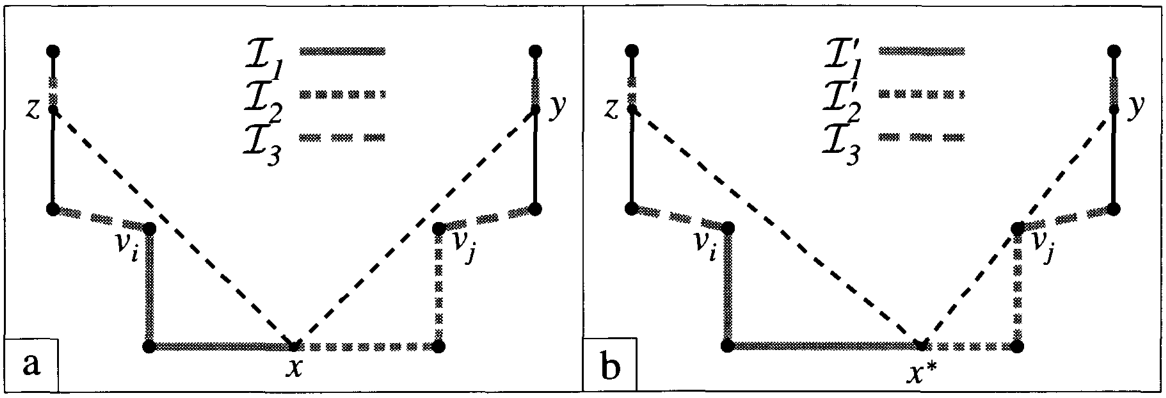


Figure 3.7: Transforming a constrained cover into C^*.

- $chain[v_{i-1}, v_{j+1}] \cap \mathcal{I}_3 = e_{i-1} \cup e_j$.

Let y be the counterclockwise endpoint of the chord of $\mathcal{I}_1$ counterclockwise from x, and z be the clockwise endpoint of the chord of $\mathcal{I}_2$ clockwise from x (as shown in Figure 3.7a). Since C is constrained, it follows from Lemma 3.8 that at least one of y and z is an extension point or a reflex vertex of P. By symmetry, we can assume that it is y.

Since $\mathcal{I}_1$ is maximal (condition 2), the point $hit(v_{j-1}, v_j)$ does not belong to $chain[v_j, y]$, and hence the point $w = hit(y, v_j)$ belongs to $chain(x, v_j)$. Let x^* be the first potential point met during a counterclockwise traversal of $bd(P)$ starting at x. We observe that $x \prec x^* \preceq w$. Finally let $\mathcal{I}'_1 = \mathcal{I}_1 \cup chain[x, x^*]$, and let $\mathcal{I}'_2 = \mathcal{I}_2 \setminus chain[x, x^*]$, as illustrated in Figure 3.7b.

Since $x \simeq y$ and $x^* \in chain[x, w]$, it follows that $x^* \simeq y$, and so $\mathcal{I}'_1$ is convex. Moreover, $x^* \in \mathcal{I}_2$, and hence $z \simeq x^*$, which implies that $\mathcal{I}'_2$ is also convex. We can thus modify C by replacing $\mathcal{I}_1$ by $\mathcal{I}'_1$ and $\mathcal{I}_2$ by $\mathcal{I}'_2$. The only subsets of $\mathcal{I}^P$ affected are those contained in $chain[v_i, v_j]$. Hence, by repeating this operation on each tip of P that contains an endpoint of a neighborly chord of C that is not a potential point, we obtain the following result.

Theorem 3.2 *If P is B_3, then there exists a potential cover of $bd(P)$.*

Let X be the set of potential points of P, and $\mathcal{I}^*$ be the partition of $bd(P)$ induced by X. We can easily compute each element of X in linear time (we will show in the next section how to compute all of them in that amount of time). We can then apply the algorithm of Section 3.1 to determine whether P is $B_3(\mathcal{I}^*)$ in $O(n^{4.752})$ time. Since every potential cover of $bd(P)$ is $\mathcal{I}^*$-elementary, this implies that we can decide whether P is B_3 in $O(n^{4.752})$ time.

3.3 Recognizing B_3 polygons efficiently

In this section, we show how to recognize B_3 polygons efficiently. This new approach is motivated by the following observation: the $O(n^{4.752})$ time algorithm only uses our characterization of the endpoints of the set of chords of a potential cover. It does not take advantage of the fact that only some of the chords that join these points are allowed. We thus modify the algorithm of Section 3.1 by discarding as many elements of $first_{a,c}$ and $last_{a,c}$ as possible. Recall that $first_{a,c}$ is a list that contains for each cover C of $\{I_a, \ldots, I_c\}$ the 3-tuple constructed from the clockwisemost intervals of each element of C, and that $last_{a,c}$ is defined similarly from the counterclockwisemost intervals of each element of C. While discarding these elements, we need to be careful to ensure that a potential cover of $bd(P)$ will be found if one exists. This technique reduces the running time of the algorithm to $O(n)$.

We start by considering some visibility properties specific to B_3 polygons. We then show how to compute all potential points of P in linear time. Next, we explain how to partition $first_{a,c}$ and $last_{a,c}$ into equivalence classes. The algorithm will only retain one element of $first_{a,c}$ and $last_{a,c}$ per equivalence class. We then prove that the number of equivalence classes thus defined is $O(1)$, and show how to compute them. Finally, we give the algorithm and prove that it runs in linear time. Throughout this section, $\mathcal{I}^*$ will refer to the partition of $bd(P)$ induced by the set of potential points of P.

3.3.1 Visibility in B_3 polygons

The algorithm that will be presented in Section 3.3.4 relies on two properties of B_3 polygons. The first property concerns their reflex vertices.

Lemma 3.9 *If P is B_3, then every two reflex vertices of P are visible.*

Proof: We prove this by contradiction. If there are two reflex vertices v_i, v_j of P that do not see each other, then there are small circles b_i, b_j of radius ε_i, ε_j centered at v_i, v_j respectively such that no point of b_i is visible from a point of b_j. Let x_{i-1} be a point of e_{i-1} whose distance from v_i is strictly positive without exceeding $\varepsilon_i/2$, and let x_i, x_{j-1} and x_j be chosen similarly on e_i, e_{j-1} and e_j respectively. No two points among x_{i-1}, x_i, x_{j-1}, and x_j see each other, and so P is not B_3. □

Before proving the second property, we require one additional fact. This fact holds for all simple polygons, even those that are not B_3.

Lemma 3.10 *Let e_i and e_j be two distinct edges of a simple polygon P, and w_i, w_j be the forward extension points of v_{i+1} and v_{j+1} respectively. If the midpoint m_i of e_i and the midpoint m_j of e_j see each other, then $m_i \prec w_i \preceq m_j \prec w_j$.*

Proof: Since no point of $chain(v_{i+1}, w_i)$ sees m_i except perhaps for reflex vertices of P that belong to $\overline{v_{i+1}w_i}$, we have $m_i \prec w_i \preceq m_j$. A symmetric argument shows that $m_j \prec w_j \preceq m_i$, which implies that $m_i \prec w_i \preceq m_j \prec w_j$, as required. □

Similarly, if w_i and w_j are the backward extension points of v_i and v_j respectively, and if the midpoint m_i of e_i sees the midpoint m_j of e_j, then $m_i \prec w_j \preceq m_j \prec w_i$. We now prove the second property required by our algorithm. It is used to derive bounds on the number of potential points of P that need to be considered.

Lemma 3.11 *If u_i and u_j are reflex vertices of P, and there are k distinct points $w_1, \ldots, w_k$ of $chain(u_i, u_j]$ that are the forward extension points of vertices $u_{i_1}, \ldots, u_{i_k}$ of $chain(u_j, u_i]$ respectively, then P is not B_k.*

Proof: Let m_j be the midpoint of e_{i_j} for each j in $\{1, \ldots, k\}$, and let m be the midpoint of $\bigcap_{j=1}^{k} chain(u_i, w_j]$. Since m is not a reflex vertex of P, and m belongs to $chain(u_{i_j+1}, w_j)$, the points m and m_j are not visible. Moreover m_x does not see m_y for $x \neq y$ because either $m_x \prec m_y \prec w_x$, or $m_y \prec m_x \prec w_y$. Hence no two among the $k+1$ points $m, m_1, \ldots, m_k$ see each other, and therefore P is not B_k. □

We note that the symmetric fact holds for backward extension points and is proved similarly.

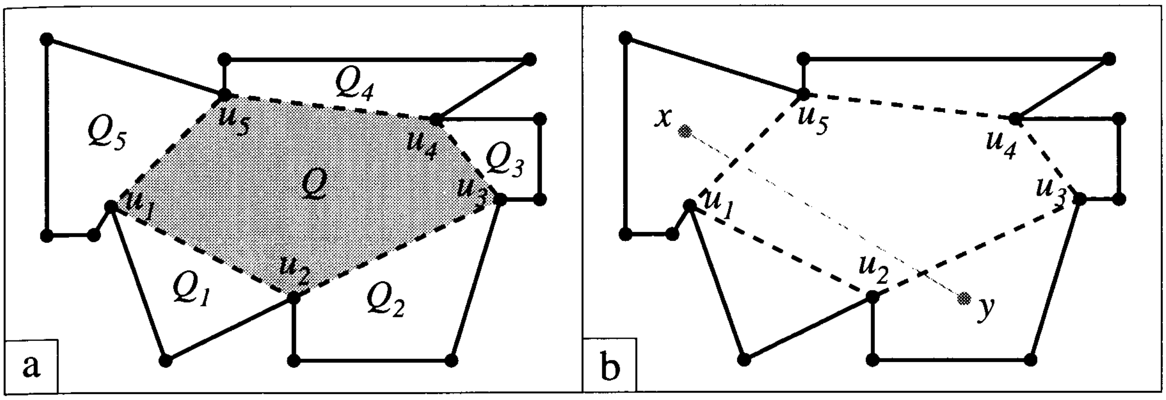


Figure 3.8: A partition of P into $Q, Q_1, Q_2, \ldots, Q_5$

3.3.2 Computing all potential points

Let r denote the number of reflex vertices of P, and let u_i be the i^{th} reflex vertex of P met in a counterclockwise traversal of $bd(P)$ starting at v_1. Consider the polygon Q whose vertices are $u_1, \ldots, u_r$. We can obtain Q in time linear in the number of vertices of P, and verify whether Q is convex in $O(r)$ time. If Q is not convex, then we can abort and report that P is not B_3 by Lemma 3.9. Hence suppose that Q is convex.

The edges of Q induce a partition of P into at most $r+1$ connected components, as shown in Figure 3.8a. One of the components is Q, and every other one is bounded by an edge $\overline{u_{i+1}u_i}$ of Q and by $chain[u_i, u_{i+1}]$. We will denote that component by Q_i; we observe that Q_i is not necessarily convex. The main property of this partition is illustrated in Figure 3.8b, and stated in the following lemma, whose simple proof will be omitted.

Lemma 3.12 *A point x of Q_i sees a point y of Q_j if and only if $\overline{xy}$ intersects both $\overline{u_iu_{i+1}}$ and $\overline{u_ju_{j+1}}$, x sees $\overline{xy} \cap \overline{u_iu_{i+1}}$ and y sees $\overline{xy} \cap \overline{u_ju_{j+1}}$.*

Lemma 3.12 allows us to use this partition of P to compute the potential points of P in $O(1)$ amortized time per point. Forward and backward extension points of P are computed separately. By symmetry, we only show how to find all forward extension points of P. Let w_i be the forward extension point of u_i for each vertex u_i of Q. We first show how to compute $Q \cap \overline{u_iw_i}$.

The basic idea is the following: we traverse the boundary of Q counterclockwise, and shoot a ray along each extensional chord as we reach its origin. When we visit an edge of Q, we check the rays that have been shot so far to see if any of them hit that edge. Let us call a ray *dangling* if it has been shot, but we have not visited the edge it hits yet. Since we can only afford to spend $O(r)$ times checking rays, we cannot verify every ray at every edge we visit. We thus use Lemma 3.11, which ensures that if P is B_3, then we never have more than two dangling rays. Hence we only check two rays per edge (and possibly fail if P is not B_3).

Let us now describe the process formally. We divide the procedure in two stages: first we shoot the rays, and then we find where they hit $bd(Q)$.

> Stage 1 : We visit $u_1, \ldots, u_r$ in this order. If $\overline{u_i w_i}$ is tangent to Q at u_i, then $Q \cap \overline{u_i w_i}$ is one of u_i, $\overline{u_{i-1} u_i}$, $\overline{u_i u_{i+1}}$. We can determine which one in constant time. If $\overline{u_i w_i}$ is not tangent to Q at u_i, then we add u_i at the end of an initially empty list $\mathcal{L}$.

> Stage 2 : We repeatedly visit the edges of Q in counterclockwise order, starting with the edge $\overline{u_1 u_2}$. We visit an edge e by examining the first two elements of $\mathcal{L}$. To examine u_i, we determine whether the line through u_i and w_i intersects e in a point other than u_i. If so, we have found $Q \cap \overline{u_i w_i}$, and we remove u_i from $\mathcal{L}$.

Stage 2 ends once every edge has been visited twice. If $\mathcal{L}$ is not empty at the end of stage 2, then we report that P is not B_3. Each stage can clearly be carried out in $O(r)$ time. We now have to prove that $Q \cap \overline{u_i w_i}$ is computed correctly for each u_i. It suffices to show that if $\mathcal{L}$ is not empty at the end of stage 2, then P is not B_3.

If $\mathcal{L}$ is not empty at the end of stage 2, then there must have been a first time in stage 2 at which we *missed* an intersection. That is, we had already visited $\overline{u_i u_{i+1}}$, we were visiting the edge $\overline{u_j u_{j+1}}$ of Q containing the endpoint of $Q \cap \overline{u_i w_i}$ distinct from u_i, but u_i was neither the first nor the second element of $\mathcal{L}$. Suppose that u_x and u_y were the first two elements of $\mathcal{L}$ when this event occurred. By construction, $x < y < i$. Moreover, since this was the first time we missed an intersection, w_x

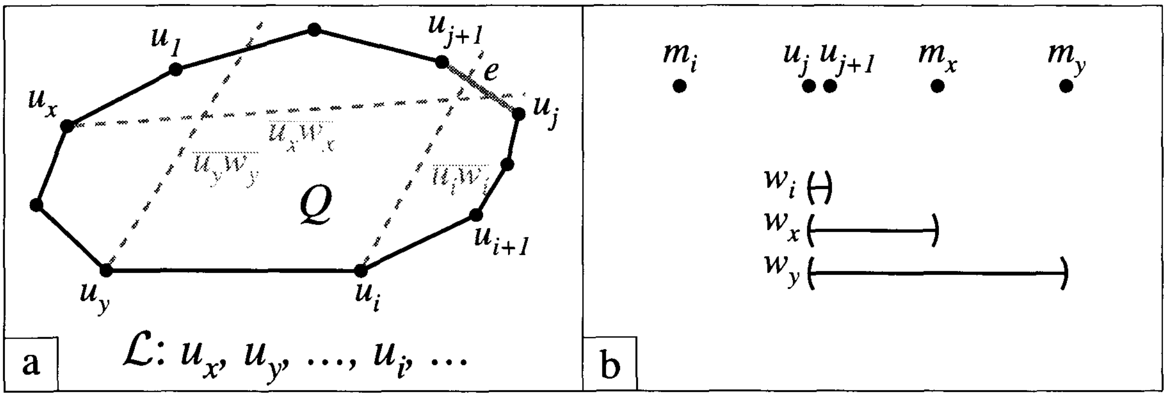


Figure 3.9: A situation in which stage 2 fails.

belongs to $chain(u_j, u_x]$, and w_y belongs to $chain(u_j, u_y]$. This situation is illustrated in Figure 3.9a.

Let m_x, m_y and m_i be the midpoints of the edges of P clockwise from u_x, u_y and u_i respectively. The possible orderings of the six points m_x, m_y, m_i, w_x, w_y and w_i around P, based on the constraints on the location of w_x and w_y, are illustrated in Figure 3.9b. In all cases, we do not have $m_x \prec w_x \prec m_y \prec w_y$, $m_x \prec w_x \prec m_i \prec w_i$, or $m_y \prec w_y \prec m_i \prec w_i$. Hence it follows from Lemma 3.10 that no two among m_x, m_y and m_i see each other. Let m be any point of $bd(P)$ that belongs to $chain(u_j, z)$ for each z in $\{w_i, w_x, w_y\}$. The point m does not belong to the interior halfplanes of the edges of P that contain m_x, m_y and m_i, and hence does not see any of m_x, m_y and m_i. Therefore P is not B_3.

Once we have determined $Q \cap \overline{u_i w_i}$, we know which of $Q_1, \ldots, Q_r$ the point w_i belongs to. We can then find w_i in time proportional to the size of that subpolygon of P. We note that Lemma 3.11 implies that no tip of P will be searched more than twice. If this occurs, we can stop and report that P is not B_3. We have thus obtained the following result.

Lemma 3.13 *In $O(n)$ time, we can either compute all forward extension points of P or report that P is not B_3.*

We can compute similarly the set of all backward extension points of P. It now remains to show how to compute the bounce points of P. We will only consider forward bounce points, since backward ones can be found using a symmetric procedure. We first compute the kernel of P. We then consider each point x that is either a reflex vertex of P or an extension point. Suppose that x belongs to $chain[u_i, u_{i+1})$. We first check that u_{i+1} belongs to $kr(P)$ and that $\overline{xu_{i+1}}$ is tangent to Q at u_{i+1}. If both conditions hold, then we compute $y = hit(x, u_{i+1})$ by searching Q_{i+1}. If P is B_3, then each tip of P contains at most 5 forward bounce points, as we shall show in Lemma 3.16, and so we need to search each tip at most 10 times. We have thus proved the main result of this section.

Lemma 3.14 *Let P be a simple polygon. In $O(n)$ time, we can either compute all potential points of P or determine that P is not B_3.*

3.3.3 Equivalence classes of $first_{a,c}$ and $last_{a,c}$

We now show how to partition $first_{a,c}$ and $last_{a,c}$ into equivalence classes. Recall that the algorithm will keep only one representative from each class, and that the fact that there are only a constant number of classes is what allows us to achieve the $O(n)$ running time. We only explain how $first_{a,c}$ is partitioned, since $last_{a,c}$ is handled symmetrically. The partition of $first_{a,c}$ is derived from a partition $first^*_{a,c}$ of $\{I_a, \ldots, I_c\}$ in the natural way: two elements $\boldsymbol{x}$, $\boldsymbol{y}$ of $first_{a,c}$ are equivalent if the i^{th} elements of $\boldsymbol{x}$ and $\boldsymbol{y}$ are equivalent under $first^*_{a,c}$ for $i = 1, 2, 3$.

Hence it suffices to explain how $first^*_{a,c}$ is constructed. Consider all queries of the form "does z see x convexly?" that may be asked by the algorithm while computing sets of covers of a superset of $\{I_a, \ldots, I_c\}$. We want to put two elements I_i, I_j of $\{I_a, \ldots, I_c\}$ in the same class if the answer to every query of the form "does z see $cw(I_i)$ convexly?" is the same as the answer to the query "does z see $cw(I_j)$ convexly?" We shall also discard every element of $\{I_a, \ldots, I_c\}$ that we know will not be queried about in the process of constructing a potential cover.

Throughout Section 3.3.3, we shall assume that $cw(I_a)$ is a reflex vertex u_a of P, and that $ccw(I_c)$ is a reflex vertex u_c of P.

The partition $\mathit{first}^*_{a,c}$ of $\{I_a, \ldots, I_c\}$

The partition of $\{I_a, \ldots, I_c\}$ into equivalence classes is based on sets $back(I_i)$ defined as follows: $back(I_i)$ is the set of all elements I of $\mathcal{I}^* \setminus \{I_a, \ldots I_c\}$ such that $\overline{ccw(I)cw(I_i)}$ is potential and $ccw(I) \simeq cw(I_i)$. If $back(I_i) = back(I_j)$, then $ccw(I) \simeq cw(I_i)$ if and only if $ccw(I) \simeq cw(I_j)$ for every element I of $\mathcal{I}^* \setminus \{I_a, \ldots I_c\}$. We thus want to say that I_i and I_j are equivalent if $back(I_i) = back(I_j)$. However, we do not know how to compute $back(I_i)$ for all elements of $\{I_a, \ldots, I_c\}$ in linear time, since each of these sets may contain $O(n)$ elements.

We thus divide the elements of $\{I_a, \ldots, I_c\}$ into three groups, and treat each group separately. We will get an equivalence relation that is similar to the one based on the values of $back(I_i)$, but some classes of the latter (the one we wanted) may be split into several classes of the former (the one that we will use). The first group contains the elements of $\{I_a, \ldots, I_c\}$ that may be used in a neighborly or extensional chord. The second group contains every element of $\{I_a, \ldots, I_c\}$ that does not belong to the first group, but whose clockwise endpoint is a reflex vertex of P. The third group contains every remaining element of $\{I_a, \ldots, I_c\}$.

Group 1 : The first group, denoted by $\mathit{first}^1_{a,c}$, contains each element I_i of $\{I_a, \ldots, I_c\}$ for which $cw(I_i)$ may be joined to the counterclockwise endpoint of an element of $\mathcal{I}^* \setminus \{I_a, \ldots I_c\}$ by a line segment that is neighborly or extensional. As we will prove that there are at most 20 elements in this set, we just put each of them as the single element of its equivalence class. An element I_i belongs to $\mathit{first}^1_{a,c}$ if it satisfies one of three conditions:

†1. $cw(I_i)$ is contained in the tip of P clockwise from u_a;

†2. $cw(I_i)$ is the forward extension point of a reflex vertex of $chain(u_c, u_a]$;

†3. $cw(I_i)$ is a reflex vertex of P and the backward extension point of $cw(I_i)$ belongs to $chain(u_c, u_a]$.

Group 2 : The second group, denoted by $\mathit{first}^2_{a,c}$, contains every element of $\{I_a, \ldots, I_c\}$ that does not belong to $\mathit{first}^1_{a,c}$ and whose clockwise endpoint is a reflex vertex of P. We can partition the elements of $\mathit{first}^2_{a,c}$ into equivalence classes without computing the sets $back(I_i)$ explicitly, as shown in the following lemma.

Lemma 3.15 *Let I_i, I_j be two elements of $first^2_{a,c}$ such that $i < j$, let $v_i = cw(I_i)$, and let $v_j = cw(I_j)$. If the following two conditions hold, then $back(I_i) = back(I_j)$:*

- *‡1. for each reflex vertex v_x equal to $ccw(I_x)$ for some $x < a$ or $x > c$, the forward extension point of v_x does not belong to $chain(v_i, v_j]$;*
- *‡2. the backward extension point of v_i belongs to $chain[u_a, v_i]$ if and only if the backward extension point of v_j belongs to $chain[u_a, v_j]$.*

Proof: Let I_k be an element of $\mathcal{I}^* \setminus \{I_a, \ldots I_c\}$, let $w = ccw(I_k)$, let $c_i = \overline{wv_i}$, and let $c_j = \overline{wv_j}$. We first show that if c_i is a potential chord of P and $w \simeq v_i$, then c_j is a potential chord of P and $w \simeq v_j$. Three facts need to be proved.

Fact 1 (c_j is a potential chord of P) : Since $I_i \in first^2_{a,c}$, the chord c_i is a reflex chord, and hence w is a reflex vertex v_y of P. It then follows from Lemma 3.9 that c_j is a chord of P, and so c_j is a potential chord of P.

Fact 2 ($\angle v_{y-1}v_yv_j$ is convex) : Let z be the forward extension point of v_y. Since $\angle v_{y-1}v_yv_i$ is convex, z belongs to $chain[v_y, v_i]$. Since condition ‡1 holds, this implies that z belongs to $chain[v_y, v_j]$, and hence that $\angle v_{y-1}v_yv_j$ is convex.

Fact 3 ($\angle v_yv_jv_{j+1}$ is convex) : Let z be the backward extension point of v_i. Since $\angle v_yv_iv_{i+1}$ is convex, z belongs to $chain[v_y, v_i]$. However $I_i \notin first^1_{a,c}$, and so z belongs to $chain[u_a, u_c]$. Hence $z \in chain[u_a, v_i]$. Since condition ‡2 holds, this implies that the backward extension point of v_j belongs to $chain[u_a, v_j]$, and therefore $\angle v_yv_jv_{j+1}$ is convex.

An identical argument shows that if c_j is a potential chord of P and $w \simeq v_j$, then c_i is a potential chord of P and $w \simeq v_i$. □

Two elements I_i, I_j of $\mathit{first}^2_{a,c}$ will be placed in the same equivalence class if they satisfy conditions ‡1 and ‡2.

Group 3 : The third group is denoted by $\mathit{first}^3_{a,c}$ and contains every element of $\{I_a, \ldots, I_c\}$ that does not belong to $\mathit{first}^1_{a,c}$ or $\mathit{first}^2_{a,c}$. We observe that, if I_i belongs to $\mathit{first}^3_{a,c}$, then $back(I_i) = \emptyset$. We therefore place every element of $\mathit{first}^3_{a,c}$ in the same equivalence class.

Counting the equivalence classes of $\mathit{first}^*_{a,c}$

Before explaining how to efficiently compute the partition of $\{I_a, \ldots, I_c\}$ described in the previous section, we prove that it contains only $O(1)$ equivalence classes. We start by counting the number of potential points contained in an individual tip of P.

Lemma 3.16 *If a simple polygon P is B_3 then every tip of P contains at most* 5 *forward bounce points and* 16 *potential points.*

Proof: Consider a tip T of P. Since P is B_3, it follows from Lemma 3.11 that T contains more than four extension points. Let us count the forward bounce points of T. Each of them is obtained from an extension point of the tip T' of P clockwise from T, or from the clockwise endpoint of T'. Hence T contains at most $4 + 1 = 5$ forward bounce points. Similarly T contains at most five backward bounce points. Therefore, T contains at most $10\,(\text{bounce points}) + 4\,(\text{extension points}) + 2\,(\text{reflex vertices}) = 16$ potential points. □

We now combine Lemmas 3.11 and 3.16 to bound the cardinality of $\mathit{first}^1_{a,c}$. By symmetry, the same bound holds for the cardinality of $\mathit{last}^1_{a,c}$.

Lemma 3.17 *If P is B_3, then* $|\mathit{first}^1_{a,c}| \leq 20$.

Proof: Since P is B_3, at most two reflex vertices of $chain(u_c, u_a]$ have forward extension points in $chain[u_a, u_c)$ by Lemma 3.11. Similarly at most two reflex vertices of $chain[u_a, u_c)$ may have a backward extension point that belongs to $chain(u_c, u_a]$. Finally it follows from Lemma 3.16 that the tip of P clockwise from u_a contains at most 16 potential points. Hence the cardinality of $\mathit{first}^1_{a,c}$ is at most 20. □

Finally, let us determine the number of equivalence classes containing elements of $\mathit{first}^2_{a,c}$. Once again, this follows from Lemma 3.11.

Lemma 3.18 *If P is B_3, then the number of equivalence classes that contain elements of $\mathit{first}^2_{a,c}$ is at most* 6.

Proof: It suffices to determine the sizes of the partitions of $\mathit{first}^2_{a,c}$ induced by conditions ‡1 and ‡2 respectively. We consider each condition separately.

> Condition ‡1 : Since P is B_3, at most two reflex vertices of $chain(u_c, u_a]$ have forward extension points in $chain[u_a, u_c)$ by Lemma 3.11. These points induce a partition of $chain[u_a, u_c]$ into at most three disjoint subsets, and two elements I_i, I_j of $\mathit{first}^2_{a,c}$ that belong to the same of these subsets satisfy conditions ‡1.
>
> Condition ‡2 : Let I be an element of $\mathit{first}^2_{a,c}$. Since I does not belong to $\mathit{first}^1_{a,c}$, its backward extension point belongs to $chain(u_a, u_c]$. Hence either it belongs to $chain[u_a, cw(I)]$, or it belongs to $chain[cw(I), u_c]$. Therefore condition ‡2 partitions $\mathit{first}^2_{a,c}$ into two subsets.

Hence conditions ‡1 and ‡2 induce a partition of $\mathit{first}^2_{a,c}$ into at most $2 \times 3 = 6$ subsets.
□

Computing the partition of $\{I_a, \ldots, I_c\}$

We now explain how to compute the partition of $\{I_a, \ldots, I_c\}$ into equivalence classes efficiently. Our algorithm relies on two results. The first one concerns $\mathit{first}^1_{a,c}$.

Lemma 3.19 $\mathit{first}^1_{a,c} \subseteq \mathit{first}^1_{a,b} \cup \mathit{first}^1_{b+1,c}$.

Proof: Let I be an element of $\mathit{first}^1_{a,c}$. If I satisfies condition †1 then I belongs to $\mathit{first}^1_{a,b}$. Suppose now that I satisfies condition †2 or condition †3. If I is in $\{I_a, \ldots, I_b\}$, then since $chain(u_{b+1}, u_a] \subseteq chain(u_c, u_a]$, I belongs to $\mathit{first}^1_{a,b}$. Similarly, if $I \in \{I_{b+1}, \ldots, I_c\}$, then I belongs to $\mathit{first}^1_{b+1,c}$. □

Given $\mathit{first}^1_{a,b}$ and $\mathit{first}^1_{b+1,c}$, we can therefore compute $\mathit{first}^1_{a,c}$ in time proportional to $|\mathit{first}^1_{a,b}| + |\mathit{first}^1_{b+1,c}|$, that is in constant time by Lemma 3.17. Let us now consider $\mathit{first}^*_{a,c}$ as a whole.

Lemma 3.20 *If I_i and I_j are equivalent under $\mathit{first}^*_{a,b}$, then they are equivalent under $\mathit{first}^*_{a,c}$.*

Proof: We can assume without loss of generality that $I_i \neq I_j$. First suppose that I_i and I_j belong to $\mathit{first}^2_{a,b}$. Since by Lemma 3.19 neither I_i nor I_j is in $\mathit{first}^1_{a,c}$, they belong to $\mathit{first}^2_{a,c}$. Since $\mathcal{I}^* \setminus \{I_a, \ldots I_c\} \subseteq \mathcal{I}^* \setminus \{I_a, \ldots I_b\}$, both of conditions ‡1 and ‡2 hold for I_i and I_j as elements of $\mathit{first}^2_{a,c}$, and so they are equivalent under $\mathit{first}^*_{a,c}$. Suppose now that I_i and I_j are elements of $\mathit{first}^3_{a,c}$. Since by Lemma 3.19 neither I_i nor I_j is in $\mathit{first}^1_{a,c}$, they belong to $\mathit{first}^3_{a,c}$, and so are equivalent under $\mathit{first}^*_{a,c}$. ◫

An identical proof shows that two elements equivalent under $\mathit{first}^*_{b+1,c}$ are also equivalent under $\mathit{first}^*_{a,c}$. Hence Lemma 3.20 ensures that each equivalence class determined by $\mathit{first}^*_{a,c}$ is the union of one or more equivalence classes determined by $\mathit{first}^*_{a,b}$ and $\mathit{first}^*_{b+1,c}$. To compute the equivalence classes of elements of $\mathit{first}^2_{a,c}$, it thus suffices to consider the following elements of $\{I_a, \ldots, I_c\}$:

- those that belong to $\mathit{first}^1_{a,b} \cup \mathit{first}^1_{b+1,c}$ but not to $\mathit{first}^1_{a,c}$;
- one element from each class of $\mathit{first}^*_{a,b}$ containing elements of $\mathit{first}^2_{a,b}$;
- one element from each class of $\mathit{first}^*_{b+1,c}$ containing elements of $\mathit{first}^2_{b+1,c}$.

Since the total number of these elements does not exceed 32 by Lemmas 3.17 and 3.18, this can be done in constant time.

3.3.4 The algorithm

The algorithm used to recognize B_3 polygons efficiently proceeds in two stages. During the first stage, it preprocesses P to compute its potential points and allow certain visibility queries to be answered in constant time. In the second stage, it computes a set $C^*_{1,m}$ of equivalence classes of covers by considering the tips of P one at a time.

The set $\mathcal{C}^*_{a,c}$ is represented in the same way as the set $\mathcal{C}_{a,c}$ of Section 3.1. If $\mathcal{C}$ belongs to $\mathcal{C}_{a,c}$, we will denote by $rep_{a,c}(\mathcal{C})$ the pair obtained from $\mathcal{C}$ by replacing each element of $first(\mathcal{C})$ by its class representative under $first^*_{a,c}$, and each element of $last(\mathcal{C})$ by its class representative under $last^*_{a,c}$.

The preprocessing phase first finds the potential points of P. We showed in Section 3.3.2 that this can be done in linear time. It then preprocesses P to allow the following query to be answered in constant time:

Given two potential points x and y, is $\overline{xy}$ potential, and if so does x see y?

Deciding whether $\overline{xy}$ is potential is trivial (recall that a potential line segment is allowed to intersect $ext(P)$). Moreover, if x does not see y, then $\overline{xy}$ is neighborly, by Lemma 3.9. To answer the query in constant time, it thus suffices to precompute all neighborly chords of P. Given x and y, it follows from Lemma 3.12 that we can determine whether x sees y by searching the tips of P that contain x and y. Moreover, the number of neighborly line segments with a given potential point x of P as an endpoint is at most 30 (one for each potential point of the two tips of P adjacent to that containing x, not counting the reflex vertices they share with the tip containing x). Hence at most $16 \times 30 = 480$ neighborly line segments of P may have an endpoint in any given tip of P, and so we can precompute all neighborly chords of P in $O(n)$ time.

We now explain how $\mathcal{C}^*_{a,c}$ is computed. The base case occurs when $chain[u_a, u_c]$ is a tip of P. The induction step constructs $\mathcal{C}^*_{a,c}$ from $\mathcal{C}^*_{a,b}$ and $\mathcal{C}^*_{b+1,c}$ where $\{I_a, \ldots, I_b\}$ is the tip of P clockwise from u_a. This is done as follows:

Base case: When $\{I_a, \ldots, I_c\}$ is a tip T of P, we start by counting the number of potential points contained in T. If this number exceeds 16, then we abort. Otherwise, we compute $\mathcal{C}_{a,c}$ in any way we like (for instance by using the algorithm of Section 3.1), and return it.

Induction step: Let $\{I_a, \ldots, I_b\}$ be the tip of P clockwise from u_a. We first compute $\mathcal{C}^*_{a,b}$ and $\mathcal{C}^*_{b+1,c}$. Next we determine the equivalence relations $first^*_{a,c}$ and $last^*_{a,c}$ from $first^*_{a,b}$, $first^*_{b+1,c}$, $last^*_{a,b}$ and $last^*_{b+1,c}$. If either contains more than 27 equivalence classes, we abort.

We then consider each pair $\mathcal{C}$, $\mathcal{C}'$ of elements of $\mathcal{C}^*_{a,b}$, $\mathcal{C}^*_{b+1,c}$ respectively. Let $\mathcal{C}^* = \mathcal{C} \circ \mathcal{C}'$. We check whether the chords joining corresponding elements of $last(\mathcal{C})$ and $first(\mathcal{C}')$ are potential, and if so whether $last(\mathcal{C}) \simeq first(\mathcal{C}')$. If this is the case we add $rep_{a,c}(\mathcal{C}^*)$ to $\mathcal{C}^*_{a,c}$, keeping two pointers to $\mathcal{C}$ and $\mathcal{C}'$.

We note that since each element of $\mathcal{C}$ or $\mathcal{C}'$ can take on at most 26 different values by Lemmas 3.17 and 3.18, each step only requires constant time, and hence the cardinality of $\mathcal{C}^*_{a,b}$ is bounded above by a constant. Once we have obtained $\mathcal{C}^*_{1,m}$, we discard from it every element $\mathcal{C}$ for which $last(\mathcal{C}) \not\simeq first(\mathcal{C})$. Since P has $O(n)$ tips, the algorithm runs in linear time. Its correctness follows from the following lemma.

Lemma 3.21 *If C is a potential cover of $bd(P)$, and $\mathcal{C}$ is the equivalence class of covers of $\{I_a, \ldots, I_c\}$ to which $rest(C, \{I_a, \ldots, I_c\})$ belongs, then $rep_{a,c}(\mathcal{C})$ belongs to $\mathcal{C}^*_{a,c}$.*

Proof: Since C is potential, Lemma 3.2 implies that $rest(C, \{I_a, \ldots, I_c\})$ is a convex cover of $\{I_a, \ldots, I_c\}$ by subsets of $\mathcal{I}^*$. Hence $\mathcal{C}$ belongs to $\mathcal{C}_{a,c}$. We now proceed by induction on the number of reflex vertices of P contained in $chain[u_a, u_c]$.

Base case: When $chain[u_a, u_c]$ is a tip of P, it contains at most 16 potential points by Lemma 3.16, and hence $\mathcal{C}^*_{a,c} = \mathcal{C}_{a,c}$. Since every potential point of $chain[u_a, u_c]$ belongs to $first^1_{a,c}$, this implies that $rep_{a,c}(\mathcal{C}) = \mathcal{C} \in \mathcal{C}^*_{a,c}$.

Induction step: Let $\mathcal{C}$, $\mathcal{C}'$ be the equivalence classes of covers of $\{I_a, \ldots, I_b\}$ and $\{I_{b+1}, \ldots, I_c\}$ that contain $rest(C, \{I_a, \ldots, I_b\})$ and $rest(C, \{I_{b+1}, \ldots, I_c\})$ respectively, and let $\mathcal{C}'' = rep_{b+1,c}(\mathcal{C}')$. It follows from the base case that $\mathcal{C} \in \mathcal{C}^*_{a,b}$, and from the induction hypothesis that $\mathcal{C}'' \in \mathcal{C}^*_{b+1,c}$. Moreover Lemmas 3.17 and 3.18 imply that each of $first^*_{a,c}$ and $last^*_{a,c}$ contains at most $20 + 6 + 1$ (for *nil*) $= 27$ equivalence classes.

Since C is potential, the chords joining corresponding elements of $last(\mathcal{C})$ and $first(\mathcal{C}')$ are potential, and moreover $last(\mathcal{C}) \simeq first(\mathcal{C}')$. Since $first(\mathcal{C}')$ and $first(\mathcal{C}'')$ are equivalent, the chords joining corresponding elements of $last(\mathcal{C})$

and $first(\mathcal{C}'')$ are potential, and $last(\mathcal{C}) \simeq first(\mathcal{C}'')$. Hence $rep_{a,c}(\mathcal{C} \circ \mathcal{C}'')$ belongs to $\mathcal{C}^*_{a,c}$. Since Lemma 3.20 implies that $rep_{a,c}(\mathcal{C} \circ \mathcal{C}'') = rep_{a,c}(\mathcal{C} \circ \mathcal{C}')$, this completes the proof. □

Hence if $\mathcal{C}_{1,m} \neq \emptyset$, then $\mathcal{C}^*_{1,m} \neq \emptyset$, which implies that a potential cover of $bd(P)$ will be found if P is B_3. We can recover a cover corresponding to a given element of $\mathcal{C}^*_{1,m}$ in the same manner as in Section 3.1 using the two pointers kept with each element. We have thus proved the following result:

Theorem 3.3 *We can determine whether P is B_3, and return a cover of $bd(P)$ by three convex subsets of P if one exists, in $O(n)$ time and space.*

3.4 Recognizing U_3 polygons

We now consider the recognition problem for U_3 polygons. We first show that a starshaped polygon is U_3 if and only if it is B_3. Hence starshaped U_3 polygons can be recognized in linear time using the algorithm of Section 3.3. We then consider simple polygons that are not starshaped, and show how to reduce the problem of determining whether they are U_3 to six instances of the recognition problem for U_2 polygons. These can therefore also be recognized in linear time [92].

Let P be a starshaped B_3 polygon, and let (Q_1, Q_2, Q_3) be a cover of $bd(P)$ by three convex subpolygons of P. Let y be a point of $kr(P)$, and let Q^*_i be the convex hull of $Q_i \cup \{y\}$ for each i. Each Q^*_i is convex, and moreover it is contained in P since $y \in kr(P)$. If x is a point of P, then $hit(y,x)$ is covered by some Q_i, and hence x belongs to Q^*_i. This shows that (Q^*_1, Q^*_2, Q^*_3) covers P, and hence we have proved that:

Lemma 3.22 *If P is starshaped, then P is U_3 if and only if it is B_3.*

When P is starshaped, we can thus determine whether it is U_3 in linear time using the algorithm presented in Section 3.3. Consider now a polygon P that is not starshaped. Even if P is B_3, there is no guarantee that it will be U_3, and hence the previous approach fails. However we observe that since P is not starshaped, the three convex polygons in a cover of P cannot have a common intersection. We will in fact show that there is a chord c of P that separates two of them. We then determine whether P is U_3 by solving two instances of the recognition problem for U_2 polygons, for each one of three possible candidates for c.

More formally, let c be a chord of P that does not contain any reflex vertex of P, and hence divides P into two subpolygons Q_α, Q_β. We will call a cover (Q_1, Q_2, Q_3) of P *c-separated* if Q_1 is a convex subset of Q_α, Q_2 is a convex subset of Q_β, and Q_3 is a convex subpolygon of P that contains c. A *separating set* for P is a set of chords of P with the property that *every* cover of P by 3 convex subsets is c-separated for some element c in the set. We now prove that P always admits a small separating set.

Lemma 3.23 *Every U_3 simple polygon that is not starshaped admits a separating set containing at most three chords.*

Proof: Let P be simple polygon that is U_3 but not starshaped. For each edge e_i of P, let m_i, H_i, and Q_i^* denote the midpoint of e_i, the interior halfplane of e_i, and the visibility region from m_i in P respectively. Since P is not starshaped, Helly's theorem [39,60] implies that there are three edges e_a, e_b, e_c of P such that $H_a \cap H_b \cap H_c = \emptyset$. Two cases now need to be considered.

Case 1 (some two of Q_a^*, Q_b^* and Q_c^* do not intersect) : We can assume that $Q_a^* \cap Q_b^* = \emptyset$. Let c_a be the chord of P that is contained in an edge of Q_a^* and separates Q_a^* from Q_b^*, and let c_b be the chord of P that is contained in an edge of Q_b^* and separates Q_a^* from Q_b^*. The chords c_a and c_b do not intersect, and hence there is a line l that separates c_a from c_b and does not contain any reflex vertex of P. Let c be a connected component of $l \cap P$ that intersects the shortest path joining m_a to m_b in P, as shown in Figure 3.10a. Since every convex subpolygon of P that covers m_a or m_b is contained in Q_a^* or Q_b^* respectively, $\{c\}$ is a separating set for P.

Case 2 (Q_a^*, Q_b^* and Q_c^* intersect pairwise) : Let T be the triangle determined by the lines containing e_a, e_b and e_c, and x be a point in the interior of T, chosen so

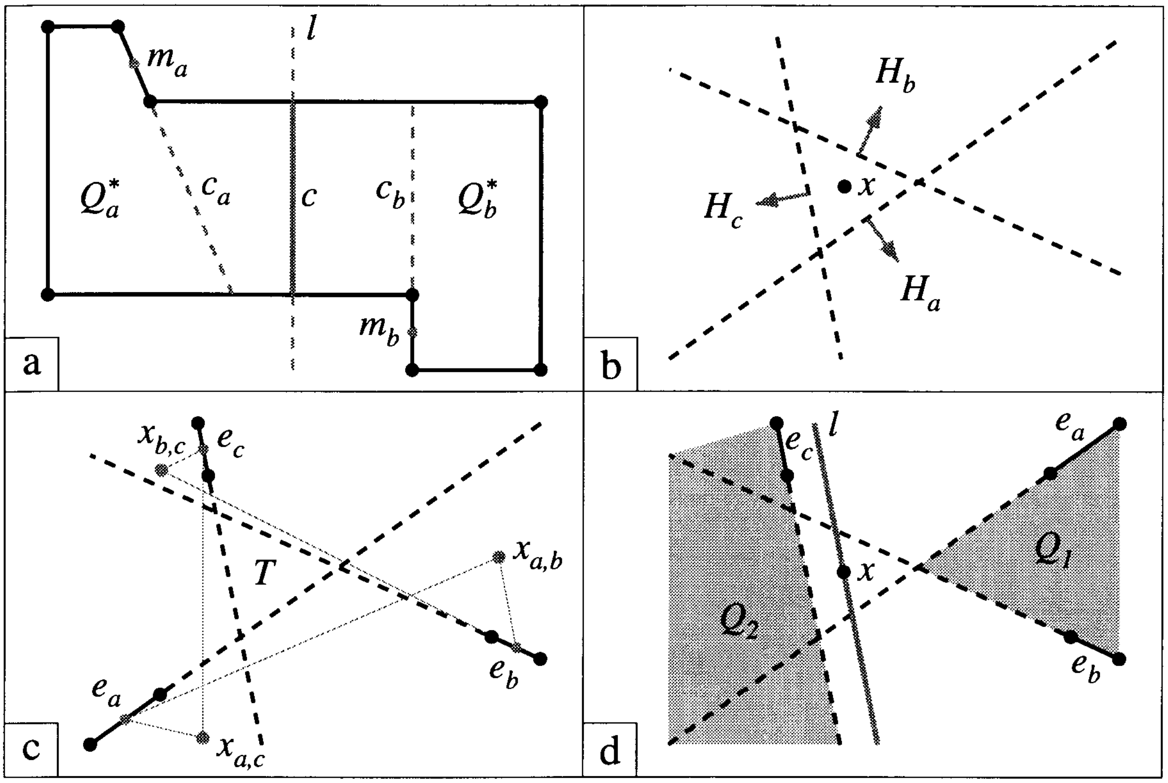


Figure 3.10: Illustrating the proof of Lemma 3.23

that the lines through x and parallel to e_a, e_b and e_c do not contain any reflex vertex of P. Since H_a, H_b and H_c do not have a common intersection, they are oriented away from x, as shown in Figure 3.10b.

Let $x_{a,b}$, $x_{a,c}$, and $x_{b,c}$ be points of $Q_a^* \cap Q_b^*$, $Q_a^* \cap Q_c^*$ and $Q_b^* \cap Q_c^*$ respectively. The path $x_{a,b}m_a x_{a,c}$ is contained in $P \cap H_a$, the path $x_{a,c}m_c x_{b,c}$ is contained in $P \cap H_c$, and the path $x_{b,c}m_b x_{a,b}$ is contained in $P \cap H_b$. Hence T is contained in P, as shown in Figure 3.10c.

Since x does not belong to any of H_a, H_b and H_c, no convex subpolygon of P that covers one of m_a, m_b and m_c also covers x. Hence in every cover C of P by three convex subsets of P, there is an element Q_1 that contains two of m_a, m_b and m_c, and a second element Q_2 that contains the third.

The line l through x and parallel to the edge containing the point covered by

Q_2 separates Q_1 from Q_2, and hence C is separated with respect to the connected component c of $l \cap P$ containing x, as shown in Figure 3.10d. Therefore the set containing the three chords of P that go through x and are parallel to e_a, e_b and e_c is a separating set for P. □

Hence every polygon admits a small separating set. The three edges e_a, e_b and e_c can be found in linear time by modifying the kernel-finding algorithm of Lee and Preparata [72, 85]. The polygons Q_a^*, Q_b^* and Q_c^* can also be computed in linear time [70,83]. To determine whether two of them intersect (say Q_a^* and Q_b^*), it suffices to verify whether the shortest link path from m_a to m_b has at most two links [95], which can also be done in linear time. Hence we can decide which of cases 1 and 2 applies in linear time. If we are in case 1, then the shortest path from m_a to m_b can be found in linear time [30, 57], as can l and c. If case 2 applies, then the three chords can be computed in linear time as well. Therefore, we can compute a separating set for P in $O(n)$ time.

Let us now show how to reduce the problem of finding a c-separated cover to two instances of the recognition problem for U_2 polygons. Let Q be one of the subpolygons of P determined by a chord c of P. We establish a relation between c-separated covers of P and covers of Q by two convex subsets of Q, one of which contains c.

Lemma 3.24 *Let P be a simple polygon, and let c be a chord for P that does not contain any reflex vertex of P. The polygon P admits a c-separated cover if and only if each of the two subpolygons of P determined by c admits a cover by two convex subsets of itself, one of which contains c.*

Proof: Let Q_α and Q_β be the two subpolygons of P determined by c. Consider first a c-separated cover (Q_1, Q_2, Q_3) of P. By symmetry, it suffices to exhibit the required cover of Q_α. Since the endpoints of c are not reflex vertices of P, the set $Q_3 \cap Q_\alpha$ is the intersection of Q_3 with one of the halfplanes determined by c, and hence it is convex. Moreover (Q_1, Q_2, Q_3) is c-separated, and so $Q_3 \cap Q_\alpha$ contains c. Therefore $(Q_1, Q_3 \cap Q_\alpha)$ is the desired cover of Q_α.

Conversely, let $(Q_{\alpha,1}, Q_{\beta,1})$ and $(Q_{\alpha,2}, Q_{\beta,2})$ be covers of Q_α and Q_β respectively in which $c \subseteq Q_{\beta,1}$ and $c \subseteq Q_{\beta,2}$. To prove that $(Q_{\alpha,1}, Q_{\beta,1}, Q_{\alpha,2} \cup Q_{\beta,2})$ is a c-separated cover of P, it suffices to show that $Q_{\alpha,2} \cup Q_{\beta,2}$ is convex. Since $Q_{\alpha,2}$ and $Q_{\beta,2}$ are convex, every reflex vertex of $Q_{\alpha,2} \cup Q_{\beta,2}$ would have to belong to $Q_{\alpha,2} \cap Q_{\beta,2} = c$ [92, 99]. However this is impossible since no endpoint of c is a reflex vertex of P. Therefore $Q_{\alpha,2} \cup Q_{\beta,2}$ is convex as claimed. □

Given c, we can thus decide whether P admits a c-separated cover, and find one if one exists, by computing covers of Q_α and Q_β by two convex subsets in which the second element contains c. This can be done in linear time using the algorithm of Shermer [92].

To summarize what we have done: to recognize U_3 polygons, we first determine whether P is starshaped, which can be done in linear time [72]. If it is, we decide whether P is B_3, which is the same as being U_3, using the linear time algorithm described in Section 3.3. Otherwise we compute a separating set for P, and then attempt to find a c-separated cover for each chord c in that set. If we succeed then P is U_3, otherwise it is not.

Theorem 3.4 *We can determine whether P is U_3, and return a cover of P by three convex subsets if one exists, in $O(n)$ time and space.*

Chapter 4

Recognizing U_2 polytopes in $\boldsymbol{E}^d$

In this section, we show how to determine whether a simple polytope P in $\boldsymbol{E}^d$ is U_2. We reduce this problem to the problem of 2-coloring a graph G defined on a partition of P. This graph contains one node for each component of the partition, and an edge joins two nodes if the corresponding components are not completely visible. We prove that P is U_2 if and only if G admits a proper 2-coloring, and show how to obtain a cover of P by 2 convex polytopes from every proper 2-coloring of G. Unless otherwise specified, each k-face of P will be considered open within the affine plane of dimension k that contains it.

We first describe the partition of P and the graph G defined on it, and prove the correspondence between covers of P and 2-colorings of G. Next we show how to compute the partition of P and the graph G in $\boldsymbol{E}^3$. We then explain how G can be found in $O(n \log n)$ time for the special case $d = 3$. Finally we prove that the problem can be solved in polynomial time in $\boldsymbol{E}^d$ for each fixed value of d.

4.1 A partition of P

The task of finding a simple algorithm to recognize U_2 polytopes will become easier if we restrict the kinds of subsets of these polytopes that need to be used in those covers. Hence we will only consider covers of a simple polytope P by *maximal* convex subsets of P, that is, convex subsets of P that are not strictly contained in any other convex subset of P.

Observation 4.1 *Every maximal convex subset of a simple polytope P contains the kernel of P.*

In order to determine whether P is U_2, it thus suffices to decide whether the points of $P \setminus kr(P)$ can be partitioned into two subsets Q_1, Q_2 such that $Q_1 \cup kr(P)$ and $Q_2 \cup kr(P)$ are both convex. We now show how to reduce this problem to that of partitioning the set $\mathcal{Q}^* = \{Q_1^*, \ldots, Q_m^*\}$ of connected components of $P \setminus kr(P)$.

Lemma 4.1 *If P is a simple polytope, $\{Q_1, Q_2\}$ is a cover of P by maximal convex subsets of P, and Q^* is an element of $\mathcal{Q}^*$, then exactly one of Q_1 and Q_2 intersects Q^*.*

Proof: Since $Q_1 \cup Q_2 = P$, either Q_1 intersects Q^* or Q_2 intersects Q^*. Suppose now that there are points x, y of Q^* that belong to Q_1 and Q_2 respectively. There is a path that joins x to y in Q^*. Since this path is closed and contained in the union of two closed sets, there is a point of the path that is contained in both of these sets. Hence a point z of Q^* belongs to $Q_1 \cap Q_2$. But then $z \in kr(P)$, which contradicts the fact that $z \in Q^*$. □

Every cover of P by two maximal convex subsets thus induces a partition of $\mathcal{Q}^*$. However the converse does not hold. Hence we need conditions that will allow us to decide whether or not a partition of $\mathcal{Q}^*$ corresponds to a cover of P by two maximal convex subsets. These conditions are provided by the next lemma.

Lemma 4.2 *Let P be a U_2 polytope, let $\{\mathcal{Q}_1^*, \mathcal{Q}_2^*\}$ be a partition of $\mathcal{Q}^*$ into two subsets, and let Q_1 and Q_2 be the union of $kr(P)$ with the elements of $\mathcal{Q}_1^*$ and $\mathcal{Q}_2^*$ respectively. The sets Q_1 and Q_2 are convex if and only if the elements of $\mathcal{Q}_1^*$ are completely visible pairwise, and the elements of $\mathcal{Q}_2^*$ are completely visible pairwise.*

Proof: If Q_1 is convex, then every two points of Q_1 are visible, and hence the elements of $\mathcal{Q}_1^*$ are completely visible pairwise. Similarly, if Q_2 is convex then the elements of $\mathcal{Q}_2^*$ are completely visible pairwise. Suppose now that the elements of $\mathcal{Q}_1^*$ are completely visible pairwise, and that the elements of $\mathcal{Q}_2^*$ are completely visible pairwise. Since P is U_2, every two points of each given element of $\mathcal{Q}^*$ see each other. Hence $\{Q_1, Q_2\}$ is a clique cover for the point visibility graph of P, or equivalently, the complement of the point visibility graph of P is 2-colorable. The lemma then follows from a result of McKinney, and of Hare and Kenelly [59, 76]. □

A partition of $\mathcal{Q}^*$ into two subsets will be called *convex* if the elements of each subset are completely visible pairwise. By Lemma 4.2, we can determine whether P is U_2 by computing a convex partition of $\mathcal{Q}^*$, and verifying that the corresponding subsets of P are convex. We can model the problem of computing a convex partition of $\mathcal{Q}^*$ using a graph $G = (V, E)$ constructed as follows:

- there is a one-to-one correspondence between the elements of $\mathcal{Q}^*$ and the nodes of V;
- two nodes of V are joined by an edge if and only if the corresponding elements of $\mathcal{Q}^*$ are not completely visible.

It can be easily verified that there is a bijection between the set of convex partitions of $\mathcal{Q}^*$ and the set of proper 2-colorings of G. We have thus obtained the following result.

Theorem 4.1 *If P is U_2, then G admits a proper 2-coloring and every proper 2-coloring of G corresponds to a cover of P by two maximal convex subsets. Conversely, if P is not U_2, then no proper 2-coloring of G induces a partition of P into two convex subsets.*

In order to obtain a recognition algorithm for U_2 polytopes in three-dimensional space, it thus suffices to show how to compute $\mathcal{Q}^*$ and G. This is done in the following two sections.

4.2 Computing $\mathcal{Q}^*$ in $\boldsymbol{E}^3$

We compute $\mathcal{Q}^*$ in two stages. During the first stage, we find $kr(P)$ and determine the faces of P that intersect each vertex, edge and facet of $kr(P)$. In the second stage, we use this information to construct a subdivision from which we can recover the faces of $f \setminus kr(P)$ for each facet f of P, and the connected components of $P \setminus kr(P)$ to which they belong. For the remainder of this section, we shall assume that P and $kr(P)$ are given by their face lattices, represented using the quad-edge data structure of Guibas and Stolfi [58]. Hence we can find the edge counterclockwise from a given edge on the boundary of a simple facet in constant time.

4.2.1 Locating subfaces of $kr(P)$ in P

We first need to compute $kr(P)$ by intersecting the positive halfspaces of each facet of P. This requires, for each facet f of P, a point that belongs to the plane H containing f and a vector normal to H pointing towards the interior of P. For the former, we choose an arbitrary vertex of f. For the latter, we use the cross product $(\vec{v}_k - \vec{v}_i) \times (\vec{v}_j - \vec{v}_i)$, where v_i, v_j and v_k are three vertices that are not all collinear and occur in this order in a counterclockwise traversal of $bd(f)$. The kernel of P can then be computed in $O(n \log n)$ time using $O(n)$ space [41]. Let us now characterize the faces of P in which faces of $kr(P)$ might be found.

Lemma 4.3 *Let P be a simple polytope and f be k-face of P. If f^{kr} is a face of $kr(P)$ that intersects f, then $f^{kr} \subseteq kr(f)$.*

Proof: Since f is a k-face of P, the affine subspace H of dimension k containing f is the intersection of $3-k$ hyperplanes supporting facets of P. Since f^{kr} is contained in the interior halfspace of each of these hyperplanes, and since it is open with respect to the affine subspace H^{kr} of dimension $dim(f^{kr})$ that contains it, $f^{kr} \subseteq H$. In the polytope drawn in Figure 4.1a, f^{kr} is not contained in H, and hence f^{kr} can not be a face of $kr(P)$. This is easily verified since f^{kr} is not contained in the interior halfspace of either of the facets of P whose boundary contains f.

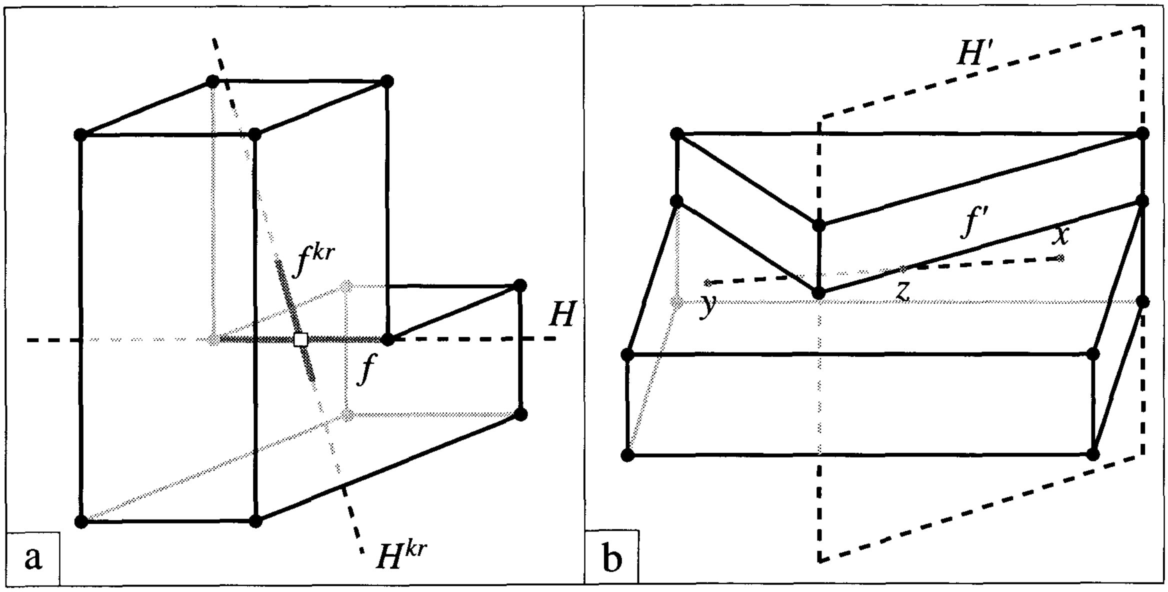


Figure 4.1: Illustrating the proof of Lemma 4.3

Consider now a point x of f^{kr}. We prove by contradiction that $x \in kr(f)$. If $x \notin kr(f)$, there is a point y of f such that $\overline{xy}$ is not contained in f, as shown in Figure 4.1b. Let z be the point closest to y on $\overline{xy}$ that is visible from x, and f' be a subface of f in whose closure z lies. Finally, let H' be an hyperplane supporting a facet of P, and whose intersection with H is the affine subspace of dimension $k-1$ that contains f'. Since $x \in kr(P)$, the line segment $\overline{zy}$ is contained in P, and hence $\overline{zy}$ belongs to the interior halfspace of H'. However this means that x does not belong to this interior halfspace, a contradiction. $\square$

Hence every face of $kr(P)$ is contained in some face of P. This means that we can find the subfaces of P intersected by a subface f^{kr} of $kr(P)$ as follows: we choose an arbitrary point x of f^{kr}, and try to locate it on the boundary of P. If x is found to lie on a subface f of $bd(P)$, then every point of f^{kr} belongs to f. Otherwise x lies in the interior of P, and hence f^{kr} is contained in $int(P)$. We thus need a way to locate a point x of $kr(P)$ in P.

One possible approach to this problem would be to project P radially from a point of $kr(P)$ onto a ball B that contains P, and then perform point location queries in the subdivision of B. At first glance, this approach seems somewhat simpler than the one that we will use. However, some complications arise when the kernel of the polygon does not have full dimension. We will therefore reduce the problem of locating x to a number of point location queries in planar subdivisions. The total size of these subdivisions and the total number of queries will both be linear, and hence the time required will be $O(n \log n)$.

We therefore proceed in three stages. We first partition the facets of P into a collection of planar subdivisions. Each planar subdivision consists of the set of facets of P contained in a common plane; we will say that this plane *supports* the subdivision. Next, we determine the faces of $kr(P)$ intersected by the planes supporting these subdivisions. Finally, we locate each face f^{kr} of $kr(P)$ by performing point location queries in the subdivisions whose supporting planes f^{kr} intersects. Stage 1 computes these subdivisions as follows:

> Stage 1 (Identify the planes containing one or more facet of P) : We store these planes in a balanced binary search tree $\mathcal{T}$. The key used for the plane H described by the equation $ax + by + cz + d = 0$ is the four-tuple $t_H = (a/t, b/t, c/t, d/t)$, where t is chosen so that the first non-zero element of t_H is 1.
>
> We thus consider each facet f of P, and insert its supporting plane H in $\mathcal{T}$. We then add f to the list of facets associated with the tree node containing H. Each insertion requires $O(\log n)$ time, and hence stage 1 runs in $O(n \log n)$ time.

Let us now show how to find the intersection with $kr(P)$ of each plane supporting one of the subdivisions computed in stage 1. We can transform the problem of intersecting a plane H with $kr(P)$ into that of locating the point $\mathcal{D}_0(H)$ on the boundary of the polytope $Q = \mathcal{D}_0(kr(P))$ ([41], Corollary 1.9). Stage 2 thus proceeds as follows:

> Stage 2 (Compute the intersection of each plane supporting facets of P with $kr(P)$) : We start by computing a plane containing each facet of Q. Next, we choose a plane H^* not parallel to any of them, and call *vertical* one of the two orientations normal to H^*. We then determine the upper and lower envelopes

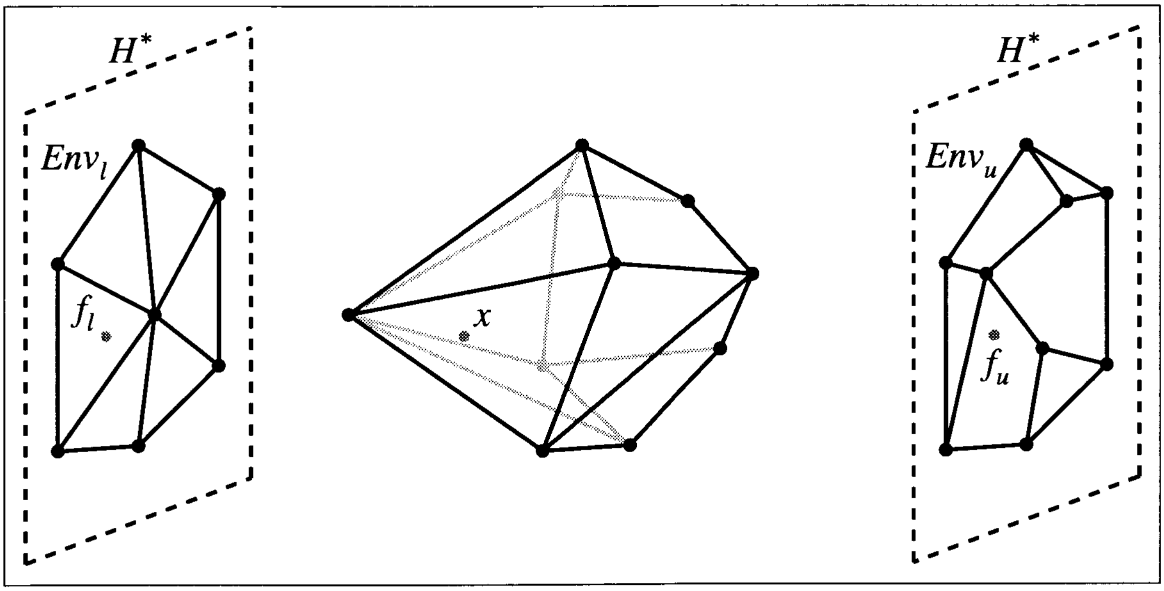


Figure 4.2: Locating a point x on a convex polytope

of Q and their projections Env_u, Env_l on H^*, as shown in Figure 4.2. Finally, we preprocess Env_u and Env_l for planar point location [42,69].

Given $x = \mathcal{D}_0(H)$, we locate x on Q by computing the perpendicular projection y of x on H^*, and locating y in each of Env_u, Env_l. Let f_u and f_l be the faces of Q that correspond to the faces of Env_u and Env_l containing y respectively. We can decide whether x belongs to f_u or f_l in constant time by determining its position relative to the planes containing f_u and f_l.

We require $O(n)$ time to compute Q; since Q is convex, H^* can also be found in linear time. The planar subdivisions Env_u and Env_l can both be computed and preprocessed in $O(n)$ time [42, 69]. We then perform $O(n)$ planar point location queries in a planar subdivision of size $O(n)$. The total time spent is therefore $O(n \log n)$.

We observe that Stage 2 solves exercise 11.5b in the book by Edelsbrunner [41]. Before detailing Stage 3, we bound the total complexity of the set of facets of a convex polytope.

Lemma 4.4 *Let P be a simple polytope with n faces, and σ_f denote the complexity of a facet f of P. The sum $\sigma_P = \sum_f \sigma_f$, where f ranges over all facets of P, is $O(n)$.*

Proof: It suffices to count the number of times that each face of P contributes to σ_P. Each facet of P is counted exactly once, and each edge of P twice. Consider now the vertices of P. The number of vertices (with multiplicity) that contribute to σ_f is the same as the number of edges. Hence the same holds for σ_P, and it therefore equals the number of facets of P plus four times the number of edges of P. □

We finally explain how the faces of $kr(P)$ are located in P. The analysis of the running time of Stage 3 relies heavily on Lemma 4.4.

> Stage 3 (Locate the faces of $kr(P)$ in P) : We locate each face f^{kr} of $kr(P)$ in order of increasing dimension. We first compute a point p of f^{kr}. If f^{kr} is a vertex of P, we choose $p = f^{kr}$. Otherwise we use the arithmetic mean of the points used for the subfaces of f^{kr}; this point belongs to f^{kr} since f^{kr} is convex. Next, we consider all subdivisions contained in a plane whose intersection with $kr(P)$ contains f^{kr}, and perform one point location query in each of them. If p is found in a face f of P, we return f. Otherwise we return the unique 3-face of P.
>
> Each facet of P is contained in exactly one subdivision, and hence the total size of these subdivisions is $O(n)$ by Lemma 4.4. Similarly, each subdivision is searched at most once for each face of $kr(P)$ contained in its supporting plane. Once again, Lemma 4.4 implies that at most $O(n)$ queries are performed. Stage 3 thus runs in $O(n \log n)$ time.

By combining Stages 1 to 3, we have proved that in $O(n \log n)$ time, we can determine for each face of $kr(P)$ the face of P in which it is contained.

4.2.2 Removing faces of $kr(P)$ from P

In this section, we show how to compute the faces of $f \setminus kr(P)$ for each facet f of P, and the connected components of $P \setminus kr(P)$ to which they belong. This is done

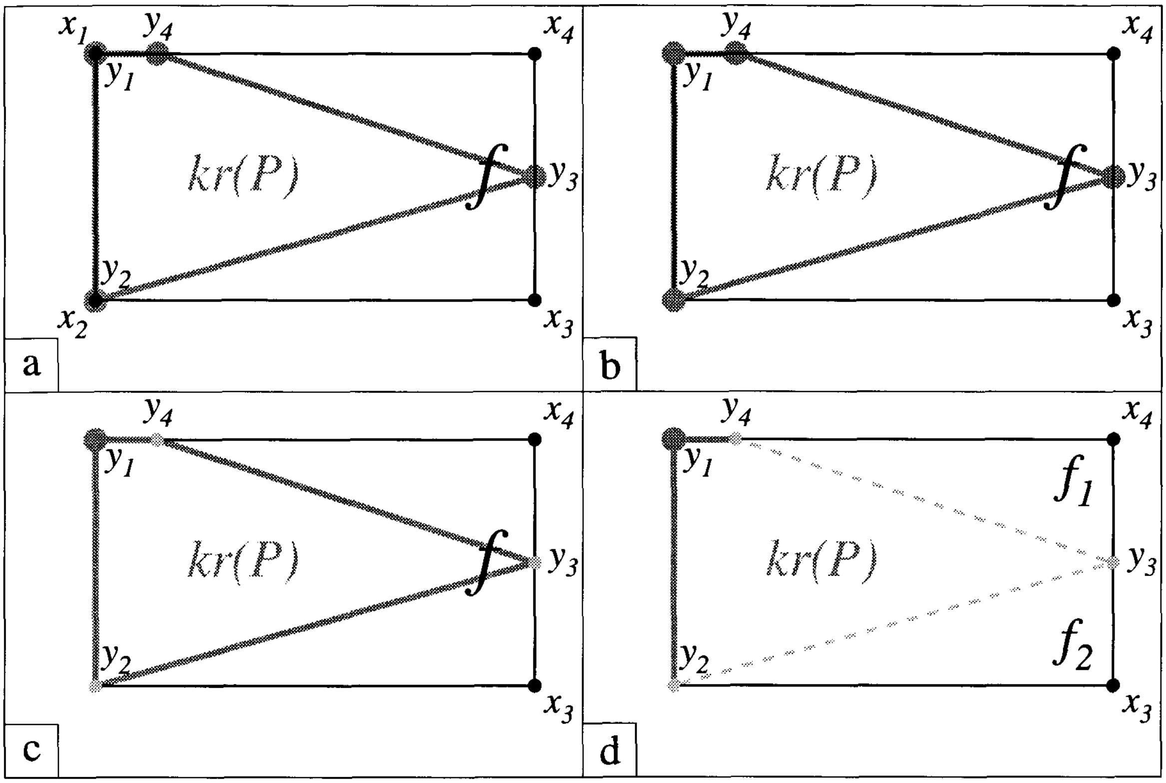


Figure 4.3: Subdividing the facets of P

by constructing a face lattice that represents the subdivision of the boundary of P created by the insertion of $kr(P)$ into P. The algorithm proceeds in two stages. The first stage is an initialization stage, and deals with the vertices of P that belong to $kr(P)$. The second stage subdivides the edges and facets of P.

While discussing these two stages, we will refer to the example illustrated in Figure 4.3. The facet f of P to be subdivided and the set $f \setminus kr(P)$ are shown in Figure 4.3a. For convenience, we shall color each face of the subdivisions created by the algorithm using one of three different colors:

- a *white* face is a face of P that has not been subdivided yet, or a face of $P \setminus kr(P)$;

- a *pink* face is a k-face of $kr(P)$ that belongs to the closure of a $(k+1)$-face of P whose superfaces have all been subdivided;

- a *red* face is a face of $kr(P)$ that is not pink.

In Figure 4.3, white, red and pink vertices and edges of f have been drawn in black, dark gray and light gray respectively. The facets contained in f have been labeled using the same colors.

Stage 1 (Initialization stage) : The initialization stage constructs a face lattice $\mathcal{F}_0$ from the face lattices $\mathcal{F}$ and $\mathcal{F}^{kr}$ of P and $kr(P)$ respectively. It proceeds in two steps.

Step 1 (Merging $\mathcal{F}$ and $\mathcal{F}^{kr}$) : We add an edge from the (-1)-face of $\mathcal{F}$ to every 0-face of $\mathcal{F}^{kr}$, and delete the (-1)-face of $\mathcal{F}^{kr}$. The faces of $\mathcal{F}_0$ that belong to $\mathcal{F}$ are white, those that belong to $\mathcal{F}^{kr}$ are red, and the (-1)-face of $\mathcal{F}_0$ is pink.

Step 2 (Merging coincident vertices) : For each white 0-face v coincident with a red 0-face v^{kr}, an edge is added from every superface of v to v^{kr}, and then v is deleted.

The subdivision of f created by the initialization stage is shown in Figure 4.3b.

Stage 2 (Induction stage) : For each $i \geq 1$, we construct $\mathcal{F}_i$ by subdividing all i-faces of $\mathcal{F}_{i-1}$. This is also done in two steps.

Step 1 (Split each i-face f of $\mathcal{F}_{i-1}$) : We first partition all white and red $(i-1)$-faces contained in the closure of f into equivalence classes: two faces are equivalent if they belong to the same connected component of $f \setminus kr(P)$. We then insert a white node N in $\mathcal{F}_i$ for each class C: the set of superfaces of N is identical to the set of superfaces of f, and the subfaces of N are the elements of C. Finally we remove f from $\mathcal{F}_{i-1}$.

Step 2 (Recolor some nodes) : We now recolor pink every $(i-1)$-face of $kr(P)$ contained in the closure of an i-face of P.

The subdivisions of f corresponding to $\mathcal{F}_i$ and $\mathcal{F}_2$ have been drawn in Figure 4.3c and 4.3d respectively.

We still need to explain how to partition the set S of all $(i-1)$-faces contained in f into equivalence classes. Since we only subdivide faces of P of dimension 1 or 2, this is easily done. Let f denote the face of P that is being subdivided.

Case 1 (f is an edge of P) : S consists of a set of vertices (at most two from P, and at most two from $kr(P)$). Two of them bound a connected component of $f \setminus kr(P)$ if and only if they are adjacent in the ordering of the elements of S along f, and are not both red. The edges $\overline{x_3x_4}$ and $\overline{x_4y_1}$ from Figure 4.3b yield the three connected components $\overline{x_3y_3}$, $\overline{y_3x_4}$ and $\overline{x_4y_4}$, as shown in Figure 4.3c.

Case 2 (f is a facet of P) : If $kr(P)$ does not intersect f, then f does not need to be subdivided (since $f \setminus kr(P) = f$). If $kr(P)$ intersects f but not $bd(f)$, then every element of S belongs to the same equivalence class. It remains to consider the case where $kr(P)$ intersects both f and $bd(f)$. The boundary of f then contains one or more pink vertices, and every element of S lies between two such vertices (not necessarily distinct).

Let f' belong to S, and v, v' be the two pink vertices between which f' lies, with v clockwise from f'. The equivalence class of f' will be the set of all elements of S that lie clockwise from v and counterclockwise from v' on either $bd(f)$ or $bd(f \cap kr(P))$. We can thus compute all such equivalence classes by simultaneously traversing $bd(f)$ and $bd(f \cap kr(P))$ counterclockwise. When this is done on the facet f of Figure 4.3c, the two new facets f_1 and f_2 shown in Figure 4.3d are obtained.

In both cases the time needed to compute the partition of S is proportional to the total number of subfaces of P and $kr(P)$ contained in the closure of f. Hence it follows from Lemma 4.4 that $\mathcal{F}_2$ is computed in $O(n)$ time. It now remains to show how to determine the partition of the set of white facets of $P \setminus kr(P)$ into its connected components. This can not be done in the same way as in case 2 above, because the edges of P are not cyclically ordered. Instead we rely on the following lemma.

Lemma 4.5 *Let f be a face of $\mathcal{F}_i$ whose subfaces have been subdivided. Two white subfaces f_1, f_2 of f belong to the same connected component of $f \setminus kr(P)$ if and only* *if there exist a path from f_1 to f_2 in $\mathcal{F}_i$ that consists only of white subfaces of f.*

Proof: Let f_1, f_2 be two white subfaces of f between which there exists a path π in $\mathcal{F}_i$ that consists only of white subfaces of f. Since each subface of f has been subdivided, no face of π intersects $kr(P)$. Hence f_1 and f_2 belong to the same connected component of $f \setminus kr(P)$.

Conversely, let f_1, f_2 be two white subfaces of f that belong to a connected component f^* of $f \setminus kr(P)$. There is a path that joins f_1 to f_2 in the closure of f^* without intersecting $kr(P)$. In fact, there is a path π that joins f_1 to f_2 in $bd(f^*)$ without intersecting $kr(P)$. Since each subface of f has been subdivided, each face of f intersected by π is white. Moreover, we can assume without loss of generality that for each face f' of f, the set $f' \cap \pi$ is connected and each endpoint of $f' \cap \pi$ belongs to a subface or a superface of f'. This corresponds to a path in $\mathcal{F}_i$ that consists only of white subfaces of f and joins f_1 to f_2. □

We can therefore use a depth-first search on $\mathcal{F}_2$ to compute these equivalence classes in $O(n)$ time. This section can therefore be summarized in the following theorem:

Theorem 4.2 *Given a simple three-dimensional polytope P of size n, we can compute in $O(n \log n)$ time the subset of $bd(P)$ contained in each element of $\mathcal{Q}^*$.*

4.3 Computing the graph G in $\boldsymbol{E}^3$

In this section, we show how to compute G given $\mathcal{Q}^*$. We thus need to determine for each pair of elements of $\mathcal{Q}^*$ whether they are completely visible. This decision is simplified by the following lemma.

Lemma 4.6 *If two elements Q_i^*, Q_j^* of $\mathcal{Q}^*$ are not completely visible, then there is a point z of P of local nonconvexity in the intersection of their closures. Moreover, in every convex neighborhood of z, there are points of Q_i^* and Q_j^* that are not visible and that belong to facets in whose closure z lies.*

Proof: If Q_i^* and Q_j^* are not completely visible, there is a point x of Q_i^*, and a point y of Q_j^* between which the shortest path $SP(x,y)$ in P is not the line segment $\overline{xy}$. Let z be the vertex of $SP(x,y)$ closest to x. No point of $SP(x,y) \cap \overline{xz}$ sees y, except possibly for z. Hence no point of $\overline{xz}$ other than z belongs to $kr(P)$, and the open line segment $\overline{xz}$ is contained in Q_i^*. Similarly, no point of $SP(x,y) \setminus \overline{xz}$ sees x, and so none of them belongs to $kr(P)$, which implies that $SP(x,y) \setminus \overline{xz}$ is contained in Q_j^*.

Hence z is a point of $kr(P)$ that belongs to the intersection of the closures of Q_i^* and Q_j^*, and since it is a vertex of $SP(x,y)$ it is a point of P of local non-convexity. Consider now a small convex neighborhood of z. There are two points x', y' of $SP(x,y)$ in that neighborhood that are not visible. Let x'' be the point of $\overline{x'y'}$ furthest from x that remains visible from x', and let y'' be define similarly with respect to y'. No point of $\overline{x'x''}$ or $\overline{y'y''}$ belongs to $kr(P)$, and hence $x'' \in Q_i^*$ and $y'' \in Q_j^*$. Moreover x'' and y'' belong to facets of P, and do not see each other. □

We use Lemma 4.6 to compute G in two stages. We consider the reflex edges of P during the first stage, and its reflex vertices in the second stage.

> Stage 1 (Edges of P) : Let e be a reflex edge of P. The kernel of P is contained in the intersection of the interior halfspaces of the two facets of P that determine e, and hence each of these two facets is entirely contained in an element of $\mathcal{Q}^*$. If they belong to different elements of $\mathcal{Q}^*$, then we add an edge in G between the nodes that correspond to these two elements.
>
> Stage 2 (Vertices of P) : Let v be a reflex vertex of P, and $f_1, \ldots, f_t$ be the white facets of $\mathcal{F}_2$ that contain edges with one endpoint at v. For every $1 \le i < j \le t$, we can determine whether f_i and f_j are contained in each other's interior halfspace in constant time. If not, and if f_i and f_j belong to different elements of $\mathcal{Q}^*$, then we add an edge in G between the nodes corresponding to f_i and f_j.

The correctness of this construction follows straightforwardly from Lemma 4.6. However, it may require $O(n^2)$ time. We now prove that if P is U_2, then the size of G is only $O(n)$.

Lemma 4.7 *If P is U_2, then G contains no more than $O(n)$ edges, which can be computed in $O(n \log n)$ time.*

Proof: Since at most one edge is added to G for each edge of P, and since P contains $O(n)$ edges, we only need to consider the edges of G that are added in the second stage. Let v be a reflex vertex of P, and $\mathcal{W} = \{f_1, \ldots, f_t\}$ be the set of white facets of $\mathcal{F}_2$ that contain edges with endpoint v. Since P is U_2, the vertex v belongs to $kr(P)$. Let H_0 be a plane strictly tangent to $kr(P)$ at v, with normal vector $\vec{n}_0$ chosen so that $kr(P) \subseteq H_0^{\geq}$. We will call a facet f of $\mathcal{W}$ *below* if it contains a point of $H_0^{<}$. We now prove two claims.

Claim : *At most two distinct elements of $\mathcal{Q}^*$ contain below facets of $\mathcal{W}$*

Proof: Consider two below facets f_i, f_j of $\mathcal{W}$ that belong to distinct elements of $\mathcal{Q}^*$. Let x_i, x_j be two points of f_i, f_j respectively that belong to $H_0^{<}$, as shown in Figure 4.4a. Since $kr(P)$ is contained in $H_0^{\geq}$ and $\overline{x_i x_j}$ is contained in $H_0^{<}$, it follows that x_i does not see x_j. Since P is U_2, at most two elements of $\mathcal{Q}^*$ may contain below facets of $\mathcal{W}$. □

Claim : *No new edge is added in stage 2 between elements of $\mathcal{Q}^*$ that do not contain below facets of $\mathcal{W}$*

Proof: Consider two facets f_i, f_j that are not below and belong to distinct elements Q_i^*, Q_j^* of $\mathcal{Q}^*$ respectively. There is a cube C contained in $H_0^{\geq}$, that does not contain any face of P except those in whose closure v lies, as illustrated in Figure 4.4b. Let x_i and x_j be two points of C that belong to f_i and f_j respectively, and do not see each other, and let H be any plane that contains v, x_i and x_j. Finally, let $P' = P \cap C \cap H$.

Since P is U_2, and both C and H are convex, P' is U_2. It is moreover connected since $v \in P'$. Hence, since x_i does not see x_j, there are adjacent tips of P' to which x_i and x_j belong. Let w be the reflex vertex of P' common to these two tips; we note that w lies on the boundary of Q_i^* and Q_j^*, and that moreover

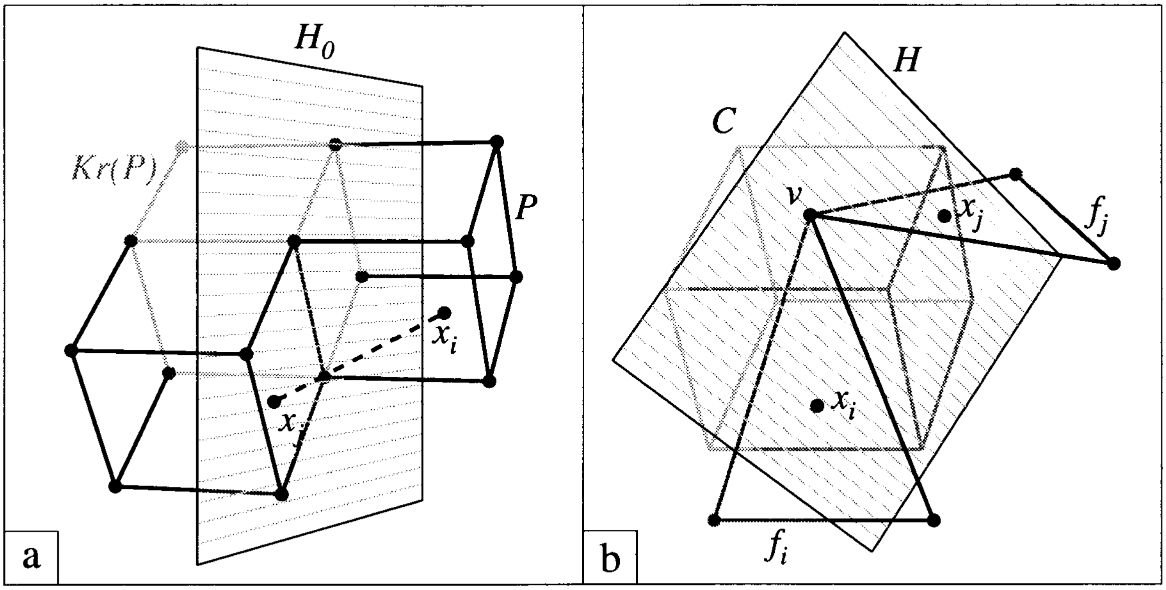


Figure 4.4: Illustrating the proof of Lemma 4.7

$w \neq v$ since v lies on a facet of C. Therefore w belongs to an edge of P whose boundary contains v, which means that the edge joining Q_i^* to Q_j^* was already added in stage 1. □

Hence every new edge added to G during the second stage joins an element of $\mathcal{Q}^*$ containing a *below* facet to another element of $\mathcal{Q}^*$ in whose boundary v lies. Hence the number of these edges does not exceed twice the number of edges that join a 0-face of P to a 1-face of P in the face lattice of P, and so it is in $O(n)$. □

We can therefore replace G by the subset G' of G computed as follows: first we perform stage 1 as previously. Then we consider each reflex vertex v of P. If v does not belong to $kr(P)$ then we abort since P is not U_2 [99]. Otherwise we compute H_0, and partition the below facets according to the element of $\mathcal{Q}^*$ to which they belong. If they belong to more than two of them, we abort. Otherwise we intersect the interior halfspaces of the below facets corresponding to each element of $\mathcal{Q}^*$, and then check whether each of the other elements of $\mathcal{W}$ is contained in this intersection in the neighborhood of v. This requires $O(t \log t)$ time, where t is the cardinality of $\mathcal{W}$.

Since the total number of elements that belong to the sets of facets to which closure each reflex vertex of P belongs is linear by Lemma 4.4, G' can be computed in $O(n \log n)$ time. If P is U_2 then $G' = G$ and every proper 2-coloring of G induces a cover of P by two convex subsets of P by Theorem 4.1. If P is not U_2, then no proper 2-coloring of G' induces such cover of P. Since $\mathcal{Q}^*$ and G can be computed in $O(n \log n)$ time, and since we can verify whether the two subsets of P corresponding to a given proper 2-coloring of G' are convex in linear time, we obtain the following result.

Theorem 4.3 *We can determine whether a simple three-dimensional polytope P is U_2, and return a cover of P by two convex subsets of P if one exists, in $O(n \log n)$ time and space.*

4.4 Computing G in $\boldsymbol{E}^d$

Let us now show how the techniques used in the preceding sections can be extended to recognize U_2 polytopes in $\boldsymbol{E}^d$. We first need to show how $\mathcal{Q}^*$ is computed. The algorithm given in Section 4.2.1 to locate the subfaces can be extended to $\boldsymbol{E}^d$, at the expense of vast amounts of both time and space. It requires n point location queries in subdivisions with up to $O(n^{\lfloor d/2 \rfloor})$ faces in $\boldsymbol{E}^{d-1}$, and up to $O(n^{\lfloor d/2 \rfloor + 1})$ point location queries in subdivisions whose total size is $O(n)$. These queries can be answered in time polynomial in n for each fixed d [29].

Similarly, the algorithm given in Section 4.2.2 to computes the faces of $f \setminus kr(P)$ for each facet f of P can be extended in a straightforward manner. It suffices to describe the manner in which the partition into equivalence classes is done in Step 1 of the induction stage.

The white $(i-1)$-faces contained in the closure of f can be partitioned in $O(n)$ time using Lemma 4.2. Let us now consider the red $(i-1)$-faces contained in the closure of f. If the closure of f contains no red i-face of $kr(P)$, then it contains at most one $(i-1)$-face of $kr(P)$, and hence there is only one equivalence class of red $(i-1)$-faces contained in f.

Suppose now that the closure of f contains an i-face f^{kr} of $kr(P)$. A proof similar to that of Lemma 4.2 can be used to show that two red $(i-1)$-faces f_1, f_2 contained in the closure of f belong to the same connected component of $f \setminus kr(P)$ if and only if there exist a path from f_1 to f_2 in $\mathcal{F}_i$ that consists only of red subfaces of f^{kr}.

Finally we need to show how to determine, for each class C_{white} of white subfaces of f, the class C_{red} of red subfaces of f that bound the connected component of $f \setminus kr(P)$ to which that class of white subfaces belongs. This can be done by considering a pink facet f' of an element of C_{white}. At least one element of C_{red} will be a superface of f'. We can determine which of the red superfaces of f' belongs to C_{red} by considering the ordering of the $(i-1)$-faces in whose boundary f' lies.

Hence each subface of P can be split easily in $O(n^{\lfloor d/2 \rfloor})$ time. The graph G can be computed in $O(n^2)$ time, and hence we have shown that U_2 polytopes can be recognized in polynomial time in $\boldsymbol{E}^d$ for each fixed value of d. We acknowledge, however, that this algorithm is very inefficient. We believe that it should be possible to determine whether P is U_2 without computing the kernel, and that there exists an efficient algorithm for that problem whose running time is not exponential in d. Our reasons for conjecturing this are outlined in the conclusion.

Chapter 5

Recognizing U_3 polytopes in $\boldsymbol{E}^3$

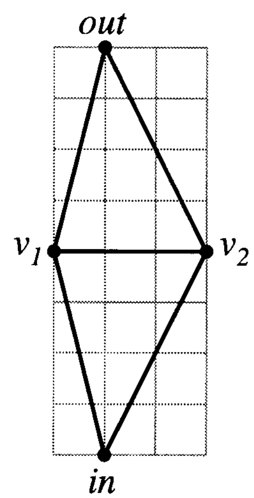
In this chapter, we prove that the problem of deciding whether a 3-dimensional star-shaped polytope is B_3 or U_3 is NP-hard. The reduction proceeds in two stages: we first prove that the problem called *Delaunay subgraph 3-colorability* is NP-complete in Section 5.1, and then use this problem in Section 5.2 to prove our main result.

5.1 Delaunay subgraph 3-colorability

In this section, we show that DS3C is NP-complete using a reduction from 3SAT. We first describe four graphs used in the reduction and state some of their basic properties. We then explain how the instance G^* of DS3C corresponding to an instance I of 3SAT is constructed, and show that the construction can be carried out in polynomial time. Finally, we prove that G^* admits a proper 3-coloring if and only if I is satisfiable, and that G^* is a subgraph of the Delaunay triangulation of its vertex set.

5.1.1 Graphs used in the construction

The construction of G^* uses several graphs taken from various sources that we shall describe in order of increasing complexity. The coordinates of each vertex are given relative to the placement of the graph's reference point. The vertices of a graph H that are labeled in, in_1 or in_2 will be called the *input* vertices of H, and the vertices



vertex	x	y
in	0	0
v_1	-1	4
v_2	2	4
out	0	8

Figure 5.1: Graph G_{copy} used to copy a color

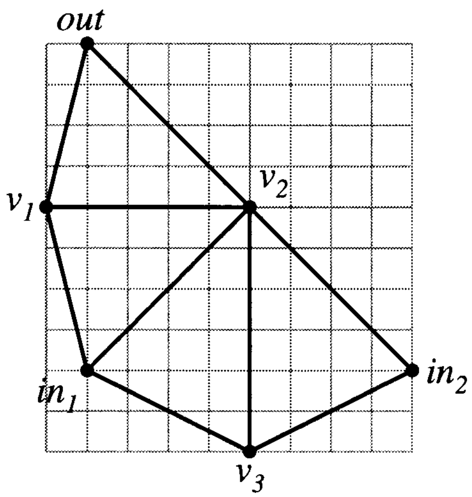
labeled *out*, out_1 or out_2 will be called the *output* vertices of H. The first graph used, shown in Figure 5.1, can be thought of as *copying* the color of its input vertex. This is formalized in the first lemma of this section.

Lemma 5.1 *Given any color c, the graph G_{copy} shown in Figure 5.1 admits a proper 3-coloring χ_c in which $\chi_c(in) = \chi_c(out) = c$. Moreover, every proper 3-coloring χ of G_{copy} satisfies $\chi(in) = \chi(out)$.*

The graph G_{merge}, shown in Figure 5.2, is used to *merge* the colors of its two input vertices. In other terms, in every proper 3-coloring of G_{merge}, both input vertices and the output vertex are assigned the same color.

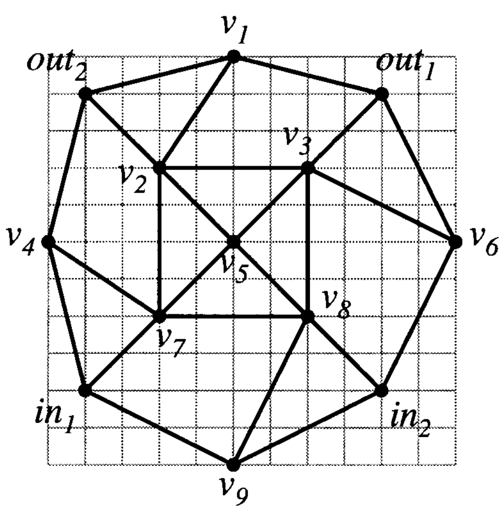
Lemma 5.2 *Given any color c, the graph G_{merge} shown in Figure 5.2 admits a proper 3-coloring χ_c in which $\chi_c(in_1) = \chi_c(in_2) = \chi_c(out) = c$. Moreover every proper 3-coloring χ of G_{merge} satisfies $\chi(in_1) = \chi(in_2) = \chi(out)$.*

The third graph, shown in figure 5.3, *exchanges* the color of its two inputs vertices. It was suggested by Fischer, and can be used to prove that planar graph 3-colorability is NP-complete [50]. Its main properties are summarized in the following lemma.



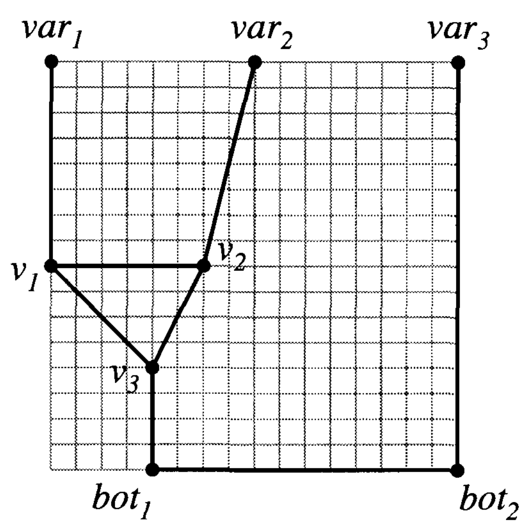
vertex	x	y
in_1	0	0
in_2	8	0
out	0	8
v_1	-1	4
v_2	4	4
v_3	4	-2

Figure 5.2: Graph G_{merge} used to merge two colors



vertex	x	y
in_1	0	0
in_2	8	0
out_1	8	8
out_2	0	8
v_1	4	9
v_2	2	6
v_3	6	6
v_4	-1	4
v_5	4	4
v_6	10	4
v_7	2	2
v_8	6	2
v_9	4	-2

Figure 5.3: Graph G_{exch} used to exchange two colors



vertex	x	y
bot_1	4	0
bot_2	16	0
v_1	0	8
v_2	6	8
v_3	4	4
var_1	0	16
var_2	8	16
var_3	16	16

Figure 5.4: Graph G_{clause} used for each clause

Lemma 5.3 *Given any two not necessarily distinct colors c_1 and c_2, the graph G_{exch} shown in Figure 5.3 admits a proper 3-coloring χ_{c_1,c_2} in which $\chi_{c_1,c_2}(in_1) = \chi_{c_1,c_2}(out_1) = c_1$ and $\chi_{c_1,c_2}(in_2) = \chi_{c_1,c_2}(out_2) = c_2$. Moreover, every proper 3-coloring χ of G_{exch} satisfies $\chi(in_1) = \chi(out_1)$ and $\chi(in_2) = \chi(out_2)$.*

The last graph used is shown in Figure 5.4. This graph can be found in the book by Cormen et al. [36] (exercise 36–2) and will represent the clauses of the 3SAT instance.

Lemma 5.4 *Let G_{clause} be the graph shown in Figure 5.4, and let χ be a 3-coloring of var_1, var_2 and var_3 in which each of them is colored 1 or 2. The function χ can be extended to a proper 3-coloring χ^* of G_{clause} in which $\chi^*(bot_1) \neq 1$ and $\chi^*(bot_2) \neq 1$ if and only if there exists $i \in \{1, 2, 3\}$ such that $\chi(var_i) = 1$.*

5.1.2 Constructing an instance of DS3C

We now describe the reduction used to transform an instance I of 3SAT into an instance G^* of DS3C. We will assume that I contains at least two distinct literals, since otherwise it is trivially satisfiable. Let $\prec$ be the total order on the literals of I in

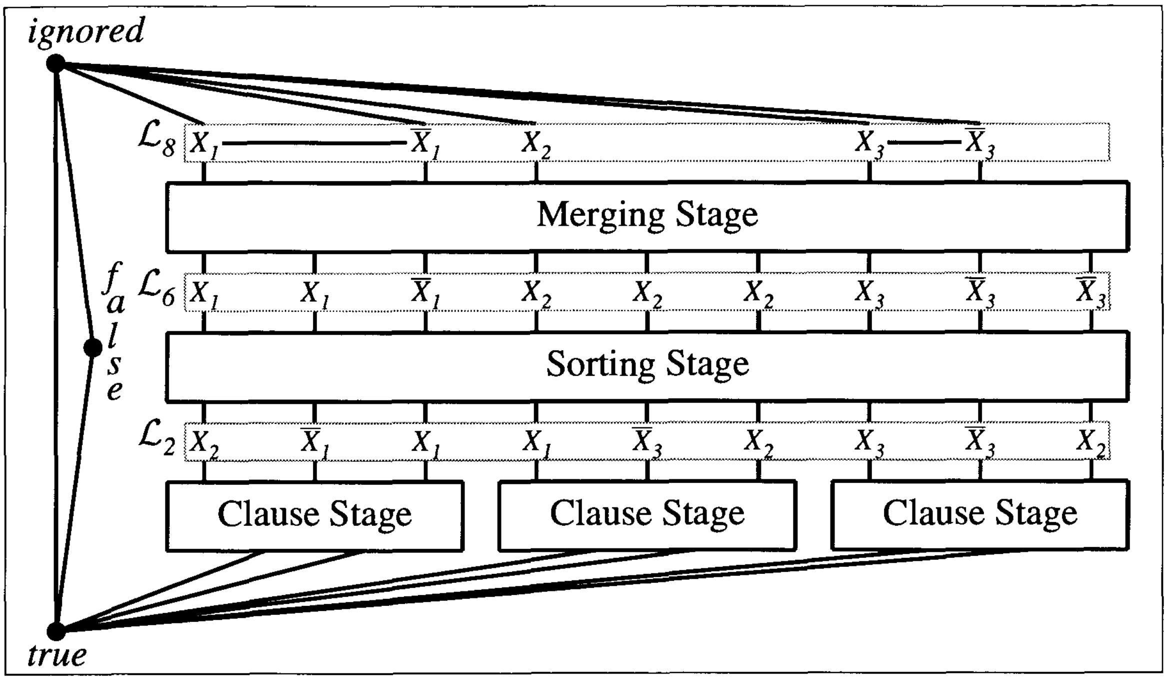


Figure 5.5: The reduction for $I = (X_2 \vee \overline{X}_1 \vee X_1) \wedge (X_1 \vee \overline{X}_3 \vee X_2) \wedge (X_3 \vee \overline{X}_3 \vee X_2)$

which $X_i \prec \overline{X}_i \prec X_{i+1}$ for each i. The construction of the graph G^* proceeds in four stages. The main data structure used by these four stages is a sequence $\mathcal{L}_1, \ldots, \mathcal{L}_m$ of lists of vertices. Each element of one of these lists will be labeled using a literal of I. The i^{th} element of $\mathcal{L}_j$ will be denoted by $v_{j,i}$.

Figure 5.5 illustrates the overall structure of G^*. The labels v_{true}, v_{false} and v_{none} indicate the truth values that will be associated with the colors assigned to these three vertices. The stages can be summarily described as follows:

Clause stage: We first construct the list $\mathcal{L}_1$; the label assigned to $v_{1,i}$ is the i^{th} literal occurring in I. We then construct $\mathcal{L}_2$ from $\mathcal{L}_1$; vertices $v_{1,i}$ and $v_{2,i}$ have the same label and are joined by an instance of G_{copy}. This stage ensures that each clause contains a *true* literal, but the truth values assigned to a given literal need not be consistent.

Sorting stage: We produce a list $\mathcal{L}_x$ with the same labels as $\mathcal{L}_2$, but in which the labels are sorted according to $\prec$. This stage proceeds in several steps, and each step consumes a list $\mathcal{L}_j$ and produces the list $\mathcal{L}_{j+1}$.

Merging stage: From $\mathcal{L}_x$, we construct a list $\mathcal{L}_m$ in which duplicate labels have been removed. This stage may also require several steps. It enforces consistency between the truth values assigned to a same literal.

Clean-up stage: Elements of $\mathcal{L}_m$ whose labels complement each other are joined by an edge to ensure that the truth values assigned to a literal and its complement are compatible.

In each of the first three stages, instances of the four graphs G_{copy}, G_{merge}, G_{exch} and G_{clause} are added to the part of G^* that already exists. Each of these instances will be called a *component* of G^*. We recall that if H is a component of G^*, and G is a graph, then we write $H \approx G$ to indicate that H is isomorphic to G. We will denote the size of $\mathcal{L}_j$ by $|\mathcal{L}_j|$, the x-coordinate of a vertex v by $v.x$, the y-coordinate of v by $v.y$, and the label of v by $v.label$.

The clause stage

The clause stage first places vertex v_{true} at the origin. It then considers the clauses of I in order; one instance of G_{clause} is added to G^* for each of them. The reference point of the i^{th} instance of G_{clause} is the point $(24i - 16, 4)$, and edges from bot_1 and bot_2 to v_{true} are added to G^*. Finally the vertices var_1, var_2, and var_3 are added to $\mathcal{L}_1$ in this order, and labeled with the first, second and third literal in the i^{th} clause of I respectively.

The list $\mathcal{L}_2$ is obtained by considering the elements of $\mathcal{L}_1$ in order. For each vertex $v_{1,i}$ of $\mathcal{L}_1$, we add an instance of G_{copy} to G^*, identify its reference point *in* with $v_{1,i}$, label *out* using $v_{1,i}.label$, and insert *out* at the end of $\mathcal{L}_2$.

The sorting stage

The sorting stage mimics the even-odd transposition parallel sorting algorithm used for linear arrays of processors [3,4]. This algorithm proceeds in several steps. During the odd-numbered steps, the value stored at each odd-numbered processor is compared

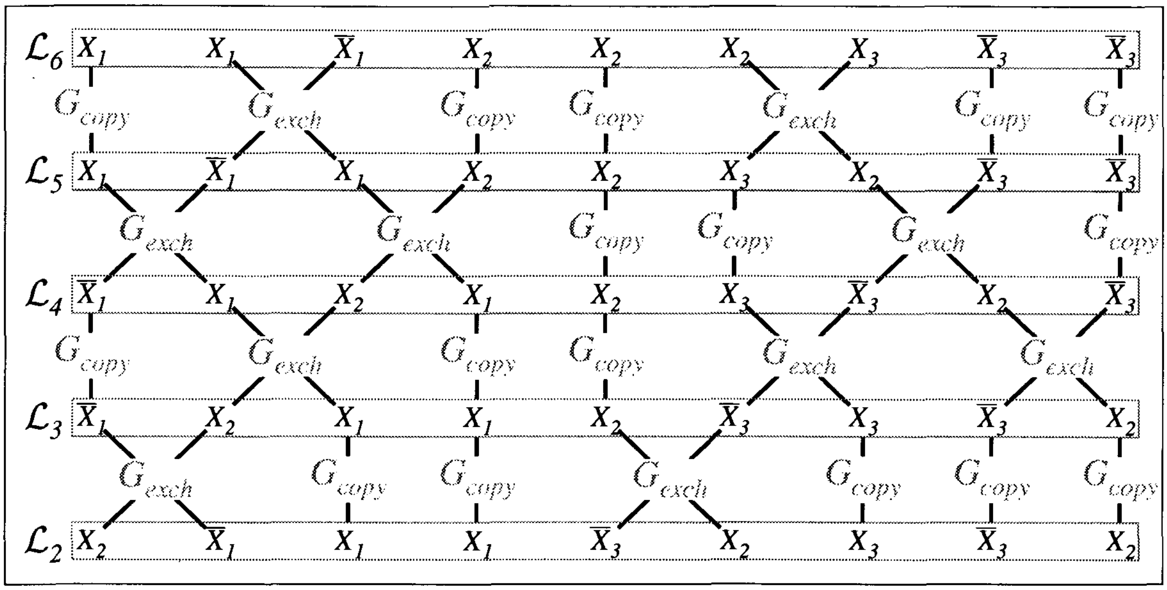


Figure 5.6: The sorting stage for $I = (X_2 \vee \overline{X}_1 \vee X_1) \wedge (X_1 \vee \overline{X}_3 \vee X_2) \wedge (X_3 \vee \overline{X}_3 \vee X_2)$

with that stored at its successor, and two compared values are exchanged if they are out of order. In the even-numbered steps, the value stored at each even-numbered processor is compared with that stored at its successor, and two compared values are exchanged if they are out of order.

The labels assigned to the elements of $\mathcal{L}_j$ correspond to the contents of a linear array of processors initialized using $v_{2,1}.label, v_{2,2}.label, \ldots, v_{2,|\mathcal{L}_2|}.label$, and to which the first $j-2$ steps of the even-odd transposition sorting algorithm have been applied. We obtain $v_{j+1,i}$ using G_{exch} if $v_{j,i}.label$ was exchanged with a neighbor in the $(j-1)^{st}$ step of the sorting algorithm, and using G_{copy} otherwise. The part of G^* constructed during the sorting stage for the example of Figure 5.5 is shown in Figure 5.6; the components isomorphic to G_{exch} and G_{copy} have been replaced by their names to improve legibility.

Let us now describe the construction formally. During each step s, the elements of $\mathcal{L}_j$ are considered in order. It suffices to explain the manner in which $v_{j,i}$ is dealt with.

Case 1 ($i \equiv s \pmod 2$, $i < |\mathcal{L}_j|$ and $v_{j,i+1}.label \prec v_{j,i}.label$) : We add an instance of G_{exch} to G^*, identify in_1 with $v_{j,i}$ and in_2 with $v_{j,i+1}$, label out_2 using $v_{j,i+1}.label$ and out_1 using $v_{j,i}.label$, and insert out_2 and out_1 in this order at the end of $\mathcal{L}_{j+1}$.

Case 2 ($i \not\equiv s \pmod 2$, $i > 1$ and $v_{j,i}.label \prec v_{j,i-1}.label$) : Nothing needs to be done.

Case 3 (all other cases) : We add an instance of G_{copy} to G^*, identify its reference point with $v_{j,i}$, label out using $v_{j,i}.label$, and insert out at the end of $\mathcal{L}_{j+1}$.

The sorting stage ends as soon as the labels of $\mathcal{L}_j$ are sorted in increasing order according to $\prec$.

The merging stage

The merging stage is performed by repeatedly considering the elements of $\mathcal{L}_j$ in order, and removing the last instance of each duplicated literal on $\mathcal{L}_j$, constructing $\mathcal{L}_{j+1}$ in the process. It thus suffices to describe the manner in which $v_{j,i}$ is dealt with. This treatment can be broken down into three cases.

Case 1 ($v_{j,i}$ is the only element of $\mathcal{L}_j$ with label $v_{j,i}.label$, or at least two such elements come after it) : We add an instance of G_{copy} to G^*, identify its reference point with $v_{j,i}$, label out using $v_{j,i}.label$, and insert out at the end of $\mathcal{L}_{j+1}$.

Case 2 (exactly one element of $\mathcal{L}_j$ with label $v_{j,i}.label$ follows $v_{j,i}$) : We add an instance of G_{merge} to G^*, identify in_1 with $v_{j,i}$, identify in_2 with $v_{j,i+1}$, label out using $v_{j,i}.label$, and insert out at the end of $\mathcal{L}_{j+1}$.

Case 3 ($v_{j,i}$ is not the only element of $\mathcal{L}_j$ with label $v_{j,i}.label$, but it is the last one) : Nothing needs to be done.

The merging stage terminates as soon as no two elements of $\mathcal{L}_j$ have the same label. The part of G^* constructed during the merging stage for the example of Figure 5.5 is shown in Figure 5.7; the components isomorphic to G_{merge} and G_{copy} have been replaced by their names to improve legibility.

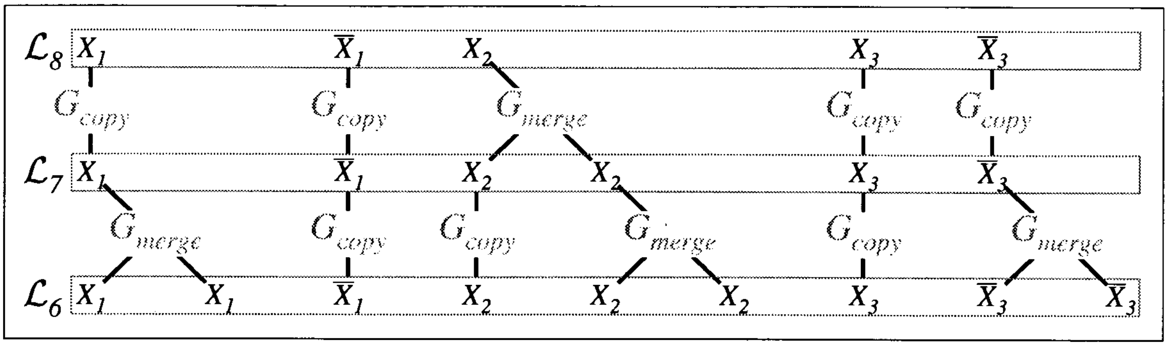


Figure 5.7: The merging stage for $I = (X_2 \vee \overline{X}_1 \vee X_1) \wedge (X_1 \vee \overline{X}_3 \vee X_2) \wedge (X_3 \vee \overline{X}_3 \vee X_2)$

The clean-up stage

The last stage of the construction places vertex v_{false} at $(0, v_{m,1}.y + 8)$ and vertex v_{none} at $(4, v_{m,1}.y/2 + 4)$. It then adds an edge from each of these two new vertices to v_{true}, and from v_{false} to v_{none}. Finally, it adds an edge from each element of $\mathcal{L}_m$ to v_{none}, and joins two consecutive elements of $\mathcal{L}_m$ if their labels complement each other.

5.1.3 Correctness of the transformation

We now show that the graph G^* obtained using the construction described in Section 5.1.2 is a subgraph of the Delaunay triangulation of its vertex set, that it is 3-colorable if and only if the instance I of 3SAT from which it was obtained admits a satisfying truth assignment, and that it can be constructed in polynomial time.

Properties of the embedding of G^*

Let us first prove certain properties of the embedding of G^* in the plane. These will be needed to show that it is a subgraph of the Delaunay triangulation of its vertex set. We start with three simple observations that follow directly from the manner in which the construction of G^* proceeds:

Observation 5.1 *If v belongs to $\mathcal{L}_j$ for some $j > 1$, then there is a vertex v' of $\mathcal{L}_{j-1}$ such that $v'.x = v.x$ and $v'.y = v.y - 8$.*

Observation 5.2 *If H is a component of G^* that is not isomorphic to G_{clause}, and (x, y) is the reference point of H, then there is a positive integer j and an element v of $\mathcal{L}_j$ such that $v.x = x$ and $v.y = y$.*

Observation 5.3 *If H is a component of G^* isomorphic to G_{merge} whose vertex in_2 belongs to $\mathcal{L}_j$, then no element of $\mathcal{L}_{j+1}$ is located at $(in_2.x, in_2.y + 8)$.*

These observations lead to a characterization of the coordinates of the reference points of the components of G^*. We first characterize the coordinates of the elements of $\mathcal{L}_j$.

Lemma 5.5 *If v belongs to $\mathcal{L}_j$, then $v.y = 8(j-1) + 20$ and there is a non-negative integer k such that $v.x = 8k + 8$.*

Proof: The proof proceeds by induction on j. When $j = 1$, the vertex v is a vertex of a component H isomorphic to G_{clause}, and is labeled var_1, var_2 or var_3. Since the reference point of H is of the form $(24i - 16, 4)$ for some positive integer i, it follows that $v.y = 8 \cdot 0 + 20$. Moreover $v.x \in \{24i - 16, 24i - 8, 24i\}$, and hence some k in $\{3i - 3, 3i - 2, 3i - 1\}$ satisfies the statement of the lemma.

Consider now the case $j > 1$. By observation 5.1 there is an element v' of $\mathcal{L}_j - 1$ such that $v'.x = v.x$ and $v'.y = v.y - 8$. The induction hypothesis thus implies that $v'.y = 8(j-1)+20$, and that there is a non-negative integer k, such that $v'.x = 8k+8$. The result of the lemma then follows from the principle of mathematical induction.□

Lemma 5.6 *Let H be a component of G^* with reference point (x, y). There are non-negative integers k, l such that*

- *if $H \approx G_{clause}$, then $x = 24k + 8$ and $y = 4$;*
- *if $H \approx G_{copy}$, then $x = 8k + 8$ and $y = 8l + 20$;*
- *if $H \approx G_{merge}$ or $H \approx G_{exch}$, then $x = 8k + 8$ and $y = 8l + 28$.*

Edges	Circle	Edges	Circle
$\overline{v_{m,1}v_{none}}$	$C_{v_{m,1},v_{m,2},v_{none}}$	$\overline{v_{m,i}v_{none}}$ for $i > 1$	$C_{v_{m,i},v_{m,i-1},v_{none}}$
$\overline{v_{true}v_{false}}$, $\overline{v_{true}v_{none}}$ $\overline{v_{false}v_{none}}$	$C_{v_{true},v_{false},v_{none}}$	$\overline{v_{true}bot_1}$, $\overline{v_{true}bot_2}$	C_{v_{true},bot_1,bot_2}

Table 5.1: Some of the circles used to cover the edges of G^*.

Proof: The case in which $H \approx G_{clause}$ follows directly from the clause stage. Hence let us suppose that $H \not\approx G_{clause}$. By observation 5.2, there is a positive integer j and a vertex v of $\mathcal{L}_j$ such that $v.x = x$ and $v.y = y$. It thus follows from Lemma 5.5 that $y = 8(j-1) + 20$, and that there is a non-negative integer k such that $x = 8k + 8$. If $H \approx G_{copy}$, then $l = j - 1$ satisfies the statement of the lemma. Otherwise $j > 1$, and hence $l = j - 2$ works. $\square$

G* is a Delaunay Subgraph

Let e be an edge of G^*. A circle C *covers* e if both endpoints of e belong to C, and *properly covers* e if it covers e, if exactly 3 vertices of G^* belong to C, and if no vertex of G^* belongs to the interior of C. A set S of circles *covers* a subgraph H of G^* if every edge of H is properly covered by at least one element of S. We will denote by $C_{x,y,z}$ the circle that goes through the points x, y and z.

To prove that G^* is a subgraph of the Delaunay triangulation of its vertex set, it suffices to exhibit a set of circles that covers G^*. We thus proceed by assigning one circle to each edge, and proving that each circle properly covers the edges to which it is assigned. The circles assigned to the edges of the components are shown in Figure 5.8, and those assigned to the remaining edges of G^* are specified in Table 5.1. Let us first consider the circles assigned to component edges. We will denote by $\mathcal{C}_H$ the union of the circles assigned to the edges of a component H of G^*.

Lemma 5.7 *Let I be an instance of 3SAT, G^* be the graph obtained from I by applying the construction described in Section 5.1.2, and G be the Delaunay triangulation of the set of vertices of G^*. If H_0 is a component of G^*, and v is a vertex of G^* that does not belong to H_0, then $v \notin \mathcal{C}_{H_0}$.*

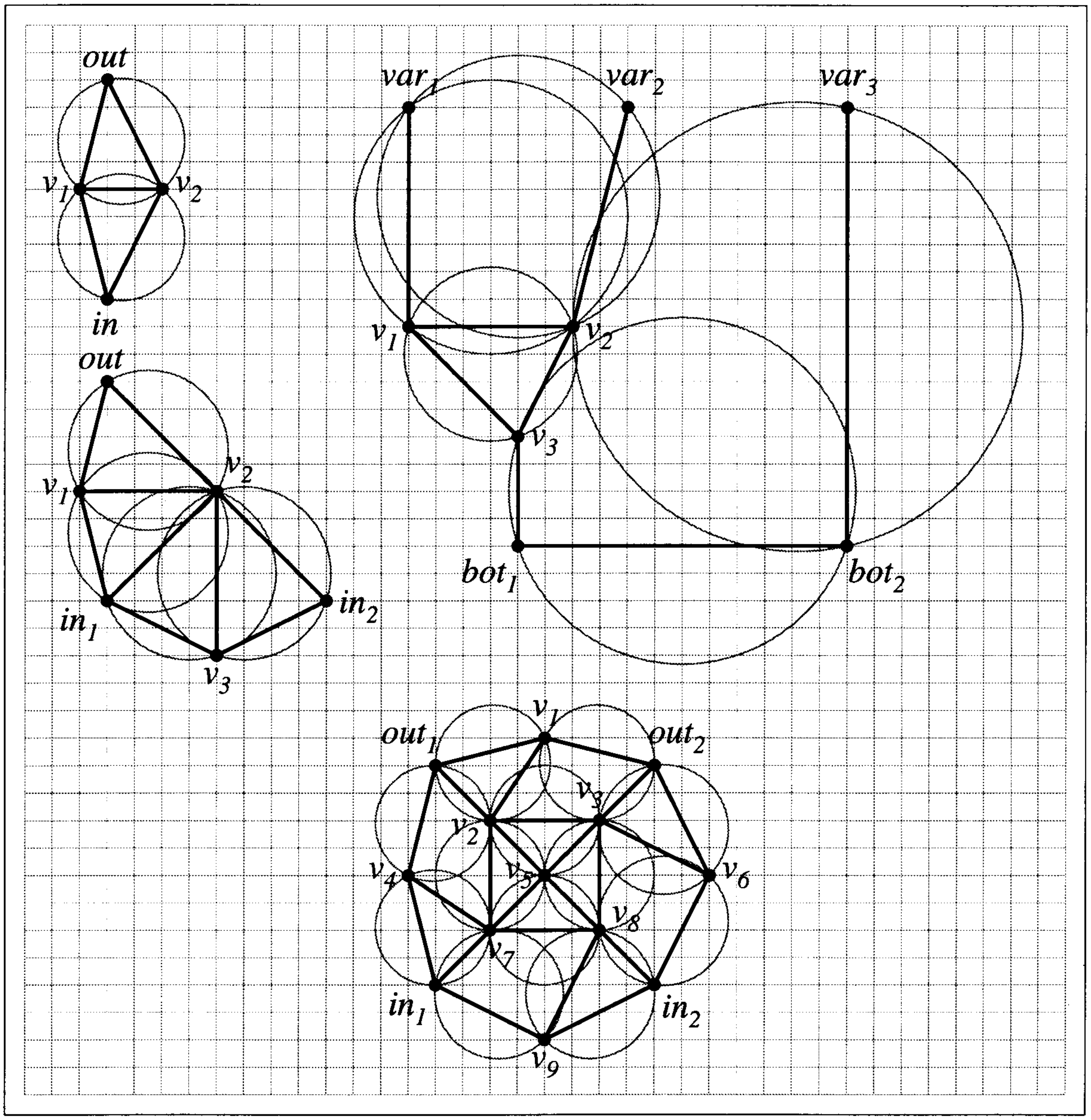


Figure 5.8: Circles assigned to the edges of the components of G^*.

Proof: Given a subset S of the plane, let $Rect(S)$ denote the smallest isothetic rectangle containing S with vertices on the integer lattice. The following table gives $Rect(H)$ and $Rect(\mathcal{C}_H)$ for a component H of G^* whose reference point (x,y) belongs to the integer lattice.

	$Rect(H)$	$Rect(\mathcal{C}_H)$
G_{copy}	$[x-1, x+2] \times [y, y+8]$	$[x-2, x+3] \times [y-1, y+9]$
G_{merge}	$[x-1, x+8] \times [y-2, y+8]$	$[x-2, x+9] \times [y-3, y+9]$
G_{exch}	$[x-1, x+10] \times [y-2, y+9]$	$[x-3, x+11] \times [y-3, y+11]$
G_{clause}	$[x, x+16] \times [y, y+16]$	$[x-2, x+23] \times [y-5, y+18]$

Let (x_0, y_0) be the reference point of H_0. First suppose that $v \in \{v_{true}, v_{false}, v_{none}\}$. Since $x_0 \geq 8$ by Lemma 5.6, the x-coordinate of the lower-left corner of $Rect(\mathcal{C}_{H_0})$ is at least 5. Since the x-coordinate of v is either 0 or 4, it follows that $v \notin \mathcal{C}_{H_0}$. If $v \notin \{v_{true}, v_{false}, v_{none}\}$, then v belongs to a component H of G^* with reference point (x,y). If $Rect(H) \cap Rect(\mathcal{C}_{H_0}) = \emptyset$, then $v \notin Rect(\mathcal{C}_H)$. Hence Lemma 5.6 implies that the only values of (x,y) left to consider (relative to (x_0, y_0)) are those shown in Table 5.2.

Since the sorting stage terminates before the merging stage begins, the cases marked with ^{1} do not occur. Since no element of $\mathcal{L}_j$ is identified with an input vertex of two distinct components, the cases marked with ^{2} never happen either. Observations 5.3 prevents the cases marked with ^{3} from occurring as well. Since two elements are never exchanged in two consecutive steps of the even-odd transposition parallel sort, the cases marked with ^{4} also do not happen. Finally, since the output vertices of a component isomorphic to G_{exch} are labeled differently, they will not appear as the two input vertices of a component isomorphic to G_{merge}, and hence the cases marked with ^{5} do not occur. It can be checked easily that, for the remaining entries of Table 5.2, no vertex of H belongs to $\mathcal{C}_{H_0}$. $\square$

Lemma 5.8 *Let I be an instance of 3SAT, and G^* be the graph obtained from I by applying the construction described in Section 5.1.2. The set of edges of G^* is a*

H_0 \ H	G_{copy}	G_{merge}	G_{exch}	G_{clause}
G_{copy}	(x_0, y_0-8) (x_0, y_0+8)	$(x_0-8, y_0-8)^3$ $(x_0-8, y_0)^2$ (x_0-8, y_0+8) (x_0, y_0-8) (x_0, y_0+8)	(x_0-8, y_0-8) $(x_0-8, y_0)^2$ (x_0-8, y_0+8) (x_0, y_0-8) (x_0, y_0+8)	(x_0-24, y_0-16) (x_0-16, y_0-16) (x_0-8, y_0-16) (x_0, y_0-16)
G_{merge}	(x_0, y_0-8) (x_0, y_0+8) (x_0+8, y_0-8) $(x_0+8, y_0)^2$ $(x_0+8, y_0+8)^3$	$(x_0-8, y_0-8)^3$ $(x_0-8, y_0)^2$ (x_0-8, y_0+8) $(x_0, y_0-8)^3$ $(x_0, y_0+8)^3$ (x_0+8, y_0-8) $(x_0+8, y_0)^2$ $(x_0+8, y_0+8)^3$	(x_0-8, y_0-8) $(x_0-8, y_0)^1$ $(x_0-8, y_0+8)^1$ $(x_0, y_0-8)^5$ $(x_0, y_0+8)^1$ (x_0+8, y_0-8) $(x_0+8, y_0)^1$ $(x_0+8, y_0+8)^1$	
G_{exch}	(x_0, y_0-8) (x_0, y_0+8) (x_0+8, y_0-8) $(x_0+8, y_0)^2$ (x_0+8, y_0+8)	$(x_0-8, y_0-8)^1$ $(x_0-8, y_0)^1$ (x_0-8, y_0+8) $(x_0, y_0-8)^1$ $(x_0, y_0+8)^5$ $(x_0+8, y_0-8)^1$ $(x_0+8, y_0)^1$ (x_0+8, y_0+8)	(x_0-8, y_0-8) $(x_0-8, y_0)^2$ (x_0-8, y_0+8) $(x_0, y_0-8)^4$ $(x_0, y_0+8)^4$ (x_0+8, y_0-8) $(x_0+8, y_0)^2$ (x_0+8, y_0+8)	
G_{clause}	(x_0, y_0+16) (x_0+8, y_0+16) (x_0+16, y_0+16) (x_0+24, y_0+16)			(x_0-24, y_0) (x_0+24, y_0)

Table 5.2: Pairs (x, y) for which $Rect(H) \cap Rect(\mathcal{C}_{H_0}) \neq \emptyset$.

subgraph of the Delaunay triangulation of the set of vertices of G^*.

Proof: It suffices to show that every circle C properly covers the edges to which it is assigned. We divide our analysis into several cases.

Case 1 ($C = C_{v_{true},v_{false},v_{none}}$) : Since every point of C has x-coordinate less than 5, and since Lemma 5.6 implies that every vertex of G^* apart from v_{true}, v_{false}, and v_{none} has x-coordinate greater than or equal to 8, C properly covers $\overline{v_{true}v_{false}}$, $\overline{v_{true}v_{none}}$ and $\overline{v_{false}v_{none}}$.

Case 2 ($C = C_{v_{true},bot_1,bot_2}$) : Let H be the component isomorphic to G_{clause} to which bot_1 and bot_2 belong, and let (x, y) be the reference point of H. Lemma 5.6 implies that $x \geq 8$, and hence no point of C has y-coordinate greater than 5. Moreover v_{true} is the only vertex of G^* with y-coordinate less than 4 by Lemma 5.6, and $C \cap \{ x \mid x = 4\} = \overline{bot_1 bot_2}$. Hence C properly covers $\overline{v_{true} bot_1}$ and $\overline{v_{true} bot_2}$.

Case 3 ($C = C_{v_{none},v_{m,i},v_{m,i+1}}$) : Let $x_1 = v_{m,i}.x$, let $x_2 = v_{m,i+1}.x$, and let $y_1 = v_{m,i}.y$. The center of C is the point $((x_1 + x_2)/2, x_1x_2/16 + y_1 + 4)$. Moreover, the slope s of the tangent to C at $v_{m,i}$ is $(8x_1 - 8x_2)/(x_1x_2 + 64)$. Since $x_1 < x_2$, it follows that $s \leq 0$. Also, $s = (-1+(x_1-8)(x_2-8))/(x_1x_2+64)$, and since $x_1 \geq 8$ and $x_2 \geq 8$ it follows that $s \geq -1$.

Consider a vertex $v = (x, y)$ of G^* distinct from $v_{m,i}$, $v_{m,i+1}$ and v_{none}. If $x < x_1$ or $x > x_2$, then since $y < y_1$ it follows that $v \notin C$. Otherwise, let $k = (x_2 - x_1)/8$ and let $v' = (x_1 + 4k, y_1 - 4k)$. There must have been k copies of the literal $v_{m,i}.label$ in I. Hence the triangle $v_{m,i}v'v_{m,i+1}$ is empty, and does not contain v. However, since $0 \leq s \leq 1$, this implies that $v \notin C$. Therefore C properly covers $\overline{v_{none}v_{m,i}}$ and $\overline{v_{none}v_{m,i+1}}$.

Case 4 ($C = C_{a,b,c}$ where a, b and c belong to a component H of G^*) : A simple case analysis (or a look at Figure 5.8) shows that a, b and c are the only three vertices of H that do not belong to the exterior of C. Moreover, Lemma 5.7 implies that every vertex of G^* that does not belong to H also lies in the exterior of C. Therefore C properly covers $\overline{ab}$, $\overline{ac}$ and $\overline{bc}$.

This concludes the case analysis and the proof of the lemma. $\square$

Remainder of the proof

Let us now show that every graph constructed from a satisfiable instance of 3SAT is 3-colorable.

Lemma 5.9 *Let I be an instance of 3SAT, and G^* be the graph obtained from I be applying the construction described in Section 5.1.2. If I is satisfiable, then G^* is 3-colorable.*

Proof: Let U be the set of variables of I, let $V = \cup_{j=1}^{m} \mathcal{L}_j$, and let $t : U \rightarrow \{true, false\}$ be a truth assignment satisfying I. Finally, let χ be the coloring of V that assigns the color 1 to every element of V whose label is a literal that is set to *true* by t, and the color 2 to every other element of V.

We need to show how to extend χ to a proper 3-coloring of G^*. We first color the vertices v_{true}, v_{false} and v_{none} of G^* using colors 1, 2 and 3 respectively. Next, we apply Lemma 5.4 to extend χ to a proper 3-coloring of each component isomorphic to G_{clause} in which $\chi(bot_1)$ and $\chi(bot_2)$ are distinct from 1. It can easily be seen that the endpoints of each edge of G^* not belonging to a component have now been colored using different colors. Finally, we can apply Lemmas 5.1 to 5.3 to properly 3-color the remainder of G^*. $\square$

The next lemma shows that in every proper 3-coloring of G^*, vertices labeled using the same literal are assigned the same color. This fact will be required in proving the converse of Lemma 5.9.

Lemma 5.10 *Let I be an instance of 3SAT, and G^* be the graph obtained from I by applying the construction described in Section 5.1.2. If χ is a proper 3-coloring of G^* and two vertices v and v' of G^* are labeled using the same literal, then $\chi(v) = \chi(v')$.*

Proof: We first claim that for every vertex $v_{j,i}$ of $\mathcal{L}_j$, there is a unique vertex v of $\mathcal{L}_m$ labeled $v_{j,i}.label$ and that moreover $\chi(v) = \chi(v_{j,i})$. This claim can be proved by induction on j. Since no two vertices of $\mathcal{L}_m$ have the same label, the base case follows.

Consider now a vertex $v_{j,i}$ of $\mathcal{L}_j$ for some $j < m$. This vertex may be connected to vertices of $\mathcal{L}_{j+1}$ using one of three types of component. Lemmas 5.1 to 5.3 imply that in each case, there is a vertex v of $\mathcal{L}_{j+1}$ labeled $v_{j,i}.label$ such that $\chi(v) = \chi(v_{j,i})$. The claim thus holds for $v_{j,i}$ by the induction hypothesis.

Since every labeled vertex of G^* belongs to $\mathcal{L}_j$ for some j in $\{1, \ldots, m\}$, this implies that if two vertices v and v' of G^* have the same label, then $\chi(v) = \chi(v')$. □

We now use the equivalence relation defined on the vertices of G^* by Lemma 5.10 to prove the converse of Lemma 5.9.

Lemma 5.11 *Let I be an instance of 3SAT, and G^* be the graph obtained from I by applying the construction described in Section 5.1.2. If G^* is 3-colorable, then I is satisfiable.*

Proof: Let χ be a proper 3-coloring of G^*, and let U be the set of literals of I. We define a truth assignment $t : U \to \{true, false\}$ as follows:

- If $x \in U$, and there exists an element v of $\mathcal{L}_m$ whose label is x and such that $\chi(v) = \chi(v_{true})$, then $t(x) = true$.
- If $x \in U$, and there exists an element v of $\mathcal{L}_m$ whose label is x and such that $\chi(v) = \chi(v_{false})$, then $t(x) = false$.

Since every element of $\mathcal{L}_m$ is adjacent to v_{none}, each one is colored using one of $\chi(v_{true})$ and $\chi(v_{false})$. Moreover every literal of I appears exactly once as the label of an element of $\mathcal{L}_m$, and hence $t(x)$ is well-defined. Finally, if u is a variable, and both u and $\overline{u}$ are literals that appear in I, then the corresponding elements of $\mathcal{L}_m$ are adjacent, and hence t is consistent. We will assume without loss of generality that $\chi(v_{true}) = 1$ and that $\chi(v_{false}) = 2$.

Consider the component H isomorphic to G_{clause} that corresponds to a clause c of I. Both bot_1 and bot_2 are adjacent to v_{true}, and so $\chi(bot_1) \neq 1$ and $\chi(bot_2) \neq 1$. Since each of var_1, var_2 and var_3 is labeled using a literal of I, Lemma 5.10 implies that each one of $\chi(var_1)$, $\chi(var_2)$ and $\chi(var_3)$ belongs to $\{1, 2\}$. Since χ induces a proper 3-coloring of H, it therefore follows from Lemma 5.4 that there exists $i \in \{1, 2, 3\}$

such that $\chi(var_i) = 1$. This implies that the corresponding literal of c is set to *true* by t. Since this holds for every clause of I, it follows that I is satisfiable. □

We conclude the proof of the correctness of our transformation by showing that it can be performed in polynomial time.

Lemma 5.12 *Given an instance I of 3SAT with n literals, the graph G^* can be constructed in $O(n^2 \log n)$ time and space.*

Proof: Since the largest x-coordinate required during the clause stage (and all subsequent stages) is $24n$, the clause stage can be performed in $O(n \log n)$ time and space using a single pass through I. The sorting stage requires no more than $O(n)$ steps [3, 4], and hence the largest y-coordinate of a vertex generated in this stage requires no more than $O(\log n)$ bits. Since each element of $\mathcal{L}_j$ can be visited in $O(\log n)$ time in each step, and since $|\mathcal{L}_j| \leq n$, the sorting stage therefore requires at most $O(n^2 \log n)$ time and space. Since each literal occurs at most n times in I, the merging step terminates after at most $O(n)$ steps. Hence no y-coordinate requires more than $O(\log n)$ bits to specify, and so each element of $\mathcal{L}_j$ can be visited in $O(\log n)$ time in each step. Therefore the merging stage needs at most $O(n^2 \log n)$ time and space. Finally, the clean-up stage requires at most $O(|\mathcal{L}_m|) \subseteq O(n)$ time. □

Since we can verify whether a given 3-coloring of a graph G is proper in time linear in the size of G, the results of Lemmas 5.8, 5.9, 5.11 and 5.12 can be summarized into the following theorem.

Theorem 5.1 *Delaunay subgraph 3-colorability is NP-complete.*

5.2 U_3 polytopes recognition

In this section, we use a reduction from DS3C to prove that the problem of determining whether a starshaped polytope is U_3 or B_3 is NP-hard. The transformation constructs a polytope P^* whose vertex visibility graph contains the complement of the graph G^* in the instance of DS3C as an induced subgraph. Visibility between the vertices of P^*

that correspond to adjacent vertices of G^* is blocked by removing pieces called *wedges* from the convex hull of P^*. In Section 5.2.1, we construct the planes and the vectors that will be used by the transformation. In Section 5.2.2 we show how wedges are constructed, and derive properties of their bounding planes. Finally, in Section 5.2.3 we describe the transformation and prove its correctness.

5.2.1 Geometric preliminaries

Suppose that we are given two concurrent planes whose normal vectors have rational coordinates (their lengths may differ). We will need to obtain vectors with rational coordinates that point into each of the four quadrants determined by these two planes. This is done as follows.

Lemma 5.13 *Let H_1 and H_2 be two planes with linearly independent normal vectors $\vec{n}_1$ and $\vec{n}_2$ respectively. If p belong to $H_1^{\geq} \cap H_2^{\geq}$ then $p + (\vec{n}_1 - \vec{n}_2) \times (\vec{n}_1 \times \vec{n}_2)$ belongs to $H_1^{>} \cap H_2^{>}$.*

Proof: Let $\vec{u} = \vec{n}_1 \times \vec{n}_2$, let $\vec{u}_1 = \vec{n}_1 \times \vec{u}$ and let $\vec{u}_2 = \vec{n}_2 \times \vec{u}$. We first establish the relation between $\vec{u}_1 \cdot \vec{u}_2$ and $\vec{n}_1 \cdot \vec{n}_2$. By expanding $(\vec{n}_1 \times \vec{u}) \cdot (\vec{n}_2 \times \vec{u})$ and using the fact that $\vec{u} \cdot \vec{n}_1 = 0$, we get

$$\vec{u}_1 \cdot \vec{u}_2 = (\vec{n}_1 \cdot \vec{n}_2)|\vec{u}|^2. \tag{5.1}$$

Let us now show how use equation 5.1 to express $\vec{n}_2$ as a linear combination of $\vec{u}_1$ and $\vec{n}_1$. By considering $\vec{u}_1 \times \vec{u}$, we find that $\vec{n}_1 = (\vec{u} \times \vec{u}_1)/|\vec{u}|^2$. Similarly, $\vec{u} = (\vec{u}_1 \times \vec{n}_1)/|\vec{n}_1|^2$. By combining these two equations, we obtain

$$\vec{n}_2 = \left(\frac{\vec{n}_1 \cdot \vec{n}_2}{|\vec{n}_1|^2}\right)\vec{n}_1 - \left(\frac{\vec{n}_1 \cdot \vec{u}_2}{|\vec{n}_1|^2|\vec{u}|^2}\right)\vec{u}_1. \tag{5.2}$$

By using the definitions of $\vec{u}$ and $\vec{u}_2$ and equation 5.2, we can also express $\vec{u}_2$ in terms of $\vec{n}_1$ and $\vec{u}_1$:

$$\vec{u}_2 = \left(|\vec{n}_2|^2 - \frac{(\vec{n}_1 \cdot \vec{n}_2)^2}{|\vec{n}_1|^2}\right)\vec{n}_1 + \left(\frac{(\vec{n}_1 \cdot \vec{n}_2)(\vec{n}_1 \cdot \vec{u}_2)}{|\vec{n}_1|^2|\vec{u}|^2}\right)\vec{u}_1. \tag{5.3}$$

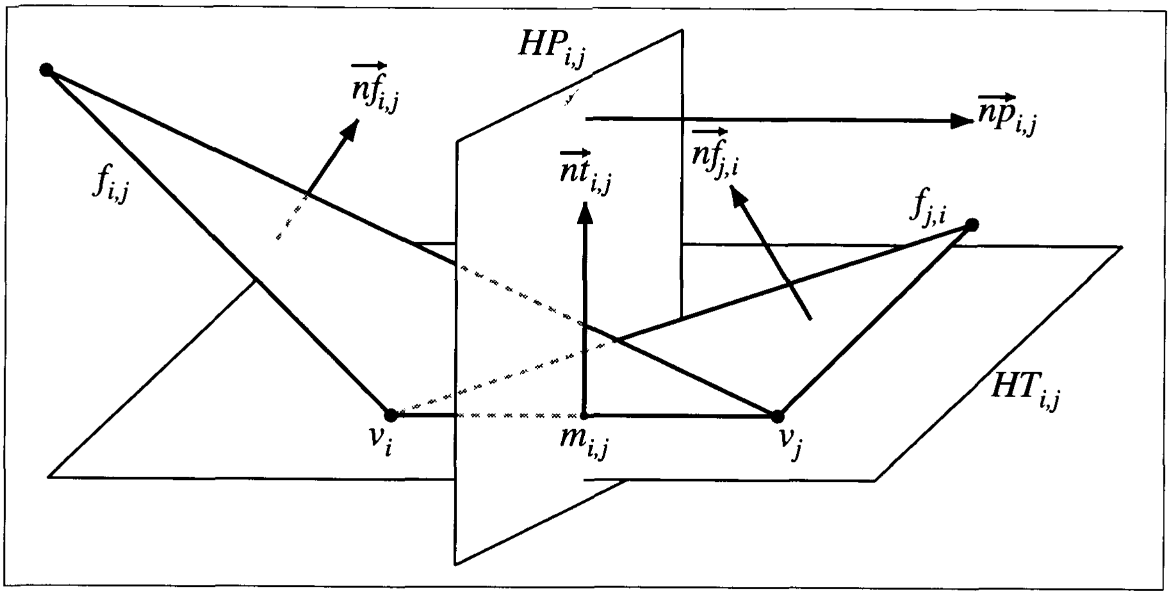


Figure 5.9: Illustrating $HP_{i,j}$ and $HT_{i,j}$.

Let $p' = p + (\vec{n}_1 - \vec{n}_2) \times \vec{u}$. Since $p' = p + \vec{u}_1 - \vec{u}_2$, it follows from equation 5.3 that $p' = p + \vec{u}_1 + (a\vec{n}_1 + b\vec{u}_1)$. Since $\vec{u}_1$ is parallel to H_1, and since the Cauchy-Schwartz inequality implies that $|\vec{n}_2|^2 - (\vec{n}_1 \cdot \vec{n}_2)^2 / |\vec{n}_1|^2 > 0$, it follows that $p' \in H_1^>$. A similar argument shows that $p' \in H_2^>$ and completes the proof of the lemma. ◫

Let us now consider the planes and the vectors that will be used to define wedges; they are shown in Figure 5.9. Let P be a convex polytope with vertex set $V = \{v_1, \ldots, v_n\}$. The edge of P whose endpoints are v_i and v_j will be denoted by $e_{i,j}$, and its midpoint by $m_{i,j}$. There is a unique facet of P in a counterclockwise traversal of whose boundary v_j immediately follows v_i, and is immediately followed by a vertex v_k of P. This facet will be denoted by $f_{i,j}$, and the plane that contains it by $HF_{i,j}$. We observe that the vector $\overrightarrow{nf}_{i,j} = (\vec{v}_k - \vec{v}_i) \times (\vec{v}_j - \vec{v}_i)$ is normal to $HF_{i,j}$ and points towards the interior of P.

Two planes that contain $m_{i,j}$ will also be used: a plane $HT_{i,j}$ tangent to P along $e_{i,j}$, and the plane $HP_{i,j}$ whose projection on $HT_{i,j}$ is the perpendicular bissector of $e_{i,j}$. The vector $\overrightarrow{np}_{i,j}$ normal to $HP_{i,j}$ is simply $\vec{v}_j - \vec{v}_i$. The following lemma shows that if we choose $\overrightarrow{nt}_{i,j} = \overrightarrow{nf}_{i,j} + \overrightarrow{nf}_{j,i}$, the plane $HT_{i,j}$ normal to $\overrightarrow{nt}_{i,j}$ and containing

$m_{i,j}$ is indeed tangent to P along $e_{i,j}$.

Lemma 5.14 *Let P be a convex polytope, and let v_i, v_j be two adjacent vertices of P. If $HT_{i,j}$ is the plane normal to $\overrightarrow{nf}_{i,j} + \overrightarrow{nf}_{j,i}$ and containing $m_{i,j}$, then $P \cap HT_{i,j} = e_{i,j}$ and $P \subseteq HT^{\geq}_{i,j}$.*

Proof: Let $\vec{n}_1 = \overrightarrow{nf}_{i,j} \times \overrightarrow{nf}_{j,i}$, and let $\vec{n}_2 = (\overrightarrow{nf}_{i,j} + \overrightarrow{nf}_{j,i}) \times \vec{n}_1$. Since $\vec{n}_1 \times \vec{n}_2 = |\vec{n}_1|^2 (\overrightarrow{nf}_{i,j} + \overrightarrow{nf}_{j,i})$, the plane $HT_{i,j}$ contains the two perpendicular vectors $\vec{n}_1$ and $\vec{n}_2$. Moreover Lemma 5.13 implies that for every $\alpha > 0$, and every point q of $HF_{i,j} \cap HF_{j,i}$, the point $q + \alpha\vec{n}_2$ belongs to $HF^{>}_{i,j} \cap HF^{<}_{j,i}$ and the point $q - \alpha\vec{n}_2$ belongs to $HF^{<}_{i,j} \cap HF^{>}_{j,i}$.

Hence consider a point p of $HT_{i,j}$. The point p can be written as $p = q + \alpha_2\vec{n}_2$, where $q = m_{i,j} + \alpha_1\vec{n}_1$. Since $\vec{n}_1$ is parallel to $e_{i,j}$, the point q belongs to $HF_{i,j} \cap HF_{j,i}$. If $\alpha_2 = 0$, then p belongs to P if and only if it belongs to $e_{i,j}$. If $\alpha_2 < 0$, then p belongs to $HF^{<}_{i,j}$, and hence does not belong to P. If $\alpha_2 > 0$, then p belongs to $HF^{<}_{j,i}$, and hence does not belong to P either.

Finally, for every point x of $f_{i,j}$, we have $x \cdot (\overrightarrow{nf}_{i,j} + \overrightarrow{nf}_{j,i}) = x \cdot \overrightarrow{nf}_{j,i} \geq 0$, which implies that $x \in HT^{\geq}_{i,j}$, and hence since $HT_{i,j}$ is tangent to P along $e_{i,j}$, $P \subseteq HT^{\geq}_{i,j}$ as required. $\square$

We conclude this subsection with a lemma that characterizes the manner in which a convex polytope interacts with two non-parallel halfspaces.

Lemma 5.15 *Let H_1 and H_2 be two planes with linearly independent normal vectors $\vec{n}_1$ and $\vec{n}_2$ respectively, let $Q = H^{\geq}_1 \cap H^{\geq}_2$, and let P be a convex polytope. If $p \in Q \cap P$, then there is a point q of $Q \cap P$ that belongs to an edge of P and such that $d(q, H_1) \geq d(p, H_1)$ and $d(q, H_2) \geq d(p, H_2)$.*

Proof: Let x belong to $H_1 \cap H_2$ and let r be the ray originating at p in direction $\overrightarrow{p - x}$. Since P is convex, there exists a point p' of $bd(P)$ such that $P \cap r = \overline{pp'}$. Furthermore, the choice of r implies that $d(p', H_1) \geq d(p, H_1)$ and $d(p', H_2) \geq d(p, H_2)$. Let f be a facet of P that contains p', let H be the plane that contains f, and let $\vec{n}$ be a vector normal to H. Since $\vec{n}_1$ and $\vec{n}_2$ are linearly independent, either H and H_1

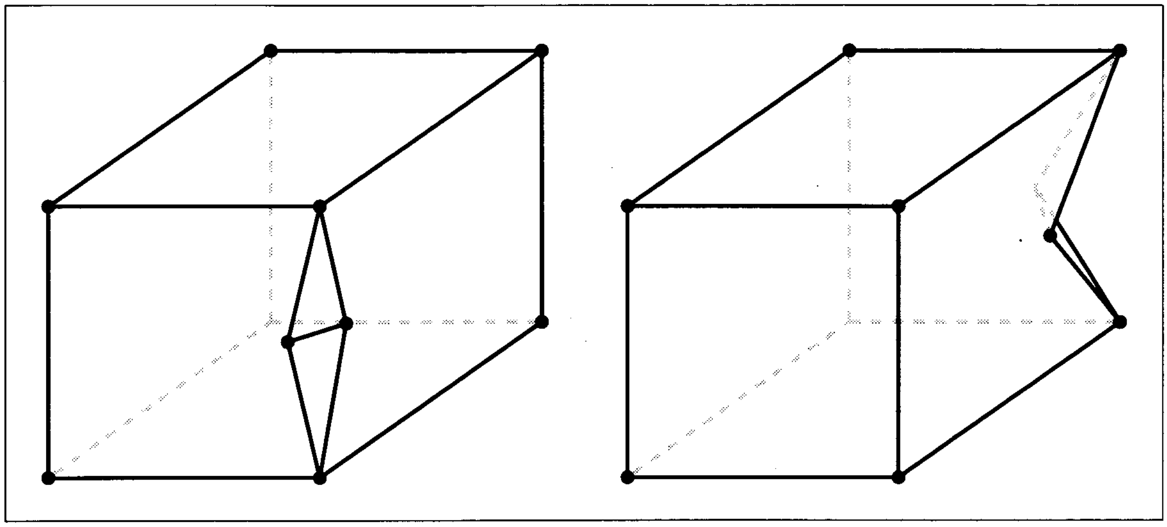


Figure 5.10: Two views of a cube from which a single wedge has been removed.

are concurrent, or H and H_2 are concurrent. By symmetry we can assume that H and H_1 are concurrent.

Let l be the line that contains p' and $p' + \vec{n} \times \vec{n}_1$. Since H and H_1 are concurrent, l is contained in H, and parallel to H_1. Therefore $d(p^*, H_1) = d(p', H_1)$ for every point p^* of l. Since l contains p', its intersection with f is a line segment $\overline{qq'}$ in which q and q' belong to edges of P. Since either $d(q, H_2) \geq d(p', H_2)$ or $d(q', H_2) \geq d(p', H_2)$, one of the endpoints of $l \cap f$ satisfies the conditions of the lemma. □

5.2.2 Constructing wedges

Let P be a convex polytope. The *wedge* associated with an edge $e_{i,j}$ of P, denoted by $W_{i,j}$, is a tetrahedral subset of P whose removal from P blocks visibility between v_i and v_j. The *size* of $W_{i,j}$ is the supremum of the perpendicular distance from $HT_{i,j}$ to a point of $W_{i,j}$. Figure 5.10 shows a cube from which a single wedge has been removed. We first determine an upper bound ε on the size of all wedges associated with edges of P. We then define the planes that bound $W_{i,j}$, and show how they interact with P and with each other. Finally, we define $W_{i,j}$ formally and show how wedges affect visibility in P.

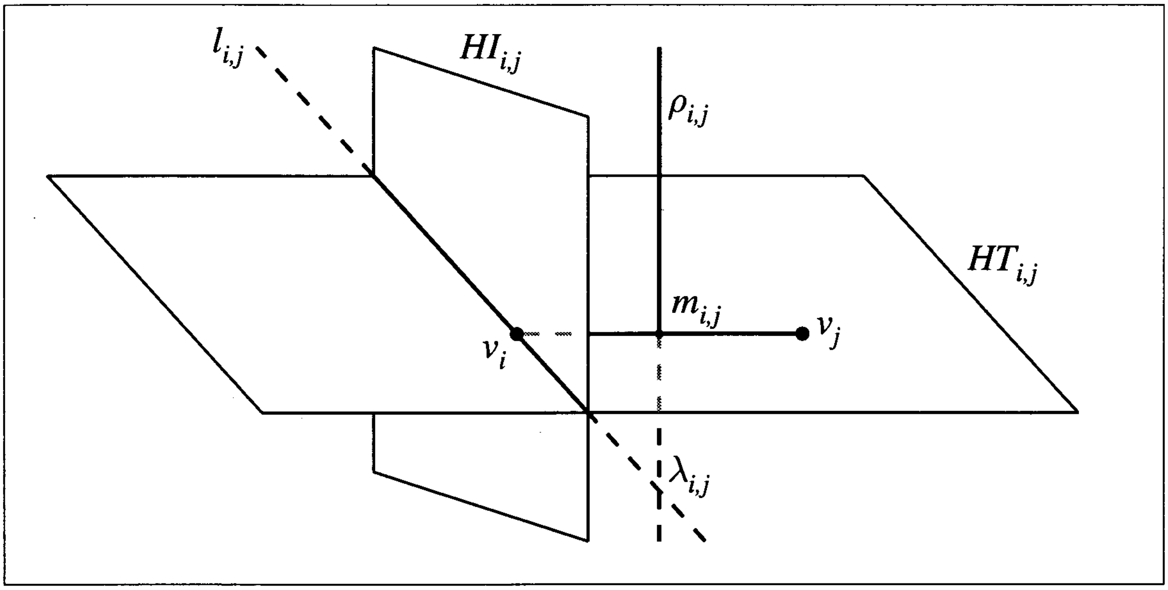


Figure 5.11: Illustrating $HI_{i,j}$, $l_{i,j}$, $\lambda_{i,j}$ and $\rho_{i,j}$.

An upper bound ε on the size of wedges

The upper bound ε on the size of the wedges is obtained by taking the minimum "distance" from certain points of $HT^{>}_{i,j}$ to $HT_{i,j}$, using a rather unusual distance function. The definitions in the following paragraph and the next two lemmas will help us define this distance function.

Consider an edge $e_{i,j}$ of P. Let $HI_{i,j}$ be the plane through v_i with normal vector $\overrightarrow{np}_{i,j}$ and let $l_{i,j} = HT_{i,j} \cap HI_{i,j}$, as illustrated in Figure 5.11. Finally, let $\lambda_{i,j}$ be the line with orientation $\overrightarrow{nt}_{i,j}$ that contains $m_{i,j}$, let $\rho_{i,j}$ be the open ray $\lambda_{i,j} \cap HT^{>}_{i,j}$, and let $\mathcal{H}_{i,j}$ be the set of all planes that contain $l_{i,j}$ and intersect $\rho_{i,j}$.

Lemma 5.16 *Let $e_{i,j}$ be an edge of P and H be a plane that contains $l_{i,j}$. If $H \neq HI_{i,j}$, then $|H \cap \lambda_{i,j}| = 1$.*

Proof: Since $H \neq HI_{i,j}$, the planes H and $HP_{i,j}$ are concurrent. Let $l = H \cap HP_{i,j}$. Since H is not parallel to $HP_{i,j}$, it does not contain any pair of concurrent lines both parallel to $HP_{i,j}$. Since H contains $l_{i,j}$, this means that it does not contain any line parallel to $\lambda_{i,j}$. This implies that $|l \cap \lambda_{i,j}| = 1$, as required. □

Lemma 5.17 *Let $e_{i,j}$ be an edge of P. Every point of $HT^{>}_{i,j} \cap HI^{>}_{i,j}$ belongs to exactly one element of $\mathcal{H}_{i,j}$.*

Proof: Let x be a point of $HT^{>}_{i,j} \cap HI^{>}_{i,j}$. Since x does not belong to $l_{i,j}$, there is a unique plane H that contains both x and $l_{i,j}$. Since $x \notin HI_{i,j}$, it follows that $H \neq HI_{i,j}$. Hence Lemma 5.16 implies that there is a point y such that $H \cap \lambda_{i,j} = \{y\}$. It remains to prove that $y \in \rho_{i,j}$. Consider the line segment $l = \overline{xy}$. Since $x \in HI^{>}_{i,j}$ and $y \in HI^{>}_{i,j}$, it follows that $l \subseteq HI^{>}_{i,j}$, and so $l \cap l_{i,j} = \emptyset$. Since $l \subseteq H$ and $H \cap HT_{i,j} = l_{i,j}$, it follows that $l \cap HT_{i,j} = \emptyset$. Since $x \in HT^{>}_{i,j}$, this implies that $y \in HT^{>}_{i,j}$, and hence that $y \in \rho_{i,j}$. □

It follows from Lemma 5.16 that $|H \cap \rho_{i,j}| = 1$ for each element H of $\mathcal{H}_{i,j}$. We will call the distance from $H \cap \rho_{i,j}$ to $HT_{i,j}$ the *slant* of H with respect to $e_{i,j}$, denoted by $slant_{i,j}(H)$. The element of $\mathcal{H}_{i,j}$ that contains a point x of $HT^{>}_{i,j} \cap HI^{>}_{i,j}$ will be denoted by $HS_{i,j}(x)$. It can be verified easily that the vector $\overrightarrow{ns}_{i,j}(x)$ defined below is normal to $HS_{i,j}(x)$:

$$\overrightarrow{ns}_{i,j}(x) = (\overrightarrow{nt}_{i,j} \times \overrightarrow{np}_{i,j}) \times (x - v_i).$$

We will call the value $slant_{i,j}(HS_{i,j}(x))$ the *slant* of x with respect to $e_{i,j}$, and denote it by $slant_{i,j}(x)$. The slant of x is the unusual distance function to which we were referring earlier. We adopt the convention that $slant_{i,j}(x) = \infty$ for every point x of $HT^{>}_{i,j} \cap HI^{\leq}_{i,j}$. These definitions are illustrated in Figure 5.12.

Let us now define the points from whose slants the value of ε will be obtained. Consider an edge $e_{i,j}$ of P. Since $HP_{i,j}$ and $HF_{i,j}$ are concurrent, they intersect in a line l. Moreover the intersection of l with $f_{i,j}$ is a line segment since $f_{i,j}$ is convex and l contains $m_{i,j}$. We will denote by $q^*_{i,j}$ the midpoint of this line segment, and by $q_{i,j}$ the point x of $\overline{v_i v_j}$ that satisfies $d(v_i, x)/d(v_i, v_j) = 1/4$.

The value of ε needs to be chosen to take two factors into account. The first is related to the angle between an edge $e_{i,j}$ of P and the facets that contain it. We need to prevent $W_{i,j}$ from intersecting these facets. A value $\alpha_{i,j}$ is thus obtained as follows for each edge $e_{i,j}$ of P:

$$\alpha_{i,j} \;=\; d(q^*_{i,j}, HT_{i,j}).$$

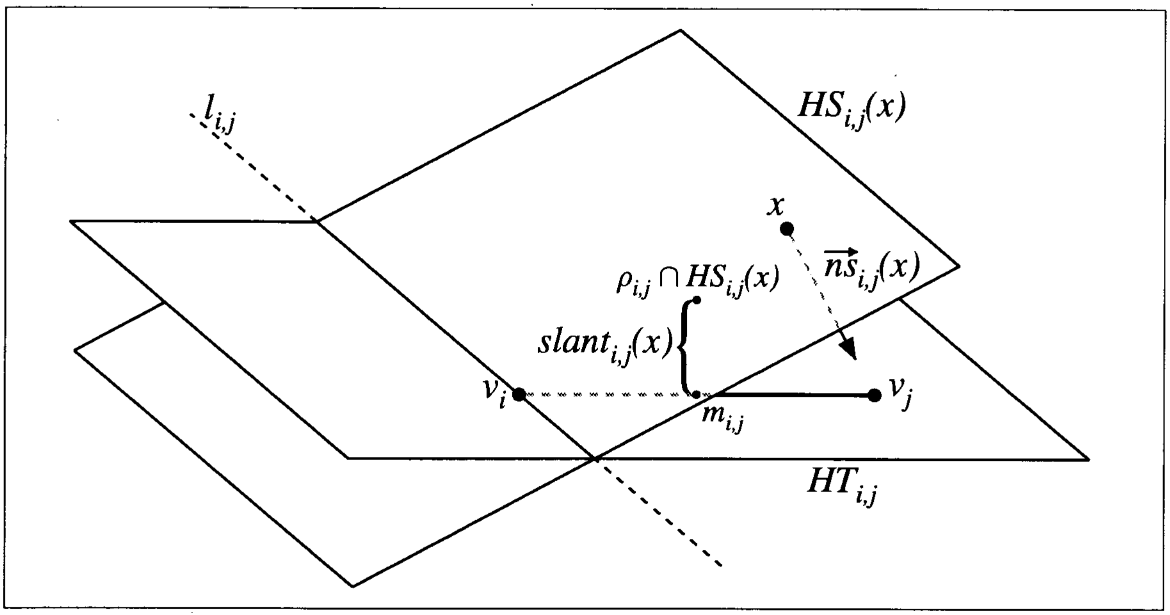


Figure 5.12: Illustrating $HS_{i,j}(x)$, $\overrightarrow{ns}_{i,j}(x)$ and $slant_{i,j}(x)$.

Secondly, we need to ensure that the removal of $W_{i,j}$ from P will only affect the visibility between the endpoints of $e_{i,j}$. This means that if e is an edge of P that shares an endpoint with $e_{i,j}$, then the midpoint of e must be strictly separated from $e_{i,j}$ by each plane defining $W_{i,j}$. Two values $\beta_{i,j}$, $\gamma_{i,j}$ are thus computed as follows for each edge $e_{i,j}$ of P:

$$\begin{aligned}
\beta_{i,j} &= \min\{slant_{i,j}(q_{i,k}), slant_{i,j}(q^*_{i,k}), slant_{i,j}(q^*_{k,i}) \mid v_k \text{ is adjacent to } v_i\}, \\
\gamma_{i,j} &= \min\{slant_{i,j}(q_{j,k}), slant_{i,j}(q^*_{j,k}), slant_{i,j}(q^*_{k,j}) \mid v_k \text{ is adjacent to } v_j\}.
\end{aligned}$$

Finally, let ε be defined as follows:

$$\varepsilon = \min\{1/2, \alpha^2_{i,j}, \beta^2_{i,j}, \gamma^2_{i,j} \mid v_i \text{ is adjacent to } v_j\}.$$

We note that, even though $\alpha_{i,j}$, $\beta_{i,j}$ and $\gamma_{i,j}$ will in general be irrational, ε is rational. Furthermore, $\varepsilon < \min\{\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j}\}$ for every edge $e_{i,j}$ of P.

We conclude this subsection with results that are related to slants and will be required in the following subsections. We first show how to compute $slant_{i,j}(x)$.

Lemma 5.18 *If $e_{i,j}$ is an edge of P and $x \in HT^{>}_{i,j} \cap HI^{>}_{i,j}$, then $slant_{i,j}(x) = ((x - m_{i,j}) \cdot \overrightarrow{nt}_{i,j})((m_{i,j} - v_i) \cdot \overrightarrow{np}_{i,j}) \,/\, (|\overrightarrow{nt}_{i,j}|\,((x - v_i) \cdot \overrightarrow{np}_{i,j}))$.*

Proof: Let H be the plane through v_i with normal vector $\overrightarrow{nt}_{i,j} \times \overrightarrow{np}_{i,j}$, let y be the perpendicular projection of x on H, and let $z = \rho_{i,j} \cap HS_{i,j}(x)$. Since v_i, $\overrightarrow{np}_{i,j}$ and $\overrightarrow{nt}_{i,j}$ are contained in H, the points $m_{i,j}$ and z belong to H. Since $HT_{i,j}$ and $HP_{i,j}$ are both perpendicular to H, it follows that $d(z, HT_{i,j}) = d(y, HT_{i,j})\, d(z, HI_{i,j}) \,/\, d(y, HI_{i,j})$. However $\overrightarrow{nt}_{i,j} \times \overrightarrow{np}_{i,j}$ is contained in $HS_{i,j}(x)$, and hence $d(y, HT_{i,j}) = d(x, HT_{i,j})$ and $d(y, HI_{i,j}) = d(x, HI_{i,j})$. By substituting for each distance function, the result follows. □

The next lemma shows how the vector $\overrightarrow{ns}_{i,j}(x)$ is oriented. It suffices to identify the side of $HS_{i,j}(x)$ to which v_j belongs.

Lemma 5.19 *If $e_{i,j}$ is an edge of P, then $v_j \in HS^{>}_{i,j}(x)$ for every point x of $HT^{>}_{i,j} \cap HI^{>}_{i,j}$.*

Proof: It suffices to show that $(v_j - v_i) \cdot \overrightarrow{ns}_{i,j}(x) > 0$. Indeed, $v_j - v_i = \overrightarrow{np}_{i,j}$, and so by using the definition of $\overrightarrow{ns}_{i,j}(x)$ we get:

$$(v_j - v_i) \cdot \overrightarrow{ns}_{i,j}(x) = ((x - v_i) \cdot \overrightarrow{nt}_{i,j})|\overrightarrow{np}_{i,j}|^2.$$

Since $v_i \in HT_{i,j}$ and $x \in HT^{>}_{i,j}$, this implies that $(x - v_i) \cdot \overrightarrow{nt}_{i,j} > 0$, and the result follows. □

The elements of $\mathcal{H}_{i,j}$ are ordered in a natural way according to the angle they make with $HI_{i,j}$. The last two lemmas of this subsection explain how the related function $slant_{i,j}(x)$ orders the point of $HT^{>}_{i,j}$.

Lemma 5.20 *Let $e_{i,j}$ be an edge of P. If x is a point of $HT^{>}_{i,j} \cap HI^{\leq}_{i,j}$ and H belongs to $\mathcal{H}_{i,j}$, then $x \in H^{<}$.*

Proof: Let $y = H \cap \rho_{i,j}$ and $n = v_i - y$. Since $x \in HT_{i,j}^{>} \cap HI_{i,j}^{\leq}$, there exist real numbers $a > 0$ and $b \leq 0$ such that $x = v_i + a\overrightarrow{nt}_{i,j} + b\overrightarrow{np}_{i,j}$. By substituting for $slant_{i,j}(H)$ and x, expanding, and using the fact that $\overrightarrow{ns}_{i,j}(y) \cdot (v_i - y) = 0$, we obtain:

$$\overrightarrow{ns}_{i,j}(y) \cdot (x - y) = a|\overrightarrow{nt}_{i,j}|^2(\vec{n} \cdot \overrightarrow{np}_{i,j}) - b|\overrightarrow{np}_{i,j}|^2(\vec{n} \cdot \overrightarrow{nt}_{i,j}).$$

Since $y \in HP_{i,j}^{>}$, it follows that $\vec{n} \cdot \overrightarrow{np}_{i,j} < 0$. Similarly $y \in HT_{i,j}^{>}$, and so $\vec{n} \cdot \overrightarrow{nt}_{i,j} < 0$. Hence $slant_{i,j}(H) \cdot (x - y) < 0$, which implies that $x \in H^{<}$. □

Lemma 5.21 *Let $e_{i,j}$ be an edge of P. If x is a point of $HT_{i,j}^{>} \cap HI_{i,j}^{>}$, and H is an element of $\mathcal{H}_{i,j}$ such that $slant_{i,j}(H) < slant_{i,j}(x)$, then $x \in H^{<}$.*

Proof: Since $v_i \in H$ and $v_j \in H^{>}$ by Lemma 5.19, the point $m_{i,j}$ belongs to $H^{>}$. Let $y = m_{i,j} + slant_{i,j}(H)\,\overrightarrow{nt}_{i,j}$, and $z = m_{i,j} + slant_{i,j}(x)\,\overrightarrow{nt}_{i,j}$. Since $slant_{i,j}(H) < slant_{i,j}(x)$, it follows that y belongs to the open line segment $\overline{m_{i,j}z}$. Since $y \in H$ and $m_{i,j} \in H^{<}$, this implies that $z \in H^{>}$. Moreover, since both x and z belong to $HS_{i,j}(x) \cap HT_{i,j}^{>}$, and since $H \cap HS_{i,j}(x) = l_{i,j}$, no point of $\overline{xz}$ belongs to H, and hence x belongs to $H^{>}$ as required. □

Bounding planes and wedges

The planes used to construct the wedges of P (called *bounding* planes) will be elements of $\mathcal{H}_{i,j}$ for each edge $e_{i,j}$ of P. Care must however be taken to ensure that they can be described using rational numbers only. Hence, for each edge $e_{i,j}$ of P, let $\delta_{i,j} = \varepsilon / \max\{1, |\overrightarrow{nt}_{i,j}|^2\}$, and let $m_{i,j}^*$ be the point $m_{i,j} + \delta_{i,j}\overrightarrow{nt}_{i,j}$. The plane associated with $e_{i,j}$ will be $HW_{i,j} = HS_{i,j}(m_{i,j}^*)$. Our first lemma that the points used to compute ε are all *above* $HW_{i,j}$ and $HW_{j,i}$ (assuming the plane $HT_{i,j}$ is horizontal).

Lemma 5.22 *Let v_j and v_k be distinct vertices of P, and let v_i be a vertex of P that is adjacent to both v_j and v_k. If $x \in \{q_{i,k}, q_{i,k}^*, q_{k,i}^*\}$, then $x \in HW_{i,j}^{<} \cap HW_{j,i}^{<}$.*

Proof: We first prove that $x \in HW_{i,j}^{<}$. The plane $HT_{i,j}$ is tangent to P along $e_{i,j}$, and so x belongs to $HT_{i,j}^{>}$. If $x \in HI_{i,j}^{\leq}$, then the result follows from Lemma 5.20 since $HW_{i,j} \in \mathcal{H}_{i,j}$. Otherwise, since $slant_{i,j}(HW_{i,j}) = \delta_{i,j}$ and since $\delta_{i,j} < \varepsilon \leq$

$\min\{1/2, slant^2_{i,j}(x)\} < slant_{i,j}(x)$, it follows from Lemma 5.21 that $x \in HW^{<}_{i,j}$. A symmetric argument shows that $x \in HW^{<}_{j,i}$. □

Lemma 5.22 allows us to determine the location of every vertex of P with respect to $HW_{i,j}$. We now prove that $HW_{i,j}$ separates v_i and v_j from the remaining vertices of P.

Lemma 5.23 *A vertex v of P belongs to $HW^{\geq}_{i,j}$ if and only if it an endpoint of $e_{i,j}$.*

Proof: Let $f : \boldsymbol{E}^3 \to \boldsymbol{E}$ be the function defined on the vertices of P by $f(v) = v \cdot \overrightarrow{ns}_{i,j}(m^*_{i,j})$. Lemma 5.19 implies that $f(v_j) > 0$, and $f(v_i) = 0$ since $v_i \in HW_{i,j}$. Moreover it follows from Lemma 5.22 that $f(v) < f(v_j)$ for every vertex v of P adjacent to v_j. Since P is convex, this implies that for each $v \neq v_j$, there is a neighbor v' of v such that $f(v') > f(v)$. Hence for each vertex v of P, there is a path from v to v_j along which the value of f increases monotonically. Let $g(v)$ be the length of the shortest such path. An easy induction on $g(v)$ proves that $f(v) < 0$ if and only if $v \notin \{v_i, v_j\}$, thus completing the proof of the lemma. □

We can use Lemma 5.23 to show that $f_{i,j}$ and $f_{j,i}$ are the only two facets of P that properly intersect both $HW_{i,j}$ and $HW_{j,i}$.

Lemma 5.24 *Let f be a facet of P that contains a point p of $HW^{\geq}_{i,j} \cap HW^{\geq}_{j,i}$. If $p \notin \{v_i, v_j\}$ then $f \in \{f_{i,j}, f_{j,i}\}$.*

Proof: Since f intersects $HW^{\leq}_{i,j}$, at least one of its vertices belongs to $HW^{\leq}_{i,j}$. If f does not contain v_j, then Lemma 5.23 implies that v_i belongs to f, and that every other vertex of f is in $HW^{>}_{i,j}$. However this implies that $f \cap HW^{\leq}_{i,j} = v_i$. Since $p \neq v_i$, it follows that v_j is a vertex of f. A symmetric argument shows that v_i is also contained in f. Since f is convex, it therefore contains $e_{i,j}$. Since P is simple, this implies that $f \in \{f_{i,j}, f_{j,i}\}$. □

Let us now consider the way in which pairs of bounding planes interact with each other and with P. We start by considering the relative position of the planes associated with the two directed edges that correspond to a given edge of P.

Lemma 5.25 *If v_i and v_j are two adjacent vertices of P, then $P \cap HW^{>}_{i,j} \cap HP^{<}_{i,j} \subseteq HW^{>}_{j,i}$.*

Proof: Consider a point x of $P \cap HW^{>}_{i,j} \cap HP^{<}_{i,j}$. If $x \in HT_{i,j}$, then $x \in HW^{>}_{j,i}$ by Lemma 5.23. Hence suppose that $x \in HT^{>}_{i,j}$, which implies that $x \in HI^{>}_{i,j}$. Since $\overrightarrow{np}_{i,j} = -\overrightarrow{np}_{j,i}$ and $x \in HP^{<}_{i,j}$, it follows that $(m_{i,j} - v_i) \cdot \overrightarrow{np}_{i,j} = (m_{i,j} - v_j) \cdot \overrightarrow{np}_{j,i} \geq 0$. Moreover $x \in HI^{>}_{i,j} \cap HI^{>}_{j,i}$, and $HP_{i,j}$ is the perpendicular bisector of $e_{i,j}$. Hence $0 < (x - v_i) \cdot \overrightarrow{np}_{i,j} < (x - v_j) \cdot \overrightarrow{np}_{j,i}$. Finally $x \in HT^{>}_{i,j}$ and hence $(x - m_{i,j}) \cdot \overrightarrow{nt}_{i,j} > 0$. Therefore Lemma 5.18 implies that $slant_{j,i}(x) < slant_{i,j}(x)$. Since $slant_{i,j}(x) \leq slant_{i,j}(HW_{i,j})$ by Lemma 5.21, and since $slant_{i,j}(HW_{i,j}) = slant_{j,i}(HW_{j,i})$, it follows that $slant_{j,i}(x) < slant_{j,i}(HW_{j,i})$, and hence $x \in HW^{>}_{j,i}$. □

We now look at the intersection of the bounding planes that correspond to two distinct edges originating from a common vertex of P, and prove that this vertex is the only point of P that they have in common.

Lemma 5.26 *Let v_i be a vertex of P. If v_j and v_k are two distinct vertices of P adjacent to v_i, then $P \cap HW^{\geq}_{i,j} \cap HW^{\geq}_{i,k} = \{v_i\}$.*

Proof: Let V denote the set of vertices of P, and $Q = HW^{\geq}_{i,j} \cap HW^{\geq}_{i,k}$. It follows from Lemmas 5.19 and 5.23 that $V \cap HW^{>}_{i,j} = \{v_j\}$, that $V \cap HW^{>}_{i,k} = \{v_k\}$, and that $V \cap Q = V \cap HW_{i,j} \cap HW_{i,k} = \{v_i\}$. Consider an arbitrary point x of $P \cap Q$. We first prove that if $x \neq v_i$ then there is a point y of Q that belongs to an edge e of P but not to V.

Case 1 ($x \in HW^{>}_{i,j} \cup HW^{>}_{i,k}$) : Let y be the point whose existence is implied by Lemma 5.15. Since $d(x, HW_{i,j}) > 0$ or $d(x, HW_{i,k}) > 0$, it follows that $d(y, HW_{i,j}) > 0$ or $d(y, HW_{i,k}) > 0$, and hence $y \neq v_i$. Since $V \cap Q = \{v_i\}$, this implies that y is the required point.

Case 2 ($x \in HW_{i,j} \cap HW_{i,k} \cap int(P)$) : Since $x \in int(P)$, there is a point x' of P in the neighborhood of x that belongs to $HW^{>}_{i,j} \cup HW^{>}_{i,k}$. The point y can then be found as in Case 1.

Case 3 ($x \in HW_{i,j} \cap HW_{i,k} \cap bd(P)$) : If x belongs to an edge of P then we can choose $y = x$. Hence assume that it is not the case. Let l be the line through v_i and x. If some point of $\overline{v_i x}$ belongs to the interior of P, then the point y can be found as in Case 2. Otherwise $\overline{v_i x}$ is contained in a face f of P. Since both v_i and x belongs to $HW_{i,j} \cap HW_{i,k}$, so does the endpoint y of $l \cap f$ distinct from v_i. Since $y \in Q$ and $y \neq v_i$, it follows that $y \notin V$, and hence y is the required point.

We now show that the existence of y leads to a contradiction. Two cases need to be considered.

Case 1 (v_i is an endpoint of e) : Since $v_i \in HW_{i,j} \cap HW_{i,k}$ and $y \in Q$, the second endpoint of e also belongs to Q, which is impossible.

Case 2 (v_i is not an endpoint of e) : Since y belongs to $HW^{\geq}_{i,j}$, at least one endpoint of e belongs to $HW^{\geq}_{i,j}$, and so v_j is an endpoint of e. A similar argument shows that v_k is the second endpoint of e. However, since $q_{j,k} \in HW^{<}_{i,j}$ and $q_{k,j} \in HW^{>}_{i,k}$, the point y must belong to $\overline{v_j q_{j,k}} \cap \overline{v_k q_{k,j}} = \emptyset$, which cannot occur.

Since the existence of a point x of $P \cap Q$ distinct from v_i leads to a contradiction, we conclude that $P \cap Q = \{v_i\}$, as required. □

Let us now consider the intersection of two arbitrary bounding planes. Six cases in which the intersection contains points of P are shown in Figure 5.13. Our next lemma proves that these are the only such cases.

Lemma 5.27 *Let $e_{i,j}$ and $e_{k,l}$ be two edges of P, and let $Q = P \cap HW^{\geq}_{i,j} \cap HW^{\geq}_{k,l}$. If $Q \neq \emptyset$, then either $i = l$, or $j = k$, or $j = l$, or $i = k$ and $Q = \{v_i\}$.*

Proof: Let V denote the set of vertices of P. If $Q \neq \emptyset$, then Lemma 5.15 implies that there is a point x of Q that belongs to an edge e of P. Two cases need to be considered.

Case 1 ($x \in V$) : Since $x \in Q$, Lemma 5.23 implies that $x \in \{v_i, v_j\}$, and that $x \in \{v_k, v_l\}$. Thus either $x = v_i = v_k$, or $x = v_i = v_l$, or $x = v_j = v_k$ or

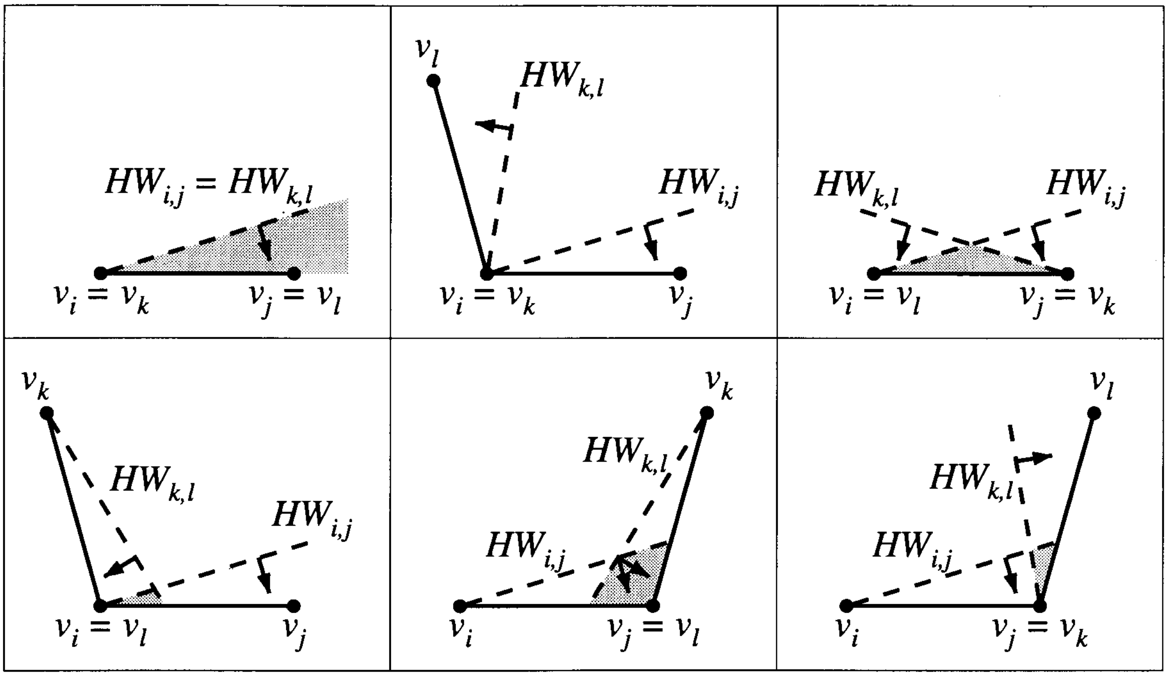


Figure 5.13: The six cases in which $P \cap HW^{\geq}_{i,j} \cap HW^{\geq}_{k,l} \neq \emptyset$.

$x = v_j = v_l$. If $i \neq l$, $j \neq k$, and $j \neq l$, then $x = v_i = v_k$ and $v_j \neq v_l$, and hence it follows from Lemma 5.26 that $Q = \{v_i\}$.

Case 2 ($x \notin V$) : Since $v_i \in HW_{i,j}$ and $V \cap HW^{\geq}_{i,j} = \{v_i, v_j\}$, it follows that e is incident upon v_j. A similar argument shows that v_l is the second endpoint of e. Suppose that $j \neq l$ (which implies that $e = e_{j,l}$) and that $i \neq l$. Lemma 5.22 implies that $q_{j,l} \in HW^{<}_{i,j}$. Since $v_j \in HW^{>}_{i,j}$, this implies that $x \in \overline{v_j q_{j,l}}$. Furthermore, since $v_l \in HW^{>}_{k,l}$, it follows that $\overline{q_{j,l} v_l} \subseteq HW^{>}_{k,l}$. Since $q_{l,j} \in \overline{q_{j,l} v_l}$, this implies that $q_{l,j} \in HW^{>}_{k,l}$, and hence from Lemma 5.22 that $j = k$.

This completes the case analysis and the proof of the lemma. □

We now present a result that characterizes more precisely the location of the intersection of the two bounding planes in the cases shown on the second row of Figure 5.13. This characterization will be required by the proof of correctness of our transformation.

Lemma 5.28 *Let v_i be a vertex of P. If v_j and v_k are two distinct vertices of P adjacent to v_i, then $P \cap HW^{>}_{i,j} \cap HW^{>}_{k,i} \subseteq HW^{>}_{j,i}$.*

Proof: Let x be a point of $P \cap HW^{>}_{i,j} \cap HW^{>}_{k,i}$. We first prove by contradiction that $x \in HP^{<}_{i,j}$. Suppose to the contrary that $x \in HP^{\geq}_{i,j}$. Let l be the line containing v_i and x, and let $y = l \cap HP_{i,j}$. Since P is convex, it follows that $y \in P$, and hence $y \in HF^{\geq}_{i,j} \cap HF^{\geq}_{j,i}$. Moreover $d(y, HT_{i,j}) = slant_{i,j}(y) = slant_{i,j}(x)$, and since $slant_{i,j}(x) \leq \min\{d(q^*_{i,j}, HT_{i,j}), d(q^*_{j,i}, HT_{i,j})$, this implies that y lies in the triangle $\triangle m_{i,j} q^*_{i,j} q^*_{j,i}$. Since $y \in HW^{>}_{k,i}$ and both $q^*_{i,j}$ and $q^*_{j,i}$ belong to $HW^{<}_{k,i}$ by Lemma 5.22, this implies that $m_{i,j} \in HW^{>}_{k,i}$. However this means that $q_{i,j} \in HW^{>}_{k,i}$, contradicting Lemma 5.22. Hence $x \in HP^{<}_{i,j}$, and so Lemma 5.25 implies that $x \in HW^{>}_{j,i}$. □

The wedge $W_{i,j}$ will be the set $P \cap HW^{>}_{i,j} \cap HW^{>}_{j,i}$. The *closed* wedge $\overline{W}_{i,j}$ will be the union of $W_{i,j}$ with its boundary, i.e. the set $P \cap HW^{\geq}_{i,j} \cap HW^{\geq}_{j,i}$. This set can also be defined in a slightly different manner. This alternate definition is presented in the next lemma.

Lemma 5.29 *If $e_{i,j}$ is an edge of P, then $\overline{W}_{i,j} = HW^{\geq}_{i,j} \cap HW^{\geq}_{j,i} \cap HF^{\geq}_{i,j} \cap HF^{\geq}_{j,i}$.*

Proof: Let $\overline{W}^* = HW^{\geq}_{i,j} \cap HW^{\geq}_{j,i} \cap HF^{\geq}_{i,j} \cap HF^{\geq}_{j,i}$. Since P is convex, $P \subseteq HF^{\geq}_{i,j} \cap HF^{\geq}_{j,i}$, and hence $\overline{W}_{i,j} \subseteq \overline{W}^*$. Let us now prove that $\overline{W}^* \subseteq \overline{W}_{i,j}$. Consider a point x of $\overline{W}^*$, and let l be the open line segment $\overline{x m_{i,j}}$. Since $\overline{W}^*$ is convex, $l \subseteq \overline{W}^*$, and so $l \subseteq HW^{\geq}_{i,j} \cap HW^{\geq}_{j,i}$. Hence Lemma 5.24 implies that no point of l belongs to a face of P other than $f_{i,j}$ or $f_{j,i}$. Since $m_{i,j} \in H^{>}$ for each plane H containing a face of P, it follows that $l \subseteq H^{>}$ for every such plane. Since $l \subseteq HF^{\geq}_{i,j} \cap HF^{\geq}_{j,i}$, this implies that $l \subseteq P$, and hence that $x \in \overline{W}_{i,j}$. Therefore $\overline{W}_{i,j} \subseteq \overline{W}^*$, and so $\overline{W}_{i,j} = \overline{W}^*$. □

We conclude this section by proving that closed wedges associated with distinct edges of P do not intersect except possibly at one common vertex of P.

Lemma 5.30 *If $e_{i,j}$ and $e_{k,l}$ are two distinct edges of P, then $\overline{W}_{i,j} \cap \overline{W}_{k,l} = e_{i,j} \cap e_{k,l}$.*

Proof: The proof can be broken down into subcases depending on the cardinality of $e_{i,j} \cap e_{k,l}$.

Case 1 ($e_{i,j} \cap e_{k,l} = \emptyset$) : This implies that $i \neq k$, $i \neq l$, $j \neq k$ and $j \neq l$, and so it follows from Lemma 5.27 that $HW^{\geq}_{i,j} \cap HW^{\geq}_{k,l} \cap P = \emptyset$. Therefore $\overline{W}_{i,j} \cap \overline{W}_{k,l} = \emptyset$.

Case 2 ($e_{i,j} \cap e_{k,l} \neq \emptyset$) : Since $e_{i,j} \neq e_{k,l}$, the edges $e_{i,j}$ and $e_{k,l}$ have exactly one common endpoint. We can assume without loss of generality that $v_i = v_k$ and $v_j \neq v_l$. Let $Q = HW^{\geq}_{i,j} \cap HW^{\geq}_{k,l} \cap P$. Since $v_i \in Q$, it follows from Lemma 5.27 that $Q = \{v_i\}$. Since $v_i \in HW^{\geq}_{j,i} \cap HW^{\geq}_{l,k}$, and since $\overline{W}_{i,j} \cap \overline{W}_{k,l} \subseteq Q$, this implies that $\overline{W}_{i,j} \cap \overline{W}_{k,l} = \{v_i\}$.

This completes the case analysis and the proof of the lemma. □

5.2.3 Transformation and proof of correctness

We now show how an instance of DS3C is transformed into an instance of the the problem of determining whether a starshaped polytope is U_3 or B_3. We first construct a polytope P^* from the instance of DS3C, prove that P^* is simple, and show that the transformation can be done in polynomial time. Then we define *neighborhoods* around each convex vertex of P^* and study their visibility properties. Finally, we prove that P^* is U_3 if and only if the instance of DS3C admits a proper 3-coloring.

Let $V^* = \{v_1^*, \ldots, v_n^*\}$ be a set of point in the plane, let $E^* = \{e_1^*, \ldots, e_m^*\}$ be a subset of the Delaunay triangulation of V^*, and let $G^* = (V^*, E^*)$. Let ϕ be the *inversion* function defined by $\phi(x, y) = (x, y, x^2 + y^2)$ [41], and let $v_i = \phi(v_i^*)$ for each i in $\{1, \ldots, n\}$. Finally, let P be the convex hull of $V = \{v_1, \ldots, v_n\}$. The following observation can be derived from observation 13.13 in the book by Edelsbrunner [41].

Observation 5.4 *If an edge $e^* = (v_i^*, v_j^*)$ belongs to E^*, then the line segment $\overline{v_i v_j}$ is an edge of P.*

By abuse of notation, when $e^* = (v_i^*, v_j^*)$ is an element of E^*, we will denote by $\phi(e^*)$ the edge $e_{i,j}$ of P. Let us now describe the manner in which our target polytope P^* is constructed from P. We will construct a sequence $P_0 = P$, P_1, ..., $P_m = P^*$ of polytopes. Given P_k, we construct P_{k+1} as follows : if $e_{k+1}^* = \overline{v_i^* v_j^*}$, the polytope P_{k+1} will be $P_k \setminus W_{i,j}$. We first need to prove that P^* is a simple polytope.

Lemma 5.31 *The polytope P^* is closed and simple. Moreover every convex vertex of P is a convex vertex of P^*.*

Proof: We prove by induction on k that P_k satisfies five properties. The lemma follows from the fact that properties 1 and 5 hold for P_m.

1. P_k is a closed simple polytope;
2. if $e^* \in E^* \setminus \{e_1^*, \ldots, e_k^*\}$, then $\phi(e^*)$ is an edge of P_k;
3. exactly one facet of P_k is contained in each facet of P;
4. if f is a facet of P_k not contained in a facet of P, there exists $e^* \in \{e_1^*, \ldots, e_k^*\}$ such that $\phi(e^*) = e_{i,j}$ and f is contained in a facet of $\overline{W}_{i,j}$;
5. every convex vertex of P is a convex vertex of P_k.

Property 1 and properties 3 to 5 hold trivially for $P_0 = P$, while property 2 follows from observation 5.4. Let us now suppose that P_k satisfies properties 1 to 5, and prove that P_{k+1} also satisfies them. Let $e_{k+1}^* = \overline{v_i^* v_j^*}$. We first require a simple observation.

Observation 5.5 *If f is a facet of P_k whose intersection with $\overline{W}_{i,j}$ contains a point distinct from v_i and v_j, then either $f \subseteq f_{i,j}$ or $f \subseteq f_{j,i}$.*

Proof: Consider a facet f of P_k contained in neither $f_{i,j}$ nor $f_{j,i}$. If f is contained in a facet of P, then Lemma 5.24 implies that $f \cap \overline{W}_{i,j} \subseteq \{v_i, v_j\}$. Otherwise it follows from property 4 that f is contained in a facet of a closed wedge corresponding to an edge of P distinct from $e_{i,j}$, and so Lemma 5.30 implies that $f \cap \overline{W}_{i,j} \subseteq \{v_i, v_j\}$. □

Since by property 3 exactly one facet of P_k is contained in $f_{i,j}$, we can use Observation 5.5 to characterize the intersection of $\overline{W}_{i,j}$ with that facet of P_k.

Observation 5.6 *The intersection of the facet f of P_k contained in $f_{i,j}$ with $\overline{W}_{i,j}$ is a convex polygon that contains $e_{i,j}$ and no other point of any edge of f.*

Proof: Since $\overline{W}_{i,j}$ is convex, the subset of $\overline{W}_{i,j}$ contained in f is a convex subpolygon of f. Moreover property 2 implies that $e_{i,j}$ is an edge of P_k, and hence $e_{i,j} \subseteq f \cap \overline{W}_{i,j}$. Finally, it follows from Observation 5.5 that no other point of any edge of f belongs to $\overline{W}_{i,j}$. $\square$

Let us now use observations 5.5 and 5.6 to prove that if properties 1 to 5 hold for P_k, then they also hold for P_{k+1}.

Property 1: Since $P_k \setminus W_{i,j} = P_k \cap (HW^{\leq}_{\bar{i},j} \cup HW^{\leq}_{i,\bar{j}})$, it follows that P_{k+1} is closed. Let $Q_k = \overline{W}_{i,j} \cap bd(P_k)$. Since by property 3 exactly one facet of P_k is contained in $f_{i,j}$, and exactly one facet of P_k is contained in $f_{j,i}$, it follows from observation 5.5 that $Q_k = (\overline{W}_{i,j} \cap f_{i,j}) \cup (\overline{W}_{i,j} \cap f_{j,i})$. Observation 5.6 thus implies that Q_k is homeomorphic to a disc, and since P_k is simple by the induction hypothesis, it follows that P_{k+1} is simple. Therefore property 1 holds for P_{k+1}.

Property 2: Let e^* be an edge of $E^* \setminus \{e^*_1, \ldots, e^*_{k+1}\}$, and e be $\phi(e^*)$. Since e is an edge of P_k by the induction hypothesis, and since $e^* \neq e^*_{k+1}$, it follows that $e \neq e_{i,j}$. Hence e belongs to a facet f of P_k not contained in either $f_{i,j}$ or $f_{j,i}$. Observation 5.5 thus implies that $e \cap W_{i,j} = \emptyset$, and so e is an edge of P_{k+1}. Hence property 2 holds for P_{k+1}.

Property 3: Consider a facet f of P. By the induction hypothesis, exactly one facet f' of P_k is contained in f. If $f \notin \{f_{i,j}, f_{j,i}\}$, then f' is a facet of P_{k+1} by Observation 5.5. Otherwise it follows from Observation 5.6 that f' contains exactly one facet of P_{k+1}. In both cases f contains exactly one facet of P_{k+1}, and so property 3 holds for P_{k+1}.

Property 4: If f is contained in a facet of P_k then Property 4 follows from the induction hypothesis. Otherwise f must be a facet of $\overline{W}_{i,j}$, and so $e^* = e^*_k$ satisfies Property 4.

Property 5: Let v be a convex vertex of P. By the induction hypothesis, v is a convex vertex of P_k. Since $v \notin W_{i,j}$, it follows that $v \in P_{k+1}$, and since P_{k+1}

is simple, v must be a convex vertex of P_{k+1}.

This completes the induction step, and therefore we conclude from the principle of mathematical induction that $P_m = P^*$ is simple, and that every convex vertex of P is a convex vertex of P^*, as required. □

Let us now prove that the simple polytope P^* constructed from G^* can be computed in time polynomial in the size of the instance of DS3C.

Lemma 5.32 *Given an instance G^* of DS3C, the polytope P^* can be computed in polynomial time.*

Proof: The inversion ϕ, and hence P, can be computed in $O(n)$ time. For each edge $e_{i,j}$ of P, the points $q_{i,j}$, $m_{i,j}$ and $q_{j,i}$, and the vectors $\overrightarrow{nt}_{i,j}$ and $\overrightarrow{np}_{i,j}$ can be computed in constant time. Each of $q^*_{i,j}$ and $q^*_{j,i}$ can be computed in linear time by traversing the boundary of $f_{i,j}$ and $f_{j,i}$. Once these points are known, $\alpha_{i,j}$ can be obtained in constant time. Since the number of edges of P is $O(n)$, there are $O(n^2)$ triplets (v_i, v_j, v_k) for which $e_{i,j}$ is an edge of P and v_k is a vertex adjacent to v_i or v_j. Lemma 5.18 then implies that $\beta_{i,j}$ and $\gamma_{i,j}$ can be computed in $O(n^2)$ time each. Consequently the value of ε can be computed in $O(n^2)$ time.

For each edge $e_{i,j}$ of P, the point $m^*_{i,j}$ and the vector $\overrightarrow{ns}_{i,j}(m^*_{i,j})$ can both be computed in constant time. Hence $HW_{i,j}$ and $HW_{j,i}$ can be computed in constant time, and Lemma 5.29 implies that P_{k+1} can be computed from P_k in constant time. Consequently P^* can be obtained from G^* in $O(n^2)$ time, as required. □

In order to obtain a cover of P^* by 3 convex polytopes from a proper 3-coloring of G^*, we need to define subsets of P^* that correspond to each element of V^*. For notational convenience, we will write $i \sim j$ if $(v^*_i, v^*_j) \in E^*$. Consider a vertex v^*_i of G^*. The vertex v_i of P^* is convex by Lemma 5.31. The *neighborhood* of v_i, denoted by $N(v_i)$, is defined as follows:

$$N(v_i) = \begin{cases} \emptyset & \text{if } v^*_i \text{ is an isolated vertex of } G^*; \\ P^* \cap \big(\bigcup_{j \sim i} HW^{>}_{j,i}\big) & \text{otherwise} \end{cases}.$$

We first show that two points of P^* that do not see each other must belong to neighborhoods associated with two distinct vertices of G^*.

Lemma 5.33 *Let x, y be two points of P^*. If x and y are not visible, then there are two integers i, j such that $i \sim j$, $x \in N(v_i)$, and $y \in N(v_j)$.*

Proof: Let l be the line segment $\overline{xy}$. Since x and y are not visible, l intersects the exterior of P^*. However, $l \subseteq P$ since P is convex, and so l intersects a wedge $W_{i,j}$. Let z belong to $l \cap W_{i,j}$. By Lemma 5.29, there exists a plane $H_x \in \{HF_{i,j}, HF_{j,i}, HW_{i,j}, HW_{j,i}\}$ such that x and z lie on different sides of H_x. Similarly, there exists a plane $H_y \in \{HF_{i,j}, HF_{j,i}, HW_{i,j}, HW_{j,i}\}$ such that z and y lie on different sides of H_y. Since $\overline{xy} \subseteq P$, neither H_x nor H_y belongs to $\{HF_{i,j}, HF_{j,i}\}$, and so either $H_x = HW_{i,j}$ and $H_y = HW_{j,i}$ or vice versa. We can assume without loss of generality that $H_x = HW_{i,j}$ and $H_y = HW_{j,i}$. This implies that $x \in HW^{\leq}_{i,j} \cap HW^{>}_{j,i}$ and that $y \in HW^{\leq}_{j,i} \cap HW^{>}_{i,j}$. Therefore $x \in N(v_i)$ and $y \in N(v_j)$, as required. □

In fact, if $x \in N(v_i)$, then there is an integer j such that $j \sim i$ and $x \in HW_{j,i}$, and hence x does not see v_j. We now prove that no two neighborhoods intersect.

Lemma 5.34 *If v_j^* and v_l^* are distinct vertices of G^*, then $N(v_j) \cap N(v_l) = \emptyset$.*

Proof: Suppose on the contrary that there are vertices v_j^*, v_l^* of G^*, and a point x of P^*, such that $x \in N(v_j) \cap N(v_l)$. Since $x \in N(v_j)$, there is an integer i such that $i \sim j$ and $x \in P^* \cap HW^{>}_{i,j}$. Similarly, there exists an integer k such that $k \sim l$ and $x \in P^* \cap HW^{>}_{k,l}$. Hence $x \in P^* \cap HW^{>}_{i,j} \cap HW^{>}_{k,l}$, which implies that $x \in P \cap HW^{>}_{i,j} \cap HW^{>}_{k,l}$. Since $j \neq l$, it follows from Lemma 5.27 that one of the three following cases occurs :

> Case 1 ($i = k$) : This implies that $P \cap HW^{\geq}_{i,j} \cap HW^{\geq}_{k,l} = \{v_i\}$. Therefore $x = v_i$, which is impossible since $v_i \notin HW^{>}_{i,j}$.
>
> Case 2 ($j = k$) : Lemma 5.28 implies that $x \in HW^{>}_{l,k}$. Hence $x \in P \cap HW^{>}_{k,l} \cap HW^{>}_{l,k}$, and so $x \in W_{k,l}$. This however contradicts the fact that $x \in P^*$.

Case 3 ($i = l$) : An argument similar to that presented in case 2 shows that this cannot occur.

Since a contradiction is reached in all cases, we conclude that $N(v_j) \cap N(v_l) = \emptyset$. □

Lemmas 5.34 implies that there is a point of P^* that does not belong to any neighborhood. It then follows from Lemma 5.33 that P^* is starshaped and that $x \in kr(P^*)$ if and only if it does not belong to any neighborhood. Hence $kr(P^*) = P^* \setminus \cup_{i=1}^{n} N(v_i)$. Each element of the cover of P^* obtained from a proper 3-coloring of G^* will be the union of $kr(P^*)$ with a number of neighborhoods. Such a union can also be defined in an alternate way that will simplify the proof of Lemma 5.36.

Lemma 5.35 *Let I be a subset of $\{1, \ldots, n\}$. If $Q = kr(P^*) \cup (\cup_{i \in I} N(v_i))$, then $Q = P^* \cap (\cap_{i \notin I, j \sim i} HW_{j,i}^{\leq})$.*

Proof: Since $P^* = kr(P^*) \cup (\cup_{i=1}^{n} N(v_i))$, we can rewrite Q as $P^* \setminus \cup_{i \notin I} N(v_i)$. By replacing $N(v_i)$ by its definition and expanding, the lemma follows. □

We are now ready to prove that P^* is U_3 if and only if G^* admits a proper 3-coloring.

Lemma 5.36 *Let G^* be the graph appearing in an instance of DS3C, and let P^* be the polytope obtained from G^*. P^* is U_3 if and only if G^* admits a proper 3-coloring.*

Proof: First suppose that P^* is U_3. Let $\mathcal{C} = \{C_1, C_2, C_3\}$ be a set of 3 convex polytopes that covers P^*. We can construct a 3-coloring χ of G^* by assigning to the vertex v_i^* of G^* the color k, where C_k is an element of $\mathcal{C}$ that contains v_i. Since $\mathcal{C}$ covers P^*, every vertex of G^* is assigned a color by χ (several assignments may be possible). Consider now two adjacent vertices v_i^*, v_j^* of G^*. Since $m_{i,j} \in W_{i,j}$, the points v_i and v_j are not visible, and hence no element of $\mathcal{C}$ contains both v_i and v_j. This implies that $\chi(v_i^*) \neq \chi(v_j^*)$. Since this holds for every pair of adjacent vertices of G^*, we conclude that χ is proper.

Suppose now that G^* admits a proper 3-coloring χ. For $i = 1, 2, 3$, let C_k be the union of $kr(P^*)$ with all neighborhoods of the vertices v_j of P^* for which $\chi(v_j^*) = i$, and let $\mathcal{C} = \{C_1, C_2, C_3\}$. Since χ is a 3-coloring of G^*, and since every point of P^* either belongs to a $kr(P^*)$ or to a neighborhood, $\mathcal{C}$ covers of P^*. Let x, y be two points of C_k. If either x or y belongs to $kr(P^*)$, or if x and y belong to the same neighborhood, then Lemma 5.33 implies that x sees y. Consider now the case where $x \in N(v_i)$ and $y \in N(v_j)$. Since $\chi(v_i^*) = \chi(v_j^*)$, it follows that $(v_i^*, v_j^*) \notin E^*$, and hence x sees y by Lemma 5.33. Therefore every pair of points of C_k is visible. It thus follows from Lemma 5.35 that C_k is closed and convex. Consequently P^* is U_3, as required. □

We observe that the proof of Lemma 5.36 remains identical if we replace "U_3" by "B_3", or in fact if we consider polytopes whose convex vertices can be covered by three convex subsets of the polytope instead. By combining Lemmas 5.32 and 5.36, we obtain the main result of this chapter.

Theorem 5.2 *Given a starshaped polytope P, and for each set S that is one of P, $bd(P)$, or the set of convex vertices of P, the problem of determining whether there are three convex subsets of P whose union contains S is NP-hard.*

Chapter 6

Conclusion

In this thesis, we considered several recognition problems for U_k and B_k polytopes. We first developed an $O(k^3|\mathcal{I}^*|^{2k-2} + T_M(k|\mathcal{I}^*|^{k-1}))$ time algorithm to determine whether a simple polygon is B_k with respect to a given partition $\mathcal{I}^*$ of its boundary into intervals. This algorithm divides sets of covers into equivalence classes, and uses matrix multiplication to merge sets of covers of two adjacent subchains of the polygon. Next, we characterized B_3 polygons by proving that every B_3 polygon admits a potential cover; that is, a cover in which every chord is potential. We then used this characterization and an adaptive partition of sets of covers into equivalence classes to obtain linear time algorithms to recognize U_3 and B_3 polygons in the plane.

Next, we showed how to reduce the problem of recognizing U_2 polytopes in $\boldsymbol{E}^d$ to that of two-coloring a graph G whose nodes are the connected components of the polytope minus its kernel. For the case $d = 3$, we gave a characterization of the way in which edges of G arise that allowed us to derive an $O(n \log n)$ time algorithm to recognize three-dimensional U_2 polytopes. The general algorithm runs in time polynomial in the size of the face lattice of the input polytope in each fixed dimension.

Finally, we proved that the recognition problem for U_k and B_k polytopes in $\boldsymbol{E}^d$ is NP-hard for every $d \geq 3$ and $k \geq 3$, and that in fact even determining whether there are three convex subsets of a polytope P in $\boldsymbol{E}^d$ that cover all convex vertices of P is NP-hard.

A number of issues remain unresolved. We now consider these issues, as well as several possible extensions of the work presented in this thesis. We first examine possible extensions of the techniques used to determine whether a simple polygon is B_k with respect to a given partition $\mathcal{I}^*$ of its boundary to covers by other types of simple polygon. We then consider the issues involved in generalizing the characterization of B_3 polygons given in Section 3.2. Next, we look at issues related to the recognition problem for U_2 polytopes in $\boldsymbol{E}^d$, and those that arise from our NP-hardness proof for the problem of recognizing U_3 and B_3 polytopes. Finally we examine the possibility of using the methods developed in this thesis to obtain good approximation algorithms.

6.1 Extending the methods developed to recognize polygons that are B_k with respect to $\mathcal{I}^*$

The methods used in Section 3.1 to determine whether a simple polygon is B_k with respect to a given partition $\mathcal{I}^*$ of its boundary into intervals can probably be extended to obtain similar algorithms for other problems in which only a constant amount of information about covers of two adjacent subchains of the polygon need to be maintained in order to determine covers of their union.

One problem to which it might be applicable is that of deciding whether there are k spiral subsets of a simple polygon (a *spiral* chain is one on which all of the reflex vertices are consecutive) that cover that polygon. We call a subset $\mathcal{I}$ of $\mathcal{I}^*$ *spiral* if its covering polygon is a spiral chain, and *semi-spiral* if that polygon minus the edge $\overline{ccw(last(\mathcal{I}))cw(first(\mathcal{I}))}$ is a spiral chain.

The main problem that needs to be solved in order to apply the techniques of Section 3.1 is to find conditions that are both necessary and sufficient for the union of two semi-spiral subsets to be semi-spiral. It would seem (intuitively) that one simply needs to know what the first and last intervals in the subset are, and whether they are part of a convex or reflex chain of the covering polygon. However in this case convex visibility can not be used to verify these conditions, and a more complicated method may be required. We think that this could be done using shortest paths.

We believe that it is likely that these details can be worked out, and that an extension of the algorithm that determines whether a simple polygon is B_k with respect to a partition $\mathcal{I}^*$ of its boundary into intervals can be used for that problem. Taking into account the slightly larger amount of information that needs to be maintained, and the corresponding increase in the number of equivalence classes, a preliminary analysis indicates that the running time of this algorithm would be in $O(k^3(8|\mathcal{I}^*|)^{2k-2} + T_M((8k|\mathcal{I}^*|)^{(k-1)}))$ time.

We note, however, that this would not solve the problem of determining whether the boundary of a simple polygon can be covered by k spiral polygons contained in that polygon, since the spiral chains thus obtained may not be simple. We are not sure whether this difficulty can be surmounted or not.

Open Problem 6.1 *Can we determine whether the boundary of a simple polygon P can be covered by k $\mathcal{I}^*$-elementary spiral subpolygons of P in time polynomial in the cardinality of $\mathcal{I}^*$ for each fixed k?*

An extension of the previous problem would be to consider covers of the boundary of P with r-spirals chains (chains whose reflex vertices can be divided into r reflex subchains) for constant r. However the problem of obtaining non-simple subsets of P seems to worsen as r increases, and we do not know how to deal with it at this point.

We might also want to consider covers by *monotone* polygons. A polytope P is *monotone* if there is a direction θ, called the *direction of monotonicity*, such that the intersection with P of every line perpendicular to a line with direction θ is either empty or connected. There are three types of cover by monotone subsets of P that we might want to consider: those in which θ is given, those in which θ is not given but all subsets have a common direction of monotonicity, and those in which there is no constraint on the direction of monotonicity.

Open Problem 6.2 *Can we determine whether the boundary of a simple polygon P can be covered by k $\mathcal{I}^*$-elementary monotone subpolygons of P (for any of the three types of cover) in time polynomial in the cardinality of $\mathcal{I}^*$ for each fixed k?*

We observe that the monotone subsets of P thus obtained may have coincident edges. We do not know, however, if their edges might cross. A simplicity problem similar to that which occurs with spiral subsets therefore needs to be considered. Another interesting problem would be to determine whether we can solve one of these covering problems using monotone polygons without knowing $\mathcal{I}^*$. This would require a characterization similar to that obtained for B_3 polygons in Section 3.2.

We also believe that the recognition algorithm for polygons that are B_k with respect to $\mathcal{I}^*$ can be extended to cover the boundary of orthogonal polygons with rectangles, and the boundary of simple polygons using $\mathcal{O}$-convex subsets. Finally, we do not think that these techniques can be extended to find covers of P by $\mathcal{I}^*$-elementary starshaped subsets. It seems that we would need to maintain the kernel of each subset, and since this may require $\Omega(n)$ information, the number of equivalence classes of covers might become exponential in n.

6.2 Recognizing B_k polygons in the plane

As seen in Chapter 2, determining whether U_k polygons can be recognized in NP has been an outstanding open problem in computational geometry for over a decade now. However determining whether P is B_k (for arbitrary fixed k) seems more tractable. This follows in part from the algorithm presented in Section 3.1. If one could compute in polynomial time a partition $\mathcal{I}^*$ of $bd(P)$ into intervals such that

1. P is B_k if and only if P is B_k with respect to $\mathcal{I}^*$,
2. the cardinality of $\mathcal{I}^*$ is in $O(n^{f(k)})$, and
3. every endpoint of an element of $\mathcal{I}^*$ can be represented using only $O(n^{g(k)})$ bits,

then we would obtain a polynomial time algorithm to recognize B_k polygons. This leads us to the following conjecture:

Conjecture 6.1 *For each fixed value of k, we can determine whether an input polygon P is B_k in time polynomial in the number of vertices of P.*

We observe that condition 3 is similar to the requirement that these endpoints have low order. The characterization proved in Section 3.2 shows that points of order 2 are sufficient for B_3 polygons.

We also note that in most NP-hardness results for covering problems [1,37,71,83, 90,91], a partition $\mathcal{I}^*$ of the boundary of the polygon obtained from the construction that satisfies conditions 1 to 3 can be found trivially (from the construction). Hence this style of construction will fail to prove that the problem of recognizing B_k polygons is NP-hard for any fixed value of k. This seems to support our conjecture.

The problem of finding such a partition for $k \geq 4$ remains open. It is possible that a generalization of the approach used in Section 3.2 for B_3 polygons might work. However, the characterization provided by Lemma 3.7 fails when $k \geq 4$, and we have not been able to find any generalization that might work. Such a generalization seems to be a prerequisite for any proof of existence of a cover of P in which the vertices of the cover elements are all located at points of the boundary of P of low order.

Open Problem 6.3 *For any value of $k \geq 4$, characterize the types of points on the boundary of P that may be required in every cover of the boundary of P by k convex subsets.*

6.3 Recognizing U_1, U_2 and U_3 polytopes

In Chapter 4, we gave an algorithm to recognize U_2 polytopes in $\boldsymbol{E}^d$. This algorithm starts by computing the kernel of P, and hence its running time can be as bad as $O(n^{\lfloor d/2 \rfloor})$ by the upper bound theorem [78,100]. However, we have reasons to believe that this is not necessary. More precisely, we can prove that if two convex subsets of a simple polytope P contain every edge of P, then they cover P.

This implies that we probably do not need to compute all connected components of $P \setminus kr(P)$, but that determining the edges of P (or pieces thereof) that belong to each of them would be enough. We observe that this can easily be done in $O(n^2)$ time, by intersecting each edge of P with the positive halfspaces determined by each facet of P. We also have the following conjecture:

Conjecture 6.2 *If x and y are not visible, then we can find points x', y' that belong to the same connected components of $P \setminus kr(P)$ as x and y, do not see each other, and lie on two edges of P contained in the boundary of adjacent facets of P.*

If Conjecture 6.2 holds, then we can compute G in low-order polynomial time, and hence we can recognize U_2 polytopes efficiently in all dimensions.

Open Problem 6.4 *Find an algorithm to recognize U_2 polytopes in $\boldsymbol{E}^d$ that runs in $O(n^\alpha)$ time for some small constant α that does not depend on d.*

On a related note, we stated in Section 2.3.4 that determining whether a simple polytope in $\boldsymbol{E}^d$ is convex can be determined in linear time by applying Tietze's theorem. However, this cannot be done so easily if the polytope is not assumed to be simple. In $\boldsymbol{E}^2$, we can nevertheless determine whether a given polygon P is convex in linear time by verifying that the vertices of P are sorted angularly around any point of P. Can an extension of this approach work in $\boldsymbol{E}^d$?

Open Problem 6.5 *Find a linear time algorithm to determine whether an input polytope in $\boldsymbol{E}^d$ that is not known to be simple is convex.*

Finally, the NP-hardness reduction given in Section 5.2 for the recognition problem for B_3 and U_3 polytopes in $\boldsymbol{E}^3$ implies that this problem is NP-hard in $\boldsymbol{E}^d$ for every $d \geq 3$, since we can use the same reduction and extend P^* into a cylinder along the remaining $d-3$ dimensions. In turn, this proof can then be easily modified by adding spikes to P^* to show that deciding whether a star-shaped polyhedron in $\boldsymbol{E}^d$ is U_k is NP-hard for every fixed values of $d \geq 3$ and $k \geq 3$. We note that every statement made about the difficulty of proving that the convex cover problem in the plane belongs to NP also holds here. We do not know, however, if the techniques used by O'Rourke to prove that the planar convex cover problem belongs to PSPACE can be extended to higher dimensions.

Open Problem 6.6 *Determine whether the convex cover problem in $\boldsymbol{E}^d$ belongs to PSPACE for $d \geq 3$.*

6.4 Approximation algorithms

An *approximation algorithm* for a covering problem is an algorithm that, given a polytope with n subfaces, returns a cover of the desired kind with at most $f(n,k)$

pieces, where k is the size of a minimum cover and f is some function. In practice, we may need solutions to a problem even if it is known to be NP-hard. Since finding an exact solution is impractical, an approximation algorithm may have to be used. Two possibly conflicting goals in the design of such an algorithm are the quality of the approximation, and the efficiency of the algorithm.

The only known approximation algorithms for covering problems are those of Ghosh [53] and of Aggarwal et al. [2] mentioned in Section 2.3.4. Both of these algorithms are for restricted starshaped subsets, however, and they only approximate the minimum cover of the given type within a factor of $O(\log n)$, i.e. with $f(n,k) \in O(k \log n)$. No approximation algorithm are known for covers by convex subsets.

Open Problem 6.7 *Find an algorithm that approximates the convex cover problem within a constant factor of the optimal ($f(n,k) \in O(k)$), or prove that none exists.*

We observe that since U_2 polytopes can be recognized in polynomial time, and since the problem of recognizing U_3 polytopes is NP-hard, Theorem 6.9 in the book by Garey and Johnson [50] implies that unless P = NP, no polynomial time approximation algorithm for the convex cover problem in $\boldsymbol{E}^d$ ($d \geq 3$) can provide a cover with less than 4/3 times as many pieces as the optimal cover.

We feel that the improved understanding of the convex cover problem gained through this thesis will lead to ways of finding good approximations. In particular, it may be possible to combine the techniques developed in Section 3.1 to determine whether a simple polygon is B_k with respect to a given partition $\mathcal{I}^*$ of its boundary into intervals with the characterization of B_3 polygons obtained in Section 3.2, and with a suitable decomposition of the input polygon, to obtain provably good convex covers of simple polygons. This is possibly the most important practical problem related to covering; our future research plans include the thorough investigation that it deserves.

Appendix A

Glossary[1]

adjacent [6] : two subfaces of a polytope P are adjacent if they have a common subface and neither is a subface of the other, or if they are the two endpoints of an edge.

approximation algorithm [131] : an approximation algorithm for a covering problem is an algorithm that, given a polytope with n subfaces, returns a cover of the desired kind with at most $f(n,k)$ pieces, where k is the size of a minimum cover and f is some function.

backward bounce point [44] : a backward bounce point of a simple polygon P is a point x of $bd(P)$ for which there is a reflex vertex v of P, and a point y of $bd(P)$ that is either a reflex vertex or an extension point of P, such that $v \in kr(P)$, both $chain[x,v]$ and $chain[v,y]$ are contained in a tip of P, and $x = hit(y,v)$.

backward extension point [44] : a backward extension point of a simple polygon P is a point x of $bd(P)$ for which there is a reflex vertex v_i of P such that $x = hit(v_{i+1}, v_i)$.

$\boldsymbol{B_k}$ [10] : a set S is B_k if there is a collection of k convex subsets of S whose union contains the boundary of S.

[1]The numbers between brackets indicate the number of the page on which the complete definition can be found.

B_k with respect to $\mathcal{I}^*$ [11] : a polygon P is B_k with respect to a partition $\mathcal{I}^*$ of its boundary into intervals if there is a collection of k convex $\mathcal{I}^*$-elementary subpolygons of P whose union contains the boundary of P.

bounce point [44] : a bounce point of a simple polygon P is a forward bounce point or a backward bounce point of P.

boundary [3] : the boundary of a set S in $\boldsymbol{E}^d$ is the set of all points of $\boldsymbol{E}^d$ that do not belong to either the interior or the exterior of S.

c-separated cover [66] : Given a chord c of a simple polygon P that divides P into two subpolygons Q_α and Q_β, a cover (Q_1, Q_2, Q_3) of P is c-separated if Q_1 is a convex subset of Q_α, Q_2 is a convex subset of Q_β, and Q_3 is a convex subpolygon of P that contains c.

chord of $\mathcal{I}$ [32] : if $\mathcal{I}$ is a subset of a partition of the boundary of a simple polygon into intervals, then a chord of $\mathcal{I}$ is the line segment joining $ccw(I)$ to $cw(succ_{\mathcal{I}}(I))$ for some element I of $\mathcal{I}$.

chord of P [5] : a chord of a simple polygon P is a line segment that does not intersect the exterior of P, and whose endpoints belong to the boundary of P.

chromatic number [14] : the chromatic number of a graph G is the integer k such that G is k-colorable but not $(k-1)$-colorable.

complete visibility [7] : two subsets S_1, S_2 of a set S are completely visible if each point of S_1 sees every point of S_2 in S.

composition [35] : the composition of two equal-length lists C, C' of subsets of $bd(P)$ is the list whose i^{th} element is the union of the i^{th} elements of C and C'.

constrained [46] : a cover of $bd(P)$ by subsets of $bd(P)$ is constrained if it contains 3 maximal elements and has minimum weight over all such covers.

convex angle [8] : an angle $\angle xyz$ is convex if z lies to the left or on the line through x and y, and oriented from x to y.

convex cover [34] : a cover of a set is convex if each of its elements is convex.

convex hull [8] : the convex hull of a set S is the smallest convex set that contains S.

convex partition [72] : Let P be a simple polytope, and $\mathcal{Q}^*$ be the set of connected components of $P \setminus kr(P)$. A partition of $\mathcal{Q}^*$ into two subsets is convex if the elements of each subset are completely visible pairwise.

convex set [8] : a set S is convex if every two points of S are visible.

convex subset of $\mathcal{I}^*$ [33] : let $\mathcal{I}^*$ be a partition of the boundary of a simple polygon into intervals. A subset of $\mathcal{I}^*$ is convex if it is semi-convex and $ccw(last(\mathcal{I}))$ sees $cw(first(\mathcal{I}))$ convexly.

convex vertex [8] : a vertex of a polytope P is convex if it is not reflex.

convex visibility [33] : a point x of $bd(P)$ sees a point y of $bd(P)$ convexly if the chord $\overline{xy}$ can be used as an edge, in a convex subpolygon of P, to join an interval of $bd(P)$ ending at x to an interval of $bd(P)$ starting at y.

counterclockwise traversal [5] : a counterclockwise traversal of the boundary of P is a traversal of $bd(P)$ in which the interior of P lies to the left of the directed line segment $\overrightarrow{v_i v_{i+1}}$ in the neighborhood of the midpoint of $\overrightarrow{v_i v_{i+1}}$.

cover [9] : a cover of a subset S' of a set S is a collection of subsets of S whose union contains S'.

covering polygon [32] : let $\mathcal{I}^*$ be a partition of the boundary of a simple polygon into intervals. The covering polygon for a subset $\mathcal{I}$ of $\mathcal{I}^*$ is the polygon obtained by connecting the counterclockwise endpoint of each element of $\mathcal{I}$ to the clockwise endpoint of the following element.

cross product [3] : the cross product of the two vectors (x_1, x_2, x_3) and (y_1, y_2, y_3) is the vector $(x_2 y_3 - y_2 x_3, x_3 y_1 - y_3 x_1, x_1 y_2 - y_1 x_2)$.

Delaunay subgraph [13] : a Delaunay subgraph is a subgraph of the Delaunay triangulation of a set of points.

Delaunay triangulation [13] : the Delaunay triangulation of a set V of points in the plane is a triangulation of V in which the circle through the three vertices of each internal face does not contain any point of V in its interior.

dent [25] : a dent of an orthogonal polygon P is an edge of P whose endpoints are both reflex.

direction of monotonicity: see *monotone.*

edge: see *polygon* or *graph.*

equivalent [36] : let $\mathcal{I}^*$ be a partition of the boundary of a simple polygon into intervals. Two lists C, C' of subsets of $\mathcal{I}^*$ are equivalent if for each pair $\mathcal{I}_i$, $\mathcal{I}_i'$ of corresponding elements of C and C', $\mathit{first}(\mathcal{I}_i) = \mathit{first}(\mathcal{I}_i')$ and $\mathit{last}(\mathcal{I}_i) = \mathit{last}(\mathcal{I}_i')$.

extension point [44] : an extension point of a simple polygon P is a forward extension point or a backward extension point of P.

extensional line segment [44] : let P be a simple polygon. A line segment is extensional if it joins an extension point x of P to the reflex vertex v_i of P for which $x = \mathit{hit}(w, v_i)$.

exterior [3] : the exterior of a set S is the interior of the complement of S.

face lattice [6] : the face lattice of a polytope P contains one vertex for each subface of P, and edges join pairs of subfaces one of which is a facet of the other.

facet: see *polytope.*

forward bounce point [44] : a forward bounce point of a simple polygon P is a point x of $bd(P)$ for which there is a reflex vertex v of P, and a point y of $bd(P)$ that is either a reflex vertex or an extension point of P , such that $v \in kr(P)$, both $\mathit{chain}[y, v]$ and $\mathit{chain}[v, x]$ are contained in a tip of P, and $x = \mathit{hit}(y, v)$.

forward extension point [44] : a forward extension point of a simple polygon P is a point x of $bd(P)$ for which there is a reflex vertex v_i of P such that $x = hit(v_{i-1}, v_i)$.

graph [12] : a graph is a pair (V, E), where V is a set of elements called vertices or nodes, and E is an irreflexive, symmetric relation on $V \times V$. Each element of E is an edge of the graph.

hidden set [10] : a hidden set inside a set S is a set of points of S, no two of which see each other.

hole [5] : a hole in a polygon P is a subpolygon of P whose boundary is disjoint from $bd(P)$.

$\mathcal{I}^*$-elementary [11] : let $\mathcal{I}^*$ be a partition of the boundary of a simple polygon P into intervals. A subpolygon of P is $\mathcal{I}^*$-elementary if its intersection with $bd(P)$ consists of a set of elements of $\mathcal{I}^*$ and reflex vertices of P.

incident [12] : an edge e of a graph G is incident upon a vertex v of G if v is one of the two vertices of G that define e.

induced subgraph [12] : an induced subgraph of a graph $G = (V, E)$ is a subgraph (V', E') of G in which E' contains every element $e = \{v, v'\}$ of E such that $v \in V'$ and $v' \in V'$.

inner product [3] : the inner product of two vectors $(x_1, \ldots, x_d)$ and $(y_1, \ldots, y_d)$ is the real number $\sum_{i=1}^{d} x_i y_i$.

interior [3] : the interior of a set S is the set of all points x of $\boldsymbol{E}^d$ for which there exists some positive real number ε such that $N_\varepsilon(x) \subseteq S$.

interior halfspace [8] : the interior halfspace of a facet f of a polytope P is the halfspace determined by f, and that contains points of P in the neighborhood of an interior point of f.

interval [5] : an interval is a closed, connected subset of the boundary of a simple polygon P that contains at least two distinct points.

isomorphic [12] : two graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* if there is a bijection $f : V \longrightarrow V'$ such that $\{v, v'\} \in E$ if and only if $\{f(v), f(v')\} \in E'$.

k-colorable [14] : a graph G is k-colorable if it admits a proper k-coloring.

k-coloring [13] : a k-coloring of a graph $G = (V, E)$ is a function $\chi : V \to \{1, \ldots, k\}$. A k-coloring of G is proper if no edge has both endpoints of the same color.

kernel [8] : the kernel of a set S is the set of all points of S from which every point of S is visible.

KR [18] : a set S is *KR* if every point of local nonconvexity of S belongs to $kr(S)$.

link visibility [7] : a point x of a set S is link-j visible from a point y of S if there are points $x = x_0, \ldots, x_j = y$ of S such that x_i sees x_{i+1} for each i in $\{0, \ldots, j-1\}$.

link-j convex [8] : a set S is link-j convex if every two points of S are link-j visible.

link-j visibility region [8] : the link-j visibility region from a point x in a set S is the set of points of S which are link-j visible from x.

local nonconvexity [7] : a point of local nonconvexity of a set S is a point x of S such that, for every $\varepsilon > 0$, there are two points of $S \cap N_\varepsilon(x)$ that do not see each other.

maximal convex set [70] : A convex subset of a polytope P is maximal if it is not strictly contained in any other convex subset of P.

maximal subset of $bd(P)$ [45] : a subset $\mathcal{I}$ of the boundary of a simple polygon P is maximal if it is $\mathcal{I}^P$-elementary, and if no $\mathcal{I}^P$-elementary superset of $\mathcal{I}$ has fewer chords, the same number of chords and more extensional chords, or the same number of chords and covers more reflex vertices of P.

monotone [128] : a polygon P is monotone if there is a direction θ such that the intersection with P of every line perpendicular to a line with direction θ is either empty or connected. The direction θ is called the direction of monotonicity.

neighborhood [122] : the neighborhood of a vertex v_i of a starshaped polytope P^* is the connected component of $P^* \setminus kr(P^*)$ that contains v_i.

neighborly line segment [44] : let P be a simple polygon. A line segment is neighborly if it is not reflex or extensional, and its endpoints belong to two adjacent tips of P.

node: see *graph.*

$\mathcal{O}$-visibility [27] : two points of a polytope are $\mathcal{O}$-visible if there is a path between them whose intersection with every line in the set $\mathcal{O}$ of orientations is either empty or connected.

orthogonal polygon [17] : a polygon is orthogonal if all of its edges are either horizontal or vertical.

orthogonal visibility [25] : two points x, y in an orthogonal polygon P are orthogonally visible if there is a path from x to y that intersects every line parallel to one of the coordinate axis in at most one connected component.

partition [10] : a partition of a set S in $\boldsymbol{E}^d$ is a cover of S in which the d-dimensional volume of the intersection of any two cover elements is zero.

planar graph [13] : a planar graph is a graph G for which there exist an embedding ψ such that (G, ψ) is a plane graph.

plane [4] : the plane through a point p with normal vector $\vec{n}$ is the set of all points x of $\boldsymbol{E}^d$ such that $\vec{n} \cdot (\overrightarrow{x - p}) = 0$.

plane graph [12] : a plane graph is a graph G together with an embedding ψ of the vertices and edges of G into the plane such that no two edges of G intersect except at a vertex upon which they are both incident.

point visibility graph [9] : the point visibility graph of a set S is the graph whose vertex set is the set of points of S, and in which two points are joined by an edge if they are visible in S.

polygon [4] : a polygon is a set defined by an ordered list of $n \geq 2$ points $v_1, \ldots, v_n$ in the plane called vertices, and n line segments $e_1, \ldots, e_n$ called edges, where e_i joins v_i to $v_{1+i \bmod n}$ for each i in $\{1, \ldots, n\}$.

polygon with holes [5] : a polygon with holes is a simple polygon, together with a collection of pairwise disjoint holes in that polygon.

polytope [6] : a polytope is a k-dimensional set bounded by a collection of $(k-1)$-dimensional sets called the facets of the polytope.

potential cover [44] : a convex cover of the boundary of a simple polygon P is potential if all of its chords are potential line segments contained in P.

potential line segment [44] : let P be a simple polygon. A line segment is potential if it is reflex or extensional, or if it is neighborly and its endpoints are potential points of P.

potential point [44] : a potential point of a simple polygon P is a reflex vertex of P, an extension point of P, or a bounce point of P.

proper k-coloring: see *k-coloring.*

property P_k [10] : a set S has property P_k if it does not contain a hidden set of size k.

reflex angle [8] : an angle $\angle xyz$ is reflex if z lies to the right of the line through x and y, and oriented from x to y.

reflex line segment [44] : let P be a simple polygon. A line segment is reflex if its endpoints are reflex vertices of P.

reflex subface [7] : a subface of a polytope P is reflex if it contains a point of local nonconvexity of P.

restriction [34] : let $\mathcal{I}^*$ be a partition of the boundary of a simple polygon into intervals. The restriction of a list C of subsets of $\mathcal{I}^*$ to a subset $\mathcal{I}$ of $\mathcal{I}^*$ is the list obtained by replacing each element of C by its intersection with $\mathcal{I}$.

semi-convex [33] : let $\mathcal{I}^*$ be a partition of the boundary of a simple polygon P into intervals. A subset of $\mathcal{I}^*$ is semi-convex if $ccw(I)$ sees $cw(succ_{\mathcal{I}}(I))$ convexly for each element I of $\mathcal{I}$ distinct from $last(\mathcal{I})$. A cover of $bd(P)$ is semi-convex if each of its elements is semi-convex.

semi-spiral [127] : let $\mathcal{I}^*$ be a partition of the boundary of a simple polygon into intervals. A subset of $\mathcal{I}^*$ is semi-spiral if its covering polygon (minus an edge) is a spiral chain.

simple polygon [4] : a polygon is simple if no two non-consecutive edges intersect, and if every pair of consecutive edges intersect in a point.

simple polytope [6] : a polytope P is simple if it is simply-connected, if no pair of non-adjacent subfaces of P intersect, and if the intersection of two adjacent subfaces of P is exactly their common subface.

simplex [8] : a d-dimensional simplex is the convex hull of $d+1$ points in $\boldsymbol{E}^d$.

simply-connected [6] : a polytope is simply-connected if no strict subset $\mathcal{F}$ of the set of facets of P has the property that each facet of an element of $\mathcal{F}$ is a facet of exactly one other element of $\mathcal{F}$.

size of a wedge [108] : the size of the wedge $W_{i,j}$ is the supremum of the perpendicular distance from $HT_{i,j}$ to a point of $W_{i,j}$.

slant of a plane [110] : the slant of an element H of $\mathcal{H}_{i,j}$ is the distance from $H \cap \rho_{i,j}$ to $HT_{i,j}$.

slant of a point [110] : the slant of a point x with respect to $e_{i,j}$ is the slant of $HS_{i,j}(x)$.

spiral [127] : a spiral chain is one on which all of the reflex vertices occur consecutively. If $\mathcal{I}^*$ is a partition of the boundary of a simple polygon into intervals, then a subset of $\mathcal{I}^*$ is spiral if its covering polygon is a spiral chain.

starshaped [8] : a set S is starshaped if there exists a point x of S from which every point of S is visible.

subface [6] : a subface of a polytope P is either P itself, or a subface of a facet of P.

subgraph [12] : a subgraph of a graph $G = (V, E)$ is a graph (V', E') such that $V' \subseteq V$ and $E' \subseteq E$.

subpolygon [5] : a subpolygon of a simple polygon P is a simple polygon contained in P.

tip [43] : a tip of P is a subset $chain[v_i, v_j]$ of $bd(P)$ where v_i and v_j are reflex vertices of P and no reflex vertex of P belongs to $chain(v_i, v_j)$.

triangulation [10] : a triangulation of a polytope P is a partition of P into simplices whose vertices are vertices of P. A triangulation of a set V of points in the plane is a maximal set E of edges for which (V, E) is a plane graph.

$\boldsymbol{U_k}$ [10] : a set S is U_k if there is a collection of k convex subsets of S whose union is S.

vertex: see *polygon* or *graph*.

vertex visibility graph [9] : the vertex visibility graph of a polytope P is the graph that has one node for every vertex of P, and in which two nodes are joined by an edge if the corresponding vertices of P are visible.

visibility [7] : two points x and y of a subset S of $\boldsymbol{E}^d$ are visible if the line segment $\overline{xy}$ is contained in S.

visibility region [8] : the visibility region from a point x in a set S is the set of points of S which are visible from x.

visually independent set: see *hidden set*.

wedge [108] : the wedge associated with an edge $e_{i,j}$ of P is a tetrahedral subset of P whose removal from P blocks visibility between v_i and v_j. The closed wedge associated with $e_{i,j}$ is the union of the wedge associated with $e_{i,j}$ with its boundary.

weight [46] : the weight of a cover of $bd(P)$ by convex subsets of P is the cardinality of the corresponding partition of $bd(P)$.

Appendix B

Notation[1]

Lowercase letters

$\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$: subscripts used for elements of $\mathcal{I}^*$.

$\boldsymbol{c}$: a chord.

$\boldsymbol{e_{i,j}}$ [106] : the edge of a polytope P whose endpoints are v_i and v_j.

$\boldsymbol{f}$: a subface of a polytope.

$\boldsymbol{f^{kr}}$: a subface of the kernel of a polytope.

$\boldsymbol{f_{i,j}}$ [106] : the unique facet of P in a counterclockwise traversal of whose boundary v_j immediately follows v_i.

$\boldsymbol{l_{i,j}}$ [109] : the line perpendicular to both $e_{i,j}$ and $\overrightarrow{nt}_{i,j}$, and containing v_i.

$\boldsymbol{m_i}$: the midpoint of edge e_i.

$\boldsymbol{m_{i,j}}$ [106] : the midpoint of edge $e_{i,j}$.

$\boldsymbol{m^*_{i,j}}$ [113] : a point of $\rho_{i,j}$ at distance at most $\delta_{i,j}$ from $HT_{i,j}$.

$\boldsymbol{n}$: the number of vertices of P.

[1]When useful, the numbers between brackets indicate the number of the page on which the notation is introduced.

$\overrightarrow{\boldsymbol{nf}}_{i,j}$ [106] : a vector normal to $HF_{i,j}$ and pointing towards the interior of P.

$\overrightarrow{\boldsymbol{np}}_{i,j}$ [106] : a vector oriented from v_i to v_j and normal to $HP_{i,j}$.

$\overrightarrow{\boldsymbol{ns}}_{i,j}(\boldsymbol{x})$ [110] : a vector normal to $HS_{i,j}(x)$.

$\overrightarrow{\boldsymbol{nt}}_{i,j}$ [106] : a vector normal to $HT_{i,j}$ and pointing towards P.

$\boldsymbol{q}_{i,j}$ [110] : the point of $e_{i,j}$ whose distance from v_i is a quarter of the length of $e_{i,j}$.

$\boldsymbol{q}^*_{i,j}$ [110] : the midpoint of the intersection of $f_{i,j}$ with $HP_{i,j}$.

$\boldsymbol{r}$: the number of reflex vertices of P.

$\boldsymbol{u}$, $\boldsymbol{u}_i$: a reflex vertex of P.

$\boldsymbol{v}$, $\boldsymbol{v}_i$: a vertex of P.

$\boldsymbol{w}_i$ [53] : the forward extension point $hit(v_{i-1}, v_i)$.

Uppercase letters

$\boldsymbol{C}$: a cover of a subset of $\mathcal{I}^*$.

$\boldsymbol{E}^*$ [119] : the set of edges of G^*.

$\boldsymbol{E}^d$ [3] : the d-dimensional Euclidean space.

$\boldsymbol{G}$: a graph.

$\boldsymbol{G}^*$ [104] : the graph in the instance of DS3C.

$\boldsymbol{H}$: a plane in $\boldsymbol{E}^3$.

$\boldsymbol{H}^{op}$ [4] : the set of all points x of $\boldsymbol{E}^d$ for which $\vec{n} \cdot (\overrightarrow{x-p})\ op\ 0$.

$\boldsymbol{HF}_{i,j}$ [106] : the plane containing $f_{i,j}$.

$\boldsymbol{HI}_{i,j}$ [109] : the plane parallel to $HP_{i,j}$ and containing v_i.

$\boldsymbol{HP}_{i,j}$ [106] : the plane normal to $e_{i,j}$ and containing $m_{i,j}$.

$\boldsymbol{HS_{i,j}(x)}$ [110] : the element of $\mathcal{H}_{i,j}$ that contains x.

$\boldsymbol{HT_{i,j}}$ [106] : a plane tangent to P along $e_{i,j}$.

$\boldsymbol{I}$: an interval, or an instance of three-satisfiability.

$\boldsymbol{M_{a,c}}$ [37] : the boolean matrix of size $|\mathit{first}_{a,c}| \times |\mathit{last}_{a,c}|$ in which an entry is true when the corresponding equivalence class contains a semi-convex cover.

$\boldsymbol{M_b}$ [38] : the matrix whose $(i,j)^{\text{th}}$ entry is 1 if and only if $\mathit{last}_{a,b}(i)$ sees $\mathit{first}_{b+1,c}(j)$ convexly.

$\boldsymbol{M'_b}$ [39] : the matrix corresponding to M_b in the general case: it encodes the convex visibility condition and the two types of consistency condition when merging sets of equivalence classes.

$\boldsymbol{N(v_i)}$ [122] : the neighborhood of a vertex v_i of P^*.

$\boldsymbol{N_\varepsilon(x)}$ [3] : an ε-neighborhood of x.

$\boldsymbol{P}$: a simple polygon or polytope.

$\boldsymbol{P^*}$ [104] : the polytope constructed by the reduction from DS3C.

$\boldsymbol{Q}$: a subpolygon of a simple polygon.

$\boldsymbol{U}$: a set of boolean variables or literals.

$\boldsymbol{V^*}$ [119] : the set of vertices of G^*.

$\boldsymbol{W_{i,j}}$ [108] : the wedge associated with $e_{i,j}$.

$\boldsymbol{X}$: a finite subset of the boundary of P.

Calligraphic letters

$\boldsymbol{\mathcal{C}}$, $\boldsymbol{\mathcal{C}'}$: two equivalence classes of covers.

$\boldsymbol{\mathcal{C}_{a,c}}$ [36] : the set of all equivalence classes of covers of $\{I_a, \ldots, I_c\}$.

$\mathcal{C}^*_{a,c}$ [63] : another set of equivalence classes of covers of $\{I_a, \ldots, I_c\}$.

$\mathcal{H}_{i,j}$ [109] : the set of all planes that contain $l_{i,j}$ and intersect $\rho_{i,j}$.

$\mathcal{I}$: a subset of $\mathcal{I}^*$.

$\mathcal{I}^*$: a partition of the boundary of a simple polygon into intervals.

$\mathcal{I}^P$ [45] : the partition of $bd(P)$ into intervals induced by a set of subsets of $bd(P)$, or a set of subpolygons of P.

$\mathcal{L}$: a list of vertices.

$\mathcal{Q}^*_i$ [71] : a connected component of a polytope minus its kernel.

$\mathcal{Q}^*$ [71] : the set of connected components of a polytope minus its kernel.

Greek letters

$\alpha_{i,j}$, $\beta_{i,j}$, $\gamma_{i,j}$ [110] : three distances used to determine ε.

$\delta_{i,j}$ [113] : a bound on the size of $W_{i,j}$.

ε [108] : an upper bound on the size of all wedges associated with edges of P.

$\lambda_{i,j}$ [109] : the line normal to $HT_{i,j}$ and containing $m_{i,j}$.

ϕ [119] : the inversion function $\phi(x,y) = (x, y, x^2 + y^2)$.

$\rho_{i,j}$ [109] : the open ray originating at $m_{i,j}$ in direction $\overrightarrow{nt}_{i,j}$.

χ: a proper k-coloring of a graph.

Other symbols

$\simeq$ [33] : $x \simeq y$ if x sees y convexly.

$\approx$ [12] : $G \approx H$ if G and H are isomorphic.

$\circ$ [35] : $C \circ C'$ is the composition of C and C'.

$|\ |$ [3] : the size of a set, or the length of a vector.

$\prec$ [43] : $x \prec y \prec z$ if x, y, z occur in this order in a counterclockwise traversal of $bd(P)$ starting at x.

Names

$\boldsymbol{last^*_{a,c}}$ [57] : a partition of $\{I_a, \ldots, I_c\}$ into equivalence classes symmetric to $first^*_{a,c}$.

$\boldsymbol{back(I_i)}$ [58] : the set of all elements I of $\mathcal{I}^* \setminus \{I_a, \ldots I_c\}$ such that $\overline{ccw(I)cw(I_i)}$ is potential and $ccw(I) \simeq cw(I_i)$.

$\boldsymbol{bd(S)}$ [3] : the boundary of a set S.

$\boldsymbol{ccw(I)}$ [32] : the counterclockwise endpoint of interval I.

$\boldsymbol{ch(S)}$ [8] : the convex hull of a set S.

$\boldsymbol{chain[x,y]}$ [43] : the set of all points z of $bd(P)$ such that $x \preceq z \preceq y$.

$\boldsymbol{chain[x,y)}$ [43] : the set of all points z of $bd(P)$ such that $x \preceq z \prec y$.

$\boldsymbol{chain(x,y]}$ [43] : the set of all points z of $bd(P)$ such that $x \prec z \preceq y$.

$\boldsymbol{chain(x,y)}$ [43] : the set of all points z of $bd(P)$ such that $x \prec z \prec y$.

$\boldsymbol{cw(I)}$ [32] : the clockwise endpoint of interval I.

$\boldsymbol{ext(S)}$ [3] : the exterior of a set S.

$\boldsymbol{first(\mathcal{I})}$ [32] : the first element of $\mathcal{I}$ in the cyclic ordering of the elements of $\mathcal{I}^*$.

$\boldsymbol{first(\mathcal{C})}$ [36] : the k-tuple containing the clockwisemost element of each subset of $\mathcal{I}^*$ in the covers in $\mathcal{C}$.

$\boldsymbol{first_i(\mathcal{C})}$ [36] : the i^{th} element of $first(\mathcal{C})$.

$\boldsymbol{first_{a,c}}$ [37] : the list of all k-tuples $\boldsymbol{x}$ for which there exists an element $\mathcal{C}$ of $\mathcal{C}_{a,c}$ such that $\boldsymbol{x} = first(\mathcal{C})$.

$\boldsymbol{first}^*_{a,c}$ [57] : a partition of $\{I_a, \ldots, I_c\}$ into equivalence classes.

$\boldsymbol{first}^1_{a,c}$, $\boldsymbol{first}^2_{a,c}$, $\boldsymbol{first}^3_{a,c}$ [58] : three subsets of $\mathit{first}_{a,c}$ used to compute $\mathit{first}^*_{a,c}$.

$\boldsymbol{hit(x,y)}$ [43] : the point z of ρ that maximizes the length of $\overline{yz}$, subject to the conditions that $\overline{yz} \subseteq P$ and $\overline{yz} \cap bd(P)$ is finite.

$\boldsymbol{int(S)}$ [3] : the interior of a set S.

$\boldsymbol{kr(S)}$ [8] : the kernel of a set S.

$\boldsymbol{last(\mathcal{I})}$ [32] : the last element of $\mathcal{I}$ in the cyclic ordering of the elements of $\mathcal{I}^*$.

$\boldsymbol{last(\mathcal{C})}$ [36] : the k-tuple containing the counterclockwisemost element of each subset of $\mathcal{I}^*$ in the covers in $\mathcal{C}$.

$\boldsymbol{last_i(\mathcal{C})}$ [36] : the i^{th} element of $\mathit{last}(\mathcal{C})$.

$\boldsymbol{last_{a,c}}$ [37] : the list of all k-tuples $\boldsymbol{x}$ for which there exists an element $\mathcal{C}$ of $\mathcal{C}_{a,c}$ such that $\boldsymbol{x} = \mathit{last}(\mathcal{C})$.

$\boldsymbol{rep_{a,c}(\mathcal{C})}$ [63] : the pair obtained from $\mathcal{C}$ by replacing each element of $\mathit{first}(\mathcal{C})$ by its class representative under $\mathit{first}^*_{a,c}$, and each element of $\mathit{last}(\mathcal{C})$ by its class representative under $\mathit{last}^*_{a,c}$.

$\boldsymbol{rest(C,\mathcal{I})}$ [34] : the restriction of C to $\mathcal{I}$

$\boldsymbol{slant_{i,j}(x)}$ [110] : the slant of x with respect to $e_{i,j}$.

$\boldsymbol{succ_{\mathcal{I}}(I)}$ [32] : the element of $\mathcal{I}$ immediately following I in the cyclic ordering of the elements of $\mathcal{I}^*$.

Bibliography

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

[2] A. Aggarwal, S. K. Ghosh, and R. K. Shyamasundar. Computational complexity of restricted polygon decompositions. In G. T. Toussaint, editor, *Computational Morphology.* North Holland, 1988.

[3] S. G. Akl. *Parallel sorting algorithms.* Academic Press, 1985.

[4] S. G. Akl. *The design and analysis of parallel algorithms.* Prentice Hall, 1989.

[5] T. Asano and H. Imai. Efficient algorithms for geometric graph search problems. *SIAM Journal on Computing,* 15:478–494, 1986.

[6] L. J. Aupperle, H. E. Conn, J. M. Keil, and J. O'Rourke. Covering orthogonal polygons with squares. In *Proceedings of the Twenty-Sixth Allerton Conference,* pages 97–106, October 1988.

[7] F. Bagemihl. On indecomposable polyhedra. *American Mathematical Monthly,* pages 411–413, 1948.

[8] R. Bar-Yehuda and E. Ben-Chanoch. An $O(n \log^* n)$ time algorithm for covering simple polygons with squares. In *Proceedings of the Second Canadian Conference on Computational Geometry,* pages 186–190, 1990.

[9] P. Belleville. Computing two-covers of simple polygons. Master's thesis, McGill University, Montréal, September 1991.

[10] P. Berman and B. DasGupta. Approximating the rectilinear polygon cover problem. In *Proceedings of the Fourth Canadian Conference on Computational Geometry*, pages 229–235, 1992.

[11] P. Berman and B. DasGupta. Results on approximation of the rectilinear cover problem. Technical Report CS-92-07, The Pennsylvania State University, 1992.

[12] J. A. Bondy and U. S. R. Murty. *Graph theory with applications.* MacMillan and American Elsevier, 1976.

[13] M. Breen. The combinatorial structure of (m, n)-convex sets. *Israel Journal of Mathematics*, 15:367–374, 1973.

[14] M. Breen. An example concerning unions of two starshaped sets in the plane. *Israel Journal of Mathematics*, 17:347–349, 1974.

[15] M. Breen. A decomposition theorem for m-convex sets. *Israel Journal of Mathematics*, 24:211–216, 1976.

[16] M. Breen. An $\mathbf{R}^d$ analogue of Valentine's theorem on 3-convex sets. *Israel Journal of Mathematics*, 24:206–210, 1976.

[17] M. Breen. Clear visibility and unions of two starshaped sets in the plane. *Pacific Journal of Mathematics*, 115(2):267–275, 1984.

[18] M. Breen. A characterization theorem for compact unions of two starshaped sets in $\mathbf{R}^3$. *Pacific Journal of Mathematics*, 128(1):63–72, 1987.

[19] M. Breen. Characterizing compact unions of two starshaped sets in $\mathbf{R}^d$. *Journal of Geometry*, 35:14–18, 1989.

[20] M. Breen. An intersection for starshaped sets in the plane. *Geometria Dedicata*, 37(3):287–294, 1991.

[21] M. Breen and D. Kay. General decomposition theorems for m-convex sets in the plane. *Israel Journal of Mathematics*, 24:217–233, 1976.

[22] M. Breen and T. Zamfirescu. A characterization theorem for certain unions of two starshaped sets in $\mathbf{R}^2$. *Geometria Dedicata*, 6:95–103, 1987.

[23] D. Bremner. Point visibility graphs and restricted-orientation polygon covering. Master's thesis, Simon Fraser University, Burnaby, April 1993.

[24] D. Bremner and T.C. Shermer. Point visibility graphs and restricted-orientation convex cover. Technical Report CMPT TR 93–07, Simon Fraser University, 1993.

[25] E. O. Buchman. Property P_3 and the union of two convex sets. *Proceedings of the American Mathematical Society*, 25:642–645, 1970.

[26] J. Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the Twentieth Annual ACM Symposium on the Theory of Computing*, pages 460–467, 1988.

[27] B. Chazelle. *Computational geometry and convexity.* PhD thesis, Yale University, Yale, 1980.

[28] B. Chazelle. Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM Journal on Computing*, 13:488–507, 1984.

[29] B. Chazelle. Fast searching in real algebraic manifold with applications to geometric complexity. In Springer-Verlag, editor, *Proceedings of the Colloquium on Trees in Algebra and Programming*, pages 145–156, 1985.

[30] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete and Computational Geometry*, 6(5):485–524, 1991.

[31] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete and Computational Geometry*, 10:377–409, 1993.

[32] B. Chazelle and D. P. Dobkin. Optimal convex decompositions. In G. T. Toussaint, editor, *Computational Geometry*. North Holland, 1985.

[33] K. L. Clarkson. Linear programming in $O(n3^{d^2})$ time. *Information Processing Letters*, 22:21–24, 1986.

[34] H. E. Conn and J. O'Rourke. Some restricted rectangle covering problems. Technical Report JHU-87/13, The Johns Hopkins University, 1987.

[35] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computations*, 9:251–280, 1990.

[36] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms.* MIT Press, 1989.

[37] J. Culberson and R. A. Reckhow. Covering polygons is NP-hard. In *Proceedings of the Twenty-Ninth IEEE Annual Symposium on the Foundations of Computer Science*, pages 601–611, 1988.

[38] J. Culberson and R. A. Reckhow. Orthogonally convex coverings of orthogonal polygons without holes. *Journal of Computer and System Sciences*, 39:166–204, 1989.

[39] L. Danzer, B. Grünbaum, and V. Klee. Helly's theorem and its relatives. In *Convexity, Proc. Symposia in Pure Mathematics.* American Mathematical Society, 1963.

[40] M. E. Dyer. On a multidimensional search technique and its application to the euclidean one-center problem. *SIAM Journal on Computing*, 15:725–738, 1986.

[41] H. Edelsbrunner. *Algorithms in combinatorial geometry.* Springer-Verlag, 1988.

[42] H. Edelsbrunner, L. Guibas, and J. Stolfi. Optimal point location in a monotone subdivision. *SIAM Journal on Computing*, 15:317–340, 1986.

[43] H. G. Eggleston. A condition for a compact plane set to be a union of finitely many convex sets. *Mathematical Proceedings of the Cambridge Philosophical Society*, 76:61–66, 1974.

[44] H. G. Eggleston. Valentine convexity in n dimensions. *Mathematical Proceedings of the Cambridge Philosophical Society*, 80:223–228, 1976.

[45] H. ElGindy. *Hierarchical decomposition of polygons with applications.* PhD thesis, McGill University, Montréal, 1985.

[46] H. ElGindy and D. Avis. A linear algorithm for computing the visibility polygon from a point. *Journal of Algorithms*, 2:186–197, 1981.

[47] I. Fary. On straight line representation of planar graphs. *Acta Sci. Math. Szeged*, 11:229–233, 1948.

[48] H-Y. F. Feng and T. Pavlidis. Decomposition of polygons into simple components: feature generation for syntactic pattern recognition. *IEEE Transactions on Computers*, C–24:636–650, 1975.

[49] D. Franzblau and D. Kleitman. An algorithm for constructing regions with rectangles. In *Proceedings of the Sixteenth Annual ACM Symposium on the Theory of Computing*, pages 167–174, 1984.

[50] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness.* W. H. Freeman and Company, 1979.

[51] M. R. Garey, D. S. Johnson, F. P. Preparata, and R. E. Tarjan. Triangulating a simple polygon. *Information Processing Letters*, 7:175–179, 1978.

[52] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1975.

[53] S. K. Ghosh. Approximation algorithms for art gallery problems. In *Proceedings of the Canadian Information Processing Society Congress*, 1987.

[54] J. E. Goodman, R. Pollack, and B. Sturmfeld. Coordinate representation of order types requires exponential storage. In *Proceedings of the Fifth ACM Symposium on Computational Geometry*, pages 405–410, 1989.

[55] B. Grünbaum. *Convex polytopes.* John Wiley & sons, 1967.

[56] M. Guay and D. Kay. On sets having finitely many points of local nonconvexity and property P_m. *Israel Journal of Mathematics*, 10:196–209, 1971.

[57] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan. Linear time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.

[58] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4:74–123, 1985.

[59] W. R. Hare Jr. and J. W. Kenelly. Sets expressible as unions of two convex sets. *Proceedings of the American Mathematical Society*, 25:379–380, 1970.

[60] E. Helly. über mengen konvexer körper mit gemeinschaftlichen punkten. *Jahresber. Deutsches Mathematiks-Verein.*, 32:175–176, 1923.

[61] J. Hershberger. Finding the visibility graph of a polygon in time proportional to its size. In *Proceedings of the Third ACM Symposium on Computational Geometry*, pages 11–20, 1987.

[62] J. F. Hurley. *Intermediate calculus.* Saunders College, 1980.

[63] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–397, 1984.

[64] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of computer computations*, pages 85–103. Plenum Press, 1972.

[65] Y. Ke. *Efficient algorithms for weak visibility and link distances problems in polygons.* PhD thesis, The Johns Hopkins University, Baltimore, September 1989.

[66] J. M. Keil. *Decomposing a polygon into simpler components.* PhD thesis, University of Toronto, Toronto, 1983.

[67] J. M. Keil. Decomposing a polygon into simpler components. *SIAM Journal on Computing*, 14:799–817, 1985.

[68] L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Math. Doklady*, 20:191–194, 1979.

[69] D. G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM Journal on Computing*, 12:28–35, 1983.

[70] D. T. Lee. Visibility of a simple polygon. *Computer Vision, Graphics, and Image Processing*, 22:207–221, 1983.

[71] D. T. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32:276–282, 1986.

[72] D. T. Lee and F. P. Preparata. An optimal algorithm for finding the kernel of a polygon. *Journal of the ACM*, 26:415–421, 1979.

[73] R. Lee, Liou W., and J. Tan. Minimum partitioning simple rectilinear polygons in $O(n \log\log n)$ time. In *Proceedings of the Fifth ACM Symposium on Computational Geometry*, pages 344–353, 1989.

[74] N. J. Lennes. Theorems on the simple finite polygon and polyhedron. *American Journal of Mathematics*, 33:37–62, 1911.

[75] W. Masek. Some NP-complete set covering problems. manuscript, MIT (cited in [83], p. 232), 1978.

[76] R. L. McKinney. On unions of two convex sets. *Canadian Journal of Mathematics*, 18:883–886, 1966.

[77] R. L. McKinney. Notices of the American Mathematical Society, 1970. Abstract number 672–575.

[78] P. McMullen. The maximum number of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.

[79] R. Motwani, A. Raghunathan, and H. Saran. Covering orthogonal polygons with star polygons. *Journal of Computer and System Sciences*, 40:19–48, 1989.

[80] J. O'Rourke. The complexity of computing minimum convex covers for polygons. In *Proceedings of the twentieth Allerton Conference*, pages 75–84, 1982.

[81] J. O'Rourke. The decidability of covering by convex polygons. Technical Report JHU/EECS-82/4, The Johns Hopkins University, 1982.

[82] J. O'Rourke. Minimum convex covers for polygons: some counterexamples. Technical Report JHU/EECS-82/1, The Johns Hopkins University, 1982.

[83] J. O'Rourke. *Art gallery theorems and algorithms.* Oxford University Press, Inc., 1987.

[84] J. O'Rourke and K. Supowit. Some NP-hard polygon decomposition problems. *IEEE Transactions on Information Theory*, 29:181–190, 1983.

[85] F. P. Preparata and M. I. Shamos. *Computational geometry, an introduction.* Springer-Verlag, second edition, 1988.

[86] A. Raghunathan. *Polygon decomposition and perfect graphs.* PhD thesis, University of California at Berkeley, Berkely, 1988.

[87] J. Ruppert and R. Seidel. On the difficulty of tetrahedralizing three-dimensional nonconvex polyhedra. *Discrete and Computational Geometry*, 7:227–253, 1982.

[88] E. Schönhardt. Über die zerlegung von dreieckspolhedern in tetraheder. *Mathematische Annalen*, 98:309–312, 1928.

[89] T. C. Shermer. Hiding people in polygons. *Computing*, 42:109–131, 1989.

[90] T. C. Shermer. *Visibility properties of polygons.* PhD thesis, McGill University, Montréal, June 1989.

[91] T. C. Shermer. Recent results in art galleries. *IEEE Proceedings*, 80(9):1384–1399, September 1992.

[92] T. C. Shermer. On recognizing unions of two convex polygons and related problems. *Pattern Recognition Letters*, 14(9):737–745, 1993.

[93] V. Soltan and A. Gorpinevich. Minimum dissection of a rectilinear polygon with arbitrary holes into rectangles. *Discrete and Computational Geometry*, 9(1):57–79, 1993.

[94] W. Stamey and J. Marr. Unions of two convex sets. *Canadian Journal of Mathematics*, 15:152–156, 1963.

[95] S. Suri. *Minimum link paths in polygons and related problems.* PhD thesis, The Johns Hopkins University, Baltimore, 1987.

[96] I. E. Sutherland and G. W. Hodgman. Reentrant polygon clipping. *Communications of the ACM*, 17(1):32–42, 1974.

[97] H. Tietze. Über konvexheit im kleinen und im grossen und über gewisse den punkten einer menge zugeordnete dimensionszahlen. *Mathematische Zeitschrift*, 28:697–707, 1928.

[98] G. T. Toussaint, C. Verbrugge, C. Wang, and B. Zhu. Tetrahedralization of simple and non-simple polyhedra. In *Proceedings of the Fifth Canadian Conference on Computational Geometry*, pages 24–29, 1993.

[99] F. Valentine. A three point convexity property. *Pacific Journal of Mathematics*, 7:1227–1235, 1957.

[100] G. M. Ziegler. *Lectures on polytopes.* Number 152 in Graduate Texts in Mathematics. Springer-Verlag, 1995.