Universidad de Alcalá
ESCUELA DE DOCTORADO

Universidad de Alcalá

Programa de Doctorado en Ciencias

Problems and applications of Discrete and Computational Geometry concerning graphs, polygons, and points in the plane

Thesis submitted by

ALEJANDRA MARTÍNEZ MORAIAN

Advisors:

DR. DAVID ORDEN MARTÍN AND DR. IVÁN MARSÁ MAESTRE

Alcalá de Henares, 2022

Acknowledgements

I want to start thanking my advisors, David Orden and Iván Marsá for their continuous support and kindness. Thank you for giving me the opportunity to start a career in academia, and for being always willing to have a discussion or a lunch with me. During these discussions, I have learned most of the tools that have allowed me to complete my PhD. Working with them has been immensely enriching. I specially want to thank David, who was also my tutor, for guiding me wisely during my PhD studies.

I am also very thankful to Paco Santos for selecting me to hold the FPI scholarship that has funded my PhD studies.

I am very grateful to Mercè Claverol and Clemens Huemer for introducing me to the world of research during my Master's studies. I have learned a lot from them and I have really enjoyed our work together. This work has continued until now, with Andrea de las Heras joining us later. I am very lucky that I had the opportunity to work with the three of them.

I am really thankful to my other coauthors: Encarnación Fernández, José Manuel Giménez, Leonidas Palios, Rene D. Reyes-Bardales, Carlos Seara, Marino Tejedor, and Paweł Żyliński. In particular, I am grateful to Paweł for hosting me during my research stay in Gdansk University. Working with him was rewarding and fun, and my experience in Gdansk was unforgettable. I am also thankful to Carlos for hosting me in Polytechnic University of Catalonia for several research visits.

The researchers at Alfréd Rényi Institute of Mathematics in Budapest have inspired me with their love for beautiful mathematics. Special thanks to Géza Tóth, Gabor Kun and Nika Salia.

Finally, I want to thank my family, and specially my husband, Diego, who is always there for me.

Abstract

This thesis deals with problems and applications of discrete and computational geometry in the plane, concerning polygons, point sets, and graphs.

After a first introductory chapter, in Chapter 2 we study a generalization of a famous visibility problem in the framework of $\mathcal{O}$-*convexity*. Given a set of orientations (angles) $\mathcal{O}$, we say that a curve is $\mathcal{O}$-*convex* if its intersection with any line parallel to an orientation in $\mathcal{O}$ is connected. When $\mathcal{O} = \{0°, 90°\}$, we find ourselves in the case of orthoconvexity, considered of special relevance. The $\mathcal{O}$-*kernel* of a polygon is the subset of points of the polygon that can be connected to any other point of the polygon with an $\mathcal{O}$-convex curve. In this work we obtain, for $\mathcal{O} = \{0°\}$ and $\mathcal{O} = \{0°, 90°\}$, an algorithm to compute and maintain the $\mathcal{O}$-kernel of a polygon as the set of orientations $\mathcal{O}$ rotates. This algorithm also provides the angles of rotation that maximize the area and perimeter of the $\mathcal{O}$-kernel.

In Chapter 3, we consider a bichromatic version of a combinatorial problem posed by Neumann-Lara and Urrutia. Specifically, we prove that every set of n blue and n red points in the plane contains a bichromatic pair of points such that every circle having them on its boundary contains at least $n(1 - \frac{1}{\sqrt{2}}) - o(n)$ points of the set in its interior. This problem is closely related to obtaining the higher order Voronoi diagrams of the point set. The edges of these diagrams contain, precisely, all the centers of the circles that pass through two points of the set. Therefore, our current line of research on this problem consists on exploring this connection by studying in detail the properties of higher order Voronoi diagrams.

In Chapters 4 and 5, we consider two applications of graph theory to sensory analysis and air traffic management, respectively. In the first case, we introduce a new method which combines geometric and statistical techniques to analyze consumer opinions, collected through projective mapping. This method is a variation of the method SensoGraph. It aims to capture

the essence of projective mapping by computing the Ecuclidean distances between pairs of samples and normalizing them to the interval $[0, 1]$. We apply the method to a real-life scenario and compare its performance with the performance of classic methods of sensory analysis over the same data set. In the second case, we use the Spectrum Graph Coloring technique to propose a model for air traffic management that aims to optimize the amount of fuel used by the airplanes, while avoiding collisions between them.

Resumen

Esta tesis trata sobre problemas y aplicaciones de la geometría discreta y computacional en el plano, relacionados con polígonos, conjuntos de puntos y grafos.

Después de un primer capítulo introductorio, en el capítulo 2 estudiamos una generalización de un famoso problema de visibilidad en el ámbito de la $\mathcal{O}$-*convexidad.* Dado un conjunto de orientaciones (ángulos) $\mathcal{O}$, decimos que una curva es $\mathcal{O}$-*convexa* si su intersección con cualquier recta paralela a una orientación de $\mathcal{O}$ es conexa. Cuando $\mathcal{O} = \{0°, 90°\}$, nos encontramos en el caso de la ortoconvexidad, considerado de especial relevancia. El $\mathcal{O}$-*núcleo* de un polígono es el conjunto de puntos del mismo que pueden ser conectados con cualquier otro punto del polígono mediante una curva $\mathcal{O}$-convexa. En este trabajo obtenemos, para $\mathcal{O} = \{0°\}$ y $\mathcal{O} = \{0°, 90°\}$, un algoritmo para calcular y mantener el $\mathcal{O}$-núcleo de un polígono conforme el conjunto de orientaciones $\mathcal{O}$ rota. Dicho algoritmo proporciona, además, los ángulos de rotación para los que el $\mathcal{O}$-núcleo tiene área y perímetro máximos.

En el capítulo 3 consideramos una versión bicromática de un problema combinatorio planteado por Neumann-Lara y Urrutia. En concreto, demostramos que todo conjunto de n puntos azules y n puntos rojos en el plano contiene un par bicromático de puntos tal que todo círculo que los tenga en su frontera contiene en su interior al menos $n(1-\frac{1}{\sqrt{2}})-o(n)$ puntos del conjunto. Este problema está fuertemente ligado al cálculo de los diagramas de Voronoi de orden superior del conjunto de puntos, pues las aristas de estos diagramas contienen precisamente todos los centros de los círculos que pasan por dos puntos del conjunto. Por ello, nuestra línea de trabajo actual en este problema consiste en explorar esta conexión realizando un estudio detallado de las propiedades de los diagramas de Voronoi de orden superior.

En los capítulos 4 y 5, planteamos dos aplicaciones de la teoría de grafos

al análisis sensorial y al control del tráfico aéreo, respectivamente. En el primer caso, presentamos un nuevo método que combina técnicas estadísticas y geométricas para analizar las opiniones de los consumidores, recogidas a través de mapeo proyectivo. Este método es una variación del método SensoGraph y pretende capturar la esencia del mapeo proyectivo mediante el cálculo de las distancias euclídeas entre los pares de muestras y su normalización en el intervalo $[0, 1]$. A continuación, aplicamos el método a un ejemplo práctico y comparamos sus resultados con los obtenidos mediante métodos clásicos de análisis sensorial sobre el mismo conjunto de datos. En el segundo caso, utilizamos la técnica del espectro-coloreado de grafos para plantear un modelo de control del tráfico aéreo que pretende optimizar el consumo de combustible de los aviones al mismo tiempo que se evitan colisiones entre ellos.

Contents

Chapter 1

Introduction

In this work, we consider several questions in the area of discrete and computational geometry. We present a summary of the research that we have carried out on these questions during the last years, which has produced the publication of four scientific articles. These articles are reproduced in Chapters 2, 3, 4, and 5, respectively.

In Chapter 2, we study a classic problem about polygons from the point of view of restricted orientation convexity. Chapter 3 is related to higher order Voronoi diagrams of point sets. And Chapters 4 and 5 deal with applications of graph theory to sensory analysis and air traffic management, respectively. For an easier reading, each of these chapters is introduced individually. We start our introduction with some notation.

1.1 Notation

Three main groups of geometric objects appear in this thesis: points in the plane, polygons, and graphs.

Points in the plane are usually required to be in *general position*. A set S of n points in the plane is in general position if no three points of S are collinear and no four points are cocircular.

A *polygon* is a closed and connected subset of the plane, bounded by a finite number of straight-line segments that form a closed broken line. The segments bounding a polygon are called *edges*, and their endpoints, *vertices*. A polygon is called *simple* if its edges do not intersect each other except at vertices. A simple polygon with k vertices is called a *k-gon*. A simple polygon (or a set) is called *convex* if its intersection with every line in the plane is either empty or connected; see Figure 1.1.

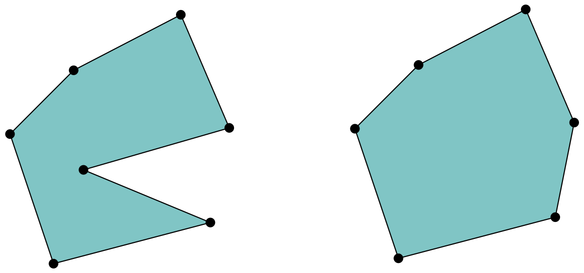


Figure 1.1: Left: a simple polygon on 7 vertices, or 7-gon. Right: a convex polygon on 7 vertices, or convex 7-gon.

The symbol $\mathcal{O}$ is reserved to denote a set of orientations. An *orientation* is an angle α measured with respect to the positive x-axis. Such an angle is associated with the oriented line ℓ_α, passing through the origin of coordinates and forming a counterclockwise angle α with the positive x-axis. A set is *$\mathcal{O}$-convex* if, for every $\alpha \in \mathcal{O}$, the intersection of the set with any line parallel to ℓ_α is either empty or connected. When $\mathcal{O} = \{0°, 90°\}$, an *$\mathcal{O}$-convex* set is called *rectilinear convex* or *orthoconvex*. Observe that, unlike in the case of usual convex sets, $\mathcal{O}$-convex sets might be disconnected; see Figure 1.2.

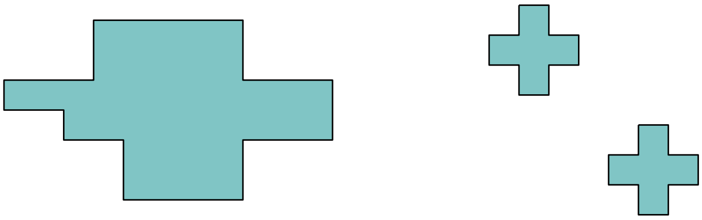


Figure 1.2: Left: a $\{0°, 90°\}$-convex set or orthoconvex set. Right: a disconnected orthoconvex set.

A graph is a pair $G = (V, E)$ where $E \subseteq V \times V$. The elements in the collection V are usually called vertices and the elements in the collection E are usually called edges. Two vertices u and v are *adjacent* if $(u, v) \in E$. A *vertex coloring* of $G = (V, E)$ is a map $c : V \to S$, where S is a collection of colors. A vertex coloring is called *proper* if, for every two adjacent vertices u and v, $c(u) \neq c(v)$.

The usual way to picture a graph is by drawing vertices as points and joining two of these points with a curve if the corresponding vertices form an edge of the graph. This is called a *drawing of a graph*; see Figure 1.3.

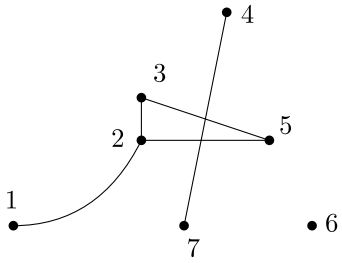


Figure 1.3: A drawing of the graph $G = (V, E)$ with $V = \{1, 2, 3, 4, 5, 6, 7\}$ and $E = \{(1, 2), (2, 3), (2, 5), (3, 5), (4, 7)\}$.

When discussing algorithmic aspects of problems, we will make use of the asymptotic notation. For two functions f and g of $n \in \mathbb{N}$, if we write $f(n) \in O(g(n))$ it means that there exist constants $C > 0$ and $n_0 > 0$ such that for all $n \geq n_0$, $|f(n)| \leq C\,|g(n)|$. Similarly, $f(n) \in o(g(n))$ means that for every $\varepsilon > 0$ there exists $n_0 > 0$ such that for all $n > n_0$, $|f(n)| < \varepsilon\,|g(n)|$. Finally, $f(n) \in \Theta(g(n))$ means that $f(n) \in O(g(n))$ and $g(n) \in O(f(n))$.

1.2 Preliminaries about restricted orientation convexity

Chapter 2 concerns restricted orientation convexity, or $\mathcal{O}$-convexity, a generalization of convexity introduced by Güting [47]. Convexity is a geometric property connected to multiple areas of mathematics. Its importance arises from its suitability to model a wide range of real-life interesting problems, and the fact that it is a great source of simple examples for properties in different areas of mathematics, including analysis, statistics, and linear algebra [38].

Restricted orientation geometry studies geometric objects whose edges are parallel to a predefined set of orientations [111]. Orthogonal geometry and Euclidean geometry can be regarded as particular cases of this when the set of predefined orientations is, respectively, the set $\{0°, 90°\}$ or the set of all possible orientations. One example motivating the interest in restricted orientation geometry is given by the fact that orthogonal objects can often be handled more efficiently than general ones, and they appear in multiple real-life scenearios. For instance, digital picture processing and VLSI design [111, 112].

Among the real-life concepts that can be studied through convexity, we note visibility problems, and, among those, the problem of guarding a room (polygon). In this problem, there is a room shaped like a simple polygon, where we need to place a number of guards so that they altogether watch the entire room, that is, without blind spots. A guard in the polygon is represented by a point. A guard sees those points in the polygon to which it can be connected with an open straight-line segment lying in the interior of the polygon [16]; see Figure 1.4.

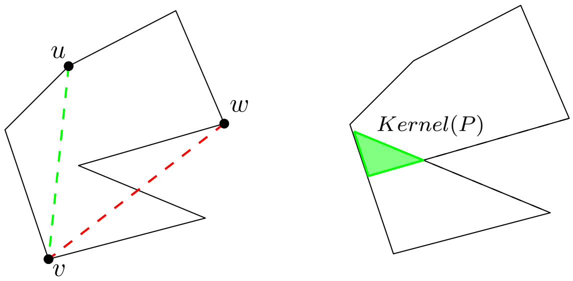


Figure 1.4: Left: Points u and v of polygon P see each other, whereas points v and w do not. Right: In green, the kernel of polygon P.

It is natural to be interested in finding out if there is any point in the polygon that can see every other point inside the polygon. If there is such a point, then it is enough to place a guard on that point to watch the whole polygon. This leads to the following definition of the kernel of a polygon; see Figure 1.4.

Definition 1.1. The kernel of a polygon P, Kernel(P), is the set of all the points in P that see every other point inside P.

Shamos and Hoey provided the first algorithm for computing the kernel of a polygon in $O(n \log n)$ time, as the intersection of all the interior half-planes determined by the edges of the polygon [120]. But the complexity of the problem was later proved to be smaller by Preparata and Lee, who obtained an optimal linear-time algorithm [105].

In Chapter 2, we study a variation of the problem of computing the kernel of a polygon in the context of $\mathcal{O}$-convexity. In particular, we reproduce an article that was published in 2021 in Journal of Global Optimization, where we compute and maintain the $\mathcal{O}$-kernel of a polygon as the set of orientations $\mathcal{O}$ rotates.

Our current line of research in the field of restricted orientation convexity focuses on the framework of orthoconvexity, in which we formulate and partially solve an Erdős-Szekeres problem. This work appears in the Book of Abstracts of the XIX Spanish Meeting on Computational Geometry [80]. We include a brief description of this work in Section A.1 of Appendix A.

1.3 Preliminaries about higher order Voronoi diagrams

Chapter 3 is dedicated to another central area in computational geometry, proximity, which involves the estimation of distances between geometric objects. Examples of proximity problems in a point set in the plane S are finding the closest pair of points in S, finding the largest empty circle inside $\mathrm{CH}(S)$, or finding the nearest neighbor in S for every point of S [106].

One of the main problems in this area is the computation and study of the Voronoi diagram of S.

Definition 1.2. The Voronoi diagram of a point set in the plane S is a subdivision of the plane into regions such that the points of the plane that lie in the same region have the same nearest point from S, also called nearest neighbor.

Note that the region associated to the neighbor $i \in S$ in the Voronoi diagram of S can be computed as the following intersection of half-planes. For $j \in S$, $j \neq i$, the perpendicular bisector of the points i and j determines two half-planes, $h(i,j)$ and $h(j,i)$, containing the points of the plane that are closer to i than to j, and vice versa. Then the region associated to the neighbor i is $f(i) = \bigcap\limits_{j \in S \setminus \{i\}} h(i,j)$; see Figure 1.5.

A Voronoi diagram can be seen as a drawing of a graph which contains unbounded edges; see Figure 1.5. In addition, faces of a Voronoi diagram are either convex polygons or unbounded convex regions. Hence, this chapter is intimately connected with the other main chapters in this thesis.

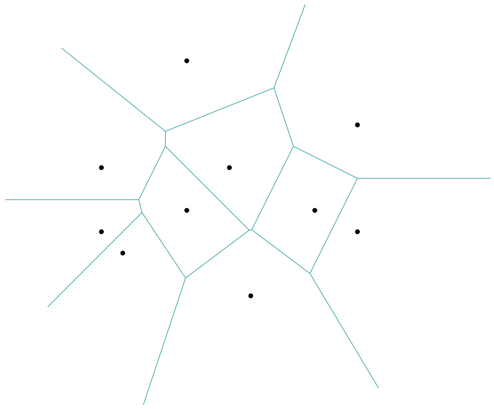


Figure 1.5: The Voronoi diagram of a point set in the plane.

Voronoi diagrams are a fundamental data structure in computational geometry, having applications in a wide range of fields inside and outside computational geometry [10, 16, 44, 94]. Perhaps their most well-known application is facility location, which aims to decide where to place a number of facilities in order to serve a number of customers distributed in a region.

The importance of Voronoi diagrams has produced several generalizations of this data structure. In some of them, the usual Euclidean distance is replaced by some other metric, such as the Manhattan distance. In some others, points are not located on the plane but on some other surface, like the sphere. And some applications study the Voronoi diagram of sets that are not formed by points, but other classes of objects, such as segments or curves. We refer to [10] for further information on these generalizations.

Chapter 3 is intimately connected to another generalization of Voronoi diagrams: higher order Voronoi diagrams or order-k Voronoi diagrams.

Definition 1.3. Let S be a set of n points in the plane. For $1 \leq k \leq n-1$, the order-k Voronoi diagram of the point set S is a subdivision of the plane into regions such that the points of the plane that lie in the same region have the same k nearest points from S, also called k nearest neighbors.

Among the applications of higher order Voronoi diagrams, we remark geometric optimization problems involving sensors or guards that need to cooperate to monitor a region. In these problems, each point of the region has to be guarded by (at least) k sensors; see, e.g., [59, 108].

In Chapter 3, order-k Voronoi diagrams are used as a tool towards a bichromatic version of the following combinatorial problem, posed by Neumann-Lara and Urrutia [92]: *Prove that every set S of n points in the plane contains two points p and q such that any circle which passes through p and q encloses "many" points of S.* Our work on this topic produced the publication of an article in the journal Discrete Mathematics in 2021. This article is reproduced in Section 3.

As for the monochromatic problem, it is believed that the best current quantification of the word "many", obtained back in 1989, still admits room for improvement [32].

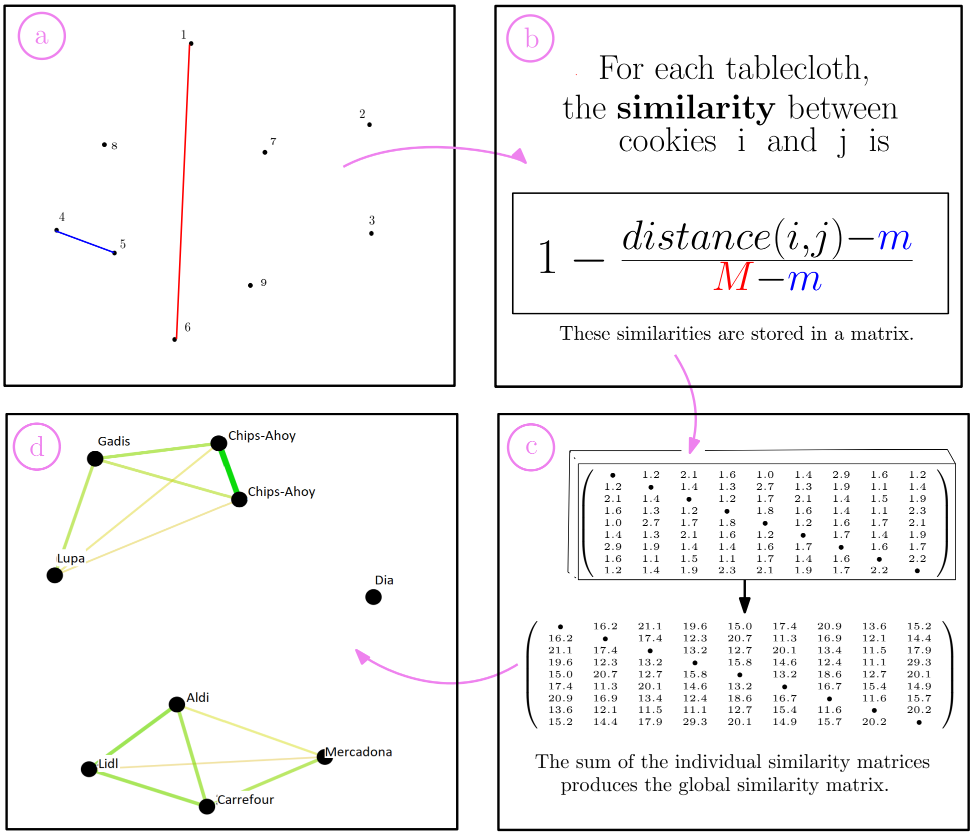
Our interest in this problem has motivated us to make an extensive study of properties of higher order Voronoi diagrams. This work is available as the preprint arXiv:2109.13002, submitted in 2021 [24], and has also been sent to a journal, where it is under review. A brief description of it is included in Section A.2 of Appendix A.

1.4 Preliminaries about sensory analysis

In Chapter 4 we explore an application of graph theory in the field of sensory analysis.

Sensory analysis is the scientific discipline that aims to analyze sensory properties of products and human response to these properties. In addition, sensory methods are widely used in industry to design or test new products before commercializing them. In this work, we evaluate consumer opinions on a number of chocolate chip cookies. These opinions were collected using the technique of projective mapping [113], which works as follows (see Figure 1.6). Each consumer is given a piece of paper or a screen, usually called tablecloth; a set of samples; and instructions to position the samples on the tablecloth. The only rule is that the more similar they consider two samples, the closer they should position them, and vice versa. The notion of similarity is not specified, but let to the consumer criteria. There are other widely used methods to collect consumers opinions, such as paired comparisons and sorting, but projective mapping has proved to be a simple and quick technique providing good results compared to the traditional methods [123].

In Chapter 4, we introduce a sensory method, that uses graph theory to analyze the data obtained from projective mapping in a testing with a large number of unexperienced consumers. Figure 1.6 pictures an outline of the main steps of this method, which works as follows.



For each tablecloth, the **similarity** between cookies i and j is

$$\boxed{1 - \frac{distance(i,j) - m}{M - m}}$$

These similarities are stored in a matrix.

$$\begin{pmatrix} \bullet & 1.2 & 2.1 & 1.6 & 1.0 & 1.4 & 2.9 & 1.6 & 1.2 \\ 1.2 & \bullet & 1.4 & 1.3 & 2.7 & 1.3 & 1.9 & 1.1 & 1.4 \\ 2.1 & 1.4 & \bullet & 1.2 & 1.7 & 2.1 & 1.4 & 1.5 & 1.9 \\ 1.6 & 1.3 & 1.2 & \bullet & 1.8 & 1.6 & 1.4 & 1.1 & 2.3 \\ 1.0 & 2.7 & 1.7 & 1.8 & \bullet & 1.2 & 1.6 & 1.7 & 2.1 \\ 1.4 & 1.3 & 2.1 & 1.6 & 1.2 & \bullet & 1.7 & 1.4 & 1.9 \\ 2.9 & 1.9 & 1.4 & 1.4 & 1.6 & 1.7 & \bullet & 1.6 & 1.7 \\ 1.6 & 1.1 & 1.5 & 1.1 & 1.7 & 1.4 & 1.6 & \bullet & 2.2 \\ 1.2 & 1.4 & 1.9 & 2.3 & 2.1 & 1.9 & 1.7 & 2.2 & \bullet \end{pmatrix}$$

$$\begin{pmatrix} \bullet & 16.2 & 21.1 & 19.6 & 15.0 & 17.4 & 20.9 & 13.6 & 15.2 \\ 16.2 & \bullet & 17.4 & 12.3 & 20.7 & 11.3 & 16.9 & 12.1 & 14.4 \\ 21.1 & 17.4 & \bullet & 13.2 & 12.7 & 20.1 & 13.4 & 11.5 & 17.9 \\ 19.6 & 12.3 & 13.2 & \bullet & 15.8 & 14.6 & 12.4 & 11.1 & 29.3 \\ 15.0 & 20.7 & 12.7 & 15.8 & \bullet & 13.2 & 18.6 & 12.7 & 20.1 \\ 17.4 & 11.3 & 20.1 & 14.6 & 13.2 & \bullet & 16.7 & 15.4 & 14.9 \\ 20.9 & 16.9 & 13.4 & 12.4 & 18.6 & 16.7 & \bullet & 11.6 & 15.7 \\ 13.6 & 12.1 & 11.5 & 11.1 & 12.7 & 15.4 & 11.6 & \bullet & 20.2 \\ 15.2 & 14.4 & 17.9 & 29.3 & 20.1 & 14.9 & 15.7 & 20.2 & \bullet \end{pmatrix}$$

The sum of the individual similarity matrices produces the global similarity matrix.

Figure 1.6: Outline of the new sensory method, applied to the evaluation of $8 + 1$ cookies (8 different cookies and a blind duplicate).

a Projective mapping: Each consumer allocates the samples on a piece of paper or a screen, creating their tablecloth. In Figure 1.6 (a), we represent one tablecloth, where samples are labeled from 1 to 9. In red, we represent the largest Euclidean distance between samples. In blue, the shortest distance between samples.

b For each consumer's tablecloth, we compute all Euclidean distances between pairs of samples. Then we define the similarity between two samples of a tablecloth by mapping their distance to the interval $[0, 1]$ as in Figure 1.6 (b).

c For each consumer, we store the similarities between samples in a matrix. The element (i, j) of this matrix is the similarity between the samples i and j, for $i \neq j$. Then, a global similarity matrix is obtained as the sum of the particular similarity matrices; see Figure 1.6 (c).

d Finally, we use the global similarity matrix to obtain a consensus plot, showing connections between the samples. This consensus plot was obtained using graph drawing techniques. The more similar two samples are according to the new method, the closer they will be positioned in the plot; see Figure 1.6 (d).

The performance of this new method is compared with the performances of a classic pure statistical method, Multiple Factor Analysis [36], and a recent geometric method, Sensograph [97].

Our work on this topic produced the publication of an article in the journal Food Quality and Preference in 2021. This article is reproduced in Chapter 4.

1.5 Preliminaries about air traffic management

In Chapter 5 we study an application of graph theory to air traffic management.

Air traffic management aims to find convenient trajectories for airplanes to travel along the airspace. There are many criteria that should be taken into account in order to model this problem. The most important restriction is that airplanes must never collide, which suggests that they should fly as far as possible from each other. But there are other considerations, such as flight duration, fuel consumption, adverse meteorology, airspace congestion, or pollution. These and other factors make the problem of air traffic management noticeably complicated [57, 73, 142].

In Chapter 5 we consider a graph model of air traffic management in which air routes are represented as the vertices of a graph, and two vertices are joined by an edge when the corresponding routes cross [14]. The goal is to allocate air routes to vertical flight levels in such a way that crossing routes are assigned to flight levels as vertically separated as possible.

In this graph model, flight levels are represented by colors and we aim to assign a color to each route (vertex) to obtain a Spectrum Graph Col-

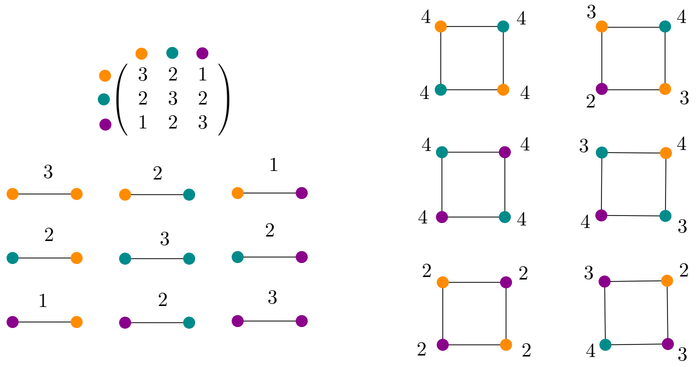


Figure 1.7: Left: Matrix of interferences between the three available colors, and the interferences induced on the possible colorings of an edge. Right: All proper 3-colorings of the cycle of length four with unlabeled vertices, and the interferences induced at each vertex. The interference at a vertex is the sum of all the interferences generated at its adjacent edges.

oring of the graph [98]. This coloring is not required to be proper, but to minimize some interferences between colors. The closer two flight levels are, the higher will be the interference between the corresponding colors. Specifically, a Spectrum Graph Coloring (SPG) of a graph requires a set of available colors and a matrix of interferences between colors. These interferences between colors induce interferences on the possible ways to color a graph; see Figure 1.7.

Our work on this topic produced the publication of an article in the journal Applied Sciences in 2020, that we reproduce in Chapter 5.

Chapter 2

Optimizing generalized kernels of polygons

The following article was originally published in 2021 in Journal of Global Optimization Volume 80, pages 887–920. It was written in collaboration with David Orden, Leonidas Palios, Carlos Seara and Paweł Żyliński. The front page of this article can be found in the Appendix B.

2.1 Introduction

The problem of computing or reaching the kernel of a polygon is a well-known visibility problem in computational geometry [53, 105, 101], closely related to the problem of guarding a polygon [100, 117, 118], and also to robot navigation inside a polygon with the restriction that the robot path must be *monotone* in some predefined set of orientations [52, 125]. The present contribution goes a step further in the latter setting, allowing the polygon or, equivalently, the set of predefined orientations to rotate. Thus, we show how to compute the orientations that maximize the region from which every point can be reached following a monotone path.

A curve $\mathcal{C}$ is $0°$-*convex* if its intersection with any line parallel to the x-axis, called $0°$-line, is connected (equivalently, if the curve $\mathcal{C}$ is y-monotone). Extending this definition, a curve $\mathcal{C}$ is α-*convex* if the intersection of $\mathcal{C}$ with any line forming a counterclockwise angle α with the positive x-axis, called α-line, is connected (equivalently, if the curve $\mathcal{C}$ is monotone with respect to the direction $\alpha^{\perp}$).

Let us now consider a set $\mathcal{O} = \{\alpha_1, \ldots, \alpha_k\}$ of k orientations in the plane, each of them given by an oriented line ℓ_i, $1 \leq i \leq k$, through the origin of the coordinate system and forming counterclockwise angle α_i with the positive x-axis. Then, a curve is *$\mathcal{O}$-convex* if it is α_i-convex for all i, $1 \leq i \leq k$, i.e., if the intersection of $\mathcal{C}$ with any line forming a counterclockwise angle α_i, $1 \leq i \leq k$, with the positive x-axis is connected (equivalently, if it is monotone with respect to all the directions $\alpha_i^{\perp}$). From now on, an $\mathcal{O}$-convex curve will be called an *$\mathcal{O}$-staircase*. See Figure 2.1 for an illustration.

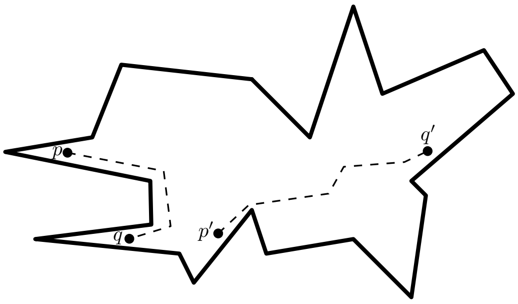


Figure 2.1: A $\{0°\}$-staircase which is not a $\{0°, 90°\}$-staircase (left) and a $\{0°, 90°\}$-staircase (right).

Observe that the orientations in $\mathcal{O}$ are between 0° and 180°. Moreover, the only $[0°, 180°)$-convex curves are lines, rays or segments. Throughout this paper, the angles of orientations in $\mathcal{O}$ will be written in degrees, while the rest of angles will be measured in radians.

Definition 2.1. Let p and q be two points inside a simple polygon P. We say that p and q *$\mathcal{O}$-see* each other or, equivalently, that they are *$\mathcal{O}$-visible* from each other, if there is an $\mathcal{O}$-staircase contained in P that connects p and q.

In the example in Figure 2.1, p and q are $\{0°\}$-visible, while p' and q' are in addition $\{0°, 90°\}$-visible. It is easy to see that p and q are not $\{90°\}$-visible.

Definition 2.2. The *$\mathcal{O}$-Kernel* of P, denoted by $\mathcal{O}$-Kernel(P), is the subset of points in P which $\mathcal{O}$-see all the other points in P. The $\mathcal{O}$-Kernel of P when the set $\mathcal{O}$ is rotated by an angle θ will be denoted by $\mathcal{O}$-Kernel$_\theta(P)$.

2.1.1 Previous related work

Schuierer, Rawlins, and Wood [117] defined the restricted-orientation visibility or $\mathcal{O}$-*visibility* in a simple polygon P with n vertices, giving an algorithm to compute the $\mathcal{O}$-Kernel(P) in time $O(k + n \log k)$, with $O(k \log k)$ preprocessing time to sort the set $\mathcal{O}$ of k orientations. In order to do so, they used the following observation.

Observation 2.3 ([117])**.** For any simple polygon P, the $\mathcal{O}$-Kernel(P) is $\mathcal{O}$-convex, connected, and

$$\mathcal{O}\text{-Kernel}(P) = \bigcap_{\alpha_i \in \mathcal{O}} \alpha_i\text{-Kernel}(P).$$

The computation of the $\mathcal{O}$-Kernel has been considered by Gewali [41] as well, who described an $O(n)$-time algorithm for orthogonal polygons without holes and an $O(n^2)$-time algorithm for orthogonal polygons with holes. The problem is a special case of the one considered by Schuierer and Wood [119] whose work implies an $O(n)$-time algorithm for orthogonal polygons without holes and an $O(n \log n + m^2)$-time algorithm for orthogonal polygons with $m \geq 1$ holes. More recently, Palios [100] gave an output-sensitive algorithm for computing the $\mathcal{O}$-Kernel of an n-vertex orthogonal polygon P with m holes, for $\mathcal{O} = \{0°, 90°\}$; his algorithm runs in $O(n + m \log m + \ell)$ time, where $\ell \in O(1 + m^2)$ is the number of connected components of $\{0°, 90°\}$-Kernel(P). Additionally, a modified version of this algorithm computes the number ℓ of connected components of the $\{0°, 90°\}$-Kernel in $O(n+m \log m)$ time [100].

2.1.2 Our contribution

We consider the problem of computing and maintaining the $\mathcal{O}$-Kernel of P while the set $\mathcal{O}$ rotates, that is, computing and maintaining $\mathcal{O}$-Kernel$_\theta(P)$ under variation of θ. For a simple polygon P and θ varying in $[-\frac{\pi}{2}, \frac{\pi}{2})$, we propose algorithms achieving the complexities in Table 2.1, where $\alpha(n)$ is the extremely-slowly-growing inverse of Ackermann's function [3]. In addition, for the case of a simple orthogonal polygon P, we propose improved algorithms to achieve the complexities in Table 2.2. Note that looking for the minimum area or perimeter only makes sense where the kernel is non-empty.

	Get the intervals of θ where the kernel is non-empty		Get a value of θ where the kernel has max/min area		Get a value of θ where the kernel has max/min perimeter	
	Time	Space	Time	Space	Time	Space
$\{0°\}$-Kernel$_\theta(P)$	$O(n \log n)$	$O(n\alpha(n))$	$O(n^2\alpha(n))$	$O(n\alpha(n))$	$O(n^2\alpha(n))$	$O(n\alpha(n))$
	(Theorem 2.11)		(Theorem 2.13)		(Theorem 2.14)	
$\{0°, 90°\}$-Kernel$_\theta(P)$	$O(n^2\alpha(n))$	$O(n^2\alpha(n))$	$O(n^2\alpha(n))$	$O(n\alpha(n))$	$O(n^2\alpha(n))$	$O(n\alpha(n))$
	(Theorem 2.18)		(Theorem 2.20)		(Theorem 2.20)	
$\mathcal{O}$-Kernel$_\theta(P)$	$O(kn^2\alpha(n))$	$O(kn^2\alpha(n))$	$O(kn^2\alpha(n))$	$O(kn\alpha(n))$	$O(kn^2\alpha(n))$	$O(kn\alpha(n))$
	(Theorem 2.19)		(Theorem 2.21)		(Theorem 2.21)	

Table 2.1: Results for P a simple polygon.

	Get the intervals of θ where the kernel is non-empty		Get a value of θ where the kernel has max/min area		Get a value of θ where the kernel has max/min perimeter	
	Time	Space	Time	Space	Time	Space
$\{0°\}$-Kernel$_\theta(P)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
	(Theorem 2.30)		(Theorem 2.32)		(Theorem 2.32)	
$\{0°, 90°\}$-Kernel$_\theta(P)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
	(Theorem 2.39)		(Theorem 2.40)		(Theorem 2.40)	
$\mathcal{O}$-Kernel$_\theta(P)$	$O(kn)$	$O(kn)$	$O(kn)$	$O(kn)$	$O(kn)$	$O(kn)$
	(Theorem 2.41)		(Theorem 2.41)		(Theorem 2.41)	

Table 2.2: Results for P a simple orthogonal polygon.

2.2 The rotated $\{0°\}$-Kernel$_\theta(P)$ in a simple polygon P

Let $(p_1, \ldots, p_n)$ be the counterclockwise sequence of vertices of a simple polygon P, which is considered to include its interior (sometimes called the *body*). In this section we deal with the rotation of the set $\mathcal{O} = \{0°\}$ by an angle $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2})$ and the computation of the corresponding $\mathcal{O}$-Kernel$_\theta(P)$, proving the results in the first row of Table 2.1.

2.2.1 The $\{0°\}$-Kernel(P), its area, and its perimeter

For the case $\mathcal{O} = \{0°\}$ and $\theta = 0$, i.e., for the $\{0°\}$-Kernel$_0(P)$ or, more simply, $\{0°\}$-Kernel(P), the kernel is composed by the points inside P which see every point in P via a y-monotone curve. Note that if P is a convex polygon, then the $\{0°\}$-Kernel(P) is the whole P. Schuierer, Rawlins, and Wood [117] presented the following definitions, observations, and results.

Definition 2.4. A reflex vertex $p_i \in P$ is a *reflex maximum* (respectively a *reflex minimum*) if p_{i-1} and p_{i+1} are both below (resp. above) p_i. Analogously, a horizontal edge with two reflex vertices is a *reflex maximum* (resp.

minimum) if its two neighbors are below (resp. above).

Note that, throughout this work, the edges are considered to be closed and, therefore, containing their endpoints. Let h_N be the horizontal line passing through a vertex p_N being a *lowest reflex minimum* of P or, if P does not have a reflex minimum, through the highest (convex) vertex of P. Let h_S be the horizontal line passing through a vertex p_S being a *highest reflex maximum* p_S of P or, if P does not have a reflex maximum, through the lowest (convex) vertex of P. Let $S(P)$ be the strip defined by the horizontal lines h_N and h_S, see Figure 2.2. Note that there are neither reflex minima nor maxima inside $S(P)$.

Lemma 2.5 ([117]). *The* $\{0°\}$-Kernel(P) *is the region defined by the intersection* $S(P) \cap P$*.*

Corollary 2.6 ([117]). *The* $\{0°\}$-Kernel(P) *can be computed in* $O(n)$ *time.*

Moreover, the horizontal lines h_N and h_S contain the segments of the *north* boundary and of the *south* boundary of the $\{0°\}$-Kernel(P); see again Figure 2.2. Lemma 2.5 is straightforward and Corollary 2.6 is trivial by computing both the lowest reflex minimum and the highest reflex maximum in linear time and then computing $S(P) \cap P$ in additional linear time.

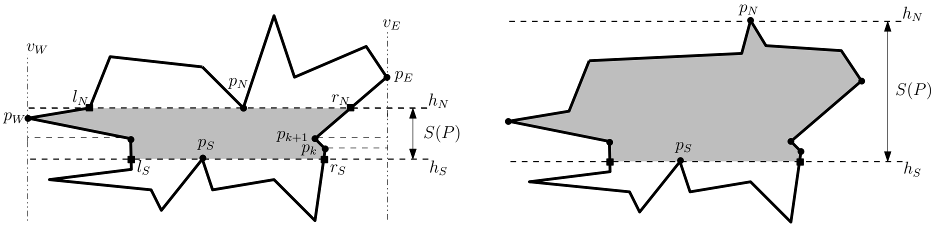


Figure 2.2: Two examples of $\{0°\}$-Kernel$_\theta(P)$ for $\theta = 0$. In the left example, the strip $S(P)$ is supported by a lowest reflex minimum p_N and a highest reflex maximum p_S. In the right example there are no reflex minima and, therefore, the strip $S(P)$ is supported by the highest (convex) vertex p_N and the highest reflex maximum p_S.

Now, let c^l and c^r denote the *left* and the *right polygonal chains* defined, respectively, by those parts of the boundary of P which are inside $S(P)$. Let $|c^l|$ and $|c^r|$ denote their number of segments. It follows from the definition of $S(P)$ and Lemma 2.5 that both chains are 0°-convex curves, i.e., y-monotone chains; see Figure 2.2 once more.

Corollary 2.7. *The area and the perimeter of the* $\{0°\}$-Kernel(P) *can be computed in* $O(n)$ *time.*

Proof. To compute the area of the $\{0°\}$-Kernel$(P) = S(P) \cap P$, we proceed as follows. The area can be decomposed into (a finite number of) horizontal trapezoids defined by pairs of vertices in $c^l \cup c^r$ with consecutive y-coordinate. The area of these trapezoids can be computed in constant time, so the area of $\{0°\}$-Kernel$(P) = S(P) \cap P$ can be computed in $O(|c^l| + |c^r|)$ time.

Computing the perimeter is even simpler, because we only need the addition of the lengths of c^l and c^r plus the lengths of the north and south boundaries of the $\{0°\}$-Kernel(P), which can also be done in $O(|c^l| + |c^r|)$ time. □

2.2.2 The existence of the $\{0°\}$-Kernel$_\theta(P)$

In this subsection, we show how to compute the intervals for θ such that the $\{0°\}$-Kernel$_\theta(P)$ is non-empty. First, we observe that we do not need a complete rotation, since $\{0°\}$-Kernel$_{-\frac{\pi}{2}}(P) = \{0°\}$-Kernel$_{\frac{\pi}{2}}(P)$. Also, notice that Definition 2.4, for reflex maxima/minima with respect to the horizontal orientation, can be easily extended to any orientation $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2})$ as follows.

Definition 2.8. A reflex vertex p_i in a simple polygon P where p_{i-1} and p_{i+1} are both below (respectively, above) p_i with respect to a given orientation θ is a *reflex maximum* (resp. a *reflex minimum*) with respect to θ. Analogously, an edge of angle θ with two reflex vertices is a *reflex maximum* (resp. *minimum*) when its two neighbors are below (resp. above) with respect to the orientation θ.

In order to know the intervals for θ such that the $\{0°\}$-Kernel$_\theta(P)$ is not empty, we need to maintain the boundary of the rotation by angle θ of the strip $S(P)$ previously defined, which will be denoted by $S_\theta(P)$; see Figure 2.3. We need to extend Lemma 2.5 to any orientation θ:

Lemma 2.9. *The* $\{0°\}$-Kernel$_\theta(P)$ *is the region defined by the intersection* $S_\theta(P) \cap P$.

Proof. The claim follows from Lemma 2.5 and the fact that $\{0°\}$-Kernel$_\theta(P) = \{0°\}$-Kernel(P_θ) and $S_\theta(P) = S(P_\theta)$, where P_θ denotes the polygon P rotated by the angle θ. See Figure 2.3. □

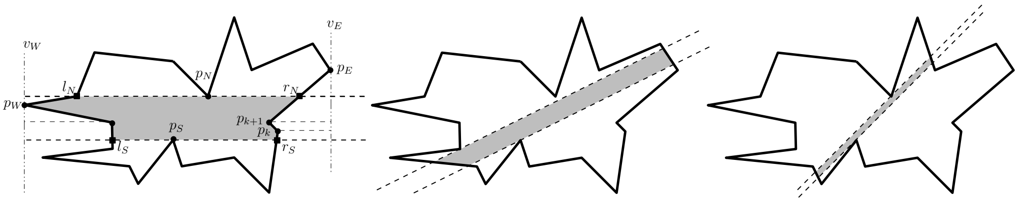


Figure 2.3: A rotating $\{0°\}$-Kernel$_\theta(P)$ for $\theta = 0$ (left), $\theta = \frac{\pi}{8}$ (middle), and $\theta = \frac{\pi}{4}$ (right).

Now, we describe the main steps of our algorithm to compute the intervals of those values of θ within $[-\frac{\pi}{2}, \frac{\pi}{2})$ such that $S_\theta(P) \neq \emptyset$ and, therefore, such that $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$.

Step 1: Angular intervals. For each vertex $p_i \in P$, if p_i is reflex, we compute the angular intervals $[\theta_1^i, \theta_2^i)$ and $[\theta_1^i + \pi, \theta_2^i + \pi)$ of orientations θ for which p_i is a reflex maximum/minimum, defined when rotating the line containing the edge $p_{i-1}p_i$ up to the line containing the edge p_ip_{i+1}. Otherwise, if p_i is convex, we compute the angular intervals $[\theta_1^i, \theta_2^i)$ and $[\theta_1^i + \pi, \theta_2^i + \pi)$ of orientations θ for which p_i is the lowest/highest vertex of the rotated polygon P_θ. Thus, in case that for some orientation θ there is no reflex maximum/minimum, the lowest/highest convex vertex for that orientation will play the role of reflex maximum/minimum. Note that an angular interval may be split into two, in case it contains the orientation $\pi/2$.

Step 2: Dualization. For the sake of efficiently handling the next step, we do the dualization of the set of vertices together with their relevant nonempty angular intervals from Step 1. The dualization function ℓ we use is as follows: If $p = (a, b)$ is a point in the primal, its dual $\ell(p)$ is the line $\ell(p) :\equiv y = ax - b$; if r is the line given by $y = ax - b$ in the primal, its dual $\ell(r)$ is the point $\ell(r) := (a, b)$. Moreover, the point $p = (a, b)$ lies below/on/above a line $l \equiv y = mx+c$ if and only if the line $\ell(p) \equiv y = ax-b$ passes above/through/below the point $\ell(l) = (m, -c)$, see [16].

In this way, for a vertex $p_i \in P$ we translate the two lines which contain the incident edges p_ip_{i-1} and p_ip_{i+1} of the polygon P into the corresponding dual points located on the dual line $\ell(p_i)$. In addition, we translate the set of lines through p_i in the angular interval of p_i into the corresponding set of dual points, which define a segment on the line $\ell(p_i)$. For an illustration, see the objects in red part in Figure 2.4. Thus, the angular interval of a point p_i is translated into the straight line segment on the line $\ell(p_i)$. Again, note that a vertex p_i may contribute two segments in the dual plane, if the corresponding angular interval contains the orientation $\pi/2$. The dualization process for all the other cases is done in an analogous way.

The dualization is performed as follows. On one hand, we dualize the reflex minima with their intervals which, in addition to the dual of intervals of the upper chain of the convex hull of P, $\mathrm{CH}(P)$, (in blue in Figure 2.4) results in an arrangement $\mathcal{D}_{\min}$ of line segments. On the other hand, we dualize the reflex maxima with their intervals (an example in red in Figure 2.4) which, together with the dual of the intervals of the lower chain of $\mathrm{CH}(P)$, gives an arrangement $\mathcal{D}_{\max}$ of line segments. Both arrangements have a linear number of line segments in the dual plane.

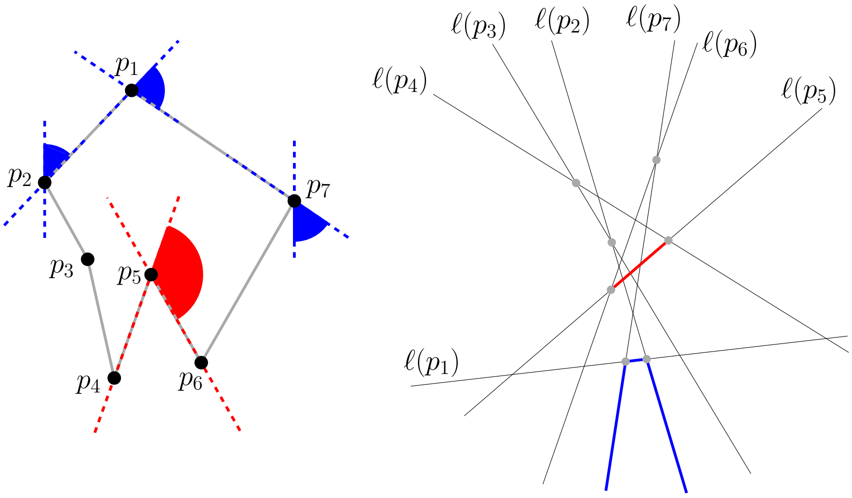


Figure 2.4: In red, dualization of the angular interval corresponding to the vertex p_5 (left) in the primal, which in the dual translates into a segment on the line $\ell(p_5)$ (right). In blue, the angular intervals of the vertices p_7, p_1, p_2 in the upper chain of the convex hull in the primal (left), translate into the lower envelope of the arrangement in the dual (right).

Step 3: Event intervals. We compute the sequence of *event intervals*, each of which is defined by a pair of orientation values $[\theta_1, \theta_2) \subset [-\frac{\pi}{2}, \frac{\pi}{2})$ such that for any value $\theta \in [\theta_1, \theta_2)$, the strip $S_\theta(P)$ is supported by the same pair of vertices of P, in other words, such that the pair of vertices of P defining the lowest reflex minimum and the highest reflex maximum does not change for $\theta \in [\theta_1, \theta_2)$, recall Figure 2.3. In order to determine the sequence of event intervals, we exploit the following observation.

Observation 2.10. The highest (resp. lowest) segment in $\mathcal{D}_{\min}$ (resp. $\mathcal{D}_{\max}$) intersected by the vertical line $x = \theta$ corresponds in the primal to the lowest reflex minimum (resp. the highest reflex maximum) with respect to the orientation θ.

Proof. It directly follows from the already mentioned fact that the dualization reverses the above-below relations between lines and/or points. □

Taking into account the above observation, we compute the upper envelope of $\mathcal{D}_{\min}$, denoted by $\mathcal{U}_{\mathcal{D}_{\min}}$, and the lower envelope of $\mathcal{D}_{\max}$, denoted by $\mathcal{L}_{\mathcal{D}_{\max}}$ [51]. Next, by sweeping the arrangement $\mathcal{U}_{\mathcal{D}_{\min}} \cup \mathcal{L}_{\mathcal{D}_{\max}}$, we obtain the sequence of pairs "lowest reflex minimum and highest reflex maximum" for all the event intervals $[\theta_1, \theta_2)$, as θ varies in $[-\frac{\pi}{2}, \frac{\pi}{2})$.

Step 4: Non-empty $\{0°\}$-Kernel$_\theta(P)$. Recall that, by Lemma 2.9, the strip $S_\theta(P)$ is empty if, with respect to θ, the lowest reflex minimum is below the highest reflex maximum. Therefore, this step relies only on scanning the relevant pairs from Step 3 and checking whether the lowest reflex minimum is above the highest reflex maximum, which results in the angular intervals $[\theta_1, \theta_2) \subset [-\frac{\pi}{2}, \frac{\pi}{2})$ such that $\{0°\}\text{-Kernel}_\theta(P) \neq \emptyset$ for all the values of $\theta \in [\theta_1, \theta_2)$.

Analysis of Algorithm 1. The correctness of Algorithm 1 follows from the discussion above, in particular from the concept of dualization together with Observation 2.10. About the complexity, STEPS 1 and 2 can be done in linear time and space, in particular, by computing the convex hull of the simple polygon P [82]. STEP 3 can be done in $O(n \log n)$ time, since the computation of the upper (and the lower) envelope of a set of n possibly-intersecting straight-line segments can be done in $O(n \log n)$ time [51]. Finally, STEP 4 can be accomplished in $O(n\alpha(n))$, since the upper envelope and the lower envelope of a set of n possibly-intersecting straight-line segments in the plane have worst-case size $O(n\alpha(n))$, where $\alpha(n)$ is the extremely-slowly-growing inverse of Ackermann's function [3].

Theorem 2.11. *For a simple polygon P with n vertices, the set of event intervals $[\theta_1, \theta_2) \subset [-\frac{\pi}{2}, \frac{\pi}{2})$ such that $\{0°\}\text{-Kernel}_\theta(P) \neq \emptyset$ for $\theta \in [\theta_1, \theta_2)$ can be computed in $O(n \log n)$ time and $O(n\alpha(n))$ space.*

Proof. The result is a direct consequence of applying Algorithm 1, whose correctness as well as time and space complexities follow from the analysis above. □

Algorithm 1 Computing the intervals of θ such that $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$

```
Input: A simple polygon $P$ with $n$ vertices
Output: Set $\mathcal{I}$ of event intervals for angles $\theta$ such that $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$

STEP 1: Angular intervals
for $i = 1$ to $n$ do
    if $p_i \in P$ is reflex then
        compute $[\theta_1^i, \theta_2^i)$ and $[\theta_1^i + \pi, \theta_2^i + \pi)$ such that $p_i$ is reflex maximum/minimum
    if $p_i \in P$ is convex then
        compute $[\theta_1^i, \theta_2^i)$ and $[\theta_1^i + \pi, \theta_2^i + \pi)$ such that $p_i$ is the lowest/highest vertex of $P_\theta$,
        proceed like $p_i$ being a vertex reflex minimum/maximum

STEP 2: Dualization of vertices with their angular events from Step 1
for $i = 1$ to $n$ do
    if $p_i$ is a reflex maximum then
        translate the angular interval of $p_i$ into the line segment on $\ell(p_i)$ and include this in an arrangement $\mathcal{D}_{\max}$
    if $p_i$ is a reflex minimum then
        translate the angular interval of $p_i$ into the line segment on $\ell(p_i)$ and include this in an arrangement $\mathcal{D}_{\min}$
    (Note that a reflex vertex may contribute two segments in the dual.)
Include in $\mathcal{D}_{\max}$ the dual of the lower chain of CH($P$) and include in $\mathcal{D}_{\min}$ the dual of the upper chain of CH($P$)

STEP 3: Event intervals
Compute the event intervals such that $S_\theta(P)$ is supported by the same pair of vertices
Compute the upper envelope $\mathcal{U}_{\mathcal{D}_{\min}}$ of $\mathcal{D}_{\min}$
Compute the lower envelope $\mathcal{L}_{\mathcal{D}_{\max}}$ of $\mathcal{D}_{\max}$
Sweep $\mathcal{U}_{\mathcal{D}_{\min}} \cup \mathcal{L}_{\mathcal{D}_{\max}}$ and compute the "lowest reflex minimum and highest reflex maximum" for the event intervals

STEP 4: Non-empty $\{0°\}$-Kernel$_\theta(P)$
Scan the vertex pairs from STEP 3, checking whether the lowest reflex minimum is above the highest reflex maximum and, if so, add the corresponding interval to an initially empty set $\mathcal{I}$
output $\mathcal{I}$
```

2.2.3 Optimizing the area of the $\{0°\}$-Kernel$_\theta(P)$

Let us consider the problem of optimizing the area of the $\{0°\}$-Kernel$_\theta(P)$, i.e., computing the value(s) of θ such that the area of $\{0°\}$-Kernel$_\theta(P)$ is maximum or minimum (note that the latter only makes sense where the kernel is non-empty). The idea of our approach is based upon Algorithm 1 for computing the set of event intervals $[\theta_1, \theta_2) \subset [-\frac{\pi}{2}, \frac{\pi}{2})$ such that $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$ for all the values of $\theta \in [\theta_1, \theta_2)$ (Theorem 2.11). Namely, we do the following:

Step A: $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$. Run STEPS 1-4 of Algorithm 1.

Step B: Vertex events. For each event interval $[\theta_1, \theta_2)$ from Step 4 (within which the highest reflex maximum and the lowest reflex minimum do not change), we subdivide $[\theta_1, \theta_2)$ every time that, as θ varies, a vertex of the simple polygon P either stops or starts contributing to the current boundary of the $\{0°\}$-Kernel$_\theta(P)$. Observe that at, every such subdivision step, the differential in the area can be decomposed into triangles, as illustrated in Figure 2.5. In particular, for each of these consecutive subintervals $[\beta_j, \beta_{j+1})$ of $[\theta_1, \theta_2)$, we have:

$$\begin{aligned} Area(\{0°\}\text{-Kernel}_\beta(P)) &= Area(\{0°\}\text{-Kernel}_{\beta_j}(P)) + A_1(\beta) \\ &\quad + A_2(\beta) - B_1(\beta) - B_2(\beta). \end{aligned} \tag{2.1}$$

Thus, for such $\beta \in [\beta_j, \beta_{j+1})$, the area of the $\{0°\}$-Kernel$_\beta(P)$ can be expressed, using simple trigonometric relations, as a function $A(\beta)$ of the angle of rotation $\beta \in [\beta_j, \beta_{j+1})$, as detailed in Section 2.5.1. Thus, it only remains to obtain the maximum value of that function in the subinterval. In the mentioned Section 2.5.1 we show how this calculation is reduced to find the real solutions of a polynomial equation in t of degree 6. The final solution to the problem is then the best one over all those computed for these consecutive subintervals $[\beta_j, \beta_{j+1})$.

Clearly, Step B requires computing and maintaining the boundary of $\{0°\}$-Kernel$_\theta(P)$, in particular, maintaining the set of vertices of the current left and right boundary chains, respectively denoted by c^l_θ and c^r_θ, of $\{0°\}$-Kernel$_\theta(P)$ as $\theta \in [\theta_1, \theta_2)$ varies (also for all the possible consecutive subintervals $[\beta_j, \beta_{j+1})$ of $[\theta_1, \theta_2)$); see again Figure 2.5. For this purpose, we compute the intersections of the lines $h_N(\theta)$ and $h_S(\theta)$ with the boundary of P, maintaining the information of the first and the last vertices of c^l_θ and c^r_θ in the current interval $[\theta_1, \theta_2)$. Now, as θ varies, the next vertex event

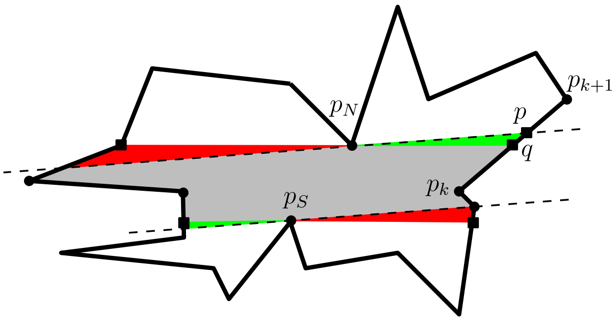


Figure 2.5: The four triangles $A_1(\beta)$, $A_2(\beta)$ (in green), and $B_1(\beta)$, $B_2(\beta)$ (in red).

can be computed in constant time by sweeping (and so modifying ad-hoc) chains c^l_θ and c^r_θ, in particular, using the circular order of the vertices of the polygon P and taking the smallest among the relevant angles defined by the current line $h_N(\theta)$ (resp. $h_S(\theta)$), the point $p_N(\theta)$ (resp. $p_S(\theta)$), and the relevant first polygon vertex on c^l_θ and the first polygon vertex after the last polygon vertex on c^r_θ (resp. the first polygon vertex on c^r_θ and the first polygon vertex after the last polygon vertex on c^l_θ).

One can wonder whether the same vertex of a simple polygon P may contribute to a vertex event for several event intervals. Surprisingly enough, there can be $\Theta(n)$ distinct vertices, each of them contributing $\Theta(n)$ vertex events, as illustrated in Figure 2.6. By Theorem 2.11 we know that the number of event intervals is at most $O(n\alpha(n))$ thus, there may be as many as $O(n^2\alpha(n))$ vertex events (consecutive subintervals) involving in total $O(n^2\alpha(n))$ non-empty kernels $\{0°\}$-Kernel$_\theta(P)$ having combinatorially different boundaries, implying the time complexity for computing the angle θ that maximizes (or minimizes) the area of $\{0°\}$-Kernel$_\theta(P)$. To see this, it is enough to construct a simple polygon P' by replicating the set of four points $\{p_1, p_2, p_3, p_4\}$ in Figure 2.6 a linear number of times, and keeping the $\Theta(n)$ vertices in the corner. As we will see later, this bound also works for the computation of the maximum (or minimum) value of the perimeter of $\{0°\}$-Kernel$_\theta(P)$. From this discussion we get the following result.

Proposition 2.12. *For a simple polygon P with n vertices, the number of vertex events or consecutive subintervals $[\beta_j, \beta_{j+1})$ where Algorithm 2 has to optimize the area of $\{0°\}$-Kernel$_\theta(P)$ is $O(n^2\alpha(n))$.*

Proof. The $O(n^2\alpha(n))$ bound comes from the simple polygon P' constructed

Algorithm 2 Computing the maximum area of $\{0°\}$-Kernel$_\theta(P)$

```
Input: A simple polygon $P$ with $n$ vertices
Output: An angle $\theta$ such that $Area(\{0°\}\text{-Kernel}_\theta(P))$ is maximum and the maximum value of the area

STEP A: $\{0°\}\text{-Kernel}_\theta(P) \neq \emptyset$
Run STEPS 1-4 from Algorithm 1

STEP B: VERTEX EVENTS.
for each $[\theta_1, \theta_2)$ from STEP 4 of Algorithm 1 do
    if a vertex of $P$ stops/starts appearing on the current boundary of the $\{0°\}\text{-Kernel}_\theta(P)$ then
        subdivide $[\theta_1, \theta_2)$ into consecutive subintervals $[\beta_j, \beta_{j+1})$ and decompose the differential of the area into triangles
for each subinterval $[\beta_j, \beta_{j+1})$ and $\beta \in [\beta_j, \beta_{j+1})$ do
    $A(\beta) = Area(\{0°\}\text{-Kernel}_\beta(P))$
    $\quad\;\;\, = Area(\{0°\}\text{-Kernel}_{\beta_j}(P)) + A_1(\beta) + A_2(\beta) - B_1(\beta) - B_2(\beta)$
    Find the real solutions of a polynomial equation, and maintain the maximum value of $A(\beta)$ and the corresponding angle
output the maximum value of the area and the corresponding angle
```

above based on Figure 2.6, taking into account the computation of the envelopes for obtaining the event intervals in Theorem 2.11. □

Analysis of Algorithm 2. The correctness of Algorithm 2 follows from the discussion above. Namely, STEP A consists on running Algorithm 1, so it takes $O(n \log n)$ time and $O(n\alpha(n))$ space, obtaining $O(n\alpha(n))$ event intervals. By Proposition 2.12, the number of vertex events or consecutive subintervals can be $O(n^2\alpha(n))$, and STEP B spends constant time for the optimization in each of them, see Section 2.5.1. Thus, this implies $O(n^2\alpha(n))$ time and $O(n\alpha(n))$ space in total. Notice that when we change from an event interval to the next event interval, we might have to manage a situation like the one illustrated in Figure 2.6, but this can be done in linear time and space since we translate one side of the kernel in parallel with the endpoints going through vertices on the boundary of P (vertices in the

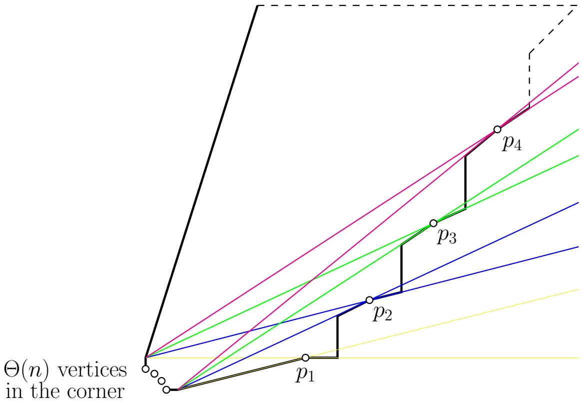


Figure 2.6: For each vertex p_i, $1 \leq i \leq 4$, all the $\Theta(n)$ vertices in the corner will be scanned again.

corner in Figure 2.6). Thus, it does not change the total time complexity because it implies an additional $O(n^2\alpha(n))$ time; also the space complexity does not change because the algorithm always reuses the linear space.

Theorem 2.13. *For a simple polygon P with n vertices, an angle θ that maximizes/minimizes the value of the area of* $\{0°\}$-Kernel$_\theta(P)$ *can be computed in $O(n^2\alpha(n))$ time and $O(n\alpha(n))$ space.*

Proof. Again, the correctness of our approach and the time and space complexities follow from the discussion above on the analysis of Algorithm 2 and Proposition 2.12. The problem of minimizing the area, where meaningful, is handled in the same way. □

2.2.4 Optimizing the perimeter of the $\{0°\}$-Kernel$_\theta(P)$

Consider now the problem of optimizing the perimeter of $\{0°\}$-Kernel$_\theta(P)$, denoted by $\Pi(\theta)$, where the goal is to compute the value(s) of θ such that $\Pi(\theta)$ is maximum or minimum (note that the latter only makes sense where the kernel is non-empty). Observe that we can apply the same approach as the one proposed for optimizing the area of $\{0°\}$-Kernel$_\theta(P)$ in Algorithm 2, with the only difference that now, when handling the vertex events (defined and computed exactly in the same way as in the case of optimizing the area in Step B), we need to handle the expression for the polygon perimeter.

Clearly, the differential in the perimeter can be decomposed as adding two segments and subtracting two other segments, see again Figure 2.5, and thus the perimeter can then be expressed, using simple trigonometric relations, as a function $\Pi(\beta)$ of the angle of rotation $\beta \in [\beta_j, \beta_{j+1})$, see Section 2.5.2. Then, it only remains to obtain the maximum value of that function in the interval $[\beta_j, \beta_{j+1})$. As detailed in Section 2.5.2, this amounts to finding the real solutions of a polynomial equation in t of constant degree. Consequently, we may conclude with the following result, where the minimization of the perimeter, if meaningful, is handled analogously.

Theorem 2.14. *For a simple polygon P with n vertices, an angle θ such that the value of the perimeter of $\{0°\}$-$\mathrm{Kernel}_\theta(P)$ is maximum/minimum can be computed in $O(n^2\alpha(n))$ time and $O(n\alpha(n))$ space.*

2.3 The rotated $\{0°, 90°\}$-Kernel$_\theta(P)$ of a simple polygon P

We continue our study on the problem of computing the $\mathcal{O}$-Kernel of a simple polygon P considering the case when $\mathcal{O}$ is given by two perpendicular orientations which rotate simultaneously, for which we prove the results in the second row of Table 2.1. Notice that the two orientations do not need to be perpendicular for the proofs nor the algorithm in this section, because we are using Observation 2.3. Moreover, since the problem for a set $\mathcal{O}$ with k orientations reduces to computing and maintaining the intersection of k different kernels, the results in the third row of Table 2.1 will follow as well.

2.3.1 The existence of the $\{0°, 90°\}$-Kernel$_\theta(P)$

Taking into account Observation 2.3, one can determine $\{0°, 90°\}$-$\mathrm{Kernel}_\theta(P)$ by computing the intersection of the two kernels $\{0°\}$-$\mathrm{Kernel}_\theta(P)$ and $\{90°\}$-$\mathrm{Kernel}_\theta(P)$, respectively. Note that, in fact, the latter equals the $\{0°\}$-$\mathrm{Kernel}_{\theta+90°}(P)$. In the following, the points $p_W(\theta)$ and $p_E(\theta)$ for the $\{90°\}$-$\mathrm{Kernel}_\theta(P)$ are analogous to the points $p_N(\theta)$ and $p_S(\theta)$ previously defined for the $\{0°\}$-$\mathrm{Kernel}_\theta(P)$. Notice that $p_N(\theta+90°) = p_W(\theta)$ and $p_S(\theta+90°) = p_E(\theta)$, recall Figure 2.2, and see Figure 2.7.

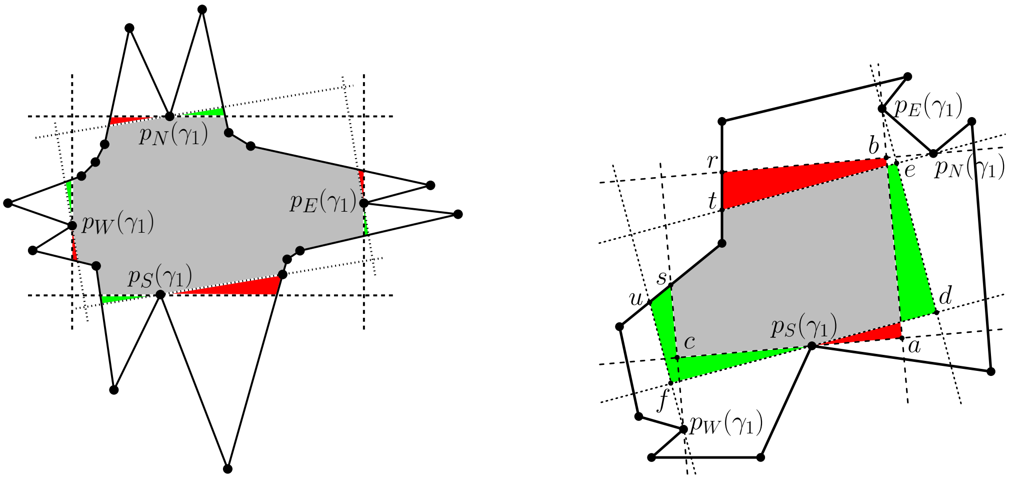


Figure 2.7: Left: A $\{0°, 90°\}$-Kernel$_\theta(P)$ and the rotated kernel in the next event, the area leaving (resp. entering) the kernel being depicted in red (resp. green). Right: A more general $\{0°, 90°\}$-Kernel$_\theta(P)$ and the rotated kernel in a slightly larger angle β, depicting the entering and leaving areas as before. Note that, in both cases, $p_i(\gamma_1) = p_i(\theta)$ for any $\theta \in [\gamma_1, \gamma_2)$, $i \in \{N, W, S, E\}$.

2.3.1.1 Floating rectangle

Let $\theta \in [0, \pi/2)$ be an angle such that both $\{0°\}$-Kernel$_\theta(P)$ and $\{90°\}$-Kernel$_\theta(P)$ are non-empty. In what follows, we refer to the intersection $S_\theta(P) \cap S_{\theta+90°}(P)$ as a *floating rectangle*, and denote it by R_θ (recall that $S_\alpha(P)$ denotes the strip defined by the lines $h_N(\alpha)$ and $h_S(\alpha)$ being, respectively, the line with slope $\tan(\alpha)$ passing through $p_N(\alpha)$ and the line with slope $\tan(\alpha)$ passing through $p_S(\alpha)$). Clearly, by combining Lemma 2.5 with Observation 2.3, we observe that

$$\{0°, 90°\}\text{-Kernel}_\theta(P) = R_\theta \cap P, \tag{2.2}$$

which immediately results in the following observation.

Observation 2.15. The $\{0°, 90°\}$-Kernel$_\theta(P)$ of a simple polygon P is empty if, and only if:

(A) either one of the two kernels $\{0°\}$-Kernel$_\theta(P)$ or $\{90°\}$-Kernel$_\theta(P)$ is empty, or

(B) the floating rectangle R_θ is lying outside P (as in Figure 2.8, left).

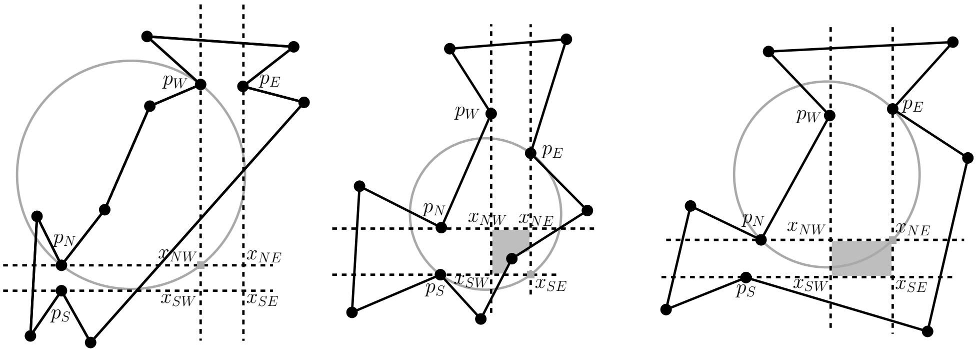


Figure 2.8: Three types of kernel with the arcs for the vertices of the floating rectangle. Note that the references to the angle θ in $p_i(\theta)$ and $x_{ij}(\theta)$ have been removed for the sake of an easier visualization.

Assume now that, following our approach proposed for the proof of Theorem 2.11, we have already computed the sequence $\mathcal{I}_{0^\circ}$ of event intervals where $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$, and in an analogous way the sequence $\mathcal{I}_{90^\circ}$ of event intervals where $\{90°\}$-Kernel$_\theta(P) \neq \emptyset$. Now, in $O(n\alpha(n))$ time and space, we obtain from these two event sequences the sequence $\mathcal{I}$ (with complexity $O(n\alpha(n))$ of the event intervals corresponding to the simultaneous rotation of both kernels, saving only those non-empty intersections $I' \cap I''$ of event intervals $I' \in \mathcal{I}_{0^\circ}$ and $I'' \in \mathcal{I}_{90^\circ}$, where both kernels are non-empty. Once we have stored this data, as a matter of fact, we have handled Case (A) in Observation 2.15.

Next, as regards Case (B) in Observation 2.15, the following lemma allows us to check whether the intersection $\{0°\}$-Kernel$_\theta(P) \cap \{90°\}$-Kernel$_\theta(P) = \{0°, 90°\}$-Kernel$_\theta(P)$ is non-empty.

Lemma 2.16. *Consider an event interval $[\gamma_1, \gamma_2) \in \mathcal{I}$ and an angle $\theta \in [\gamma_1, \gamma_2)$. Then the $\{0°, 90°\}$-*Kernel$_\theta(P)$ *is non-empty in the following cases:*

(B.1) *At least one point among the current $p_N(\gamma_1)$, $p_S(\gamma_1)$, $p_E(\gamma_1)$, $p_W(\gamma_1)$ belongs to the floating rectangle R_{γ_1} (see Figure 2.7).*

(B.2) *The polygon P contains at least one of the corners of the floating rectangle R_θ (see Figure 2.8).*

Proof. First, if at least one of the cases (B.1), (B.2) holds then the $\{0°, 90°\}$-Kernel$_{\gamma_1}(P)$ is non-empty. Assume now that the kernel is non-empty and suppose, for contradiction, that neither (B.1) nor (B.2) holds. Then, the fact that (B.2) does not hold implies that all 4 corners of the rectangle R_{γ_1} lie outside P. Consider two adjacent corners r, r' of R_{γ_1} lying on the line $h_N(\gamma_1)$ that goes through $p_N(\gamma_1)$. The fact that (B.1) does not hold implies that $p_N(\gamma_1)$ does not belong to the line segment connecting r, r', But then, if there were a point $q \in P$ on the segment rr', then the definition of the $\{0°\}$-Kernel$_\theta(P)$ (see Definition 2.2) implies that q should be γ_1-visible from $p_N(\gamma_1)$. Then, Definition 2.1 implies that there is a γ_1-staircase C in P connecting $p_N(\gamma_1)$ and q; this is a contradiction because the intersection of C with the line $h_N(\gamma_1)$ which has slope $\tan(\gamma_1)$ is not connected. Thus, the entire edge rr' of R_{γ_1} lies outside P.

Similarly, the other edges of R_{γ_1} lie outside P as well. Then, for any point q' inside R_{γ_1}, we can apply the same argument by using a line parallel to $h_N(\gamma_1)$ that goes through q' (note that such a line intersects the strip S_{γ_1}), proving that $q' \notin P$. Therefore, the entire R_{γ_1} lies outside P, in contradiction to the fact that the $\{0°, 90°\}$-Kernel$_{\gamma_1}(P)$ is non-empty. □

Clearly, Case (B.1) can be checked in constant time, by the orientation test with the point considered and the two lines forming the relevant strip. Notice that the situation of these four points cannot change during the event interval $[\gamma_1, \gamma_2)$, since $p_i(\gamma_1) = p_i(\theta)$ for any $\theta \in [\gamma_1, \gamma_2)$, $i \in \{N, W, S, E\}$.

2.3.1.2 Arc events

For $i \in \{N, S\}$, let $x_{iW}(\theta)$ (resp. $x_{iE}(\theta)$) denote the intersection point of the line $h_i(\theta)$ with the line $h_N(\theta + 90°)$ (resp. $h_S(\theta + 90°)$), see Figure 2.8. In other words, the points $x_{ij}(\theta)$, $i \in \{N, S\}$ and $j \in \{W, E\}$, are the relevant four corners of the floating rectangle R_θ. Next, for $i \in \{N, S\}$ and $j \in \{W, E\}$, let $C_{ij}(\theta)$ denote the circle passing through the points $p_i(\theta), p_j(\theta)$ and $x_{ij}(\theta)$, again see Figure 2.8. Finally, let $\breve{a}_{ij}(\theta)$ denote the arc of $C_{ij}(\theta)$ between $p_i(\theta)$ and $p_j(\theta)$ such that $x_{ij}(\theta)$ belongs to $\breve{a}_{ij}(\theta)$. Notice that the angle between points $p_i(\theta), x_{ij}(\theta)$ and $p_j(\theta)$ is the right angle, and so the point $x_{ij}(\theta)$ describes the semicircle having as diameter the segment $\overline{p_i(\theta)p_j(\theta)}$ (see again see Figure 2.8), thus impyling $C_{ij}(\theta) = C_{ij}(\gamma_1)$ and $\breve{a}_{ij}(\theta) = \breve{a}_{ij}(\gamma_1)$ for any $\theta \in [\gamma_1, \gamma_2)$. Consequently, as θ varies in $[\gamma_1, \gamma_2)$, the point $x_{ij}(\theta)$ continuously moves along the arc $\breve{a}_{ij}(\gamma_1)$. Moreover, we have the following observation.

Observation 2.17. As θ varies in $[\gamma_1, \gamma_2)$, the point $x_{ij}(\theta)$ can change several times from the exterior to the interior of the polygon P or vice versa.

The claim follows from the interval $[\gamma_1, \gamma_2)$ being the intersection of event intervals and the fact that the boundary of the simple polygon P can be a polyline of size $\Theta(n)$, as the one in Figure 2.6. Taking into account Observation 2.17, for an event interval $[\gamma_1, \gamma_2)$, we can handle the case (B.2) in linear time. Because there are at most $O(n\alpha(n))$ event intervals, the total complexity for this step will be $O(n^2\alpha(n))$. Therefore, we can outline Algorithm 3.

Algorithm 3 Computing the intervals of θ such that $\{0°, 90°\}$-Kernel$_\theta(P) \neq \emptyset$

```
Input: A simple polygon $P$ with $n$ vertices
Output: Sequence $\mathcal{E}$ of intervals for angles $\theta$ such that $\{0°, 90°\}$-Kernel$_\theta(P) \neq \emptyset$

STEP I: EVENT INTERVALS: CHECKING CASE (A) IN OBSERVATION 2.15
Apply Algorithm 1 to compute the sequence $\mathcal{I}_{0°}$ of event intervals where $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$
Apply Algorithm 1 to compute the sequence $\mathcal{I}_{90°}$ of event intervals where $\{90°\}$-Kernel$_\theta(P) \neq \emptyset$
Combine $\mathcal{I}_{0°}$ and $\mathcal{I}_{90°}$ into the sequence $\mathcal{I} = \{I' \cap I'' = [\gamma_j, \gamma_{j+1}) \mid I' \in \mathcal{I}_{0°}, I'' \in \mathcal{I}_{90°}\}$

STEP II: FLOATING RECTANGLE: CHECKING CASE (B)
for each event interval $[\gamma_1, \gamma_2) \in \mathcal{I}$ do
    if $p_N(\gamma_1)$ or $p_S(\gamma_1)$ or $p_E(\gamma_1)$ or $p_W(\gamma_1)$ belongs to $R_{\gamma_1}$ then
        Case (B.1) holds and $\{0°, 90°\}$-Kernel$_\theta(P) \neq \emptyset$ for $\theta \in [\gamma_1, \gamma_2)$. Insert $[\gamma_1, \gamma_2)$ in an initially empty sequence $\mathcal{E}$
    else
        for each vertex event $[\beta_1, \beta_2) \subseteq [\gamma_1, \gamma_2)$ do
            if $x_{ij}(\theta)$, $i \in \{N, S\}$, $j \in \{W, E\}$, on $\breve{a}_{ij}(\gamma_1)$ as $\theta \in [\beta_1, \beta_2)$, is in the interior of $P$ then
                Case (B.2) holds and $\{0°, 90°\}$-Kernel$_\theta(P) \neq \emptyset$ for $\theta \in [\beta_1, \beta_2)$. Insert $[\beta_1, \beta_2)$ in $\mathcal{E}$
output $\mathcal{E}$
```

Analysis of Algorithm 3. For STEP I we only need to apply twice Algorithm 1, and then do a refinement of two sequences of sizes $O(n\alpha(n))$, getting a sequence of size $O(n\alpha(n))$ in $O(n \log n)$ time and $O(n\alpha(n))$ space. STEP II has two cases: Case (B.1) takes only constant time to check whether some of the points belongs to the floating rectangle, and it is done $O(n\alpha(n))$ times, giving $O(n\alpha(n))$ total time complexity. Case (B.2) is also done $O(n\alpha(n))$ times but in each of them, we might have to check (in constant time) at most a linear number of vertex events or consecutive subintervals for each of the four vertices of the current floating rectangle. Therefore the total complexities of STEP II are $O(n^2\alpha(n))$ time and space. Notice that the space complexity of the algorithm is $O(n^2\alpha(n))$ because we are storing a sequence $\mathcal{E}$ of (possible) size $O(n^2\alpha(n))$.

Theorem 2.18. *For a simple polygon P with n vertices, the sequence of consecutive intervals for the angles θ such that $\{0°, 90°\}$-$\mathrm{Kernel}_\theta(P) \neq \emptyset$ can be computed in $O(n^2\alpha(n))$ time and space.*

Proof. The discussion above and the analysis of the complexities in Algorithm 3 provide the proof of this theorem. □

2.3.1.3 Generalization to k orientations

One can extend Theorem 2.18 to the case of a set $\mathcal{O} = \{\alpha_1, \ldots, \alpha_k\}$ of k orientations. In particular, Lemma 2.16 can be extended as follows. Instead of the four points $p_N(\gamma_1)$, $p_S(\gamma_1)$, $p_E(\gamma_1)$, and $p_W(\gamma_1)$, we have $2k$ highest/lowest maximum/minimum reflex vertices according to the k different orientations. The extended version of Condition (B.1) requires at least one of them to be inside the convex polygon defined by the intersection of the k strips, what can be checked in $O(k)$ time and space, whereas Condition (B.2) holds if at least one vertex of this convex polygon is inside P, what can be checked in $O(kn^2\alpha(n))$ time and space. Thus, we get the following result.

Theorem 2.19. *For a simple polygon P with n vertices, the sequence of consecutive intervals for the angles θ such that $\{\alpha_1, \ldots, \alpha_k\}$-$\mathrm{Kernel}_\theta(P) \neq \emptyset$ can be computed in $O(kn^2\alpha(n))$ time and space.*

2.3.2 Optimizing the area and perimeter of $\{0°, 90°\}$-Kernel$_\theta(P)$ of simple polygons

Let us consider a $\{0°, 90°\}$-Kernel$_\theta(P)$ at some angle $\theta = \gamma_1$ and suppose that the orientations are rotated to a slightly larger angle β so that the kernels at angle γ_1 and β are defined by the same reflex minima and maxima $p_i(\gamma_i)$, $i \in \{N, W, S, E\}$, and are bounded by the same edges of the polygon. The differential in the area of the kernels in the case shown in Figure 2.7, left, can be expressed in terms of 8 triangles similar to the ones we saw for the $\{0°\}$-Kernel$_\theta(P)$. The case we show in Figure 2.7, right, is more general and we have (for simplicity, we use here p_i instead of $p_i(\gamma_i)$ for $i \in \{N, W, S, E\}$):

$$\begin{aligned} A(\beta) \;=\; & A(\gamma_1) + (A_T(p_S\, d\, p_E) - A_T(p_S\, a\, p_E)) + (A_T(p_E\, e\, p_N) \\ & - A_T(p_E\, b\, p_N)) - A_T(p_N\, r\, t) + A_T(p_E\, s\, u) \\ & - (A_T(p_E\, f\, p_S) - A_T(p_E\, c\, p_S)), \end{aligned} \tag{2.3}$$

where by $A_T(a\,b\,c)$ we denote the area of the triangle with vertices a, b, c. Thus, the differential in the area can be expressed using the area of at most 8 triangles with 1 edge on a polygon edge and at most 4 differences of two triangles with common base and whose third vertex moves along a circular arc. The differential in the perimeter is (see Figure 2.7, right):

$$\begin{aligned} \Pi(\beta) \;=\; & \Pi(\gamma_1) + (\Pi_T(p_S\, d\, p_E) - \Pi_T(p_S\, a\, p_E)) - (\Pi_T(p_E\, e\, p_N) \\ & - \Pi_T(p_E\, b\, p_N)) + \Delta\Pi_T^-(p_N\, r\, t) + \Delta\Pi_T^+(p_E\, s\, u) + (\Pi_T(p_E\, f\, p_S) \\ & - \Pi_T(p_E\, c\, p_S)) \end{aligned} \tag{2.4}$$

where $\Pi_T(a\,b\,c)$ is the perimeter of the triangle with vertices a, b, c and $\Delta\Pi_T^+(a\,b\,c)$ (resp. $\Delta\Pi_T^-(a\,b\,c)$) is the sum (resp. difference) of the difference of the lengths of the edges at angle β and γ_1 plus (resp. minus) the length of the third edge.

To compute and maintain the optimal values for the area and perimeter of the $\{0°, 90°\}$-Kernel$_\theta(P)$, we can use the data computed in Section 2.3.1 about the intervals where this kernel is non-empty. Moreover, we can assume that in each of these intervals there are neither changes in the points of P defining the kernel, nor changes in the vertices of the intersection rectangle of the two kernel strips. In particular, following Lemma 2.16 and Observation 2.17, we compute different intervals for the cases when one, two, three, or the four vertices of the rectangle lie inside the kernel. This only implies a multiplicative constant factor in the number of event intervals. Thus, again a total of $O(n^2\alpha(n))$ intervals arise.

Next, we can analyze the method and formulas to compute the area or the perimeter according to the different types of intervals. We can always assume that we have computed the area or the perimeter of the previous interval, i.e., if we are going to analyze the interval $[\gamma_1, \gamma_2)$, then we know the values of the area and the perimeter for the previous interval $[\gamma_1', \gamma_2')$.

Thus, for the area or perimeter in Case (B.1) of Lemma 2.16, if these four points are inside the kernel as illustrated in Figure 2.7, left, then we have to consider the 8 triangles involved with the formulas for the area or perimeter, in an analogous way as for the case of one orientation $\{0°\}$-Kernel$_\theta(P)$ in Subsections 2.2.3 and 2.2.4. If there are three, two, or only one of the points inside the kernel, it is enough to incorporate the corresponding new formulas for these cases. For the sake of easier reading, and since the complexity of the algorithm does not increase, the details for those cases are omitted.

An analogous situation arises for Case (B.2) of Lemma 2.16: If all four rectangle corners are inside the polygon P, then it is easy to describe the formulas for the area and perimeter. We would have to add new formulas for the cases where there are three, two, or only one corner of the rectangle, but again the complexity of the algorithm does not change and details are omitted.

Thus, it is clear that the relevant issue for the algorithms optimizing area or perimeter is the total time for computing all of the $O(n^2\alpha(n))$ intervals (each one of them can be handled in constant time), which is $O(n^2\alpha(n))$. The space complexity is $O(n\alpha(n))$ because we only maintain the maximum/minimum values of the area or the perimeter but no all of the computed values, thus the used space is essentially for computing the set of event intervals. Notice that when we change from an event interval to the next event interval, we may have to manage a situation like the one illustrated in Figure 2.6 from Proposition 2.12, but this can be done in linear time and space since we translate one side of the kernel in parallel with a endpoint going through vertices on the boundary of P, and it does not change the total time and space complexities. Therefore, we have the following result.

Theorem 2.20. *For a simple polygon P with n vertices, an angle θ such that the area or the perimeter of the $\{0°, 90°\}$-Kernel$_\theta(P)$ are maximum/minimum can be computed in $O(n^2\alpha(n))$ time and $O(n\alpha(n))$ space.*

2.3.2.1 Generalization to k orientations

In a similar way as above, we can extend Theorem 2.20 to the case of a set $\mathcal{O} = \{\alpha_1, \ldots, \alpha_k\}$ of k orientations. Thus, we get the following result[1].

Theorem 2.21. *For a simple polygon P with n vertices, an angle θ such that the area or the perimeter of $\{\alpha_1, \ldots, \alpha_k\}$-*Kernel$_\theta(P)$ *are maximum/minimum can be computed in $O(kn^2\alpha(n))$ time and $O(kn\alpha(n))$ space.*

2.4 Simple orthogonal polygons

In this section, we confine our study to simple orthogonal polygons, showing how the results in Table 2.1 can be improved to those in Table 2.2 for this case.

Each edge of an orthogonal polygon is a N-*edge*, S-*edge*, E-*edge*, or W-*edge* depending on whether it bounds the polygon from the north, south, east, or west, respectively. In particular, for $D \in \{\mathrm{N}, \mathrm{S}, \mathrm{E}, \mathrm{W}\}$, a D-*dent* is a D-edge whose both endpoints are reflex vertices of the polygon. We call a sequence of alternating N- and E-edges a NE-*staircase*, and similarly we define the NW-*staircase*, SE-*staircase*, and SW-*staircase*; clearly, each of these staircases is both x- and y-monotone. Additionally, we characterize the reflex vertices of an orthogonal polygon based on the type of incident edges; more specifically, each reflex vertex incident to a N-edge and an E-edge is called a NE-*reflex vertex*, and analogously we have the NW-, SE- and SW-*reflex vertices*. See Figure 2.9, left. The definition of reflex maxima/minima with respect to some orientation (Definition 2.8) and the angles of the lines L such that both neighbors of a reflex vertex are both below or both above L imply the following observation.

Observation 2.22. (i) For $\theta = 0$ (resp. $\theta = -\frac{\pi}{2}$), only the S- and N-dents (resp. W- and E-dents) contribute reflex minima and maxima, respectively.

(ii) With respect to an orientation $\theta \in (0, \frac{\pi}{2})$, *every* SE-reflex vertex of an orthogonal polygon is a reflex maximum and *every* NW-reflex vertex is a reflex minimum, whereas for $\theta \in (\frac{\pi}{2}, \pi)$, *every* SW-reflex vertex is

[1] Actually, we can compute all angles θ maximizing/minimizing the area/perimeter in of $O(kn^2\alpha(n))$ space.

> a reflex maximum and *every* NE-reflex vertex is a reflex minimum. Analogously, with respect to the orientation $\theta + 90°$, for $\theta \in (0, \frac{\pi}{2})$, *every* SW-reflex vertex is a reflex maximum and *every* NE-reflex vertex is a reflex minimum, whereas for $\theta \in (\frac{\pi}{2}, \pi)$, *every* SE-reflex vertex is a reflex maximum and *every* NW-reflex vertex is a reflex minimum.

As not all SE-reflex and NW-reflex vertices are corners of dents, Observation 2.22 implies that there may be a discontinuity in the area or perimeter of the $\{0°\}$-Kernel$_\theta(P)$ at $\theta = 0$ and $\theta = \frac{\pi}{2}$; these two cases need to be treated separately. Furthermore, it points out a crucial advantage of the orthogonal polygons over simple polygons stated in the following observation.

Observation 2.23. In an orthogonal polygon P, for any $\theta \in (0, \frac{\pi}{2})$ (and similarly for any $\theta \in (-\frac{\pi}{2}, 0)$), the set of reflex minima/maxima does not change, and thus the lines bounding the strip $S_\theta(P)$ rotate in a continuous fashion.

This directly implies that a situation like the one depicted in Figure 2.6 cannot occur. Finally, statement (ii) of Observation 2.22 implies the following corollary.

Corollary 2.24. *Let P be a simple orthogonal polygon. If there are a SE-reflex vertex $u = (x_u, y_u)$ and a NW-reflex vertex $v = (x_v, y_v)$ of P such that $x_u \leq x_v$ and $y_u \geq y_v$, then the $\{0°\}$-*Kernel$_\theta(P)$ *is empty for each $\theta \in (0, \frac{\pi}{2})$. Similarly, if there are a SW-reflex vertex $u' = (x_{u'}, y_{u'})$ and a NE-reflex vertex $v' = (x_{v'}, y_{v'})$ of P such that $x_{u'} \geq x_{v'}$ and $y_{u'} \geq y_{v'}$, then the $\{0°\}$-*Kernel$_\theta(P)$ *is empty for each $\theta \in (\frac{\pi}{2}, \pi)$.*

Proof. To see this, note that for any u, v as in the statement of the corollary, for any $\theta \in (\frac{\pi}{2}, \pi)$, the line through u at angle θ is above the line through v at angle θ; see Figure 2.9, right. Then, because u and v contribute a reflex maximum and a reflex minimum respectively (Observation 2.22(ii)), the strip S_θ is empty and so is the $\{0°\}$-Kernel$_\theta(P)$ by Lemma 2.9. A similar argument works for the vertices u', v'. □

Notation. We denote by $\vartheta_P(a, b)$ the counterclockwise (CCW) boundary chain of polygon P from point a to point b where a and b are located on the boundary of P.

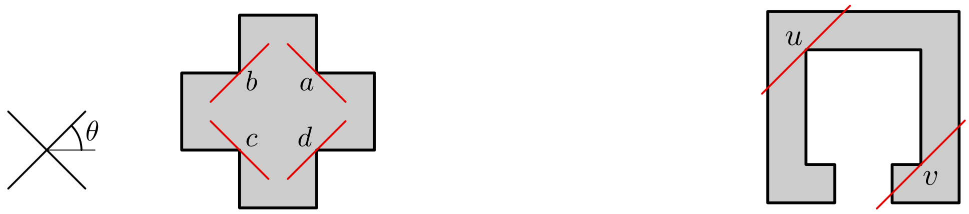


Figure 2.9: Left: The NW-reflex vertex b and the SE-reflex vertex d are a reflex minimum and a reflex maximum with respect to the orientation θ, respectively, whereas the NE-reflex vertex a and the SW-reflex vertex c are a reflex minimum and a reflex maximum with respect to the orientation $\theta+\frac{\pi}{2}$, respectively. Right: Illustration for Corollary 2.24.

2.4.1 The $\{0°\}$-Kernel$_\theta(P)$ of simple orthogonal polygons

We now prove the results in the first row of Table 2.2, focusing on the case for $\theta \in [0, \frac{\pi}{2})$ since the case for $\theta \in [-\frac{\pi}{2}, 0)$ is similar. Observation 2.22 implies that for $\theta = 0$, the $\{0°\}$-Kernel$_\theta(P)$, if non-empty, is determined by a lowest N-dent and a highest S-dent and that for $\theta \in (0, \frac{\pi}{2})$, only the SE-reflex (NW-reflex respectively) vertices contribute reflex maxima (resp. minima).

Let P be a simple orthogonal polygon and suppose that there is at least one SE-reflex vertex in P. Let u be the leftmost SE-reflex vertex of P (in case of ties, take the topmost such vertex), consider the downward-pointing ray $\vec{r}$ emanating from u, and, among its intersections with S-edges of P extending to the left of $\vec{r}$, let s_{SE} be the closest one to u. Similarly, let u' be the topmost SE-reflex vertex of P (in case of ties, take the leftmost such vertex) and let t_{SE} be, among the points of intersection of the rightward-pointing $\vec{r}'$ emanating from u' with an E-edge extending above $\vec{r}'$, the one closest to u'; see Figure 2.10, left.

Next, let C_{SE} be the upper hull of the CCW boundary chain $\vartheta_P(s_{SE}, t_{SE})$; the chain C_{SE} is the blue dashed line in Figure 2.10, left. Similarly, by working with the NW-reflex vertices, we locate the (in case of ties, topmost) rightmost and the (in case of ties, leftmost) bottom-most NW-reflex vertices and we define the points s_{NW} and t_{NW}, and the lower hull C_{NW} of the CCW boundary chain $\vartheta_P(s_{NW}, t_{NW})$. The definition of the chain C_{SE} which states that C_{SE} is the upper hull of $\vartheta_P(s_{SE}, t_{SE})$ and implies that all the vertices of C_{SE} except for s_{SE}, t_{SE} are SE-reflex vertices and the corresponding arguments for the chain C_{NW} imply the following lemma.

Lemma 2.25. *Let s_{SE}, t_{SE}, C_{SE}, s_{NW}, t_{NW}, and C_{NW} of a simple orthogonal polygon P be as defined earlier. If all SE-reflex vertices belong to the CCW boundary chain $\vartheta_P(s_{SE}, t_{SE})$ and all NW-reflex vertices belong to the CCW boundary chain $\vartheta_P(s_{NW}, t_{NW})$, then for any angle $\theta \in (0, \frac{\pi}{2})$, any vertex of C_{SE} (resp. C_{NW}) at which a line at angle θ is tangent to C_{SE} (resp. C_{NW}) is a topmost reflex maximum (resp. lowest reflex minimum) with respect to the orientation at angle θ.*

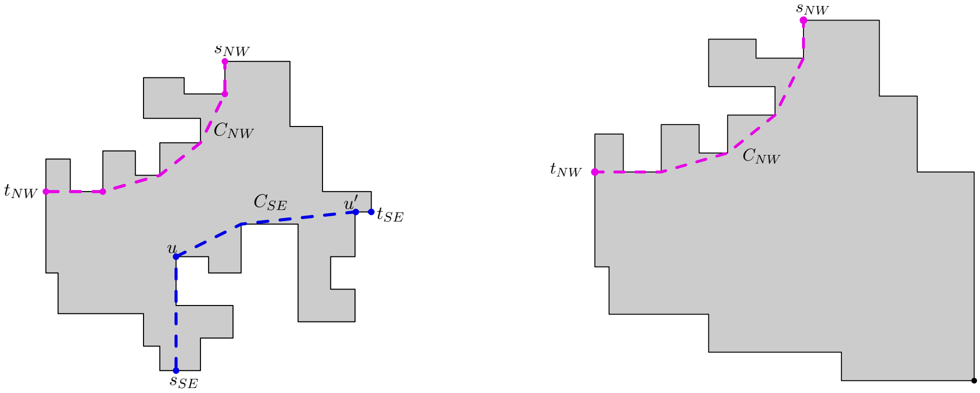


Figure 2.10: Left: An orthogonal polygon and the corresponding convex chains C_{SE} and C_{NW}. Right: An orthogonal polygon without SE-reflex vertices in which we can consider that the convex chain C_{SE} degenerates into vertex v.

Additionally, assuming that the CCW ordering of s_{SE}, t_{SE}, s_{NW}, and t_{NW} around the boundary of P is precisely $s_{SE}, t_{SE}, s_{NW}, t_{NW}$, we can prove the following property of the CCW boundary chains of P from t_{NW} to s_{SE} and from t_{SE} to s_{NW}.

Lemma 2.26. *Let s_{SE}, t_{SE}, s_{NW}, and t_{NW} of a simple polygon P be as defined earlier, and assume that the CCW ordering of s_{SE}, t_{SE}, s_{NW}, and t_{NW} around the boundary of P is precisely $s_{SE}, t_{SE}, s_{NW}, t_{NW}$ and that all SE-reflex vertices belong to the CCW boundary chain $\vartheta_P(s_{SE}, t_{SE})$ and all NW-reflex vertices belong to the CCW boundary chain $\vartheta_P(s_{NW}, t_{NW})$. Then, the CCW boundary chain $\vartheta_P(t_{NW}, s_{SE})$ of P from t_{NW} to s_{SE} is a SW-staircase and the CCW boundary chain $\vartheta_P(t_{SE}, s_{NW})$ from t_{SE} to s_{NW} is a NE-staircase.*

Proof. Let us consider the case of the CCW boundary chain $\vartheta_P(t_{NW}, s_{SE})$ (see Figure 2.10, left); the proof for the chain $\vartheta_P(t_{SE}, s_{NW})$ is symmetric. Since all SE-reflex vertices belong to the CCW boundary chain $\vartheta_P(s_{SE}, t_{SE})$ and all NW-reflex vertices belong to the CCW boundary chain $\vartheta_P(s_{NW}, t_{NW})$, the chain $\vartheta_P(t_{NW}, s_{SE})$ contains neither SE-reflex nor NW-reflex vertices.

Suppose that we start at the W-edge to which t_{NW} belongs (let this edge be uv with v below u) and proceed in CCW order. The edge following the W-edge uv is not a N-edge, otherwise the vertex v would be a NW-reflex vertex, a contradiction. Thus, the edge following the W-edge uv is a S-edge, let it be vw. If $s_{SE} \in vw$, then we are done and the lemma holds. Otherwise, if the edge following the edge vw was an E-edge, then the top vertex of the leftmost edge in the CCW boundary chain $\vartheta_P(w, s_{SE})$ would be a SE-reflex vertex (note that the E-edge incident on w belongs to this chain), a contradiction. Therefore, the edge following the S-edge vw is a W-edge. Then, the above argument can be repeated until we reach the point s_{SE}, implying that the CCW boundary chain $\vartheta_P(t_{NW}, s_{SE})$ is a NW-staircase. □

Lemma 2.26 implies that if the given polygon P has no SE-reflex vertices, then the CCW boundary chain $\vartheta_P(t_{NW}, s_{NW})$ consists of a SW-staircase followed by a NE-staircase; see Figure 2.10, right. A similar result holds if there are no NW-reflex vertices.

2.4.1.1 The existence of the $\{0°\}$-Kernel$_\theta(P)$ for a simple orthogonal polygon P

In this subsection, we give an algorithm to determine when $\{0°\}$-Kernel$_\theta(P)$ for a simple orthogonal polygon P is non-empty. First, if no SE-reflex vertex exists, then no S-dents exist and as mentioned, the chain C_{SE} degenerates into the rightmost lowest vertex (see Figure 2.10, right) which thus belongs to the $\{0°\}$-Kernel$_\theta(P)$ for all $\theta \in (0, \frac{\pi}{2})$; thus, the $\{0°\}$-Kernel$_\theta(P)$ is non-empty for all $\theta \in [0, \frac{\pi}{2})$. A similar argument holds if no NW-reflex vertex exists. So, in the following, we assume that the polygon P has SE-reflex and NW-reflex vertices. Then, we show the following lemma.

Lemma 2.27. *Let s_{SE}, t_{SE}, C_{SE}, s_{NW}, t_{NW}, and C_{NW} of a simple orthogonal polygon P be as defined earlier.*

(i) Let Q_{SE} be the convex part of the plane bounded from the left and

above by C_{SE}, the downward-pointing ray emanating from s_{SE}, and the rightward-pointing ray emanating from t_{SE}. Similarly, let Q_{NW} be the convex part of the plane bounded from the right and below by C_{NW}, the upward-pointing ray emanating from s_{NW}, and the leftward-pointing ray emanating from t_{NW}.

(a) *If the interiors of Q_{SE} and Q_{NW} intersect, then $\{0°\}$-Kernel$_\theta(P)$ is empty for each $\theta \in (0, \frac{\pi}{2})$.*

(b) *If the interiors of Q_{SE} and Q_{NW} do not intersect but Q_{SE} and Q_{NW} touch at a common point z, then the $\{0°\}$-Kernel$_\theta(P)$ degenerates to a line segment for each θ equal to the angle of each common interior tangent of C_{SE} and C_{NW} at z, and is empty for all other values of θ.*

(ii) If there exists a SE-reflex vertex not belonging to the CCW boundary chain $\vartheta_P(s_{SE}, t_{SE})$ or a NW-reflex vertex not belonging to the CCW boundary chain $\vartheta_P(s_{SE}, t_{SE})$, then the $\{0°\}$-Kernel$_\theta(P)$ is empty for each $\theta \in (0, \frac{\pi}{2})$.

Proof. (i.a) Let p be a point in the intersection of the interiors of the unbounded convex polygons Q_{SE} and Q_{NW}. Then, for any angle $\theta \in [0, \frac{\pi}{2})$, p lies below the tangent to C_{SE} at angle θ and above the tangent to C_{NW} at angle θ and thus the strip $S_\theta(P)$ is empty. Therefore, for each $\theta \in [0, \frac{\pi}{2})$, the strip S_θ is empty and, by Lemma 2.9, so is the $\{0°\}$-Kernel$_\theta(P)$.

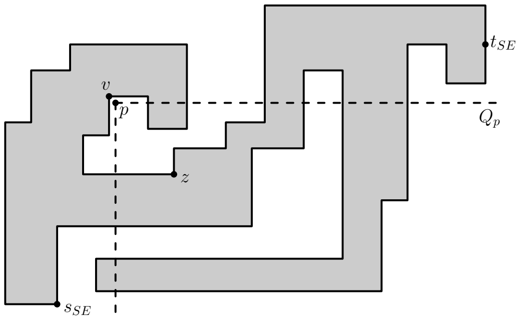


Figure 2.11: The quadrant Q_p in the proof of Lemma 2.27(ii).

(i.b) If Q_{SE} and Q_{NW} touch along their horizontal rays, then the $\{0°\}$-Kernel$_\theta(P)$ is a horizontal line segment if $\theta = 0$, otherwise it is empty. Similarly, if they touch along their vertical rays, then the $\{0°\}$-Kernel$_\theta(P)$ is a vertical line segment if $\theta = \frac{\pi}{2}$, otherwise it is empty. Next, assume that

Q_{SE} and Q_{NW} touch at a point of C_{SE} and C_{NW}. Then, because Q_{SE} and Q_{NW} are convex, they touch at a connected portion of C_{SE} and C_{NW}, that is, they touch at a point or a line segment. In either case, for any angle θ of any common interior tangent to C_{SE} and C_{NW}, the $\{0°\}$-Kernel$_\theta(P)$ is a line segment, otherwise it is empty.

(ii) Let us concentrate on the case of a SE-reflex vertex of P, say v, not belonging to the CCW boundary chain $\theta_P(s_{SE}, t_{SE})$. (The case of a NW-reflex vertex not belonging to the CCW boundary chain $\vartheta_P(s_{NW}, t_{NW})$ is similar.) Let p be a point infinitesimally to the right and below v so that p is outside P. Since the chain $\vartheta_P(s_{SE}, t_{SE})$ is determined by the leftmost and the topmost SE-reflex vertices, the x-coordinate of v is larger than the x-coordinate of s_{SE} while the y-coordinate of v is smaller than the y-coordinate of t_{SE}. This implies that $\vartheta_P(s_{SE}, t_{SE})$ intersects both the rightward-pointing horizontal ray $\vec{r}_\rightarrow$ emanating from p and the downward-pointing vertical ray $\vec{r}_\downarrow$ emanating from p. Let Q_p be the closed quadrant delimited by the rays $\vec{r}_\rightarrow$ and $\vec{r}_\downarrow$ (see Figure 2.11). Consider the set A_C of all minimal boundary chains of P that lie in Q_P and are delimited by a point on $\vec{r}_\rightarrow$ and a point on $\vec{r}_\downarrow$ (the minimality implies that no point in such a chain other than its endpoints belongs to either $\vec{r}_\rightarrow$ or $\vec{r}_\downarrow$); these chains do not intersect, therefore they are totally ordered and the ordering is the same as the ordering of their endpoints on $\vec{r}_\rightarrow$ or $\vec{r}_\downarrow$. Among the chains in A_C, let C be the chain with endpoint on $\vec{r}_\rightarrow$ closest to p. Since p is outside the polygon P, then the interior of P is to the left of C as we walk along it from its endpoint on $\vec{r}_\rightarrow$ to its endpoint on $\vec{r}_\downarrow$; see Figure 2.11. Then, the right vertex of a lowest horizontal edge in C is a NW-reflex vertex (e.g., vertex z in Figure 2.11). Since this vertex is below and to the right of the SE-reflex vertex v, Corollary 2.24 implies that the $\{0°\}$-Kernel$_\theta(P)$ is empty for each $\theta \in (0, \frac{\pi}{2})$. □

So, assume that none of the cases of Lemma 2.27 holds. Then, the chains C_{SE} and C_{NW} do neither intersect nor touch, and the inner common tangents to C_{SE} and C_{NW} are well defined; let them be T_1 and T_2 with the slope of T_1 being smaller than the slope of T_2, and let ϕ_1, ϕ_2 be the CCW angle with respect to the positive x-axis of T_1 and T_2, respectively. If the y-coordinate of t_{NW} is greater than the y-coordinate of t_{SE}, then we set $\theta_{min} = 0$, otherwise $\theta_{min} = \phi_1$. Similarly, we define θ_{max} to be equal to $\frac{\pi}{2}$ if the x-coordinate of s_{SE} is greater than the x-coordinate of s_{NW}, otherwise $\theta_{max} = \phi_2$. For example, in Figure 2.10, left, $\theta_{min} = 0$ and

$\theta_{max} < \frac{\pi}{2}$. Then, since for $\theta \in (0, \frac{\pi}{2})$ the strip $S_\theta(P)$ is non-empty if and only if $\theta \in [\theta_{min}, \theta_{max}] \cap (0, \frac{\pi}{2})$, by Lemma 2.5 we have:

Lemma 2.28. *Let P be a simple orthogonal polygon such that none of the cases of Lemma 2.27 hold and consider that $\theta \in (0, \frac{\pi}{2})$. Then, if no SE-reflex or NW-reflex vertices exist, the $\{0°\}$-Kernel$_\theta(P)$ is non-empty for each $\theta \in (0, \frac{\pi}{2})$, otherwise the $\{0°\}$-Kernel$_\theta(P)$ is non-empty if and only if $\theta \in [\theta_{min}, \theta_{max}] \cap (0, \frac{\pi}{2})$.*

Corollary 2.29. *The values of the angle $\theta \in [0, \frac{\pi}{2})$ for which the $\{0°\}$-*Kernel$_\theta(P)$ *of a simple orthogonal polygon P is non-empty form a single interval and potentially the value $\theta = 0$.*

Based on the above discussion, we outline our algorithm in Algorithm 4.

Analysis of Algorithm 4. The correctness of Algorithm 4 follows from the fact that if no SE-reflex or no NW-reflex vertices exist, the $\{0°\}$-Kernel$_\theta(P)$ is non-empty for all $\theta \in [0, \frac{\pi}{2})$, and from Observation 2.22 and Lemmas 2.27 and 2.28.

Computing the SE-reflex and NW-reflex vertices, the N- and S-dents, and then finding a lowest N-dent and a highest S-dent can be done in $O(n)$ time. Thus, STEP 1 can be completed in $O(n)$ time and $O(1)$ space. Computing the points $s_{SE}, t_{SE}, s_{NW}, t_{NW}$ can be done in $O(n)$ time. The chains C_{SE} and C_{NW} can be computed in $O(n)$ time as well [83]. As the size of C_{SE} and C_{NW} is $O(n)$ and they are x-monotone, we can check whether they cross or touch in $O(n)$ time by walking along them from their leftmost to their rightmost endpoint in lockstep fashion. Computing the angle of the line supporting the segment I and the angle ranges of the tangents at z can be done in $O(1)$ time. The inner common tangents to C_{SE} and C_{NW} can be computed in $O(\log n)$ time (in a fashion similar to computing the outer ones [63]), from which we can compute θ_{min} and θ_{max} in $O(1)$ time. Hence, STEP 2 requires $O(n)$ time and $O(n)$ space. Finally, STEP 3 takes $O(1)$ time and space. In summary, we have:

Theorem 2.30. *For a simple orthogonal polygon P with n vertices, the intervals of $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2})$ for which $\{0°\}$-*Kernel$_\theta(P) \neq \emptyset$ *can be computed in $O(n)$ time and space.*

Algorithm 4 Computing the intervals of θ such that $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$ for a simple orthogonal polygon P

```
Input: A simple orthogonal polygon $P$ with $n$ vertices
Output: The intervals of the angle $\theta$ such that $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$

STEP 1: CHECK IF $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$ FOR $\theta = 0$
if no SE-reflex vertices exist or no NW-reflex vertices exist then
    output $[0, \frac{\pi}{2})$ and stop
compute the N- and S-dents of $P$ and let $e_N$ and $e_S$ be a lowest N-dent
    and a highest S-dent, respectively
if the $y$-coordinate of $e_N$ is smaller than the $y$-coordinate of $e_S$ then
    $solution\_for\_0 \leftarrow \emptyset$
else
    $solution\_for\_0 \leftarrow [0, 0]$

STEP 2: CHECK IF $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$ FOR $\theta \in (0, \frac{\pi}{2})$
compute the points $s_{SE}, t_{SE}, s_{NW}, t_{NW}$ and the convex chains $C_{SE}$
    and $C_{NW}$
check whether $C_{SE}$ and $C_{NW}$ cross, touch, or do not intersect
if there exists a SE-reflex vertex not in $\vartheta_P(s_{SE}, t_{SE})$ or a NW-reflex
    vertex not in $\vartheta_P(s_{NW}, t_{NW})$ or the chains $C_{SE}$ and $C_{NW}$ cross then
    $solution\_for\_0\text{-}\frac{\pi}{2} \leftarrow \emptyset$
else if the chains $C_{SE}$ and $C_{NW}$ share a line segment $I$ then
    $solution\_for\_0\text{-}\frac{\pi}{2} \leftarrow [\theta_I, \theta_I] \cap (0, \frac{\pi}{2})$ where $\theta_I$ is the angle of the line
    supporting the line segment $I$
else if the chains $C_{SE}$ and $C_{NW}$ touch at a single point $z$ then
    let $R_{SE}(z)$ ($R_{NW}(z)$, resp.) be the angle interval of the tangent to
    $C_{SE}$ (to $C_{NW}$ resp.) at $z$
    $solution\_for\_0\text{-}\frac{\pi}{2} \leftarrow R_{SE}(z) \cap R_{NW}(z) \cap (0, \frac{\pi}{2})$
else
    compute the inner common tangents to $C_{SE}$ and $C_{NW}$
    compute angles $\theta_{min}$ and $\theta_{max}$ as explained in the paragraph preceding
    Lemma 2.28
    $solution\_for\_0\text{-}\frac{\pi}{2} \leftarrow [\theta_{min}, \theta_{max}] \cap (0, \frac{\pi}{2})$

STEP 3: OUTPUT RESULTS
output $solution\_for\_0 \cup solution\_for\_0\text{-}\frac{\pi}{2}$
```

2.4.1.2 Optimizing the area and perimeter of the $\{0°\}$-Kernel$_\theta(P)$ for a simple orthogonal polygon P

In this subsection, we present an algorithm that computes an angle $\theta \in [0, \frac{\pi}{2})$ such that the area (or perimeter) of the $\{0°\}$-Kernel$_\theta(P)$ is maximized; minimization works similarly.

If the $\{0°\}$-Kernel$_\theta(P)$ for $\theta = 0$ is non-empty, we compute its area (or perimeter) and we use these to set the current maximum value and the current angle of the maximum that we maintain; if the $\{0°\}$-Kernel$_0(P)$ is empty, then its area and perimeter are set to 0. Next, we work for $\theta \in (0, \frac{\pi}{2})$. For the sake of generality, in the following, we consider that the polygon P has both SE-reflex and NW-reflex vertices; if one of these two vertex types is missing, then we skip the computations involving that vertex type, whereas if both vertex types are missing, then the kernel is the entire polygon P and we simply need to compute the area or perimeter of P.

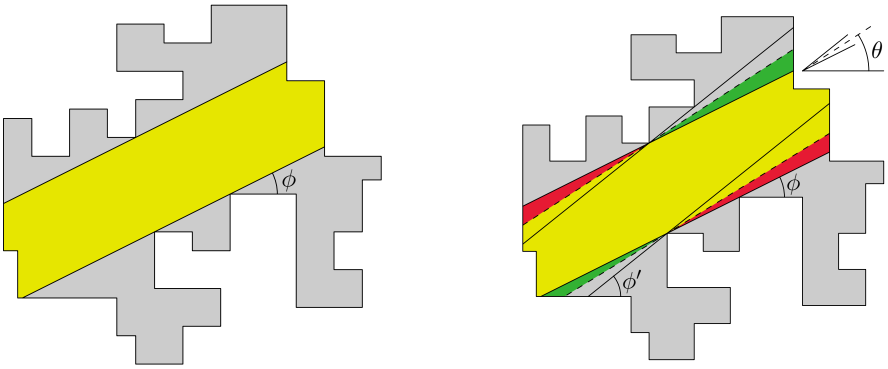


Figure 2.12: Left: $\{0°\}$-Kernel$_\theta(P)$ for $\theta = \phi$. Right: Optimizing the area/perimeter of the $\{0°\}$-Kernel$_\theta(P)$ for $\theta \in [\phi, \phi')$.

Next, we check whether the conditions of Lemma 2.27 hold; if they do, the area of each of the degenerate kernels that arise is equal to 0, whereas, whenever the kernel is non-empty, its perimeter can be computed in $O(1)$ time. Otherwise, we compute the interval $A = [\theta_{min}, \theta_{max}] \cap [0, \frac{\pi}{2})$ as in Algorithm 4; we need to maximize the area or perimeter of the $\{0°\}$-Kernel$_\theta(P)$ for any angle $\theta \in A$. We start at $\theta = \theta_{min}$ and we explicitly compute the $\{0°\}$-Kernel$_{\theta_{min}}(P)$ and its area (perimeter), which is the current area (perimeter) maximum (we note that if $\theta_{min} = 0$, we compute the area of the intersection of the polygon P with a horizontal strip defined by the highest SE-reflex vertex from below and the lowest NW-reflex vertex from above).

Subsequently, as in Section 2.2.2, we partition the interval A into angular subintervals, in each of which the following property holds:

Property 1 The kernel involves the same topmost reflex maximum and lowest reflex minimum and the same edges of the polygon.

For the resulting partition, say P_A, the following lemma holds.

Lemma 2.31. *For an orthogonal polygon P with n vertices, the size of the partition P_a of $A = [\theta_{min}, \theta_{max}] \cap [0, \frac{\pi}{2})$ is $O(n)$.*

Proof. Let P_{SE} be the partition of the interval $A = [\theta_{min}, \theta_{max}] \cap [0, \frac{\pi}{2})$ based on which vertex of the chain C_{SE} is the current topmost reflex maximum and on which edges of the polygon bound the lower segment of the strip S_θ. Then, an angle $\theta \in A$ is a partition point if it is

- the angle of an edge of the chain C_{SE} (see Lemma 2.25 because at that angle the topmost reflex maximum changes, or
- the angle of the tangent from a vertex of P to the chain C_{SE} because at that point the lower segment of the strip S_θ moves to another edge.

Because the segments bounding the strip S_θ rotate in a continuous fashion (Observation 2.23), the number of vertices of the chain C_{SE} and the polygon is $O(n)$, the size of P_{SE} is $O(n)$. Similarly, the size of the corresponding partition P_{NW} related to the current lowest reflex minimum and the chain C_{NW} is $O(n)$ as well. Then, the partition P_A is the refinement of the partition P_{SE} by means of the partition P_{NW}, which yields that its size is $O(n)$. □

After the partition P_A has been computed, we process the subintervals in increasing angle value and in each such interval $[\beta_j, \beta_{j+1})$, we maximize the area/perimeter as a function of an angle $\beta \in [\beta_j, \beta_{j+1})$ by taking into account the area/perimeter of $\{0°\}$-Kernel$_{\beta_j}(P)$ and of the two green triangles and the two red triangles in the spirit of Equation 2.1, as shown in Figure 2.12, right. The area (respectively perimeter) of each of these four triangles depends linearly on $\tan\beta$ and $\cot\beta$ (resp. linearly on $(1 \pm \cos\beta)/\sin\beta$ and $(1 \pm \sin\beta)/\cos\beta$), see Section 2.5.

Based on the above discussion, we outline our algorithm to maximize the area of $\{0°\}$-Kernel$_\theta(P)$ in Algorithm 5.

Algorithm 5 Computing the maximum area of $\{0°\}$-Kernel$_\theta(P)$ for a simple orthogonal polygon P

Input: A simple orthogonal polygon P with n vertices
Output: A value of the angle θ such that the area of $\{0°\}$-Kernel$_\theta(P)$ is maximum

```
STEP 1: CHECK SPECIAL CASES
execute Algorithm 4 to compute the set $T$ of values of $\theta$ for which $\{0°\}$-Kernel$_\theta(P) \neq \emptyset$, and $\theta_{min}, \theta_{max}$, if they can be defined
$current\_angle \leftarrow 0$
if $0 \in T$ then
    compute the area of $\{0°\}$-Kernel$_\theta(P)$ for $\theta = 0$
    $current\_max \leftarrow$ computed area
else
    $current\_max \leftarrow 0$
if any of the conditions of Lemma 2.27 holds then
    output $current\_max$ and stop

STEP 2: MAXIMIZE THE AREA OF $\{0°\}$-Kernel$_\theta(P)$ FOR $\theta \in A = [\theta_{min}, \theta_{max}] \cap [0, \frac{\pi}{2})$
compute the area of $\{0°\}$-Kernel$_\theta(P)$ for $\theta = \theta_{min}$
$\theta \leftarrow \theta_{min}$
while $\theta < \theta_{max}$ do
    compute the angles for which the highest reflex maximum and the lowest reflex minimum change
    compute the angles for each of the segments bounding the strip $S_\theta$ to reach the next vertex of the polygon $P$
    $\delta \leftarrow$ the minimum among the angles computed in the 2 preceding lines
    maximize the area of the $\{0°\}$-Kernel$_\theta(P)$ for $\theta \in [\theta, \theta + \delta)$ by using the expressions for the area in Section 2.5.
    update, if needed, the current maximum area value $current\_max$ and the corresponding angle $current\_angle$
output $current\_max$ and $current\_angle$
```

Analysis of Algorithm 5. The correctness of Algorithm 5 follows from Observation 2.22, Lemmas 2.26, 2.27, and 2.28, and the preceding discussion. Algorithm 4 requires $O(n)$ time and space, as do the computation of the $\{0°\}\text{-Kernel}_\theta(P)$ for $\theta = 0$ and its area, and checking the conditions of Lemma 2.27. Thus, STEP 1 takes $O(n)$ time and space. Computing the $\{0°\}\text{-Kernel}_\theta(P)$ for $\theta = \theta_{min}$ can be explicitly done in $O(n)$ time and space. Each iteration of the `while` loop in STEP 2 takes $O(1)$ time as it involves accessing and processing in $O(1)$ time at most 8 neighboring vertices and maximizing a constant-degree polynomial. Moreover, it is important to note that the subintervals processed in the `while` loop precisely form the partition P_A. Since a different subinterval is processed in each iteration of the while loop and since the number of subintervals is $O(n)$ (Lemma 2.31), the execution of the `while` loop in STEP 2 takes $O(n)$ time. Hence, by also taking into account that minimization, where meaningful, can be handled analogously, we get:

Theorem 2.32. *For a simple orthogonal polygon P with n vertices, the values of θ such that the area or the perimeter of the $\{0°\}\text{-Kernel}_\theta(P)$ are maximum/minimum can be computed in $O(n)$ time and space.*

2.4.2 The rotated $\{0°, 90°\}\text{-Kernel}_\theta(P)$ of a simple orthogonal polygon P

We now extend our study to $\mathcal{O} = \{0°, 90°\}$ for a simple orthogonal polygon P, proving the results in the second row of Table 2.2. Observe that it suffices to consider $\theta \in [0, \frac{\pi}{2})$ since $\{0°, 90°\}\text{-Kernel}_0(P) = \{0°, 90°\}\text{-Kernel}_{\frac{\pi}{2}}(P)$. Again, Observation 2.3 and Lemma 2.9 imply that

$$\begin{aligned} \{0°, 90°\}\text{-Kernel}_\theta(P) &= \{0°\}\text{-Kernel}_\theta(P) \ \cap \ \{90°\}\text{-Kernel}_\theta(P) \\ &= (\, S_\theta(P) \cap P \,) \ \cap \ (\, S_{\theta+90°}(P) \cap P \,) \end{aligned} \tag{2.5}$$

and therefore $\{0°, 90°\}\text{-Kernel}_\theta(P) = S_\theta(P) \cap S_{\theta+90°}(P) \cap P$, that is, the case is an extension of the $\{0°\}\text{-Kernel}_\theta(P)$ with two strips $S_\theta(P)$ and $S_{\theta+90°}(P)$, which are perpendicular to each other.

For $\theta = 0$, the $\{0°, 90°\}\text{-Kernel}_\theta(P)$ is the intersection of the polygon P with the horizontal strip determined above by the lowest N-dent and below by the topmost S-dent, and with the vertical strip determined to the left by the rightmost W-dent and to the right by the leftmost E-dent. Thus, the $\{0°, 90°\}$-Kernel$_0(P)$ may have reflex vertices (but no dents) at the top left, top right, bottom left or bottom right corners and is orthogonally convex.

Below we consider the case for $\theta \in (0, \frac{\pi}{2})$. In accordance with Observation 2.22, all reflex vertices are reflex maxima or minima with respect to one of the orientations in $\mathcal{O}_\theta$; then, the definition of the $\{0°, 90°\}$-Kernel$_\theta(P)$ implies that:

Observation 2.33. For $\theta \in (0, \frac{\pi}{2})$, the $\{0°, 90°\}$-Kernel$_\theta(P)$ is a convex polygon.

See Figure 2.13, right for an example. Recall that for $\theta = 0$, the $\{0°, 90°\}$-Kernel$_\theta(P)$ is not necessarily convex, but it is orthogonally convex.

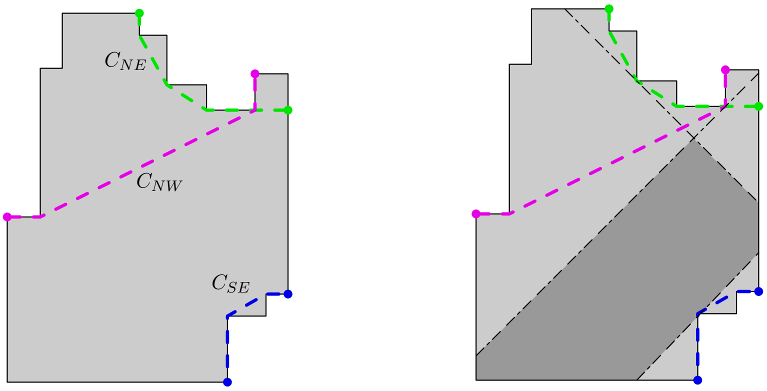


Figure 2.13: Left: A simple orthogonal polygon P and the convex chains C_{SE}, C_{NE}, C_{NW}; no SW-reflex vertices exist. Right: The $\{0°, 90°\}$-Kernel$_\theta(P)$ for $\theta = \frac{\pi}{4}$ is shown darker.

As mentioned above, in this case, the kernel is in general defined by the two perpendicular strips $S_\theta(P)$ and $S_{\theta+90°}(P)$. Let us investigate the cases that may arise for the points of intersection of the lines bounding these strips. So, consider an angle $\theta \in (0, \frac{\pi}{2})$ such that there is at least one reflex maximum in the orientation θ and at least one reflex minimum in the orientation $\theta + 90°$, and let $\ell_{\theta,\downarrow}$ (resp. $\ell_{90°+\theta,\uparrow}$) be the bottom (resp. top) segment bounding $S_\theta(P)$ (resp. $S_{\theta+90°}(P)$). Moreover, let p (resp. q) be the right endpoint of $\ell_{\theta,\downarrow}$ (resp. $\ell_{90°+\theta,\uparrow}$). Clearly p belongs to a N- or an E-edge, and similarly, q belongs to a S- or an E-edge. Each of the above possibilities for p and q may well arise if the segments $\ell_{\theta,\downarrow}$ and $\ell_{90°+\theta,\uparrow}$ intersect (see

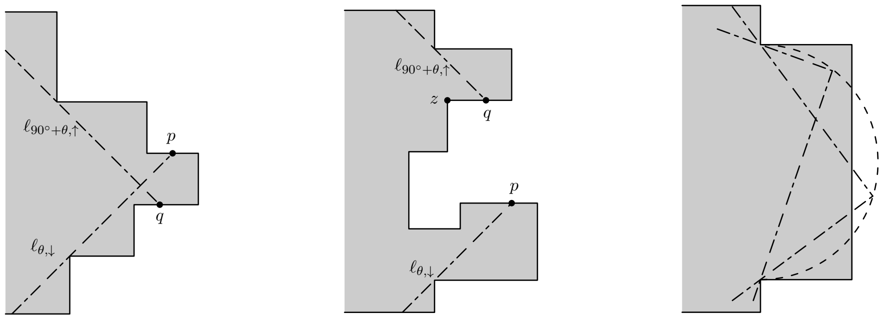

$\ell_{90°+\theta,\uparrow}$
p
q
$\ell_{\theta,\downarrow}$
$\ell_{90°+\theta,\uparrow}$
z
q
p
$\ell_{\theta,\downarrow}$

Figure 2.14: Left: The segments $\ell_{\theta,\downarrow}$ and $\ell_{90°+\theta,\uparrow}$ intersect. Middle: For Lemma 2.34; an impossible configuration. Right: As the angle θ increases, the segments $\ell_{\theta,\downarrow}$ and $\ell_{90°+\theta,\uparrow}$ intersect, later they stop doing so, and later they intersect again.

Figure 2.14, left); the point of intersection lies in the polygon P and, as the strips rotate, it moves along an arc of a circle whose diameter is the line segment connecting the reflex maximum and the reflex minimum about which $\ell_{\theta,\downarrow}$ and $\ell_{90°+\theta,\uparrow}$, respectively, rotate. However, if these two segments do not intersect, then only one case for the relative location of p and q is possible, as we show in the following lemma.

Lemma 2.34. *Let P be a simple orthogonal polygon and suppose that the conditions of Lemma 2.27 hold neither for the SE-reflex and NW-reflex vertices, nor for the SW-reflex and the NE-reflex vertices. Let segments $\ell_{\theta,\downarrow}$ and $\ell_{90°+\theta,\uparrow}$ and points p, q be defined as above. If $\ell_{\theta,\downarrow}$ and $\ell_{90°+\theta,\uparrow}$ do not intersect, then p and q belong to the same E-edge of P.*

Proof. The tangency of the segments $\ell_{\theta,\downarrow}$ and $\ell_{90°+\theta,\uparrow}$ to the chains C_{SE} and C_{NE}, respectively, implies that p, q belong (in fact, in that order) to the CCW boundary chain $\vartheta_P(t_{SE}, s_{NE})$ of P. Suppose, for contradiction, that the point p belongs to a N-edge. Then, no matter whether q belongs to an E-edge or a S-edge, the left vertex of the topmost edge of the CCW boundary chain from p to q is a SE-reflex vertex that is higher than p and thus higher than t_{SE} (see vertex z in Figure 2.14, middle), in contradiction to the assumption that Lemma 2.27, statement (ii), does not hold for the chain C_{SE}. Thus, p belongs to an E-edge. The exact same argument enables us to show that q belongs to an E-edge, and in fact that p, q belong to the same E-edge. □

Since a circular arc (the locus of the intersection points of the lines supporting the rotating segments $\ell_{\theta,\downarrow}$ and $\ell_{90^\circ+\theta,\uparrow}$) and a line segment (e.g., an E-edge) intersect in at most two points (see Figure 2.14, right), the above lemma implies that we may need to consider at most 3 angular subintervals for the at most 3 different cases to consider for the pair of $\ell_{\theta,\downarrow}$ and $\ell_{90^\circ+\theta,\uparrow}$). As there are at most 4 such pairs, we have:

Observation 2.35. An angle interval satisfying Property 1 (Section 2.4.1.2) may need to be broken into at most 12 sub-intervals.

Additionally, Lemma 2.34 readily implies that if the segments $\ell_{\theta,\downarrow}$ and $\ell_{90^\circ+\theta,\uparrow}$ do not intersect, then, in the boundary of the $\{0^\circ, 90^\circ\}$-Kernel$_\theta(P)$, p and q are connected by a part of an edge of P. Note that the kernel has one fewer edge if $\ell_{\theta,\downarrow}$ and $\ell_{90^\circ+\theta,\uparrow}$ intersect or if exactly one of these segments rotates around a degenerate chain, that is a point (see Figure 2.13, right); similar results hold for the remaining 4 pairs of “consecutive" segments and more occurrences of the above cases result into a kernel of even fewer edges. Therefore:

Corollary 2.36. *For $\theta \in (0, \frac{\pi}{2})$, the rotated $\{0^\circ, 90^\circ\}$-*Kernel$_\theta(P)$ *of an orthogonal polygon P has at most 8 edges.*

2.4.2.1 The existence of the $\{0^\circ, 90^\circ\}$-Kernel$_\theta(P)$ of a simple orthogonal polygon P

In this subsection, we give an algorithm to determine when the $\{0^\circ, 90^\circ\}$-Kernel$_\theta(P)$ for a simple orthogonal polygon P is non-empty. The algorithm relies on the following lemma, which is an extension of Lemma 2.27.

Lemma 2.37. *If the conditions of Lemma 2.27 hold for either the SE-reflex and NW-reflex vertices and the chains C_{SE} and C_{NW}, or the SW-reflex and NE-reflex vertices and the chains C_{SW} and C_{NE}, then the $\{0^\circ, 90^\circ\}$-*Kernel$_\theta(P)$ *either is empty or degenerates to a point or a line segment.*

If Lemma 2.37 does not apply, then we work as in Section 2.3.1, namely, we determine a sequence $\mathcal{I}$ of angle intervals such that each interval in $\mathcal{I}$ satisfies Property 1 (Section 2.4.1.2). Then, for each such event interval $[\gamma, \gamma')$, we find the values of $\theta \in [\gamma, \gamma')$ such that at least one of the corners of the floating rectangle R_θ lies in P. For a corner r, this computation can be easily done by comparing the position of r with the position of the corresponding endpoint, say s, of the line segment that bounds the strip $S_\theta(P)$ and whose supporting line defines r, that is by comparing the locus of the corner for $\theta \in [\gamma, \gamma')$ (which is a circular arc) with the edge on which s lies; see Figure 2.14, left and right. Our algorithm is outlined in Algorithm 6.

Computing the $\{0°, 90°\}$-Kernel$_\theta(P)$ for $\theta = 0$ can be done in $O(n)$ and space; recall that it is defined by the lowest N-dent, by the topmost S-dent, by the rightmost W-dent, and by the leftmost E-dent. $O(n)$ time is also need to check the conditions of Lemma 2.37 and $O(n)$ time and space suffice to compute the $\{0°, 90°\}$-Kernel$_\theta(P)$ if any one of these conditions holds. Thus, STEP 1 can be completed in $O(n)$ time ans space. STEP 2 of Algorithm 5 takes $O(n)$ time and space and hence, and so does the entire STEP 2 of Algorithm 6; note that the refinement of two interval sequences of $O(n)$ size each (Lemma 2.31) can be done in $O(n)$ time and produces a sequence of $O(n)$ size. In STEP 3, checking case (B.1) in Observation 2.16 can be done in $O(1)$ time by locating each of the $p_N(\ ), p_S(\ )$ against the strip $S_{\theta+90°}$ and each of the $p_E(\ ), p_W(\ )$ against the strip S_θ. For case (B.2), for each of the 4 corners, we determine the values of θ, for which the circular arc traced by the corner for $\theta \in [\delta, \delta')$ intersects any of the (at most 8) edges of the polygon that delimit the segments bounding the strips S_θ and $S_{\theta+90°}$ (see Figure 2.14); then, by taking into account whether the corner at $\theta = \delta$ lies in P or not, we can find the values of the angle θ for which the corner lies in P. Then, for case (B.2), the values of θ sought are precisely the union of the angle values computed for each corner of the rectangle R_θ; this takes $O(1)$ time as well. Moreover, since the sequence $\mathcal{I}$ is of $O(n)$ size and because of Observation 2.35, we have:

Observation 2.38. The total number of subintervals in the sequence $\mathcal{E}$ is $O(n)$.

Finally, since the `for`-loop in STEP 3 is repeated $O(n)$ times, STEP 3 is completed in $O(n)$ time and space. Thus:

Theorem 2.39. *For a simple orthogonal polygon P with n vertices, the intervals of $\theta \in [0, \frac{\pi}{2})$ for which $\{0°, 90°\}$-Kernel$_\theta(P) \neq \emptyset$ can be computed in $O(n)$ time and space.*

Algorithm 6 Computing intervals of θ such that $\{0°, 90°\}$-Kernel$_\theta(P) \neq \emptyset$ for a simple orthogonal polygon P

```
Input: A simple orthogonal polygon $P$ with $n$ vertices
Output: Sequence $\mathcal{E}$ of intervals for angles $\theta$ such that $\{0°, 90°\}$-Kernel$_\theta(P) \neq \emptyset$

STEP 1: CASE FOR $\theta = 0$ AND APPLY LEMMA 2.37
compute the $\{0°, 90°\}$-Kernel$_\theta(P)$ for $\theta = 0$
if $\{0°, 90°\}$-Kernel$_0(P) \neq \emptyset$ then
    $solution\_for\_0 \leftarrow [0, 0]$
if the conditions of Lemma 2.37 hold then
    apply STEP 2 of Algorithm 4 to the $\{0°\}$-Kernel$_\theta(P)$ or $\{90°\}$ - Kernel$_\theta(P)$ that is degenerate, compute the values of the angle $\theta$ (if any) for which it is non-empty, and compute the subset $solution2$ of these values for which the intersection of these kernels is non-empty
    output $solution\_for\_0 \cup solution2$ and stop

STEP 2: COMPUTE EVENT INTERVALS
compute the sequence $\mathcal{I}_{0°}$ of subintervals having Property 1 as in STEP 2 of Algorithm 5
compute the sequence $\mathcal{I}_{90°}$ of subintervals having Property 1 as in STEP 2 of Algorithm 5
Refine $\mathcal{I}_{0°}$ by using $\mathcal{I}_{90°}$ into the sequence $\mathcal{I} = \{I' \cap I'' = [\gamma, \gamma') \mid I' \in \mathcal{I}_{0°}$ and $I'' \in \mathcal{I}_{90°}\}$

STEP 3: CHECK CORNERS OF FLOATING RECTANGLE
for each angle interval $[\phi, \phi') \in \mathcal{I}$ do
    by determining the values of the angle $\theta \in [\phi, \phi')$ for which at least one of the cases (B.1), (B.2) in Lemma 2.16 holds, find the values of the angle $\theta$ for which the corner lies in $P$
    insert these values, if any, in an initially empty sequence $\mathcal{E}$
output $\mathcal{E}$
```

2.4.2.2 Optimizing the area and perimeter of the $\{0°, 90°\}$-$\mathrm{Kernel}_\theta(P)$ of a simple orthogonal polygon P

Our algorithm for the problem of optimizing the area/perimeter of the $\{0°, 90°\}$-$\mathrm{Kernel}_\theta(P)$ for a simple orthogonal polygon P follows the steps of Algorithm 6. It treats the case for $\theta = 0$ as a special case and computes its area or perimeter, which it uses to initialize the current maximum value. Next, it checks the conditions of Lemma 2.37 and computes the values of area/perimeter in these degenerate cases (see Lemma 2.37). Subsequently, it performs STEP 2 of Algorithm 6 and proceeds to STEP 3, except that in each small angular interval for which at least one corner of the rectangle R_θ lies in P, it works incrementally maximizing the area or perimeter as in the algorithm in Section 2.3.2, which takes $O(1)$ time; the area (resp. perimeter) depends linearly on $\tan\beta$, $\cot\beta$, and $\sin\beta\cos\beta$ (resp. linearly on $(1 \pm \cos\beta)/\sin\beta$, $(1 \pm \sin\beta)/\cos\beta$, and $(\sin\beta + \cos\beta)$), see Section 2.5. It is important to note that for each angle interval $[\gamma, \gamma')$ with $\gamma \neq 0$ and $\gamma' \neq \frac{\pi}{2}$ for which the $\{0°, 90°\}$-$\mathrm{Kernel}_\theta(P)$ is non-empty such that the kernel is empty for $\theta = \gamma - \varepsilon$ for a small enough ε, the kernel for $\theta = \gamma$ is degenerate, i.e., it is a point or a line segment, so that its area and perimeter can be computed in $O(1)$ time.

The above discussion and the fact that the algorithm for the $\{0°, 90°\}$-$\mathrm{Kernel}_\theta(P)$ is very similar to that for the $\{0°\}$-$\mathrm{Kernel}_\theta(P)$ lead to the following result.

Theorem 2.40. *For a simple orthogonal polygon P, computing the $\{0°, 90°\}$-$\mathrm{Kernel}_\theta(P)$ as well as finding an angle θ such that its area or perimeter is maximized or minimized can be done in $O(n)$ time and space.*

2.4.2.3 Generalization to k orientations

For a set $\mathcal{O}$ with k orientations $\alpha_1, \ldots, \alpha_k$, computing the intervals of the angle θ such that $\{\alpha_1, \ldots, \alpha_k\}$-$\mathrm{Kernel}_\theta(P) \neq \emptyset$ or an angle θ such that the area or the perimeter of this kernel is reduced to computing and maintaining the intersection of P with k different strips. As mentioned in Section 2.3.1.3, Lemma 2.16 appropriately extends and the incremental construction of the kernel involves work on $O(k)$ triangles. As a result, Theorems 2.39 and 2.40 extend to the following theorem that leads to the results in the third row of Table 2.2.

Theorem 2.41. *Given a simple orthogonal polygon P with n vertices, the intervals of θ such that $\{\alpha_1, \ldots, \alpha_k\}$-$\mathrm{Kernel}_\theta(P) \neq \emptyset$ or an angle such that the area or perimeter of $\{\alpha_1, \ldots, \alpha_k\}$-$\mathrm{Kernel}_\theta(P)$ are maximum/minimum can be computed in $O(kn)$ time and space.*

2.5 Appendix to the chapter: Trigonometric formulas

2.5.1 Trigonometric formulas for the area of the $\{0°\}$-$\mathrm{Kernel}_\theta(P)$ and the $\{0°, 90°\}$-$\mathrm{Kernel}_\theta(P)$

We consider angle $\beta \in [\theta_i, \theta_{i+1}] \subseteq (0, \frac{\pi}{2})$ and triangles with two edges on lines forming angles β, θ_i with the positive x-axis. For the third edge, we distinguish two cases: it is horizontal or it is on a line forming angle α with the positive x-axis (see Figure 2.15).

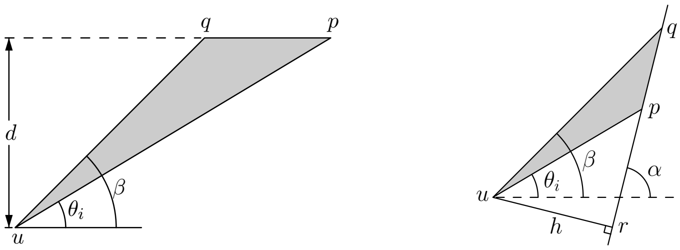


Figure 2.15: For the formulas of the area and perimeter of the $\{0°\}$-$\mathrm{Kernel}_\theta(P)$.

From Figure 2.15 (left), the area of a triangle T with edges at angles 0 (horizontal edge), θ_i, and β ($0 < \theta_i \leq \beta \leq \theta_{i+1} < \frac{\pi}{2}$) is equal to

$$A_T = \frac{1}{2}\, d\, |\overline{pq}| = \frac{1}{2}\, d\, (d \cot\theta_i - d \cot\beta) = \frac{1}{2}\, d^2\, (\cot\theta_i - \frac{\cos\beta}{\sin\beta}). \quad (2.6)$$

Let us now consider a triangle T with edges at angles θ_i, β, and α ($0 < \theta_i \leq \beta < \theta_{i+1} \leq \frac{\pi}{2}$) (see Figure 2.15, right). Then, $\widehat{upr} = \alpha - \phi$ and $\widehat{uqr} = \alpha - \theta_i$ which imply that $|\overline{pq}| = |\overline{rq}| - |\overline{rp}| = h \tan(\frac{\pi}{2} - \alpha + \beta) - h \tan(\frac{\pi}{2} - \alpha + \theta_i) = h \cot(\alpha - \beta) - h \cot(\alpha - \theta_i)$ where $h = |\overline{ur}|$ is the (perpendicular) distance of u from the line through p, q. Then, the area of the triangle is equal to

$$\begin{aligned} A_T &= \tfrac{1}{2}h\,|\overline{pq}| \\ &= \tfrac{1}{2}h^2\,(\cot(\alpha-\beta)-\cot(\alpha-\theta_i)) \\ &= \tfrac{1}{2}h^2\left(\tfrac{1+\cot\alpha\cot\beta}{\cot\beta-\cot\alpha}-\cot(\alpha-\theta_i)\right) \\ &= \tfrac{1}{2}h^2\left(\tfrac{\sin\beta+\cot\alpha\cos\beta}{\cos\beta-\cot\alpha\sin\beta}-\cot(\alpha-\theta_i)\right). \end{aligned} \tag{2.7}$$

Expression of the area of the $\{0°\}$**-Kernel**$_\theta(P)$**.** By using Equation 2.1 and Equations 2.6 and 2.7 and since θ_i is fixed, the area $A(\beta)$ of the $\{0°\}$-Kernel$_\theta(P)$ in terms of $\beta \in [\theta_i, \theta_{i+1})$ is

$$\begin{aligned} A(\beta) &= A(\theta_i) + (A_1(\beta) + A_2(\beta) - B_1(\beta) - B_2(\beta)) \\ &= A(\theta_i) + \sum_{i=1}^{4}\left(\frac{C_i\sin\beta + D_i\cos\beta}{E_i\sin\beta + F_i\cos\beta} + G_i\right), \end{aligned}$$

where $A(\theta_i)$ is the known value of the current area, and C_i, D_i, E_i, F_i, and G_i are all constants for every $i = 1, \ldots, 4$.

Then, by setting the derivative equal to zero, we get

$$A'(\beta) = \sum_{i=1}^{4} \frac{C_iF_i - D_iE_i}{(E_i\sin\beta + F_i\cos\beta)^2} = 0,$$

implying that

$$\sum_{i=1}^{4}\left[(C_iF_i - D_iE_i)\prod_{\substack{j=1\\ j\neq i}}^{4}(E_j\sin\beta + F_j\cos\beta)^2\right] = 0.$$

Expanding the product, we find three types of terms depending on $\sin^2\beta$, $\cos^2\beta$, and $\sin\beta\cos\beta$. Now using the trigonometric transformations

$$\sin^2\beta = \frac{\tan^2\beta}{1+\tan^2\beta}, \quad \cos^2\beta = \frac{1}{1+\tan^2\beta}, \quad \sin\beta\cos\beta = \frac{\tan\beta}{1+\tan^2\beta},$$

and making the change $\tan\beta = t$ we get a rational function in t. Then, the derivative function for the area is now a function on the variable t, $A'(t)$, and it is a rational function having as numerator a polynomial in t of degree 6 and as denominator a polynomial of degree 12. So we can compute the real solutions of a polynomial equation in t of degree 6.

Orthogonal polygons: For the case of the $\{0°\}$-Kernel$_\theta(P)$ for orthogonal polygons P, the triangles $A_i(\beta)$ have a horizontal or a vertical base. Since for $\alpha = \frac{\pi}{2}$, Equation 2.7 becomes

$$A_T = \frac{1}{2}h^2\left(\frac{\sin\beta}{\cos\beta} - \cot(0-\theta_i)\right) = \frac{1}{2}h^2\left(\tan\beta - \tan\theta_i\right), \tag{2.8}$$

Equations 2.6 and 2.8 imply that in this case we have

$$A(\beta) = A(\theta_i) + C\,\tan\beta + D\,\cot\beta + G$$

for appropriate constants C, D, G.

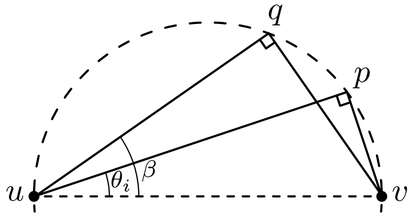


Figure 2.16: For the formulas of the area and perimeter of the $\{0°, 90°\}$-Kernel$_\theta(P)$.

Expression of the area of the $\{0°, 90°\}$**-Kernel**$_\theta(P)$**.** Consider the case in which the corner of the floating rectangle $R_{\theta)}$ lies in P (then it is a vertex of the kernel) and does so for all the angles $\beta \in [\theta_i, \theta_{i+1})$. Then, the corner moves along a circular arc with diameter the distance of the reflex minima/maxima that define the corner; see Figure 2.16. Then, the differential in the area is

$$\begin{aligned}\Delta A_T &= A_T(u\,v\,q) - A_T(u\,v\,p) = \frac{1}{2}\,|\overline{uq}|\,|\overline{vq}| - \frac{1}{2}\,|\overline{up}|\,|\overline{vp}| \\ &= \frac{1}{2}\,(|\overline{uv}|\,\cos\beta)\,(|\overline{uv}|\,\sin\beta) - \frac{1}{2}\,(|\overline{uv}|\,\cos\theta_i)\,(|\overline{uv}|\,\sin\theta_i) \\ &= \frac{1}{2}\,|\overline{uv}|^2\,(\sin\beta\,\cos\beta - \sin\theta_i\,\cos\theta_i)\,.\end{aligned}$$

Thus, in this case, for simple polygons, the differential in the area involves at most 4 terms, each being either

$$\frac{C_i \sin\beta + D_i \cos\beta}{E_i \sin\beta + F_i \cos\beta} \quad \text{or} \quad K_i \sin\beta\, \cos\beta.$$

Orthogonal polygons: For the case of orthogonal polygons, similarly we have at most 4 terms, each being $C_i \tan\beta$, $D_i \cot\beta$, or $K_i \sin\beta \cos\beta$ and thus we have we have

$$A(\beta) = A(\theta_i) + C\, \tan\beta + D\, \cot\beta + K\, \sin\beta\, \cos\beta + G$$

for appropriate constants C, D, K, G.

2.5.2 Trigonometric formulas for the perimeter of the $\{0°\}$-Kernel$_\theta(P)$ and the $\{0°, 90°\}$-Kernel$_\theta(P)$

As in the previous section, let us first consider the case of Figure 2.15, left. We want to compute $\Delta\Pi_T^+$ ($\Delta\Pi_T^-$ which is the difference of the length of the edge at angle β minus the length of the edge at angle θ_i plus (minus resp.) the length of the side at angle 0. Thus:

$$\begin{aligned}
\Delta\Pi_T^{\pm} &= |\overline{uq}| - |\overline{up}| \pm |\overline{pq}| \\
&= \tfrac{d}{\sin\beta} - \tfrac{d}{\sin\theta_i} \pm (d\, \cot\theta_i - d\, \cot\beta) \\
&= d\left(\tfrac{1}{\sin\beta} - \tfrac{1}{\sin\theta_i} \pm \tfrac{\cos\theta_i}{\sin\theta_i} \mp \tfrac{\cos\beta}{\sin\beta}\right) \\
&= d\left(\tfrac{1\mp\cos\beta}{\sin\beta} - \tfrac{1\mp\cos\theta_i}{\sin\theta_i}\right).
\end{aligned} \tag{2.9}$$

Next, let us consider a triangle T with edges at angles θ_i, β, and α ($0 < \theta_i \leq \beta < \theta_{i+1} \leq \frac{\pi}{2}$) (see Figure 2.15, right). Recall that $\widehat{upr} = \alpha - \phi$ and $\widehat{uqr} = \alpha - \theta_i$ which imply that

$$|\overline{up}| = \frac{h}{\cos(\frac{\pi}{2} - \alpha + \theta_i)} = \frac{h}{\sin(\alpha - \theta_i)}, \quad |\overline{uq}| = \frac{h}{\cos(\frac{\pi}{2} - \alpha + \beta} = \frac{h}{\sin(\alpha - \beta},$$

Then, since $|\overline{pq}| = h\ (\cot(\alpha-\beta) - \cot(\alpha-\theta_i))$, the differential $\Delta\Pi$ in the perimeter is equal to:

$$\begin{aligned} \Delta\Pi_T^{\pm} &= |\overline{uq}| - |\overline{up}| \pm |\overline{pq}| \\ &= \tfrac{h}{\sin(\alpha-\beta)} - \tfrac{h}{\sin(\alpha-\theta_i)} \pm h\ (\cot(\alpha-\beta) - \cot(\alpha-\theta_i)) \\ &= h\left(\tfrac{1\pm\cos(\alpha-\beta)}{\sin(\alpha-\beta)} - \tfrac{1\pm\cos(\alpha-\theta_i)}{\sin(\alpha-\theta_i)}\right) \\ &= h\left(\tfrac{1\pm\cos\alpha\cos\beta\pm\sin\alpha\sin\beta}{\sin\alpha\cos\beta-\cos\alpha\sin\beta)} - \tfrac{1\pm\cos(\alpha-\theta_i)}{\sin(\alpha-\theta_i)}\right). \end{aligned} \tag{2.10}$$

Expression of the perimeter of the $\{0°\}$-Kernel$_\theta(P)$. In Figure 2.5, the green (red, resp.) triangles which result in an increase (a decrease resp.) in the perimeter contribute a $\Delta\Pi_T^+$ ($\Delta\Pi_T^-$ resp.) term, and thus we use both the differentials $\Delta\Pi_T^{\pm}$. So, from Equations 2.9 and 2.10, for the perimeter $\Pi(\beta)$ of the $\{0°\}$-Kernel$_\theta(P)$ as a function of $\beta \in [\theta_i, \theta_{i+1})$ we can write

$$\Pi(\beta) = \Pi(\theta_i) + \sum_{i=1}^{4}\left(\frac{C_i\sin\beta + D_i\cos\beta + H_i}{E_i\sin\beta + F_i\cos\beta} + G_i\right),$$

where $\Pi(\theta_i)$ is the known value of the current perimeter, and C_i, D_i, E_i, F_i, G_i, and H_i are all constants for every $i = 1, \ldots, 4$.

Orthogonal polygons: For the case of the $\{0°\}$-Kernel$_\theta(P)$ for an orthogonal polygons P, the triangles $A_i(\beta)$ have a horizontal or a vertical base. Then from Equation 2.10 for $\alpha = \frac{\pi}{2}$, we have

$$\begin{aligned} \Delta\Pi_T^{\pm} &= h\left(\tfrac{1\pm\sin\beta}{\cos\beta} - \tfrac{1\pm\cos(\frac{\pi}{2}-\theta_i)}{\sin(\frac{\pi}{2}-\theta_i)}\right) \\ &= h\left(\tfrac{1\pm\sin\beta}{\cos\beta} - \tfrac{1\pm\sin\theta_i}{\cos\theta_i}\right) \end{aligned} \tag{2.11}$$

and Equations 2.9 and 2.11 imply that in this case the perimeter $\Pi(\beta)$ of the $\{0°\}$-Kernel$_\theta(P)$ in terms of $\beta \in [\theta_i, \theta_{i+1})$ is equal to

$$\Pi(\beta) = \Pi(\theta_i) + C\,\frac{1+\cos\beta}{\sin\beta} + D\,\frac{1-\cos\beta}{\sin\beta} + E\,\frac{1+\sin\beta}{\cos\beta} + F\,\frac{1-\sin\beta}{\cos\beta} + G$$

for appropriate constants C, D, E, F, G.

Expression of the perimeter of the $\{0°, 90°\}$-Kernel$_\theta(P)$. In this case, we may also have corners of the kernel moving along a circular arc as shown in Figure 2.16. From this figure, we observe that the differential in the perimeter is

$$\begin{aligned} \Delta\Pi_M &= (|\overline{uq}| + |\overline{vq}|) - (|\overline{up}| + |\overline{vp}|) \\ &= (|\overline{uv}|\,\cos\beta + |\overline{uv}|\,\sin\beta) - (|\overline{uv}|\,\cos\theta_i + |\overline{uv}|\,\sin\theta_i) \\ &= |\overline{uv}|\,(\sin\beta + \cos\beta) - |\overline{uv}|\,(\sin\theta_i + \cos\theta_i). \end{aligned}$$

Thus, for simple polygons, the differential in the perimeter involves at most 4 terms, each being

$$\frac{C_i \sin\beta + D_i \cos\beta + H_i}{E_i \sin\beta + F_i \cos\beta} \qquad \text{or} \quad K_i\,(\sin\beta + \cos\beta).$$

Orthogonal polygons: For the case of orthogonal polygons, similarly we have at most 4 terms, each being $C_i\,(1 \pm \cos\beta)/\sin\beta$, $D_i\,(1 \pm \sin\beta)/\cos\beta$, or $K_i\,(\sin\beta + \cos\beta)$ and thus

$$\begin{aligned} \Pi(\beta) &= \Pi(\theta_i) + C\,\tfrac{1+\cos\beta}{\sin\beta} + D\,\tfrac{1-\cos\beta}{\sin\beta} + E\,\tfrac{1+\sin\beta}{\cos\beta} \\ &\quad + F\,\tfrac{1-\sin\beta}{\cos\beta} + K\,(\sin\beta + \cos\beta) + G \end{aligned}$$

for appropriate constants C, D, E, F, G, K.

The above expressions of the perimeter $\Pi(\beta)$ of the $\{0°\}$-Kernel$_\theta(P)$ and the $\{0°, 90°\}$-Kernel$_\theta(P)$ in terms of the angle β can be maximized as we showed for the area of the $\{0°\}$-Kernel$_\theta(P)$ in Section 2.5 by computing the real solutions of a polynomial of constant degree.

Chapter 3

On circles enclosing many points

The following article was published in 2021 in Discrete Mathematics Volume 344, Issue 10, 112541. It was written in collaboration with Mercè Claverol and Clemens Huemer. The front page of this article can be found in Appendix B.

3.1 Introduction

Neumann-Lara and Urrutia [92] posed the following problem: Prove that every set S of n points in the plane contains two points p and q such that any circle which passes through p and q encloses “many" points of S. The question is to quantify this number of enclosed points which can always be guaranteed. We only consider point sets without three collinear points and without four cocircular points. We say that such a point set is in *general position*. Almost all the results on this question date back to the late 1980’s. The first bound $\lceil \frac{n-2}{60} \rceil$, by Neumann-Lara and Urrutia [92], was improved in a series of papers [11, 49, 50]. The best bound was obtained by Edelsbrunner et al. [32] who proved that any set of n points in the plane contains two points such that any circle through those two points encloses at least $n(\frac{1}{2} - \frac{1}{\sqrt{12}}) + O(1) \approx \frac{n}{4.7}$ points. Their proof makes use of properties of higher order Voronoi diagrams. The order-k Voronoi diagram of a point set S is a subdivision of the plane into convex regions, such that a region consists of those points of the plane which have the same k nearest points of S [10]. Twenty years later, Ramos and Viaña [110] made progress on that problem and proved a stronger statement: There is always a pair of points

such that any circle through them has, both inside and outside, at least $n(\frac{1}{2} - \frac{1}{\sqrt{12}}) + O(1) \approx \frac{n}{4.7}$ points. To prove their result, they transformed the problem from circles in the plane to planes in $\mathbb{R}^3$ and used results on the number of j-facets of point sets in $\mathbb{R}^3$.
The known upper bound on the number of enclosed points is $\lceil \frac{n}{4} \rceil - 1$, due to a construction by Hayward et al. [50]. Urrutia [134] conjectured that $\frac{n}{4} \pm c$, for some constant c, is the tight bound. For sets of n points in convex position, a tight bound of $\lceil \frac{n}{3} \rceil - 1$ is known [50].
Our contributions to this problem are the following ones:
• In Section 3.2 we present a modified and shorter version of the proof of Edelsbrunner et al. [32] which also leads to the result of Ramos and Viaña [110]. As in [32], the proof makes use of properties of higher order Voronoi diagrams.
• The proposed modification gives rise to the following result, shown in Section 3.3. Every set of n points in the plane contains two points such that any circle through those two points encloses at most $\lfloor \frac{2n-1}{3} \rfloor$ points of S.
• In Section 3.4 we show that sets S of n red points and $m = \lfloor cn \rfloor$, for $c \in (0, 1]$, blue points contain a red point and a blue point such that any circle passing through them encloses at least $\frac{n+m-\sqrt{n^2+m^2}}{2} - o(n+m)$ points of S. For $n = m$ this gives the bound $n(1 - \frac{1}{\sqrt{2}}) - o(n) \approx \frac{n}{3.4}$. The colored version of the problem was studied by Prodromou [107] for any dimension d and $\lfloor \frac{d+3}{2} \rfloor$ colors. The particular case $d = 2$ in [107], Theorem 1.1, gives a lower bound of $\frac{n+m}{36}$ on the number of enclosed points. Our result improves this bound. We also present an upper bound construction with n red and n blue points in convex position such that for every pair of points, one red and the other blue, there is a circle through them which encloses at most $\lfloor \frac{n}{2} \rfloor$ points of S.
• In Section 3.5 we study how many circles passing through two given points p and q (and a third point of S) enclose the same number of points of S. This is equivalent to study how many edges of the order-k Voronoi diagram of S can lie on the same line. Apart from [65], where it is proved that the order-k Voronoi diagram contains at most $\min(k, n-k)$ collinear edges, not much seems to be known on this question. We present some constructions with many collinear edges in higher order Voronoi diagrams. We believe that this line of research can also lead to an improvement for the problem posed by Neumann-Lara and Urrutia.

As for related works, in [5] the problem of Neumann-Lara and Urrutia is studied when instead of allowing all possible circles through two points of the given point set S, only circles which have two points of S as diametral endpoints are considered. Generalizations of the problem to higher dimensions are studied in [122].

3.2 An adaption of the proof by Edelsbrunner, Hasan, Seidel and Shen

Let S be a set of n points in general position. For each pair of points p, q of S, let b_{pq} be the *perpendicular bisector* of the segment $\overline{pq}$. That is, b_{pq} is the line equidistant to p and q. Let B be the set of all $\binom{n}{2}$ perpendicular bisectors defined by the points in S. For each pair of points p, q of S, define C_{pq} as the set of circles passing through them. The center of each circle in C_{pq} is on b_{pq}. Any line b_{pq} in B is cut into $n-1$ open segments (two of them are unbounded) which are delimited by the center points of the circles passing through p, q, and one of the $n-2$ remaining points of S. Any circle in C_{pq} with center on one of these segments encloses the same subset of points of S; its cardinality is the *weight* of the segment. Figure 3.1 shows an example with four points p, q, r and s, where the bisector b_{pq} is divided into three segments. The segments are separated by the centers of the circles through p, q, r and through p, q, s, respectively (denoted as pqr and pqs in the figure). Notice that any circle in C_{pq} with center on one of these segments encloses the same subset of points (weight of the segment).

It is well known that if the weight of such a segment is $k-1$, then the segment is an edge of the order-k Voronoi diagram; see, e.g., [71]. The order-3 and order-4 Voronoi diagrams of a set of 8 points are shown in Figure 3.2. The proof idea of Edelsbrunner et al. [32] is as follows. They show that the sum of the numbers of edges of the order-k Voronoi diagrams, for k from 1 up to approximately $(1/2 - 1/\sqrt{12})n$, is less than $\binom{n}{2}$. This implies that one of the lines in B, say b_{pq}, does not have any edge of any order-k Voronoi diagram, for $k < n(1/2 - 1/\sqrt{12}) + O(1)$, among its $n-1$ segments. Consequently, for the two points p and q, that define b_{pq}, any circle through them encloses at least $n(1/2 - 1/\sqrt{12}) + O(1)$ points of S. We present a similar proof.

Observation 3.1. When moving the center of a circle in C_{pq} along b_{pq} from one segment to the consecutive one, the number of points contained in the two corresponding circles differs by ±1. Equivalently: The weights of two

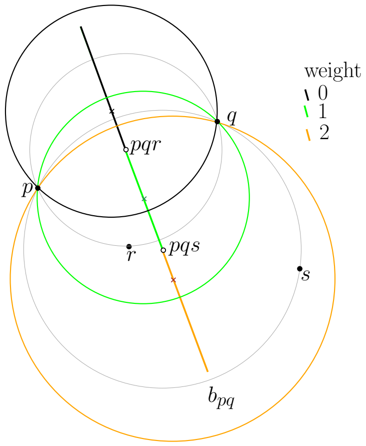


Figure 3.1: Weights of consecutive segments along the perpendicular bisector b_{pq} of points p and q from the set $S = \{p, q, r, s\}$.

consecutive segments on b_{pq} differ by ±1.

For $0 \leq j \leq \frac{n-2}{2}$, a segment $\overline{pq}$ connecting two points p, q of S is a *j-edge* of S, if the line through p and q divides the plane into two open half-planes, such that one of them contains exactly j points of S. The number of j-edges of S is denoted by e_j.

Observation 3.2. Let $\overline{pq}$ be a j-edge of S, and let b_{pq} be the perpendicular bisector of $\overline{pq}$. Then the two unbounded segments of b_{pq} have weights j and $n-j-2$, respectively. See Figure 3.2.

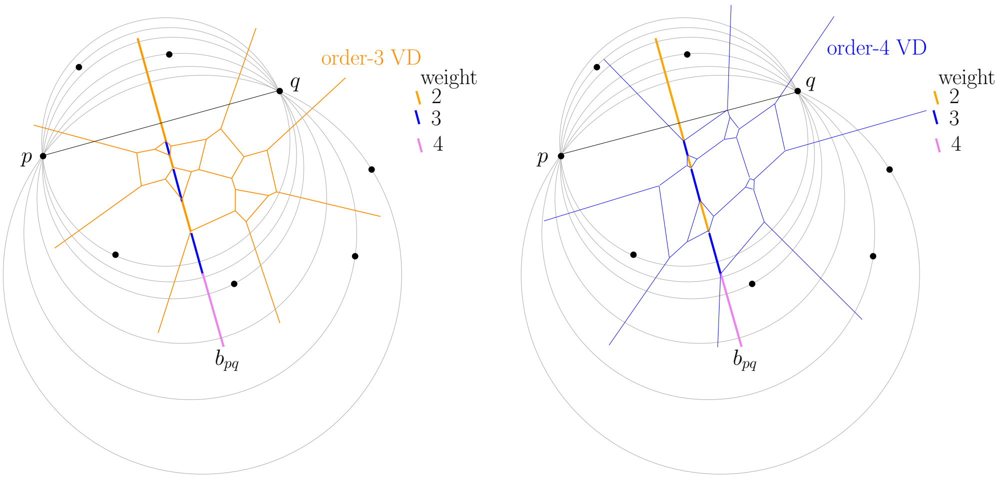
As a consequence, by an intermediate value argument:

Observation 3.3. Let $\overline{pq}$ be a j-edge of S, and let b_{pq} be the perpendicular bisector of $\overline{pq}$. For every k with $j \leq k \leq n-j-2$, the line b_{pq} contains at least one segment of weight k. See Figure 3.2.

Let c_k be the number of circles passing through three points of S that enclose exactly k points of S. Differing from the proof of [32], we will use the following property, obtained by Lee [71]; later proofs were given in [9, 23, 72]:

$$c_k + c_{n-k-3} = 2(k+1)(n-k-2). \tag{3.1}$$

A direct correspondence between the numbers c_k and the number of faces of higher order Voronoi diagrams is given for instance in [72]. The numbers c_{k-1}



(a) (b)

Figure 3.2: A 2-edge $\overline{pq}$ of a set S of eight points, and the circles through p, q and a third point of S. The two unbounded segments of the bisector b_{pq} have weight 2 and 4. Also the relation between segments of weight $k-1$ on the perpendicular bisector of two points and the edges of the order-k Voronoi diagram is illustrated: (a) The segments of weight 2 are edges of the order-3 Voronoi diagram; (b) The segments of weight 3 are edges of the order-4 Voronoi diagram.

and c_{k-2} also give the number of vertices of the order-k Voronoi diagram, denoted as v_k, via the following relation [72]

$$v_k = c_{k-1} + c_{k-2}. \tag{3.2}$$

More precisely, each vertex of the order-k Voronoi diagram of S is the center of a circle through three points of S which encloses $k-1$ or $k-2$ points of S. The vertices of this Voronoi diagram have degree three (because no four points of S are cocircular), and its edges are the segments of weight $k-1$ (among all the bisectors in B).

Observation 3.4. A circle passing through points $a, b, c \in S$ corresponds to three segments, one on b_{ab}, one on b_{ac} and one on b_{bc}. In other words, for each vertex of the order-k Voronoi of S, we count its three incident edges. Each bounded segment is counted for two such circles, and each unbounded segment is counted for one such circle.

Let s_k denote the number of segments of weight k, among all $\binom{n}{2}$ lines in B.

Observation 3.5. The number of edges of the order-$(k+1)$ Voronoi diagram of S equals s_k. The number of unbounded edges (of weight k) of the order-$(k+1)$ Voronoi diagram of S equals the number of k-edges of S, e_k. The number of unbounded edges of the order-$(n-k-1)$ Voronoi diagram of S (which is the number of unbounded segments of weight $n-k-2$ among all lines in B) is e_k, too.

Lemma 3.6. *The number of segments of weight k plus the number of segments of weight $n-k-2$, among all $\binom{n}{2}$ lines in B, equals* $6(k+1)(n-k-1)-3n+e_k$.

Proof. By Observations 3.4 and 3.5, we can double count the pairs (v,e), where v is a vertex of the order-$(k+1)$ Voronoi diagram and e is an edge incident to v, obtaining $3v_{k+1}=2s_k-e_k$. The same argument for the order-$(n-k-1)$ Voronoi diagram gives $3v_{n-k-1}=2s_{n-k-2}-e_k$. Then,

$$3v_{k+1}+3v_{n-k-1}=2s_k+2s_{n-k-2}-2e_k.$$

Using Equation (3.2), this further equals

$$3(c_k+c_{k-1})+3(c_{n-k-2}+c_{n-k-3})=2s_k+2s_{n-k-2}-2e_k.$$

And using Equation (3.1) for k and $k-1$ we get

$$6(k+1)(n-k-2)+6k(n-k-1)=2s_k+2s_{n-k-2}-2e_k.$$

The result follows. □

We are now ready to give another proof of the theorem by Ramos and Viaña [110], when neglecting sublinear terms.

Theorem 3.7 ([110])**.** *Every set S of n points in general position in the plane contains two points such that each circle passing through them encloses at least k and at most $n-k-2$ points of S, for $k=\left(\frac{1}{2}-\frac{1}{\sqrt{12}}\right)n-o(n)$.*

Proof. We first state the proof idea: Using Lemma 3.6, we show that at least one of the lines in B, say b_{pq}, neither contains a segment of weight $k-1$ nor one of weight $n-k-1$, for some fixed value $k<\frac{n-4}{2}$ to be determined. Then, each of the $n-1$ segments of b_{pq} has weight between k and $n-k-2$. Therefore, any circle passing through points p and q, which define b_{pq}, encloses between k and $n-k-2$ points of S. It turns out that $k \approx \left(\frac{1}{2} - \frac{1}{\sqrt{12}}\right) n$ is the best choice for k in the proof.

Claim: There exist two points $p, q \in S$ such that the perpendicular bisector b_{pq} of $\overline{pq}$ neither contains a segment of weight $k-1$ nor one of weight $n-k-1$, for $k \leq \left(\frac{1}{2} - \frac{1}{\sqrt{12}}\right) n - o(n)$.

Assume towards a contradiction, that every line in B contains a segment of weight $k-1$ or of weight $n-k-1$. Partition the lines in B into two classes: (1) those whose defining points $p, q \in S$ form a j-edge for $j \leq k-1$, and (2) those whose defining points $p, q \in S$ form a j-edge for $j \geq k$.

We show that each line in B contains at least two segments with weight in the set $\{k, n-k-2\}$. By Observation 3.3, each line of type (1) contains at least one segment of weight k and at least one of weight $n-k-2$. For any line b_{pq} of type (2), its unbounded segments have weights $j \geq k$ and $n-j-2 \leq n-k-2$, respectively. By assumption, b_{pq} also contains a segment of weight $k-1$ or of weight $n-k-1$. If b_{pq} contains a segment of weight $k-1$, then b_{pq} contains two segments of weight k; indeed, when traversing b_{pq}, we go from a segment of weight $j \geq k$ via one of weight $k-1$ to one of weight $n-j-2 \geq k+1$; since the changes of the weights of consecutive segments are ±1, we encounter a subsequence of weights $k, k-1, k, k+1$ among the segments of b_{pq}. In the same way, if b_{pq} contains a segment of weight $n-k-1$, we encounter a subsequence of weights $n-k-3, n-k-2, n-k-1, n-k-2$ when traversing b_{pq}. Hence, in this case b_{pq} contains two segments of weight $n-k-2$. We conclude that each line of type (1) and of type (2) in B contains at least two segments with weight in the set $\{k, n-k-2\}$.

By Lemma 3.6, the number of segments of weight k or of weight $n-k-2$ among all lines in B is

$$6(k+1)(n-k-1) - 3n + e_k.$$

The number of k-edges e_k is at most $O(n\sqrt[3]{k+1})$ [31]. We get a contradiction if

$$6(k+1)(n-k-1) - 3n + O\left(n\sqrt[3]{k+1}\right) < 2\binom{n}{2},$$

because then there would not be enough segments to cover all the lines in

B with two segments. The largest value of k which gives a contradiction is $k = \left(\frac{1}{2} - \frac{1}{\sqrt{12}}\right) n - o(n)$. This proves the claim.

Therefore, there exist two points $p, q \in S$ such that b_{pq} contains no segment of weight $k-1$ and no segment of weight $n-k-1$. It follows from Observations 3.1, 3.2, and 3.3, that this line b_{pq} is of type (2), and that b_{pq} cannot contain a segment of weight i for $i < k-1$ and for $i > n-k-1$ either. □

3.3 Circles enclosing not too many points

The following lemma is implied by known results on higher order Voronoi diagrams and bounds on $(\leq k)$-*sets*. A k-*set* of a point set S is a subset of k points of S which can be strictly separated from the remaining points of S by a straight line. And a $(\leq k)$-set of a point set S is a subset of at most k points of S which can be strictly separated from the remaining points of S by a straight line.

Lemma 3.8. *Let S be a set of n points and let $k < \frac{n-3}{2}$. Then $c_k \geq (k+1)(n-k-2)$ and $c_{n-k-3} \leq (k+1)(n-k-2)$.*

Proof. Denote with f_k^∞ the number of unbounded regions in the order-k Voronoi diagram of S. Also define $f_0^\infty = 0$ and $c_{-1} = 0$. These numbers f_k^∞ are related to circles enclosing points via the following relation, when $k \geq 1$, see [72].

$$\sum_{i=1}^{k} f_{i-1}^\infty = (k-1)(2n-k) - c_{k-2}.$$

On the other hand, it is well known that each unbounded region counted in f_k^∞ corresponds to a k-set. The number of $(\leq k)$-sets of S is known to be at most $k \cdot n$ for $k < \frac{n}{2}$ [8]. Therefore, $\sum_{i=1}^{k} f_{i-1}^\infty \leq (k-1) \cdot n$ and

$$c_{k-2} \geq (k-1)(2n-k) - (k-1)n.$$

Then, $c_k \geq (k+1)(2n-k-2) - (k+1)n = (k+1)(n-k-2)$. Finally, $c_{n-k-3} \leq (k+1)(n-k-2)$ follows from Property (3.1). □

Lemma 3.9. *Let S be a set of n points and let $k \leq \frac{n-3}{2}$. The number of segments of weight $n-k-3$ among all lines in B, is at most $\frac{3}{2}(2k+3)(n-k-2) - \frac{3}{2}(k+2) + \frac{e_{k+1}}{2}$.*

Proof. The number of segments of weight $n-k-3$, s_{n-k-3}, is the number of edges of the order-$(n-k-2)$ Voronoi diagram of S. By Observations 3.4 and 3.5, $3v_{n-k-2} = 2s_{n-k-3} - e_{k+1}$. By (3.2), $v_{n-k-2} = c_{n-k-3} + c_{n-k-4}$. By Lemma 3.8, $c_{n-k-3} \leq (k+1)(n-k-2)$ and $c_{n-k-4} \leq (k+2)(n-k-3)$. Then,

$$s_{n-k-3} = \frac{3}{2}v_{n-k-2} + \frac{e_{k+1}}{2} \leq \frac{3}{2}((k+1)(n-k-2)+(k+2)(n-k-3)) + \frac{e_{k+1}}{2}$$
$$= \frac{3}{2}(2k+3)(n-k-2) - \frac{3}{2}(k+2) + \frac{e_{k+1}}{2}.$$

□

A slight variation of the proof of Theorem 3.7, and using the bound on the number of ($\leq k$)-sets from [8] leads to the following Theorem 3.10. We show, using Lemmas 3.8 and 3.9, that there is a line in B such that all its segments have weight at most $n-k-3$, for some $k \leq \frac{n-4}{2}$. The best choice for k turns out to be $k = \lfloor \frac{n}{3} \rfloor - 2$.

Theorem 3.10. *Let S be a set of $n \geq 3$ points in general position in the plane. Then S contains two points such that every circle passing through them encloses at most $\lfloor \frac{2n-1}{3} \rfloor$ points of S.*

Proof. The statement is easily verified for $n \leq 5$. Then, assume $n \geq 6$. The proof is by contradiction: Suppose that every line in B contains a segment of weight $n-k-2$, for some fixed $k \leq \frac{n-4}{2}$; we will see that the largest value of k which gives a contradiction is $k = \lfloor \frac{n}{3} \rfloor - 2$.
Partition the lines in B into two classes: (1) those whose defining points $p, q \in S$ form a j-edge for $j \geq k+1$, and (2) those whose defining points $p, q \in S$ form a j-edge for $j \leq k$.

Claim: Every line of type (1) contains two segments of weight $n-k-3$.

By assumption, each line b_{pq} in B has a segment of weight $n-k-2$. Let b_{pq} be of type (1). By Observation 3.2, b_{pq} has an unbounded segment of weight $n-j-2 \leq n-k-3$. When traversing b_{pq} we go from an unbounded segment of weight $j \leq (n-2)/2$, via a segment of weight $n-k-3$ to a segment of weight $n-k-2$ and then to the other unbounded segment of weight at most $n-k-3$. Hence, b_{pq} contains two segments of weight

$n-k-3$. This proves the claim.

Since each line of type (2) contains a segment of weight $n-k-3$ and each line of type (1) contains two segments of weight $n-k-3$, we get

$$\frac{3}{2}(2k{+}3)(n{-}k{-}2){-}\frac{3}{2}(k{+}2){+}\frac{e_{k+1}}{2} \geq s_{n-k-3} \geq 1\sum_{j=0}^{k} e_j{+}2\left(\binom{n}{2}-\sum_{j=0}^{k} e_j\right).$$

We have $\frac{e_{k+1}}{2} < e_{k+1}$ and move this term to the other side of the inequality. Then,

$$\frac{3}{2}(2k+3)(n-k-2)-\frac{3}{2}(k+2) \geq 2\binom{n}{2}-\sum_{j=0}^{k+1} e_j.$$

The sum $\sum_{j=0}^{k+1} e_j$ is at most $(k+2)\cdot n$, because the number of $(\leq k)$-sets of S is known to be at most $k \cdot n$ [8] and the number of j-edges, e_j, equals the number of $(j+1)$-sets [8, 103]. Then,

$$\frac{3}{2}(2k+3)(n-k-2)-\frac{3}{2}(k+2) \geq 2\binom{n}{2}-(k+2)\cdot n.$$

But this only holds if $k \approx\geq \frac{n}{3}$, and gives a contradiction for $k \leq \lfloor\frac{n}{3}\rfloor - 2$. Therefore B contains a line b_{pq} that does not contain a segment of weight $n-k-2$, for $k = \lfloor\frac{n}{3}\rfloor - 2$. It follows from Observations 3.1, 3.2, and 3.3, that every segment on b_{pq} has weight at most $n-k-3 = n-\left(\lfloor\frac{n}{3}\rfloor-2\right)-3 = \lfloor\frac{2n-1}{3}\rfloor$. Then, every circle through p and q encloses at most $\lfloor\frac{2n-1}{3}\rfloor$ points of S. □

3.4 Two-colored point sets

Theorem 3.11. *Every set S of n red points and $m = \lfloor cn \rfloor$, for $c \in (0,1]$, blue points in general position in the plane contains a red point p and a blue point q such that any circle passing through them encloses at least $\frac{n+m-\sqrt{n^2+m^2}}{2} - o(n+m)$ points of S.*

Proof. The proof is very similar to the one of Theorem 3.7. The difference is that we now only consider perpendicular bisectors b_{pq} for points p and q

of different color. Let B^* be this set of nm *bichromatic bisectors*. Each b_{pq} in B^* is cut into $n+m-1$ open segments, which are delimited by the center points of the circles passing through p, q, and one of the $n+m-2$ remaining points of S. We say that the segments on $b_{pq} \in B^*$ are bichromatic. We then need a bound on the number of bichromatic segments of weight k plus the number of bichromatic segments of weight $n+m-k-2$, among all lines in B^*. The following observation is analogous to Observation 3.4.

Observation 3.12. A circle passing through points $p_1, p_2, q \in S$, with q of different color than p_1 and p_2, corresponds to two bichromatic segments, one on $b_{p_1 q}$ and one on $b_{p_2 q}$. Hence, each vertex of the order-$(k+1)$ Voronoi diagram of S that is the center of one such circle, is incident to two bichromatic segments of weight k.

Let s_k^* denote the number of bichromatic segments of weight k, among all nm lines in B^*. Then s_k^* equals the number of bichromatic segments of the order-$(k+1)$ Voronoi diagram of S. Let e_k^* denote the number of unbounded bichromatic segments of the order-$(k+1)$ Voronoi diagram of S.

Lemma 3.13. *Let S be a set of n red and m blue points. Then,*

$$s_k^* + s_{n+m-k-2}^* \leq 4(k+1)(n+m-k-1) - 2(n+m) + e_k^*.$$

Proof. By Observation 3.12, we can double count pairs (v, e), where v is a vertex of the order-$(k+1)$ Voronoi diagram of S and e is a bichromatic segment incident to v. Since each vertex v is incident to zero or two bichromatic segments, we get the inequality $2v_{k+1} \geq 2s_k^* - e_k^*$. The same argument for the order-$(n+m-k-1)$ Voronoi diagram of S gives $2v_{n+m-k-1} \geq 2s_{n+m-k-2}^* - e_k^*$. Then,

$$2v_{k+1} + 2v_{n+m-k-1} + 2e_k^* \geq 2s_k^* + 2s_{n+m-k-2}^*.$$

Using Equation (3.2), we further have

$$(c_k + c_{k-1}) + (c_{n+m-k-2} + c_{n+m-k-3}) + e_k^* \geq s_k^* + s_{n+m-k-2}^*.$$

And using Equation (3.1) for k and $k-1$ we get

$$2(k+1)(n+m-k-2) + 2k(n+m-k-1) + e_k^* \geq s_k^* + s_{n+m-k-2}^*.$$

The result follows. □

Then, following the steps of the proof of Theorem 3.7, we get that each line of type (1) or of type (2) in B^* contains at least two segments with weight in the set $\{k, n+m-k-2\}$. Recall that b_{pq} is of type (1) if $\overline{pq}$ is a j-edge for $j \leq k-1$ and it is of type (2) if $\overline{pq}$ is a j-edge for $j \geq k$. We get a contradiction if

$$4(k+1)(n+m-k-1) - 2(n+m) + e_k^* < 2nm,$$

because then there would not be enough segments to cover all the lines of type (1) and of type (2) in B^* with two segments. By [31], the number e_k^* is at most $O((n+m)\sqrt[3]{k+1})$.

Then we can set

$$k = \frac{n+m-\sqrt{n^2+m^2}}{2} - o(n+m).$$

□

Figure 3.3 shows an upper bound construction, for a set S of n red and n blue points in convex position. It is constructed as follows: Each one of the four groups of points in the figure is placed very close to a corner of a unit square, over one of its edges. Then, to prevent degenerate cases, the points are moved slightly in such a way that convexity is preserved. For every pair of points, one red and the other blue, there is a circle through them which encloses at most $\lfloor \frac{n}{2} \rfloor$ points of S.

3.5 Many segments of repeated weights

Proposition 3.14. *There exists a set S of n points in general position in the plane which satisfies: Let b_k be the number of bisectors among pairs of points of S that contribute with four edges to the order-k Voronoi diagram of S. Then $\sum_{k=1}^{2n/7} b_k \geq \frac{4n}{7} - o(n)$ and $b_k \neq 0$ for at least $\frac{n}{6} - o(n)$ values of k. Further, only a subset of $O(\log(n))$ points is needed to define these bisectors.*

Proof. The set of points S is obtained recursively in the following way. Let p, q, r be the vertices of an equilateral triangle and consider the supporting lines through them dividing the plane into seven regions, see Figure 3.4. In each of these regions there is a group of $n/7$ points such that the circle by

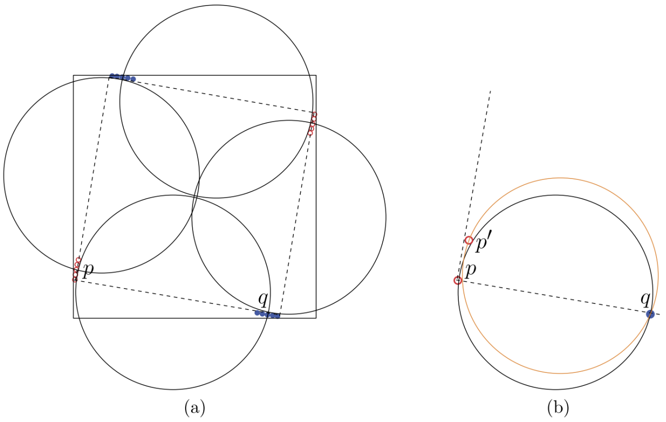


Figure 3.3: (a) Convex configuration S of n red points and n blue points, consisting of four groups of $\lfloor \frac{n}{2} \rfloor$ or $\lceil \frac{n}{2} \rceil$ points. For any choice of a blue point q and a red point p, there exists a circle passing through them enclosing only points of one color. (b) Detail of the construction: Red point p' above p is placed outside the circle with diameter $\overline{pq}$ (drawn in black) and such that p' and q are on the same side of the line through p that is perpendicular to segment $\overline{pq}$. The point p' can be arbitrary close to p.

p, q and r only contains points of the central region. We assign the points p, q and r to the central region. This is the initial configuration.

The $(n/7) - 3$ points from the interior of the triangle $\triangle pqr$ give rise to seven other groups of $(n - 3 \cdot 7)/7^2$ points, all of them arranged in the same way as the initial configuration, that is, its central region has again three points p', q' and r', which delimit another central region and so on. Note that p', q', and r' are placed very close to the center of the triangle $\triangle pqr$, on the lines from this center to p, q and r, respectively. Further, we can arrange the points symmetrically with respect to the lines through $\overline{pp'}$, $\overline{qq'}$ and $\overline{rr'}$; then we can move the points slightly to guarantee general position.

First, let's see how many weights are repeated in any of the bisectors b_{pq}, b_{pr} or b_{qr}. If we go through any of them, we obtain the list of weights $[3n/7, \overset{\downarrow}{\ldots}, n/7, \overset{\uparrow}{\ldots}\ (2n/7) - 3, \overset{\downarrow}{\ldots}, (n/7) - 3, \overset{\uparrow}{\ldots}, (4n/7) - 2]$, where consecutive values differ by one. Therefore, all values between $(2n/7) - 3$ and $n/7$ are repeated four times. This already shows that there exists a bisector which contributes with four edges to $(n/7) - 3$ different higher order Voronoi diagrams. Note that there is an interval of b_{pq} in which the cir-

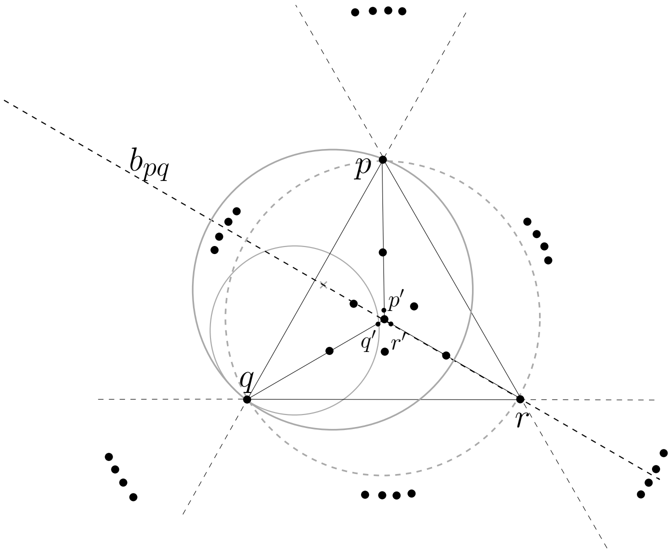


Figure 3.4: A recursive construction with many repeated weights.

cles through p and q that have their centers in this interval do not contain points outside of the central region defined by p, q and r. The same happens in the different central regions obtained recursively. Hence, considering this interval of the bisector $b_{p'q'}$ (analogously for $b_{p'r'}$, $b_{q'r'}$), the list of weights is: $[\ldots, 3((n/7)-3)/7, .\overset{\downarrow}{.}., ((n/7)-3)/7, .\overset{\uparrow}{.}. \; (2((n/7)-3)/7)-3, .\overset{\downarrow}{.}. \; , ((n/7)-3)/7-3, .\overset{\uparrow}{.}., 4((n/7)-3)/7-2, \ldots]$, where $(((n/7)-3)/7)-3$ of them are repeated four times. This recursion give a lower bound on the number of bisectors contributing with at least four edges to different higher order Voronoi diagrams. After j steps (where the first step $j = 1$ counts for b_{pq}, b_{pr}, and b_{qr}) we have

$$3\,(n/7 - 3 + (n/7-3)/7 - 3 + ((n/7-3)/7-3)/7 - 3 + \ldots) =$$

$$= 3\sum_{i=1}^{j} \left(n/7^i - 3(j+1-i)/7^{i-1}\right)$$

weights repeated four times.

Second, we consider any of the three perpendicular bisectors of the segments $\overline{pp'}$, $\overline{qq'}$, $\overline{rr'}$. By construction, these three segments lie on *halving lines*. A halving line of S is a straight line which leaves the same number of points of S on each of its sides. The list of weights on each of the three perpendicular bisectors of the segments $\overline{pp'}$, $\overline{qq'}$, $\overline{rr'}$ is $[(n/2)-2, \overset{\downarrow}{..}, 2((n/7)-3))/7, \overset{\uparrow}{..}\, 3((n/7)-3))/7, \overset{\downarrow}{..}, 2((n/7)-3))/7, \overset{\uparrow}{..}, (n/2)-2]$, where $((n/7)-3)/7$ values are repeated four times. Considering the corresponding three halving lines in each step of the recursion we obtain after j steps (where the first step $j=1$ counts for $b_{pp'}$, $b_{qq'}$, and $b_{rr'}$):

$$3\left((n/7-3)/7 + ((n/7-3)/7-3)/7 + \ldots\right) = \\ 3\textstyle\sum_{i=1}^{j}\left(n/7^{i+1} - 3(j+1-i)/7^{i}\right)$$

weights repeated four times.

Adding up, we obtain

$$\sum_{k=1}^{2n/7} b_k \geq 3\sum_{i=1}^{j}\left(n/7^{i} - 3(j+1-i)/7^{i-1}\right) + 3\sum_{i=1}^{j}\left(n/7^{i+1} - 3(j+1-i)/7^{i}\right)$$

$$= 4n/7 - (4n+14)/7^{j+1} - 12j + 2,$$

which is $4n/7 - o(n)$; note that we can take $j \in \Theta(\log(n))$ steps in the recursion. From the construction we get $b_k \neq 0$ for at least $n/7 + n/7^2 + \ldots + n/7^{\Theta(\log(n))} - o(n) = n/6 - o(n)$ values of k. □

For sets of $2n$ cocircular points, the segments on the perpendicular bisector of any $(n-1)$-edge (such a $(n-1)$-edge lies on a halving line) have weight $n-1$. Another, not so elementary, construction without four cocircular points is given in Proposition 3.15, see Figure 3.5.

Proposition 3.15. *There exists a set S of $2n$ points in general position in the plane such that every pair of points p, q of S which defines a halving line satisfies: Every circle passing through p and q encloses $n-2, n-1$ or n points of S.*

Proof. The construction is as follows. First, we place two points, p_1 and q_1; let m be their midpoint. We consider the lines $\ell_1, \ldots, \ell_n$, where ℓ_1 is the supporting line of $\overline{p_1q_1}$, ℓ_n is $b_{p_1q_1}$, and the remaining ones are obtained from ℓ_1 after successive rotations of angle $\pi/(2n)$ and center m. In the following we define the points q_i, p_i for $i = n, \ldots, 3, 2$ such that all p_i are above ℓ_1 and all q_i are below ℓ_1. Both q_i and p_i will lie on ℓ_i. Now we place the points p_n and q_n on $b_{p_1q_1}$, in such a way that p_n is close to m, and q_n is cocircular with p_1, q_1 and p_n.

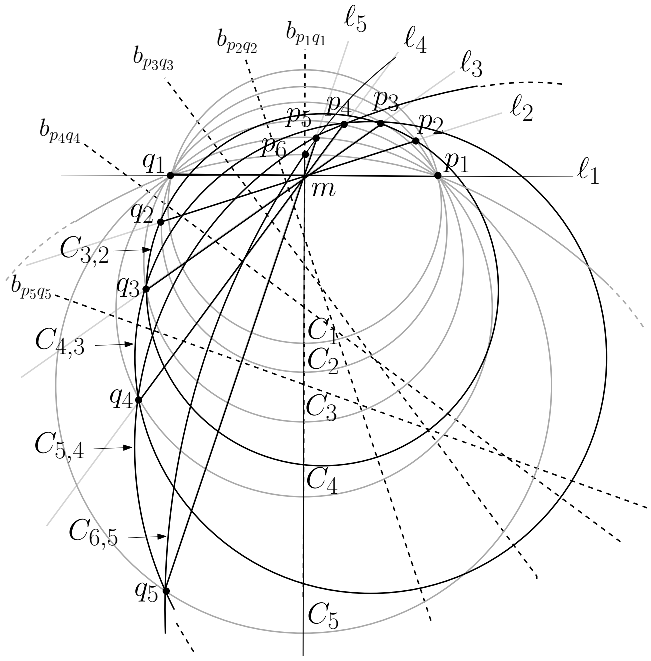


Figure 3.5: A set of $2n$ points. All $2n-1$ segments on the bisector of any $(n-1)$-edge have weight $n-2$, $n-1$, or n.

Let $2 \leq k \leq n$. From $C_{p_1q_1}$, we denote C_k the circle through p_1, q_1 and the point p_k on the bisector $b_{p_1q_1}$. Hence, q_n is on C_n. Note that these circles are well defined, once p_k is defined.

Next, we place the point q_{n-1} on line ℓ_{n-1}, in the interior of C_n. We draw a circle through the points q_n, p_n and q_{n-1}. This circle cuts ℓ_{n-1} in the point q_{n-1} and in another one, where we put p_{n-1}. We denote $C_{i+1,i}$ the circle through q_{i+1}, p_{i+1}, q_i and p_i for $i = n-1, \ldots, 3, 2$.

In a general step of this construction, we place the point q_{n-i} on the line ℓ_{n-i}, in the interior of C_{n-i+1} and outside the circle $C_{n-i+2,n-i+1}$. Then we repeat the process to obtain the point p_{n-i}. See Figure 3.5.

Note that the points q_{i+1}, p_{i+1}, q_i and p_i are cocircular, they belong to $C_{i+1,i}$, by construction. Furthermore, p_i and q_i are the intersection points of $C_{i+1,i}$ with $C_{i,i-1}$. These points p_i and q_i define in $C_{i,i-1}$ two arcs, the one containing q_{i-1} is exterior to $C_{i+1,i}$, the other one is in the interior of $C_{i+1,i}$. Therefore, if we go through any of the bisectors $b_{p_iq_i}$, the corresponding circle is reaching the pairs p_i and q_i in order. That is, if the circle passes through q_{i-1} it can not reach q_j (or p_j), where $j > i + 1$, before q_{i+1} (or p_{i+1}) is reached.

Note that we can move the points q_i slightly to achieve general position, while keeping the described properties.

It is easy to check that the line through $\overline{p_i q_i}$ is a halving line and if we go through any of the bisectors $b_{p_i q_i}$, the corresponding circle contains $n-2$, $n-1$ or n points. □

Chapter 4

Geometric and statistical techniques for projective mapping of chocolate chip cookies

In this chapter, we reproduce an article that was published in 2021 in the journal Food Quality and Preference Volume 87, 104068. This article was written in collaboration with David Orden, Encarnación Fernández-Fernández and Marino Tejedor-Romero. The front page of this article can be found in Appendix B.

4.1 Introduction

Traditionally, the sensory evaluation of foods has been carried out using generic descriptive analysis with trained panels [68] and consumer sensory analysis with a 9-point hedonic scale, with different purposes and objectives [136]. As an alternative to QDA and consumer sensory analysis with hedonic scale, a number of new sensory techniques have arisen in the last couple of decades [136, 137, 135]. These methods, sometimes called rapid sensory methods as opposed to the fact that generic descriptive analysis is quite time consuming, have proved to be useful in order to obtain accurate and reliable information from consumers.

Among these rapid methods, the present work focuses on Projective Mapping [113], later used in Napping [99]. Some of the interesting characteristics of this method are being a holistic methodology, based on consumers' perception of the global similarities and differences among a set of samples, as well as allowing its use both with trained panellists and with unexperienced consumers. Data from rapid methods are traditionally analyzed using statistical techniques, with multiple factor analysis (MFA) [99] being the most common choice for Projective Mapping. Very recently, some works have also proposed the use of geometric techniques from graph theory [97, 66]. A first contribution of the present work is to deepen in this recent line of research, introducing a substantial modification of the SensoGraph method proposed in [96] and further comparing the results and their stability between this new variant, the original SensoGraph, and MFA. Our hypothesis is that mixing geometric techniques from graph theory with common tools in statistics, like distances between samples or dendrograms, would provide further insight in sensory studies.

The number of assessors required to perform Projective Mapping differs depending upon the study, the products, and the level of expertise of the participants. Previous research has shown that the minimum number of consumers needed to obtain stable maps in Projective Mapping using MFA strongly depends on the number of samples and their degree of difference [139, 140]. Although stable configurations might be reached by just 20 assessors, larger numbers are often required [135].

Despite the simplicity of Projective Mapping makes it especially suitable to be used by unexperienced consumers, to the best of our knowledge no work seems to have carried out Projective Mapping with significantly more than 100 consumers, although other techniques like Sorting have been used with up to 389 consumers [128]. A second contribution of the present work is to perform a Projective Mapping study with a large number of consumers, in particular $n = 349$. This large number properly allows the use of resampling techniques in order to analyze the stability of both MFA and SensoGraph for a panel of unexperienced consumers. The data collected are made publicly available [95], seeking that the community can benefit from a dataset with such a large number of consumers.

As for the product under study, Projective Mapping has been pre- viously used to analyze a variety of foods with different sensory complexities, from white wines [99, 15] to cheeses [13, 90]. For the present work we have chosen to study commercial chocolate chip cookies. Cookies are one of the most common snack foods [42] due to their general acceptability, convenience and shelf-life. In particular, chocolate chip cookies are present in

Sample	Brand	Private label of supermarket	Manufacturer	Percentage of Chocolate
1	Hacendado	Mercadona	Grupo Siro	37% chocolate chips
2	Chips-Ahoy		Mondelez España Commercial, S.L.	25.6% chocolate chips
3	Carrefour	Carrefour	Aurly S.L.	37% chocolate chips
4	Grandino	Lidl	Übach-Palenberg	29% chocolate chips and 11% milk chocolate chip
5	Chips-Ahoy		Mondelez España	Commercial, S.L. 25.6% chocolate chip
6	Alteza	Lupa	Galletas Gullón S.A.	25% chocolate chips
7	American Cookies	Aldi	Banketbakkerij Merba B.V.	29% chocolate chips and 11% milk chocolate chips
8	Dia	Dia	Don Cake S.A.	26.7% chocolate chips and 10.3% milk chocolate chips
9	Ifa Eliges	Gadis	Galletas Gullón S.A.	25% chocolate chips

Table 4.1: Chocolate chip cookies samples information.

most supermarket shelves, food stalls, and service stations. Their easy acquisition and low price help them become part of the diet, especially among young people [84], which are the target of the present study. A Scopus search provided no previous works using Projective Mapping with chocolate chip cookies and a single one with commercial cookies [127]. Thus, a third contribution of the present work is the analysis of commercial chocolate chip cookies, from a multinational brand and several private labels. Our hypothesis is that unexperienced consumers can provide consistent results about this popular type of cookies.

4.2 Material and methods

4.2.1 Samples

Nine commercial chocolate chip cookies were used in this study, bought at supermarkets in Palencia (Spain). Products differed in terms of brand, being or not a private label, manufacturer, and percentage of chocolate (Table 4.1). Blind duplicates were used within the product set, with samples 2 and 5 being the same product, since the ability of positioning close blind duplicates is widely considered an indicator of the reliability of the method and the accuracy of the panel [87, 138, 116, 86].

The chocolate chip cookies were presented in plastic cups, labeled with a three-digit random code, and served in randomized order following a balanced block experimental design. Due to the significant differences in the size of the cookies and their external appearance, they were served as halves in order to minimize the possibility of the consumers recognizing them. To make sure that the assessors could re-test several times if needed, labeled plates with extra samples were at their disposal. Water was also provided to all consumers to rinse between samples.

4.2.2 Consumers

A total of three hundred forty-nine ($n = 349$) consumers participated in this study, recruited among seven educational centers and two university fairs along the academic years 2017–18 and 2018–19; more details are provided in the dataset description [95]. The participants, of which 53% were female and 47% male, ranged between 14 and 30 years old. All the participants agreed to take part in this study and no identifiable or sensitive information was collected.

4.2.3 Projective mapping

Prior to starting the tasting session, all participants were sensitized about the importance of sensory analysis by a visual presentation. An aim underlying this study was to promote the sensory analysis among young people, at different high schools and during the open days of the University of Valladolid. Subsequently, the basis of Projective Mapping was explained to the participants, using an illustration which showed cookies of different shapes (rectangular and round) and colors (cream and brown).

After the explanation of the technique, the participants received an A2 (60×40 cm) sheet of paper to allocate the samples. Samples were to be placed close to each other if, according to the assessor's own criteria, they seemed sensorially similar and vice versa, i.e., two chocolate chip cookies were to be distant from one another if they seemed different. The participants had to observe, smell, and taste the chocolate chip cookies, and then position the samples on the A2 sheet, trying to use as much of the tablecloth as possible. Once they had decided on the positioning, they were asked to write the codes on the sheet. The dataset with the x- and y-coordinates of chocolate chip cookies from the individual perceptual spaces is made publicly available [95], for the sake of facilitating replicability in research.

4.2.4 Data analysis with existing methods

4.2.4.1 Statistical analysis

The data obtained from Projective Mapping [113] was then analyzed by MFA [99], using the R language [109] and the FactoMineR package [69].

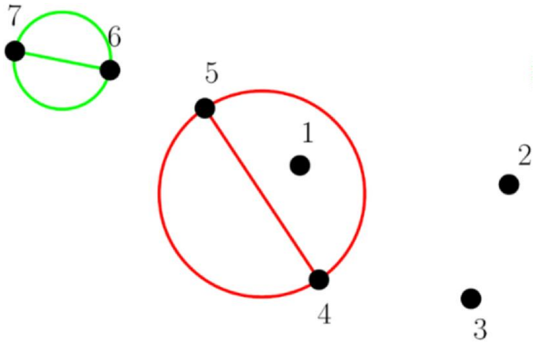


Figure 4.1: Clustering with the Gabriel graph. The connection 6–7 will be drawn, since no third sample lies inside the circle having that connection as diameter. On the contrary, the connection 4–5 will not be drawn, because sample 1 lies inside the corresponding circle.

Confidence ellipses were constructed using truncated total bootstrapping [20] with the SensoMineR package [69].

4.2.4.2 Geometric analysis using SensoGraph with a clustering method

The x- and y-coordinates from the tablecloths were imported and analyzed using a web app [96] implementation of the SensoGraph method introduced in [97]. The first step there was to perform a clustering on each tablecloth using the Gabriel graph [39], a tool from Computational Geometry. With this technique, two samples become connected if, and only if, there is no third sample contained in the circle having that potential connection as diameter. See Figure 4.1.

The aim of this clustering was to connect some pairs of samples at each tablecloth, so that a global similarity matrix could be constructed by counting, for each pair of samples, in how many tablecloths they became connected.

In other words, for each tablecloth the clustering induced a matrix with entries in the set 0,1 standing, respectively, for the corresponding pair of samples being connected or not by the clustering. Then, the global similarity matrix was just the addition of the particular similarity matrices from each tablecloth. See Figure 4.2 for an illustration.

The second step in [97] was the use of the Kamada-Kawai force-directed algorithm from graph drawing [61] to obtain a positioning of the samples (consensus graphic) based on the entries of the global similarity matrix.

Interested readers can follow the descriptions in [97] to implement the SensoGraph method in R, using the command gg in the package cccd [76] for the Gabriel graph and the command layout_with_kk of the package igraph [26] for the Kamada-Kawai graph drawing algorithm. The corresponding author can also be contacted for further details.

Several improvements were implemented with respect to the software used in [97] and the features of the above-mentioned commands in R. First, a color code from red (smallest) to green (largest) was incorporated to both the connections between samples in the consensus graphic and the global similarity matrix, to illustrate the strength of those connections following the lines of recent successful visualization tools [143]. Second, a dendrogram for the data of the global similarity matrix was obtained using hierarchical clustering [48] and the matrix was then rearrange according to the order in the dendrogram, for the sake of an easier visualization of the groups of samples.

In addition, the possibility of displaying only the most relevant connections was implemented. The strength of a particular connection was normalized to the interval between the smallest and largest strengths in the global similarity matrix, according to the following formula

$$normalized\ strength = \frac{current\ strength - smallest\ strength}{largest\ strength - smallest\ strength}.$$

Then, these normalized strengths were grouped into deciles, allowing to display only the $10k\%$ of largest normalized strengths, for k an integer from 1 to 10.

4.2.5 Data analysis with a new method

The present work introduces and tests an alternative way of obtaining the global similarity matrix. As in the original SensoGraph proposal, the global matrix will also be the sum of particular similarity matrices for each tablecloth, but the difference is that the values in these particular matrices now lie in the interval $[0, 1]$. See Figure 4.3 for an illustration.

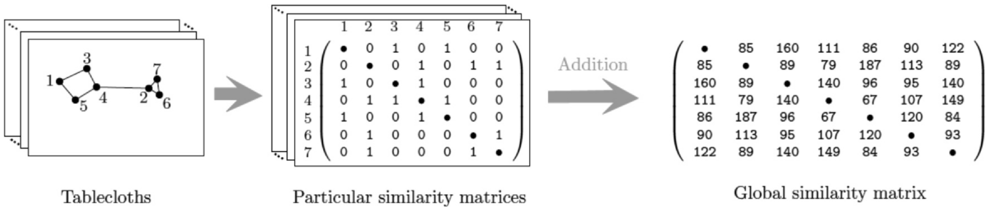


Figure 4.2: Illustration of the process for SensoGraph with Gabriel [97].

Specifically, the values of the particular similarity matrices correspond to the distances between the pairs of samples, normalized to the interval $[0, 1]$ as follows: The interval between the smallest and the largest distances obtained is linearly mapped to a similarity in the interval $[0, 1]$ in an inverse way, according to the formula

$$similarity = 1 - \frac{distance - smallest\ distance}{largest\ distance - smallest\ distance}.$$

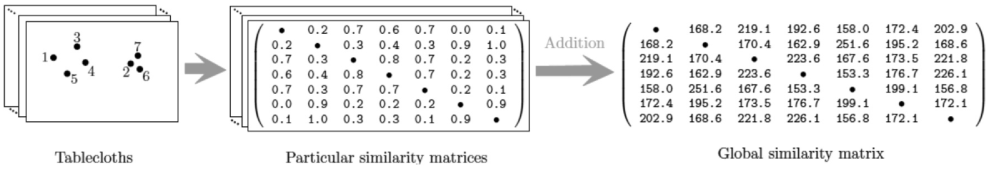
which assigns similarity 1 to the smallest distance, 0 to the largest distance, smaller similarities (between 0 and 1) to the larger distances, and vice versa.

This approach aims to capture the essence of Projective Mapping, which asks the assessors to position closer those samples perceived as more similar and vice versa, by assigning larger similarities to (hence, considering more important) those pairs of samples positioned closer, and vice versa. In order to emphasize the importance of those samples which were more clearly positioned closer or further, the linear mapping mentioned above was tuned using the following formula, which depends on a parameter $p \geq 1$.

$$tuned\ similarity = \begin{cases} 2^{p-1}(similarity)^p & when\ similarity < 1/2, \\ 1 - 2^{p-1}|similarity - 1|^p & when\ similarity \geq 1/2. \end{cases}$$

Figure 4.4 shows the effect of this function for several values of p. The value $p = 1$ gives the aforementioned linear mapping, while increasing the value of p results in the graph combing towards an S-shape, hence emphasizing the effect of extreme values corresponding to more clearly positioned samples. After checking different values of p between 1 and 3 for the function of tuned similarity, obtaining analogous results, a compromise value of $p = 2$ was chosen for the sake of emphasizing extreme values without an excessive distortion of the inputs, see again Figure 4.4.

This new method was also implemented in the web app [96], using Python as backend language on the server side, Flask as microframework,



$$\begin{pmatrix} \bullet & 0.2 & 0.7 & 0.6 & 0.7 & 0.0 & 0.1 \\ 0.2 & \bullet & 0.3 & 0.4 & 0.3 & 0.9 & 1.0 \\ 0.7 & 0.3 & \bullet & 0.8 & 0.7 & 0.2 & 0.3 \\ 0.6 & 0.4 & 0.8 & \bullet & 0.7 & 0.2 & 0.3 \\ 0.7 & 0.3 & 0.7 & 0.7 & \bullet & 0.2 & 0.1 \\ 0.0 & 0.9 & 0.2 & 0.2 & 0.2 & \bullet & 0.9 \\ 0.1 & 1.0 & 0.3 & 0.3 & 0.1 & 0.9 & \bullet \end{pmatrix}$$

$$\begin{pmatrix} \bullet & 168.2 & 219.1 & 192.6 & 158.0 & 172.4 & 202.9 \\ 168.2 & \bullet & 170.4 & 162.9 & 251.6 & 195.2 & 168.6 \\ 219.1 & 170.4 & \bullet & 223.6 & 167.6 & 173.5 & 221.8 \\ 192.6 & 162.9 & 223.6 & \bullet & 153.3 & 176.7 & 226.1 \\ 158.0 & 251.6 & 167.6 & 153.3 & \bullet & 199.1 & 156.8 \\ 172.4 & 195.2 & 173.5 & 176.7 & 199.1 & \bullet & 172.1 \\ 202.9 & 168.6 & 221.8 & 226.1 & 156.8 & 172.1 & \bullet \end{pmatrix}$$

Figure 4.3: Illustration of the process for SensoGraph with distances.

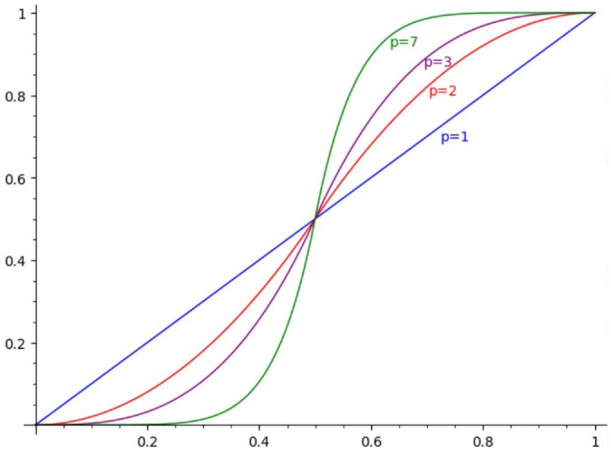


Figure 4.4: Effect of linear ($p = 1$) and non-linear mappings ($p > 1$) over the interval $[0, 1]$.

and MongoDB to manage the databases, as well as JavaScript, HTML, and CSS on the user side. Again, replicability of the results can be ensured by alternative implementations using the techniques and commands detailed above. Interested readers can also contact the corresponding author for further details.

4.2.6 Stability of the results

The stability of the results obtained was analyzed using boot-strapping resampling in order to simulate repetition of the experiments [121], as used by [139] for the study of the stability of sample configurations from projective mapping. For each value of $m = 10, 20, 30, \ldots, n$, subsets of m assessors were randomly drawn with replacement from the original data set. As in previous works [37, 17], a collection of 100 subsets was generated for each value of m.

For the study of the stability of MFA results three analyses were performed, using independent bootstrapping resamplings: With the first two dimensions of the MFA and with the first four dimensions as in the stability study by [139] and, in addition, with eight dimensions for better comparison with the results for SensoGraph.

In order to measure the agreement between the MFA consensus map for each subset and that of the original panel, the RV coefficient was computed [35, 131, 60]. This is a popular similarity measure between point configurations or matrices, whose values range between 0 and 1 and for which, the more similar two items are, the higher is the corresponding RV coefficient.

The RV coefficient was computed using the FactoMineR [69] function coeffRV. The mean and the standard deviation for RV coefficients of subsets of the same size were then obtained using the commands mean and sd of the R language [109].

For the study of the stability of SensoGraph results, the only meaningful analysis is considering the global similarity matrices. A graph is defined by a set of vertices and a set of pairs (i, j) of vertices [46], which are encoded as the set of entries (i, j) of a matrix. The two-dimensional representation of a global similarity matrix is a plotting artifact based on the graph-drawing algorithm considered. Furthermore, the random nature of the seeds chosen by the Kamada-Kawai algorithm used by SensoGraph could result in different coordinates in the consensus graph.

In order to measure the agreement between the SensoGraph similarity matrix for each subset and that of the original panel, the Mantel coefficient was computed [75], which evaluates the similarity between two configurations by measuring the correlation between two matrices of distances between the samples [2]. The Mantel coefficient is the usual Pearson correlation coefficient based on two vectors of size $n(n-1)/2$ (n being the number of products) containing the off-diagonal elements of the two dissimilarity matrices being compared. Its values range between -1 and 1 and, the more similar two items are, the higher is the corresponding Mantel coefficient. Widely used in ecology, the Mantel coefficient has also been used in sensory analysis, e.g., by [17] to investigate the stability of sorting maps.

The mantel coefficient was computed using the Python function skbio.stats.distance.mantel [129] using Pearson's product-moment correlation coefficient. The mean and the standard deviation for Mantel coefficients of subsets of the same size were then obtained using the commands mean dataset.mean and dataset.std of the pandas library [130].

4.3 Results and discussion

4.3.1 Results for MFA

The data obtained from the Projective Mapping tablecloths were exported as a CSV file and were processed using MFA as mentioned in Section 4.2.4.1. The computation took an average of 90.2 s, with a standard deviation of 11.7 s, measured over 5 independent runs on a computer with an Intel(R) Core(TM)2 Duo CPU @2.13 GHz with 4 GB of RAM.

The consensus representation of the similarities and differences among samples is shown in Figure 4.5. In this consensus graphic of MFA, the first two dimensions accounted for 41.12% of the explained variance (25.64% Dim1 and 15.48% Dim2). Low percentages of explained variance have been observed for panels composed by unexperienced consumers [90], with some works reporting such panels not positioning close together blind duplicate samples [89].

In our case, the panel of unexperienced consumers was able to position the blind duplicate samples 2 and 5, from the same brand (Chips-Ahoy),

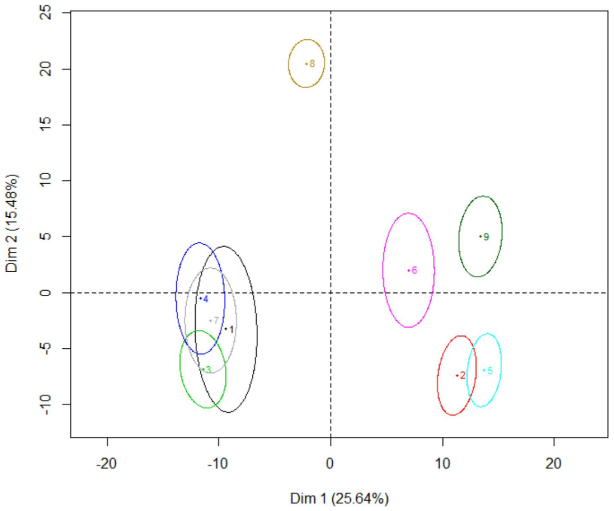


Figure 4.5: Consensus plot from MFA, dimensions 1 and 2.

close together in the lower-right quadrant. Their confidence ellipses overlapping means that no significant differences were perceived and, therefore, the consumers were capable to detect their similarity.

In addition to that, samples 1 (Hacendado), 3 (Carrefour), 4 (Grandino), and 7 (American Cookies) became positioned in the lower-left quadrant of the perceptual space, with their confidence ellipses overlapping indicating that consumers did not perceive statistically significant differences between them. These were the cookies with the highest percentage of chocolate: Samples 1 (Hacendado) and 3 (Carrefour) had a 37% of chocolate chips, while samples 4 (Grandino) and 7 (American Cookies) had a 29% of chocolate chips plus an 11% of milk chocolate chips. Recall Table 4.1.

Further, sample 6 (Alteza) and sample 9 (Ifa Eliges) appear separated, with their ellipses not overlapping but not far to each other and to samples 2 and 5. Probably, this can be explained because the samples 6 and 9 have a percentage of chocolate similar to samples 2 and 5 (25% and 25.6% of chocolate chips, respectively, see Table 4.1). Samples 6 and 9 are from a private label, but produced by the same manufacturer, while the duplicate samples 2 and 5 are from a best-selling multinational brand (see Table 4.1).

Finally, sample 8 (Dia), was isolated at the upper-left quadrant despite its proportion of chocolate chips being similar to others (26.7% of chocolate chips and 10.3% of milk chocolate chips, see Table 4.1). A possible explanation is that this sample appears as the farthest one to samples 2 and 5,

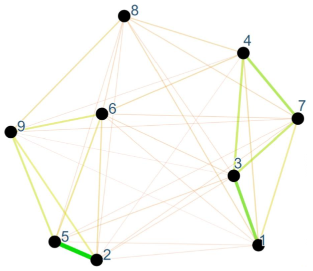


Figure 4.6: Consensus plot obtained using SensoGraph with Gabriel (Orden et al., 2019).

those from a best-selling multinational brand, so the private label cookie 8 was perceived as the least similar to a best-selling cookie.

The above discussion is to be considered taking into account that, due to the lack of batch tracking, changes in product recipes might have arisen over the two years of data collection.

4.3.2 Results for SensoGraph with Gabriel clustering

The data from the Projective Mapping tablecloths was uploaded to the SensoGraph software [96], which processed them as described in Section 4.2.4.2. The computation took less than one second with the same computer mentioned in Section 4.3.1.

Figure 4.6 shows the consensus graphic obtained, with all the connections between samples, obtained using SensoGraph with the Gabriel graph clustering as in [97], see Section 4.2.4.2. Figure 4.7 shows only the 60% of most relevant connections, while Figure 4.8 shows the dendrogram (left) obtained by hierarchical clustering from the global similarity matrix with the strengths of those connections (right). For the consensus plots and the global similarity matrix, a color code from red (smallest strength) to green (largest strength) has been used for the sake of an easier visualization.

These graphics show that the positioning of the samples provided by SensoGraph is similar to that in the consensus map given by MFA, the RV coefficient between the point configurations in Figures 4.5 and 4.6 being 0.8785. Groups 2–5-6–9, 1–3-4–7, and 8 can be identified in Figures 4.5 and 4.6, with the global similarity matrix in Figure 4.8 (right) easily showing that the connection 2–5 is the strongest (greenest) one, corresponding to

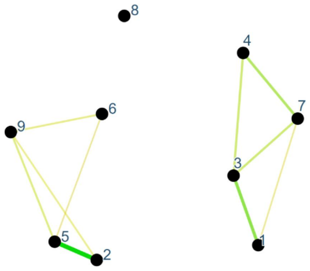
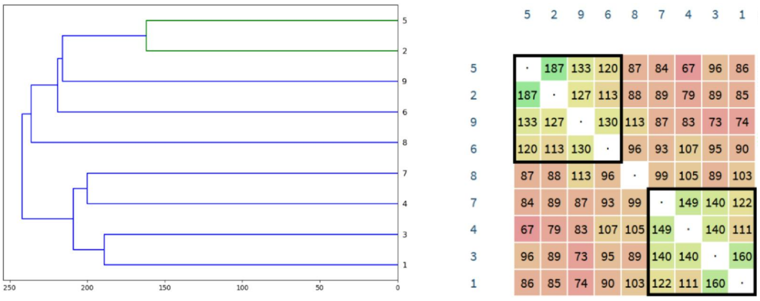


Figure 4.7: Consensus plot from SensoGraph with Gabriel, showing only the 60% of most relevant edges.



	5	2	9	6	8	7	4	3	1
5	·	187	133	120	87	84	67	96	86
2	187	·	127	113	88	89	79	89	85
9	133	127	·	130	113	87	83	73	74
6	120	113	130	·	96	93	107	95	90
8	87	88	113	96	·	99	105	89	103
7	84	89	87	93	99	·	149	140	122
4	67	79	83	107	105	149	·	140	111
3	96	89	73	95	89	140	140	·	160
1	86	85	74	90	103	122	111	160	·

Figure 4.8: Dendrogram (left) and global similarity matrix (right) for SensoGraph with Gabriel. The matrix is rearranged according to the result of the dendrogram, so that the groups obtained appear as submatrices with similar colors (framed).

samples 2 and 5 being the blind duplicates. It is interesting to note that SensoGraph identifies these blind duplicates as the most similar samples, both by the consensus map in Figure 4.6 and by the global similarity matrix in Figure 4.8, while in the MFA graphic for the first two dimensions (Figure 4.5) the pair 1–7 is instead the closest one (although the confidence ellipses for 2–5 do overlap, as discussed above). Nevertheless, note that the value of the connection 2–5 in the SensoGraph similarity matrix is just 187 out of the 349 consumers considered.

In addition, SensoGraph shows samples 6 and 9 appearing in the same group as the aforementioned samples 2 and 5, although the strength of the other connections in this group is quite smaller than that of the connection 2–5. Samples in this group share a similar percentage of chocolate chips since, as commented above, samples 6 (Alteza) and 9 (Ifa Eliges) have a 25% of chocolate chips, and samples 2 and 5 (Chips-Ahoy) have a 25.6% of

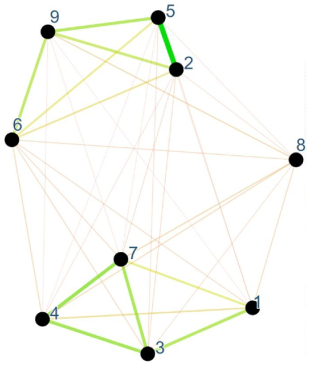


Figure 4.9: Consensus plot obtained using SensoGraph with distances, as introduced in Section 4.2.5.

chocolate chips (Table 4.1).

A second group observed in the SensoGraph graphic is composed by samples 1–3–4–7, with the corresponding ellipses overlapping in MFA as well. The connection 1–3 is the second strongest one and the connection 4–7 is the third strongest one, while connections 1–4 and 1–7 are not so strong, and the connections 3–4 and 3–7 have an intermediate strength. This group can be explained because the samples 1 (Hacendado) and 3 (Carrefour) have the same percentage of chocolate chips, 37%, while the sample 4 (Grandino) and the sample 7 (American Cookies) also have both the same percentage of chocolate (29% of chocolate chips plus 11% of milk chocolate chips).

Finally, the sample 8 (Dia) being isolated in MFA coincides with its connections to all other samples being weak in SensoGraph, and with this sample being actually isolated when showing only the 60% of most relevant edges (Figure 4.7). As happened for MFA, the sample 8 appears as the farthest one to samples 2 and 5 also in SensoGraph, both for the consensus map (Figure 4.6) and for the global similarity matrix (Figure 4.8, right).

4.3.3 Results for SensoGraph with distances

In this case, the data was uploaded to the software by [96] and the option of using SensoGraph with distances was selected, in order to process the data as described in Section 4.2.5. The computation took less than one second with the same computer mentioned in Section 4.3.1.

Figure 4.9 shows the consensus graphic obtained, with all the connec-

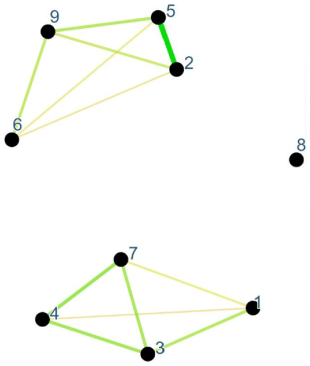


Figure 4.10: Consensus plot from SensoGraph with distances, showing only the 60% of most relevant edges.

tions between samples, using SensoGraph with distances, as introduced in Section 4.2.5. Figure 4.10 shows only the 60% of most relevant connections. Finally, Figure 4.11 depicts the dendrogram (left) obtained from the global similarity matrix (right). Again, the color code from red to green has been used for the consensus graphics and the similarity matrix.

The consensus map obtained (Figure 4.9) is similar to those given by the previous methods, the RV coefficient with MFA for the first two dimensions (Figure 4.5) being 0.7407 and that with SensoGraph with Gabriel (Figure 4.6) being 0.6390. Both the consensus maps (Figures 4.5, 4.6 and 4.9) and the dendrograms and global similarity matrices (Figures 4.8 and 4.11) show the same groups, being slightly more clear for SensoGraph with distances than for the variant with Gabriel (Figures 4.7 and 4.10, also Figures 4.8 and 4.11).

4.3.4 Stability of the results

As detailed in Section 4.2.6, bootstrapping was performed to analyze the stability of the results given in the previous subsections by the three methods considered.

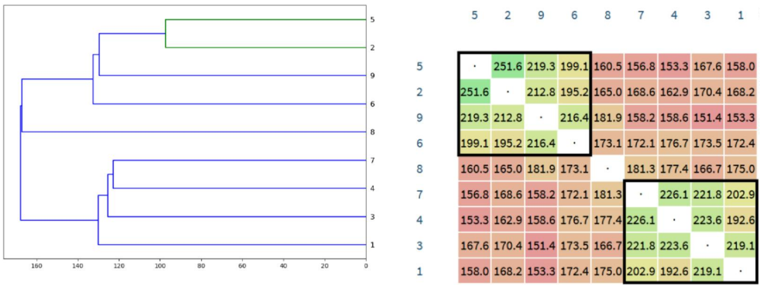
Figures 4.12-4.16 show, for different cases, the evolution of the RV and Mantel coefficients between virtual panels, composed by subsets of consumers randomly drawn with replacement, and the true panel. The vertical axis corresponds to the average RV or Mantel coefficient, incorporating standard deviations as vertical bars, while the horizontal axis corresponds to the number of consumers in the virtual panel. As expected, in all the cases increasing the number of consumers leads to an increase of the RV or Mantel coefficients and a decrease of the standard deviations.

Previous works [37, 17, 139] have considered different values of the RV coefficient as the threshold above which the results are considered to be stable, the most restrictive one being the value 0.95 proposed by [17], which is depicted as a red horizontal dashed line. There is no agreement in the literature about which value of the RV coefficient indicates a good agreement, with [139] reporting works which consider values that range between 0.65 and 0.95. For the Mantel coefficient there is no standard threshold value as well, although [17] observed a strong linear relationship between both types of coefficient, with the Mantel coefficient tending to provide slightly lower values than the RV coefficient.

For the case of MFA, considering the first four dimensions instead of the first two leads to better results, with higher initial RV coefficients and smaller standard deviations. [139] observed similar results where, their work considering up to 100 consumers, some panels did not reach the 0.95 RV coefficient suggested by [17]. The results we obtain show that for this panel of unexperienced consumers with the first two dimensions accounting for only 41.12% of the explained variance, the 0.95 threshold is achieved for around 200 consumers (Figure 4.12). When considering the first four dimensions, accounting for 65,95% of the explained variance, the 0.95 threshold is achieved for around 150 consumers (Figure 4.13). Considering the eight dimensions, which obviously account for the 100% of the explained variance, allows to achieve the 0.95 threshold for 40 consumers (Figure 4.14).

It should be noticed that [88] analyzed the corpus of 46 publications in Food Quality and Preference and Food Research International dealing with projective mapping until then, concluding that most of the papers considered only the first two dimensions of the MFA, with just a few of them (6 out of 46) going further and using up to the first four dimensions. To the best of our knowledge, no previous work used the full (eight) dimensions of MFA to study the stability of the samples configuration.

As for SensoGraph with Gabriel, the results in Figure 4.15 show smaller deviations than those for MFA with the first two dimensions (Figure 4.12), although more consumers are needed to achieve the 0.95 threshold, around 300 consumers. This could be explained because, as observed by [17], the Mantel coefficient tends to be slightly smaller than the RV coefficient. Ac-



	5	2	9	6	8	7	4	3	1
5	·	251.6	219.3	199.1	160.5	156.8	153.3	167.6	158.0
2	251.6	·	212.8	195.2	165.0	168.6	162.9	170.4	168.2
9	219.3	212.8	·	216.4	181.9	158.2	158.6	151.4	153.3
6	199.1	195.2	216.4	·	173.1	172.1	176.7	173.5	172.4
8	160.5	165.0	181.9	173.1	·	181.3	177.4	166.7	175.0
7	156.8	168.6	158.2	172.1	181.3	·	226.1	221.8	202.9
4	153.3	162.9	158.6	176.7	177.4	226.1	·	223.6	192.6
3	167.6	170.4	151.4	173.5	166.7	221.8	223.6	·	219.1
1	158.0	168.2	153.3	172.4	175.0	202.9	192.6	219.1	·

Figure 4.11: Dendrogram (left) and global similarity matrix (right) for SensoGraph with distances. The matrix is rearranged according to the result of the dendrogram, so that the groups obtained appear as submatrices with similar colors (framed).

tually, in this case, considering 200 consumers achieves a 0.92 threshold.

It is worth noticing that, although the dimensionality of the 9×9 global similarity matrix is eight (it defines nine points in nine dimensions but the fact that entries (i, i) are zero implies that those points do actually lie on an 8-dimensional hyperplane), the global similarity matrix can also be depicted in a single 2-dimensional graphic (Figure 4.8, right).

Finally, SensoGraph with distances leads to the results depicted in Figure 4.16, which show higher Mantel coefficients and smaller standard deviations than the version using Gabriel. In particular, the 0.95 threshold is achieved for around 200 consumers, like for MFA with the first two dimensions (Figure 4.12), but with smaller deviations. Further studies with different panels and products, as in [139], should be performed in order to confirm this behavior.

4.4 Conclusions

This work used Projective Mapping for the evaluation of commercial chocolate chip cookies by a large number of $n = 349$ unexperienced consumers, analyzing the data with statistical (MFA), geometric (SensoGraph-Gabriel) and mixed (SensoGraph-distances) methods. All of them provided the same groups of samples, with the two blind duplicates being positioned close together. The identification of these duplicates was clearer for the geometric and mixed techniques than for the MFA consensus map, where a different pair of samples appeared as the closest one.

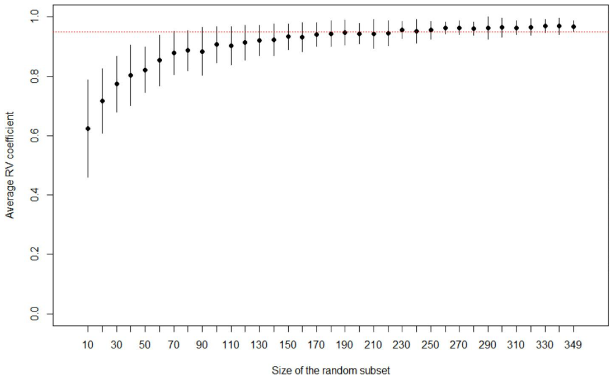


Figure 4.12: Evolution of the RV coefficient between subsets of consumers randomly drawn with re- placement and the whole panel, for the first two dimensions of MFA (Figure 4.5). The red horizontal dashed line corresponds to a 0.95 threshold.

The stability of the results was studied with bootstrapping resampling, randomly drawing 100 subsets of $m = 10, 20, 30, \ldots, n$ assessors from the original data set and measuring the agreement with the original panel by the RV coefficient for MFA and the Mantel coefficient for SensoGraph. MFA achieved the highly restrictive RV 0.95 stability threshold for around 200 consumers when using the first two dimensions, for around 150 consumers when considering the first four dimensions, and for 40 consumers when all the eight dimensions are considered. For SensoGraph with Gabriel, the Mantel 0.95 stability threshold was achieved around 300 consumers, while SensoGraph with distances led to values beyond the Mantel 0.95 threshold for around 200 consumers. These values are to be considered taking into account that Mantel coefficients have been previously observed to be slightly smaller than RV coefficients.

Further research, with different panels and products, would be needed in order to confirm these behaviors which suggest that, on one hand, global similarity matrices are useful for Projective Mapping data analysis and, on the other hand, graph drawing techniques provide reliable consensus maps.

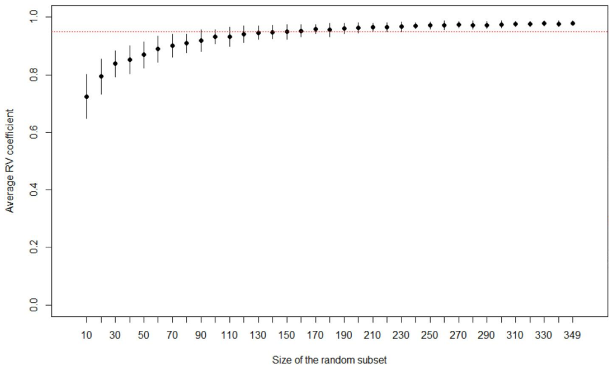


Figure 4.13: Evolution of the RV coefficient between subsets of consumers randomly drawn with replacement and the whole panel, for the first four dimensions of MFA. The red horizontal dashed line corresponds to a 0.95 threshold.

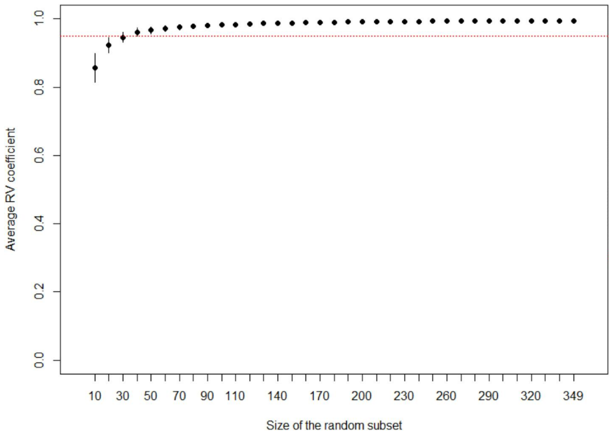


Figure 4.14: Evolution of the RV coefficient between subsets of consumers randomly drawn with replacement and the whole panel, for the eight dimensions of MFA. The red horizontal dashed line corresponds to a 0.95 threshold.

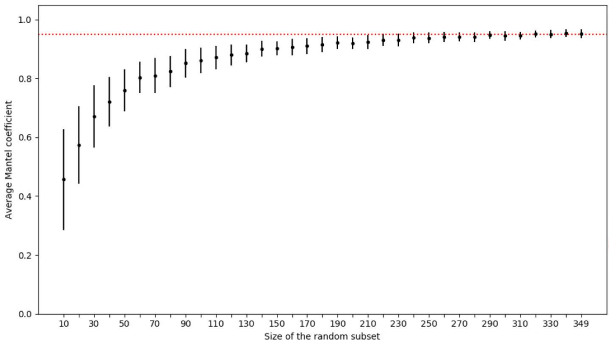


Figure 4.15: Evolution of the Mantel coefficient between subsets of consumers randomly drawn with replacement and the whole panel, for the SensoGraph with Gabriel global similarity matrix (Figure 4.8). The red horizontal dashed line corresponds to a 0.95 threshold.

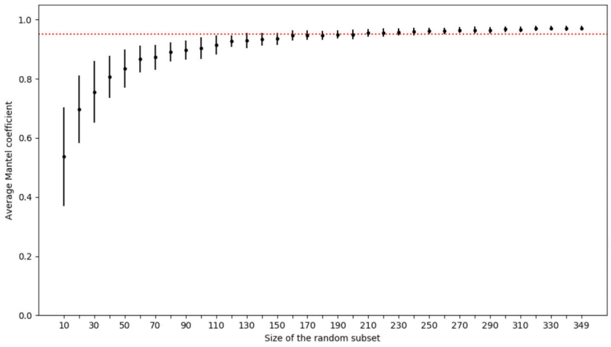


Figure 4.16: Evolution of the Mantel coefficient between subsets of consumers randomly drawn with replacement and the whole panel, for the SensoGraph with distances global similarity matrix (Figure 4.11). The red horizontal dashed line corresponds to a 0.95 threshold.

Chapter 5

Flight level assignment using graph coloring

This chapter contains an article that appeared originally in 2020 in the journal Applied Sciences Volume 10, Issue 18, 6157. It was written in collaboration with Jose Manuel Gimenez-Guzman, Rene D. Reyes-Bardales, David Orden and Ivan Marsa-Maestre. The front page of this article can be found in Appendix B.

5.1 Introduction

The increasing congestion of the airspace is widely accepted to be one of the outstanding problems in today's society. Researchers agree in considering the state-of-the-art methods for air traffic management (ATM) to be insufficient to handle the expected growth, see the survey by Vossen et al. [142]. Barnier and Brisset [14] describe how the airspace is divided into smaller regions, each one handled by air traffic control (ATC). Therefore, the assignment of flight levels (FLs) is carried out locally within each region, resulting in a suboptimal solution both for individual routes and for the overall network. This is mainly due to the fact that ATM often requires level changes, which are costly in terms of fuel and emissions, as well as deviations from the optimal path from origin to destination, which are also costly in terms of time and fuel. This is why allocation of flight levels is the goal in [14], where a graph coloring problem is solved by a combination of a greedy algorithm and constraint programming.

Being a well-established tool coming from discrete mathematics, graph coloring has proved to be useful in many different applications. There are a number of different types of graph coloring problems. For an exhaustive description of most of them, the reader is kindly redirected to the surveys in [74, 133]. For the purposes of this paper, we will discuss the classic graph coloring problem, which is the one used by Barnier and Brisset in [14], and the spectrum coloring problem, which we proposed in [98]. The classic graph coloring problem assigns to each vertex in the graph a label or color, with the constraint that two vertices of the graph cannot have the same color if there is an edge connecting them -this kind of edges are called monochromatic edges, and the colorings which avoid them, proper colorings. If we envision a graph where aircraft routes are vertices, edges connect intersecting routes, and colors represent FLs, this problem can be used to ensure that no two crossing routes are assigned the same FL [14]. A limitation of this approach is that routes may cross at adjacent FLs. Safety considerations, however, should favor those colorings where crossing routes use further separated FLs, so that height changes due to flight conditions are less likely to result in accidents. Taking this into account, we consider here the general spectrum graph coloring problem [98], where there is a set of available colors and a matrix of interferences between colors, and the goal is to minimize the total interference. We have achieved significantly good solutions using this technique in domains as diverse as Wi-Fi channel assignment [98] and reactive network resiliency [79]. In this paper we evaluate its applicability to traffic management, by mapping interference to closeness between flight levels -i.e. the closer the FLs, the higher the interference.

There are a number of papers on flight level assignment in the literature. Sasso et al. [28] propose simulated annealing to optimize air traffic through small reroutes and en-route level changes, as well as flight delays to handle intersecting routes. Flying at an optimal altitude can help to save a significant amount of fuel during the cruise phase of a trip, but there is a fuel cost in reaching that altitude. This yields a trade-off between optimizing fuel expenditure by flying at optimum altitude and the extra fuel spent to reach that altitude. The length of the air route arises as a significant variable since, the longer the flight, the smaller percentage of total fuel spent is due to reaching optimum flight level [124].

Both fuel consumption (e.g., [7, 27, 132]) and flight level assignment (e.g., [14, 25, 28]) have been addressed in the literature but, to the best of our knowledge, this is the first work considering both of them simultaneously for sets of routes. In doing so, we provide a real-world example of an instance of behavior optimization in a complex self-interested network, in the sense that there are interdependent decision makers (the flight pilots or the airlines governing these flights), whose individual interests (minimizing fuel consumption for their own flights) are different from the social goal (ensuring safety distances between FLs for all the flights). There is, of course, an individual incentive to position a flight in its optimal fuel consumption level, but not all flights can fly at their optimum without this being a great risk in terms of safety. The coexistence of different interests between individual goals which solely depend on individual actions (i.e., how far an aircraft is from its optimal FL), and social goals which depend on the interaction between actions (i.e., at which FLs are conflicting aircrafts flying), suggests to model the problem as a factorized optimization, and to evaluate techniques similar to the ones we have used successfully in other complex network domains [43].

To optimize fuel consumption together with flight level assignment, we can resort to a number of optimization techniques that can be used for graph coloring. Examples of such optimization techniques that we have already successfully used to color graphs are briefly described next. Harmony Search [40] is an evolutionary nonlinear optimization technique inspired in musical composition that we have used in [29]. Another family of optimization techniques that are suited for this problem are the ones based on Particle Swarms [104], which we successfully used for that purpose in [30]. More specifically, in [30] we used an implementation that relies on augmented Lagrange multipliers to deal with restrictions [56]. Another more recent nonlinear optimization technique we have used for graph coloring in [22] is based on Coral Reefs [115]. More specifically, it is a bio-inspired technique which simulates how a coral reef evolves, considering aspects like the fight for space in the reef and specific features of the corals' reproduction. Although all the aforementioned techniques have shown to be useful to color graphs, in this paper we have chosen Simulated Annealing (SA) [1] as it has been shown it is able to obtain high-utility results very efficiently [67]. In addition to SA, we have also considered Hill-Climbing (HC) as an optimization technique because, as discussed in [64], the comparison between HC and SA will let us to assess if the problem under study is highly complex, because in those scenarios greedy optimizers like HC tend to be blocked in local optima, while the exploration probability of SA allows to escape from those local optima.

The present paper contributes to this goal in the following ways:

- We derive functions for the individual and social utilities, which we use later in the model (Section 5.2.1).
- We model the joint optimization of fuel consumption and flight level assignment as a factorized optimization problem (Section 5.2.2).
- We provide an approximate optimization solution to the problem as an instance of a wide class of coloring problems (Section 5.2.3).

To validate our model and evaluate the significance of our contributions, we have performed a set of experiments over a graph representing the Spanish air space, to see the relative performance of different optimization approaches. The experiment settings and the discussion of results are provided in Section 5.3. Finally, Section 5.4 summarizes our conclusions and sheds light on future lines of research.

5.2 Flight-Level Assignment as Spectrum Coloring Problem with Hard Constraints

In this section we describe the model we have developed for the problem in this work. We start by stating the utilities for the different actors in the model, including the individual utilities (each flight operator wants to minimize fuel consumption) and the social utility (keep crossing airways separate enough to ensure certain levels of safety). Next, we model the flight level assignment as a factorized optimization problem. Finally, we propose to approximate the solution to the problem as an instance of spectrum graph coloring.

5.2.1 Individual and social utility models

As stated above, the model considers the interaction between the individual actors (who have the goal to minimize their individual fuel consumption) and the social goal (of ensuring safety by keeping intersecting airways far away from each other).

5.2.1.1 Individual utility model: fuel consumption

Individuals (e.g., flight operators) will want to minimize their utility functions, defined as the fuel consumption for each flight. Fuel consumption is a function of both the distance traveled and the cruise flight level, and takes into account fuel consumption at the cruise phase and during the climb phase. For longer flights, more time will be spent in the cruise phase, so flying at sub-optimal altitude will have a more noticeable effect on fuel consumption than in a shorter flight. In this work, we will neglect the fuel used during the descent phase as literature suggests that the descent profile for most flights is approximately the same regardless of total distance [124], and the idle setting of engines during this phase makes fuel expenditure invariant to distance and flight level.

With the aforementioned considerations, the individual utility function corresponding to fuel consumption that we will use for a flight of distance d and assigned flight level x is:

$$F(d,x) = F_{climb}(x) + F_{cruise}(d-\varepsilon, x), \tag{5.1}$$

where F_{climb} represents the fuel consumption during the climb phase, F_{cruise} is the fuel consumption during the cruise phase, and ε is the horizontal distance traversed during the climb. F_{climb} and F_{cruise} are defined as piecewise functions depending on the aircraft used.

In this work, we will focus on the Airbus A320 aircraft, which is one of the most common aircrafts used for Spanish domestic routes. There exists a trade-off between the better Specific Air Range (SAR)[1] provided by higher flight levels (until the optimal FL for the A320 is reached), and the fuel required to reach a given FL. In other words, some flights may be so short that reaching their optimum flight level may not be worthy, since the fuel savings during the time spent at cruise level do not compensate the fuel cost of reaching that altitude.

To pre-calculate these optimum flight levels, we will define the functions as piecewise linear approximations based on existing literature. For the climb costs, we use [85] to construct Table 5.1.

In view that the relationship between the flight level number and the fuel consumption is approximately linear, we can then define the following

[1] Specific Air Range (SAR) is the distance travelled per fuel unit.

Table 5.1: Climb fuel cost (extracted from [85]).

FL (100s ft)	Fuel (kg)	$\frac{\Delta Fuel}{\Delta FL}$
200	600	n/a
240	710	2.75
280	820	2.75
300	875	2.75
320	940	3.25

piecewise linear approximation:

$$\begin{cases} F_{climb}(x) = 2.75(x - 200) + 600, & 200 \leq x \leq 300 \\ F_{climb}(x) = 3(x - 300) + 875, & 300 \leq x \leq 400 \end{cases} \tag{5.2}$$

For the cruise phase, we use [73] to obtain the results in Table 5.2, where different FLs are listed, along with the SAR goodness (expressed as % deviation from optimal SAR) and the slope resulting from comparing to the previous (lower) FL. We use these slopes to make a piecewise-linear approximation defining an auxiliary function $S(x)$, which gives an approximate value for this percentage deviation from the optimum SAR corresponding for a given flight level:

Table 5.2: Cruise fuel cost (extracted from [73]).

FL (100s ft)	% off from optimum SAR	$\frac{\Delta Percentage}{\Delta FL}$
300	-12	n/a
320	-8	0.2
340	-4	0.2
360	-1.5	0.125
380	0	0.075
400	-3	0.15

$$\begin{cases} S(x) = -4 - 0.2(340 - x), & 200 \leq x \leq 340 \\ S(x) = -4 + 0.125(x - 340), & 340 < x \leq 360 \\ S(x) = -1.5 + 0.125(x - 360), & 360 < x \leq 380 \\ S(x) = -0.15(x - 380), & 380 < x \leq 400 \end{cases} \tag{5.3}$$

Based on the $S(x)$ function and the fact that

$$\text{SAR} = \frac{\text{distance}}{\text{Unit Fuel}} \Rightarrow F_{cruise} = \frac{\text{distance}}{\text{SAR}}, \tag{5.4}$$

we can deduce an expression for the fuel required in the cruise phase, based on distance d, flight level x and a constant φ that represents the optimum SAR:

$$F_{cruise}(d, x) = \frac{d}{(1 + S(x))\varphi}. \tag{5.5}$$

We can now find the total fuel required, based on these piecewise functions. We recall the previous expression

$$F(d, x) = F_{climb}(d) + F_{cruise}(d - \varepsilon, x). \tag{5.6}$$

Since we now have explicit expressions for climb and cruise, we can calculate the utility function, just needing to set the parameters ε and φ to a realistic value. The values that are used will be set appropriately to the type of aircraft.

As stated above, for the purpose of this paper the aircraft considered will be an Airbus A320. Based on the slopes observed in the tables above, both the cruise and climb utility functions are linear approximations. The tables provide the operating points and ranges from literature, but given that we may use certain operating points below or above the limits, we make an extrapolation for these points using a linear model. This is viable because, in the case of the climb, the fuel expenditure is almost perfectly linear, with a rate of change around 3 (observed at 3 ± 0.25). Then, for the cruise phase, we need to extrapolate FLs in the range below FL300, and the percentage off from optimum SAR starts to behave linearly.

Finally, we obtain the values used for ε and φ based on the literature. From [4], we get that at Cost Index equal to zero (CI $= 0$, maximum fuel efficiency), 150NM (278km) are travelled when climbing to FL330. Assuming a constant rate of climb, we have, for a given flight level x:

$$\varepsilon(x) = 0.84x. \tag{5.7}$$

For the parameter φ, we use the data in [19], which shows that as the length of a flight increases (which means cruise phase fuel expenditure becomes a larger percentage of the total used), the average fuel spent per kilometer tends to 3.3kg/km. This means that each kg of fuel yields approximately 0.303km, so we can set $\varphi = 0.303$.

Based on the proposed model, Fig. 5.1 shows the total fuel consumption of the Airbus A320 for different values of distance (d) and FL (x) according to our model.

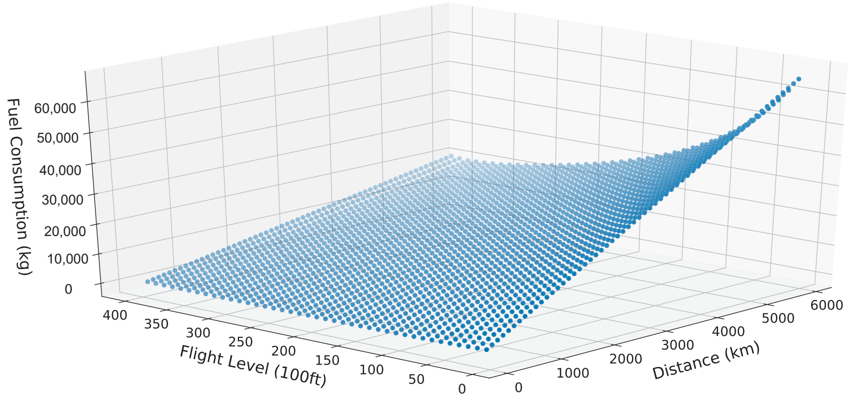


Figure 5.1: Total fuel consumption of Airbus A320 as a function of distance and FL.

5.2.1.2 Social utility: interference between crossing airways

In this section we bring social safety considerations to flight level assignment, considering flight levels as corridors in which crossings arise with the largest possible vertical distance between the two flight levels involved. Thus, there is no need to handle the possibility of two aircrafts passing through a crossing at the same time, hence not needing to take into account the actual position of aircrafts within the corridor in every time unit.

We translate the search for the largest possible vertical distance at crossings of routes to an instance of the the Spectrum Coloring problem we studied in [98]. There, we introduced the idea of having an abstract graph G and a spectrum of colors $C = \{c_1, \ldots, c_n\}$ together with a matrix W storing weights $W_{ij} = W(c_i, c_j)$ for each pair of colors. Thus, a coloring c of the graph G will induce, via the matrix W, an *interference* I_v at each vertex v, according to the following expression:

$$I_v(G, W, c) = \sum_{u \in N(v)} W(c(u), c(v)), \tag{5.8}$$

where $N(v)$ is the set of neighbors of vertex v.

The applicability of Spectrum Coloring to the problem presented in this paper is clear when we map the colors in the spectrum C to the FLs available for our flights. Here, the entry W_{ij} of the matrix of interferences represents the impact on safety of assigning FLs c_i and c_j to a pair of crossing routes. Clearly, this interference must have a maximum value when the separation between FLs is zero (that is, in the diagonal of the matrix), and could have decreasing values as the separation between FLs increases.

One can devise different approaches to define the matrix W_{ij}: We could take an academic approach and use linear or exponential decays for the interferences, as we did in [81, 98], or we could take a more risk-based approach and use the probabilities for a collision when the corresponding FLs are used (in a similar way to what we did in [79] in a different domain). Finally, we can take a more conservative approach and assume that crossing routes must have FLs with a minimum separation, so that the interference induced by FLs closer than that minimum distance threshold must be infinite. As we will discuss later, this is the approach we have taken in this work.

5.2.2 Factorized Optimization Model

The two utility models defined in the previous section have inherent differences. Not only do they represent the utilities of different actors (individual utility *vs.* social utility), but they also differ in the dimensions of the problem they depend on. Fuel consumption is a per-flight utility value which depends solely on an autonomous decision performed for each individual flight, the FL chosen for the cruise phase. The social utility yielded by the interference matrix W, instead, depends on the interplay between these individual autonomous decisions.

Given this coexistence of individual contributions to utility and also interactions between actions, it makes sense to model the whole problem as a *factorized optimization problem*, that is, a problem P of the form

$$\text{minimize} \sum_{i \in V} \Phi_i(x_i) + \sum_{s \in S} \Psi_s(x_s), \tag{5.9}$$

where V is a finite set of *variables* and S is a finite collection of subsets of V representing constraints. Φ_i and Ψ_s are real-valued functions called, respectively, *variable functions* and *factor functions*, representing the impact on the objective function of the value of each independent variable, and of combinations of variable values (i.e., interdependences between variables).

In our setting, V would be the set of routes in our scenario and, for each $i \in V$, x_i represents the FL chosen for that route. Hence, the Φ_i functions would represent fuel consumption and would be defined as $\Phi_i(x_i) = F(d_i, x_i)$, where d_i is constant for each route i and represents the distance for that route.

In the same way, S represents the pairs of routes in which the trajectories intersect, and for each $s \in S$, defined as $s = (i, j)$ the function Ψ_s is defined as $\Psi_s(s) = W_{ij} = W(x_i, x_j)$, taking values directly from the interference matrix.

This mapping allows to generalize techniques that we have applied successfully to very different domains, such as Wi-Fi channel assignment [77]. These techniques assume an augmented graph coloring problem, which we describe in the following section.

5.2.3 Graph coloring model

As stated above, the safety considerations in the FL assignment scenario can be translated to an instance of the Spectrum Coloring problems we studied in [98]. Problems in this class are defined by an abstract graph G, a spectrum of colors C, and an interference matrix W. The construction of the graph G requires further attention and will be detailed in Section 5.3.1. For the spectrum C we consider 40 colors, corresponding to the 40 FLs ranging from FL10 (1,000 feet) to FL400 (40,000 feet), with 1,000 feet separation between adjacent FLs. Finally, in order to define the interference matrix W we will use a conservative approach, which is coherent with the serious safety considerations in aviation scenarios. We will assume that there is a distance threshold δ, expressed in multiples of 1,000 feet, below which the risks of incidents are not tolerable, so we will assign an infinite value to the entries of W corresponding to pairs of colors in which corresponding FLs are closer than the threshold δ. For $\delta = 0$ the interference matrix W^0 will be the null matrix. Recalling Eq. 5.9, in the $\delta = 0$ case only fuel consumption is considered, without taking interferences into account. For $\delta = 1$, the interference matrix W^1 has the value ∞ at all the entries of the main diagonal, that is,

$$W^1 = \begin{pmatrix} \infty & 0 & \dots & 0 & 0 \\ 0 & \infty & \dots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & \infty & 0 \\ 0 & 0 & \dots & 0 & \infty \end{pmatrix}. \tag{5.10}$$

This corresponds to the classic Proper Coloring problem where no monochromatic edges are allowed. In our setting, this means that no two crossing routes are allowed to be assigned to the same FL. In a similar way, for $\delta = 2$ the interference matrix W^2 is a tridiagonal matrix which adds values ∞ at the superdiagonal and subdiagonal, as follows:

$$W^2 = \begin{pmatrix} \infty & \infty & 0 & 0 & \dots & 0 \\ \infty & \infty & \infty & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ 0 & \dots & \infty & \infty & \infty & 0 \\ 0 & \dots & 0 & \infty & \infty & \infty \\ 0 & \dots & 0 & 0 & \infty & \infty \end{pmatrix}. \tag{5.11}$$

This corresponds to an $L(h, k)$-labelling [21] with $h = 2$ and $k = 0$. In our problem, this means that equal or adjacent FLs are not allowed for crossing routes. Analogously, similar matrices W^δ can be derived for higher values of δ.

Our spectrum coloring model has great expressivity, allowing a number of different possibilities, which we leave for future work. One example is to define a matrix $\widetilde{W}^2$ having infinite values at the diagonal as before, but with finite values at the superdiagonal and the subdiagonal. In such a model, we would be ensuring a separation of at least one flight level between crossing routes and encouraging (instead of forcing) a separation of two FLs between such routes. This would give better flexibilities to pilots, which would be able to move to an adjacent different flight level during the cruise stage in case of, for example, turbulences. Another example would be different decision makers (pilots or airlines) having matrices W with different finite values, modeling different risk tolerances.

5.3 Experimental Evaluation

5.3.1 Graphs used

For the sake of considering a tractable but sufficiently complex problem, we restricted ourselves to the Spanish airspace. The set of routes for our experiments has been built collecting those routes having both origin and destination in Spain from OpenFlights [102]. The database used as dataset from OpenFlights [102] is composed by 67,663 routes operated by 548 different airlines and between 3,321 airports. Among other information, the database includes the source and destination airports, along with their planar coordinates in the map. Among these, we extracted intersecting routes, for which the aim is to ensure different FLs at intersections. The airport locations are considered given by their planar coordinates in the Mercator projection of Spain and the routes are modeled as straight segments from origin to destination, as Barnier and Brisset do [14] (Fig. 5.2).

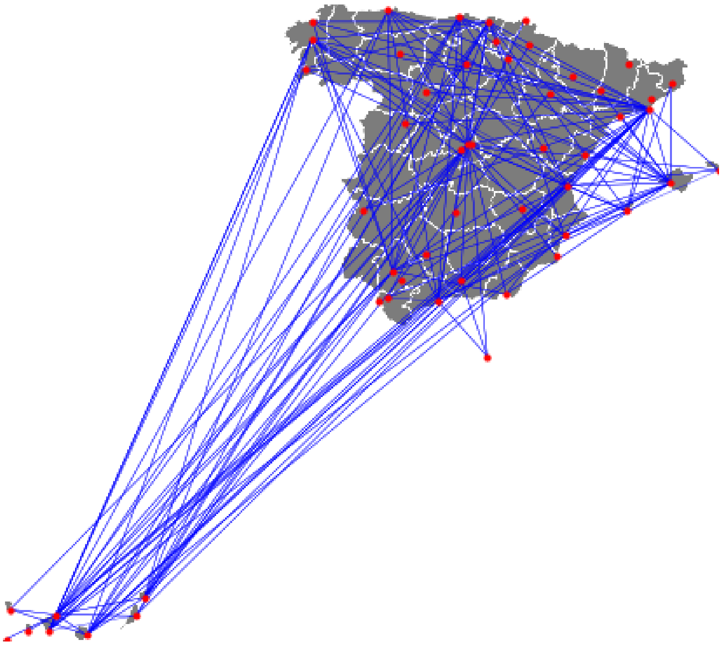


Figure 5.2: Airports and intersecting airways on the Mercator projection of Spain.

The route graph as described will not be used directly as input for the graph coloring described in Section 5.2.3. Instead, as in [14], we will use an intersection graph $G = (V, E)$ where each of those airways is a vertex in V and two vertices are joined by an edge if, and only if, the corresponding routes do intersect (Fig. 5.3).

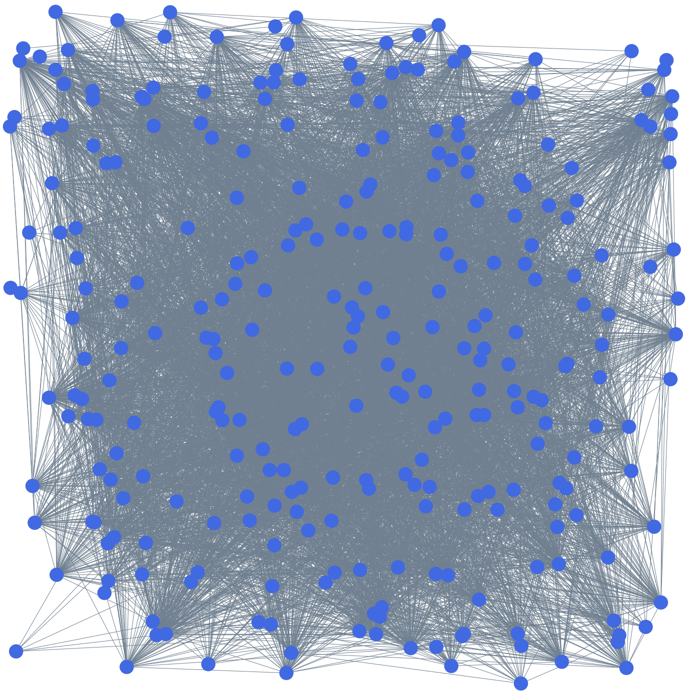


Figure 5.3: Intersection graph $G = (V, E)$ considered, where V (blue nodes) is the set of routes and the (grey) edges in E connect two of those routes when they intersect.

The implementation of the graphs has been done in a Python environment using the *networkX* library [91].

To construct the set E of edges in such a graph, each pair of routes has to be tested in order to determine whether the routes intersect or not. These intersection tests are performed assuming that the two routes are not bidirectional, because in that case they would certainly intersect. The following two conditions need to be fulfilled by a pair of routes to intersect. First, it is necessary that the convex hull of the four airports (smallest convex set containing them) is a quadrilateral. Otherwise, the convex hull is a triangle and the routes do not intersect (Fig. 5.4). Second, when the convex hull of the four airports is a quadrilateral, the routes intersect if, and only if, the routes represent diagonals of that quadrilateral (Fig. 5.5).

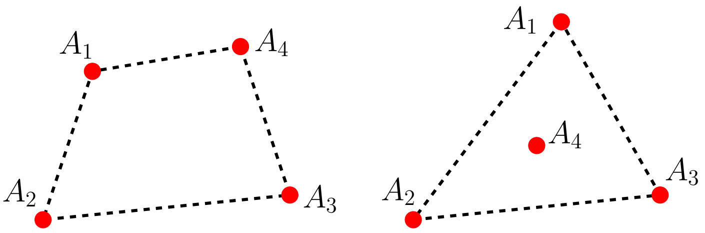


Figure 5.4: Possibilities for the convex hull of the four airports involved in two routes.

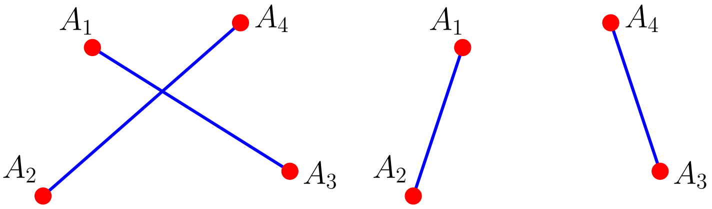


Figure 5.5: Cases with the convex hull being a quadrilateral.

In order to distinguish between triangular and quadrilateral convex hulls, we first compute the sign of each triplet, defined as the determinant

$$\operatorname{sign}(A_i, A_j, A_k) = \begin{vmatrix} 1 & 1 & 1 \\ A_{i,x} & A_{j,x} & A_{k,x} \\ A_{i,y} & A_{j,y} & A_{k,y} \end{vmatrix}, \tag{5.12}$$

where $A_{i,x}$ and $A_{i,x}$ are, respectively, the horizontal and vertical coordinates of the location for airport A_i.

That sign equals respectively $+1$ or -1 when the turn A_i, A_j, A_k is, respectively, counterclockwise or clockwise. Thus, the outcome of the product $\operatorname{sign}(A_1, A_2, A_3) \cdot \operatorname{sign}(A_1, A_2, A_4) \cdot \operatorname{sign}(A_1, A_3, A_4) \cdot \operatorname{sign}(A_2, A_3, A_4)$ is respectively $+1$ or -1 when the convex hull of the four points is, respectively, a quadrilateral or a triangle. Then, we just check if the routes are diagonals of the quadrilateral (hence crossing) or not.

Performing these intersection tests leads to the intersection graph $G = (V, E)$ depicted in Fig. 5.3, with 6,597 edges and 288 vertices. Observe that nodes in this graph correspond to routes in Fig. 5.2.

Finally, it is interesting to note that each node (route) in the graph does not represent a single flight, but all the flights that take the same route. This means that, in order to compute the fuel consumption for a given FL assignment, the fuel cost for each route is multiplied by the frequency of that route (extracted from the OpenFlights data [102]). We have considered all the flights that take place in a day for internal routes in Spain.

5.3.2 Algorithms Used

Given the intersection graph obtained from the air routes, we will use some variations of the optimization techniques proposed in [77], with utility functions and hard constraints adapted to the context of ATM. Four algorithms are used to address the graph coloring problem, two are inspired by Simulated Annealing (SA), and two by Hill-Climber (HC) optimization approaches. Similarly, two of the algorithms use a more greedy approach, where FLs are assigned as close to the optimum as possible for each route, whereas the other two use a global approach, where any FL may be assigned.

There is a hard constraint on this graph coloring problem, which is that nodes sharing an edge should have a separation of at least δ FLs. This is due to the fact that these nodes represent intersecting routes, so having them at the same FL would be unsafe. This illustrates how such an optimization pivots around finding a trade-off between safe and fuel-efficient FL assignments.

The versions of SA and HC used work basically by starting at a random solution point (in our scenario, a random assignment of FLs to all flights), and iterating through single issue (in our case, single route) random mutation. Accordingly, at each iteration, the FL of a randomly selected airway is assigned a random new value. The key difference between HC and SA is that, for the hill-climber, the new mutated solution is only *accepted* (and therefore used as a basis for the next iteration) when it yields an improvement in terms of utility, while for the annealer there is a finite probability to accept a solution even when it causes utility losses. That finite probability depends on a parameter called *annealing temperature*, and decreases as the algorithm iterates. The rationale is that this randomized acceptance approach allows the annealer to escape local optima, which gives it an advantage in highly rugged utility spaces [30].

For the problem studied in this paper, the algorithms have been modified to enforce a separation of at least δ FLs between crossing routes, and to take into account the utility function based on fuel efficiency. In both the SA and HC algorithms, conflicts are avoided at every color assignment. To prevent conflicts, the algorithm follows these steps when assigning a new random color to a node:

1. Create a list of available colors representing FLs (this depends mostly on whether it is a global or greedy approach).
2. Iterate through the node's neighbors (that is, all intersecting routes), and remove from the list of available colors all colors within separation δ.
3. Once this process is done, if there are still colors available, one is chosen randomly.
4. For the Greedy HC and Greedy SA: If none of the colors within the original range are available, the algorithm assigns the next closest available FL (i.e, if FL_{opt} is the optimal FL for a given route and the original range is $\text{FL}_{opt} \pm 5$, then the algorithm will assign the first available FL out of $\text{FL}_{opt} \pm 6, \text{FL}_{opt} \pm 7$, etc.).
5. If none of the 40 FLs are available, this means the current node cannot be changed.

Then, the utility function for fuel efficiency depends on the distance of the route and the assigned FL. Since the length of the route is fixed, utility optimization becomes completely dependent on FLs. Since not all routes can be assigned their optimum FL, part of the optimization implicitly revolves around determining which routes have the least negative effect when assigned a suboptimal FL.

5.3.3 Experimental Results

In this section we evaluate the performance in terms of fuel consumption for the four different algorithms stated before. To conduct this study, simulations were run with 100, 500, 1,000, 2,000, 3,000, 4,000, 5,000 and 10,000

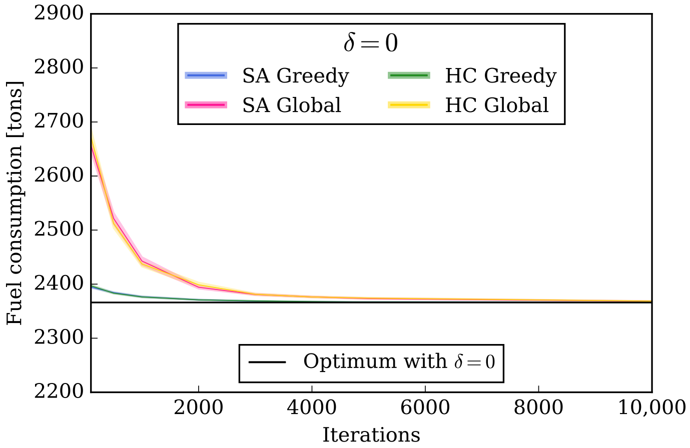


Figure 5.6: Total fuel consumption for the different techniques and $\delta = 0$.

iterations, making 20 runs for each of them, to ensure statistical relevance of the results. With respect to the complexity of the algorithms used, for both HC and SA and for each iteration of the algorithm we require an approximately constant time for mutating a color. This is followed by an evaluation of the utility function, for which the complexity depends on the problem size. When we change the color of a vertex, the utility only changes for that vertex and all its neighbors. For that reason, being K the number of iterations of the algorithm and $deg(G)$ the average degree of the graph, the expected complexity will be $O(K \cdot deg(G))$. Furthermore, the worst-case complexity is $O(K \cdot |V|)$, for $|V|$ the number of vertices in the graph.

After running the experiments, and as expected, we have observed that the computation time increases linearly with the number K of iterations. For example, using an Intel Core i7-2600 with 8 CPUs@3.40GHz and 8 GB RAM running Ubuntu 14.04.4 LTS, we have measured that the mean time required to run 1,000 iterations of SA is 6.84 s with a 95% confidence interval of 0.24 s.

For $\delta = 0$ (i.e., when crossing routes can use the same FL), airplane behavior is independent, as all the routes can use their optimal FL independently from the rest of routes. Although this scenario with $\delta = 0$ does not represent a socially acceptable approach, due to its inherent risk, it is interesting as a baseline for three reasons. First, it will provide the minimum total fuel consumption, as each route can use its optimal FL. Second, as each flight is independent from the others, it will allow us to compute that optimum value exactly. Third, as for this scenario we are able to know the optimum value, we can test the validity of the proposed algorithms comparing their solutions with that optimum value.

Thus, for $\delta = 0$, Fig. 5.6 shows the evolution of the four algorithms studied, as a function of the number of iterations. For each algorithm, we show both the mean and the 95% confidence intervals of the results. The main conclusion from this figure is that all four approaches converge to the computed optimum fuel consumption, the value of which is 2,366 tons of fuel. It must be noted that the convergence to the optimum of Global approaches is slower than that of Greedy approaches. Moreover, it is also interesting to note that there are no statistically significant differences between the results of SA and HC, telling us that SA is not able to take advantage from its ability to escape from local optima, suggesting that the state space is not highly rugged but instead has a monotonic structure.

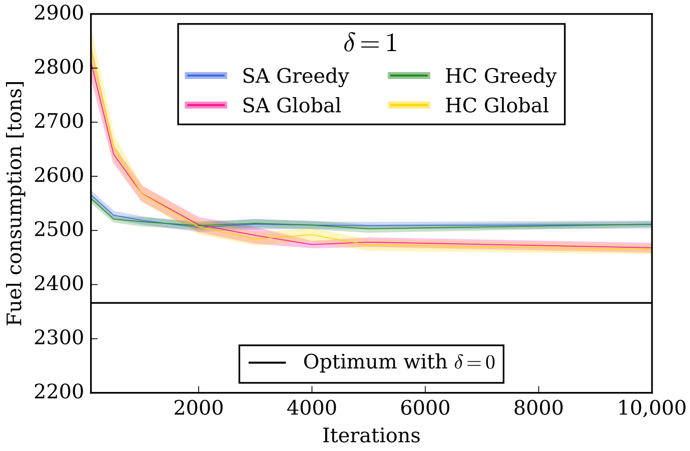
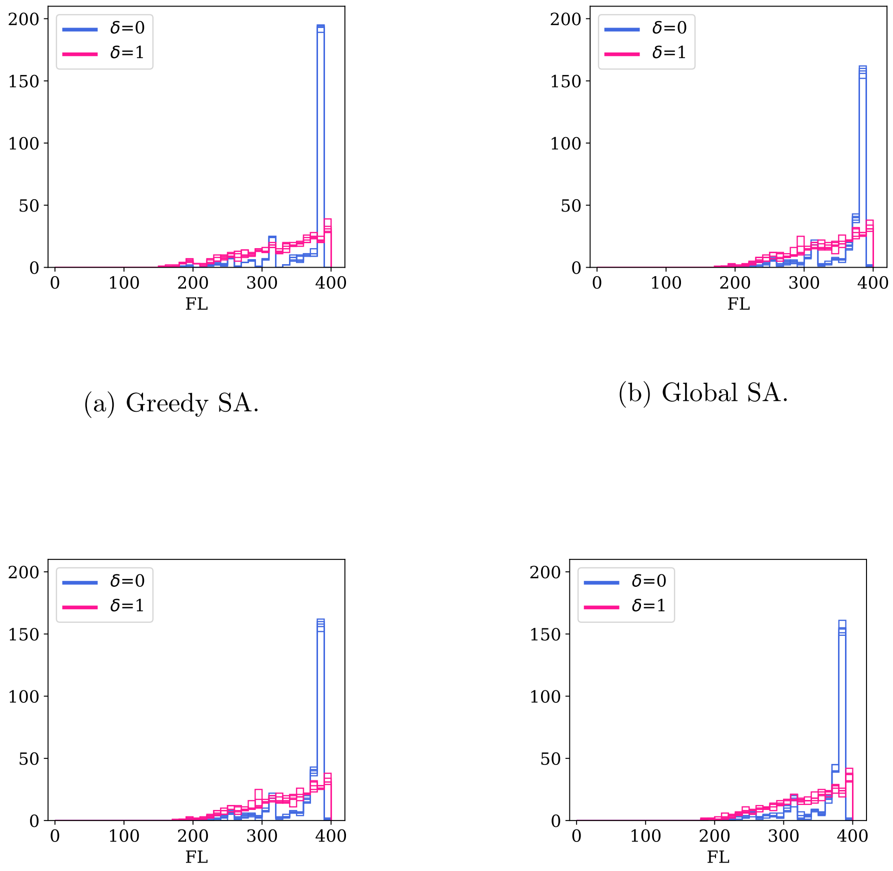


Figure 5.7: Total fuel consumption for the different techniques and $\delta = 1$.

Figure 5.7 shows the results for the case $\delta = 1$. This is a much more realistic setting than $\delta = 0$, since crossing routes are now restricted not to use the same FL, i.e., these solutions do consider the social safety goal. Note that this figure still shows the optimum fuel consumption for $\delta = 0$, as a baseline and since the complexity of the case $\delta = 1$ (due to the interdependences among routes) makes unfeasible to compute the optimum fuel consumption in this case. First of all, we can observe that, as expected, solutions with $\delta = 1$ are not able to reach the optimum fuel consumption of the case $\delta = 0$, because when $\delta = 1$ not all flights are able to use their optimum FL to minimize fuel consumption. However, from Fig. 5.7 we can conclude that the cost of forbidding crossing routes to use the same FL is around a 5% in fuel consumption. More specifically, this loss is 4% for the Global approaches and 6% for the Greedy ones. Comparing the different approaches, we see again that the performance of HC and SA is fairly similar and that the solutions obtained by the Global algorithms are better, with a lower fuel consumption, but they need more iterations to reach those improvements. It is interesting to observe that, for both $\delta = 0$ and $\delta = 1$, the Greedy approaches reach reasonably good solutions in a much shorter time than the Global approaches. For $\delta = 2$, our experiments have shown that the complexity of the graph does not allow to obtain feasible colorings, i.e., it is unfeasible to obtain FL assignments that ensure a separation of two flight levels when a pair of routes cross.

Thus, we move the focus from the total fuel consumption to an in-depth analysis of the assignment of flight levels obtained by the different algorithms. In Fig. 5.8 we show how FLs are assigned to routes. In all cases, we have considered the results obtained after 10,000 iterations of each algorithm. Moreover, although we ran 20 experiments, for the sake of clarity and without loss of generality, we only show the results of the first five of them. More specifically, the figure shows the number of times each flight level is used, for the cases $\delta = 0$ and $\delta = 1$. It is interesting to note that for $\delta = 1$ the FLs are more evenly used than for $\delta = 0$, and it is worth highlighting the peak in FL380 for $\delta = 0$, where the Airbus A320 has its optimum for the cruise (recall Table 5.2). Moreover, it is also interesting to note that, although for $\delta = 0$ all the algorithms converge to the optimal solution (as shown in Fig. 5.6), the number of times that FL380 is used turns out to be lower in the Global approaches.

Tables 5.3 and 5.4 deepen this analysis of the FL assignment provided by different techniques for $\delta = 0$ and $\delta = 1$, respectively, and using 10,000 iterations and 20 runs for each algorithm. First, the column labeled as 'FL use' describes the average number of times that each FL in use is used. For example, a value of 288 stands for the fact that flight levels in use are used, on average, 288 times each. The standard deviation of this value is key to understand if FLs are evenly used or not. Analyzing this performance parameter in both tables, we conclude that FLs are re-used less times in $\delta = 1$ and, as the standard deviation is much lower, the use of FLs is more even in $\delta = 1$, as it was also expected from Fig. 5.8. The column labeled



(a) Greedy SA.

(b) Global SA.

(c) Greedy HC.

(d) Global HC.

Figure 5.8: Distribution of FLs for different optimization methods.

as ‘FLs used’ in Tables 5.3 and 5.4 represents the number of FLs used at least once in the solutions provided by the different techniques. We can observe that the mean number of FLs used is clearly higher in $\delta = 1$ than in $\delta = 0$. Finally, the column labeled as ‘FL separation’ represents the separation between crossing routes measured in FLs. For example, if two routes cross each other, one is assigned FL350, and the other FL370, then their separation is 2 FLs. This performance parameter is key for improving

Table 5.3: Analysis of the FLs used for the different algorithms and $\delta = 0$.

	FL use		**FLs used**		**FL separation**	
	Mean	std dev	Mean	std dev	Mean	std dev
SA Global	261.82	652.78	19.30	0.90	1.37	0.03
SA Greedy	288.00	825.16	16.70	1.05	1.14	0.01
HC Global	250.43	629.56	19.35	0.96	1.38	0.04
HC Greedy	320.00	871.28	15.45	0.59	1.15	0.01

Table 5.4: Analysis of the FLs used for the different algorithms and $\delta = 1$.

	FL use		**FLs used**		**FL separation**	
	Mean	std dev	Mean	std dev	Mean	std dev
SA Global	250.43	183.46	21.55	0.59	6.03	0.35
SA Greedy	221.54	168.50	23.60	1.16	7.61	0.27
HC Global	261.82	178.75	21.60	0.58	6.14	0.22
HC Greedy	213.33	170.51	23.75	0.89	7.59	0.35

safety. The differences in this parameter between $\delta = 0$ and $\delta = 1$ are clear, as the mean separation between intersecting routes in $\delta = 0$ (resp. $\delta = 1$) is around 1.2 (resp. 7) FLs. It is worth noting that, for the most interesting setting $\delta = 1$, although the fuel consumption of the Greedy approaches is slightly higher, the separation between FLs of crossing routes is also higher, which again confirms the trade-off between safety and fuel consumption.

5.4 Discussion

We study the problem of behavior optimization in the context of flight level assignment. Apart from being a relevant problem due to the expected growth (both in size and complexity) of the air traffic networks, it is very interesting due to the existence of different individual and social goals: the incentive for fuel consumption minimization by the flight operators, and the safety constraints demanded by regulators and by society as a whole.

In this paper, we use a spectrum graph coloring model for behavior optimization, which relies on a factorized optimization model and realistic utility models for the individual and social goals. After modeling the problem, we propose an adaptation of hill-climber and simulated annealing techniques, including some versions which are more greedy than the ones we had used before, and we evaluate their relative performance in a setting representing the Spanish air space. The experiments show that there is an inherent trade-off between fuel consumption and safety, and that the proposed techniques are suitable for finding flight level assignments that take into account both individual and social goals. The results also reveal that the greedy versions of the algorithms are able to reach acceptable solutions in much less time, which enables an additional trade-off between fuel consumption and response time, which can be very significant for real-time flight level assignment. This motivates us to pursue as our immediate future work a real-time operation model, with flight level assignments dynamically depending on the actual position of aircrafts within the routes in real time. In the same line, we will study the performance of message-passing optimization algorithms like belief propagation [78], which have demonstrated a better real-time response. Moreover, the study of message-passing optimization algorithms would be especially suitable if we assume distributed settings where there are a number of agents (for example, airlines) that compete to achieve optimum FLs. We are also interested in introducing risk aversion as a parameter in the utility model, since recent history has shown that different airlines follow very different policies in this regard. Finally, we plan to apply the proposed techniques in other research problems. One of such problems is the management of routes for unmanned aircrafts or drone operations in urban areas [126]. Another important problem that we plan to explore with these techniques is the reduction of pollution in cities by means of ride-sharing [55].

Chapter 6

Conclusions

First, we obtained an algorithm to compute and maintain $\mathcal{O}$-Kernel(P) as the set of orientations $\mathcal{O}$ rotates, providing the angle intervals such that $\mathcal{O}$-Kernel(P) has maximum area and perimeter. For the remarkable case of P being a simple orthogonal polygon, we improved those algorithms.

Second, we studied a problem related to the higher order Voronoi diagrams of a point set. These diagrams were used by Edelsbrunner, Hasan, Seidel and Shen to prove the following classic theorem in combinatorial geometry [32]. Every set of n points in the plane contains two points such that all the circles having those two points in their boundary contain at least $n(\frac{1}{2} - \frac{1}{\sqrt{12}}) + O(1)$ points in their interior. We continued using higher order Voronoi diagrams to prove the following bichromatic version of this result. Every set of n red and n blue points in the plane contains one red and one blue point such that every circle through them encloses at least $n(1 - \frac{1}{\sqrt{2}}) - o(n)$ points of the set. Our theorem improves the previous results on this bichromatic problem, due to Prodromou [107].

Next, we applied graph theory to two real-life problems. In the first problem, we introduced a new method for sensory analysis, combining techniques from statistics and geometry. We compared the performance of this method with the performance of classic statistical methods in sensory analysis over the same data set. Our conclusion is that geometric and mixed methods provided similar results than those obtained with pure statistical techniques. Some small differences in the results were appreciated, showing that these new methods can provide useful complementary information to statistical analysis.

Finally, the problem of air traffic management was a challenging framework for behaviour optimization, combining social and individual goals. We modeled this problem using the technique of spectrum graph coloring, that penalizes aircrafts from flying close to each other through a matrix of interferences. We observed that air traffic management is an immensely complicated task, involving many factors, such as passengers safety, weather conditions, and airspace congestion. In our model, we simplified the problem by only considering safety and fuel consumption. In the future, we wish to explore the performance of this model in other cases of use, such as the case of drones operating in an urban area.

Chapter 7

Bibliography

[1] Aarts, E., Korst, J., Michiels, W., *Simulated annealing*, Search methodologies, Springer, Boston, MA, 187–210 2005.

[2] Abdi, H., *Congruence: congruence coefficient, RV-coefficient and Mantel coefficient*. In N. J. Salkind, D. M. Dougherty, B. Frey (Eds.). Encyclopedia of research design. Thousand Oaks, CA: Sage, 222—229, 2010.

[3] Ackermann, W., *Zum Hilbertschen Aufbau der reellen Zahlen*, Mathematische Annalen 99, 118–133, 1928.

[4] Airbus Customer Services. *Flight Operations Support & Line Assistance*, https://ansperformance.eu/library/airbus-cost-index.pdf, accessed September 2019.

[5] Akiyama, J., Ishigami, Y., Urabe, M., Urrutia, J., *On circles containing the maximum number of points*, Discrete Mathematics 151, 15–18, 1996.

[6] Akl, S.G., Toussaint, G.T., *Efficient convex hull algorithms for pattern recognition*, Proceedings of the 4-th International Joint Conference on Pattern Recognition, Kyoto, Japan, 483–487, 1978.

[7] Alizadeh, A., Uzun, M., Koyuncu, E., *Optimal trajectory planning based on wind-optimal cost index*, Aviation Technology, Integration, and Operations Conference, 2018.

[8] Alon, N., Gyori, E., *The number of small semi-spaces of a finite set of points in the plane*, Journal of Combinatorial Theory, Series A 41, 154–157, 1986.

[9] Ardila, F., *The number of halving circles*, American Mathematical Monthly 111 586–591, 2004.

[10] Aurenhamer, A., *Voronoi diagrams - A survey of a fundamental data structure*, ACM Computing Surveys 23(3), 345–405, 1991.

[11] Bárány, I., Schmerl, H, Sidney, S.J., Urrutia, J., *A combinatorial result about points and balls in Euclidean space*, Discrete and Computational Geometry 4, 259–262, 1989.

[12] Barber, C.B., Dobkin, D.P., Huhdanpaa, H., *The quickhull algorithm for convex hulls*, ACM Transactions on Mathematical Software 22 (4), 469–483, 1996.

[13] Barcenas, P., Elortondo, F. P., Albisu, M., *Projective mapping in sensory analysis of ewes milk cheeses: A study on consumers and trained panel performance*. Food Research International, 37(7), 723–729. https://doi.org/10.1016/j.foodres.2004.02.015, 2004.

[14] Barnier, N., Brisset, P., *Graph coloring for air traffic flow management*, Annals of Operations Research 130, 163–178, 2004.

[15] Barton, A., Hayward, L., Richardson, C. D., McSweeney, M. B., *Use of different panellists (experienced, trained, consumers and experts) and the projective mapping task to evaluate white wine*. Food Quality and Preference, 83, Article 103900. https://doi.org/10.1016/j.foodqual.2020.103900, 2020.

[16] de Berg, M., Cheong, O., van Kreveld, M., Overmars, M., *Computational geometry: algorithms and applications*. Springer, Berlin (2008).

[17] Blancher, G., Clavier, B., Egoroff, C., Duineveld, K., Parcon, J., *A method to investigate the stability of a sorting map*. Food Quality and Preference, 23, 36—43. https://doi.org/10.1016/j.foodqual.2011.06.010, 2012.

[18] Bondy, J.A., *Small cycle double covers of graphs*, Cycles and Rays, NATO ASI Series C (Kluwer), 21–40, 1990.

[19] Burzlaff, M., *Aircraft fuel consumption*, https://www.fzt.haw-hamburg.de/pers/Scholz/arbeiten/LandingBurzlaff.html, accessed February 2019.

[20] Cadoret, M., Husson, F., *Construction and evaluation of confidence ellipses applied at sensory data.* Food Quality and Preference, 28(1), 106—115. https://doi.org/10.1016/j.foodqual.2012.09.005, 2013.

[21] Calamoneri, T., *The $L(h,k)$-labelling problem: An updated survey and annotated bibliography*, The Computer Journal 54:8, 1344–1371, 2011.

[22] Camacho-Gómez, C., Marsa-Maestre, I., Gimenez-Guzman, J. M., Salcedo-Sanz, S., *A coral reefs optimization algorithm with substrate layer for robust Wi-Fi channel assignment*, Soft Computing 23, 12621–12640, 2019.

[23] Clarkson, K.L., Shor, P.W., *Applications of random sampling in computational geometry*, II, Discrete and Computational Geometry 4, 387–421, 1989.

[24] Claverol, M., de las Heras, A., Huemer, C., Martínez-Moraian, A., *The edge labeling of higher order Voronoi diagrams*, https://doi.org/10.48550/arXiv.2109.13002

[25] Constans, S., Gadenne, N., Fondacci, R., *Applying genetic techniques to the tactical flight level assignment*, 16th Mini-EURO Conference and 10th Meeting of EWGT Proceedings, 13–16, 2005.

[26] Csardi, G., *igraph.* http://igraph.org/r/, 2015. Accessed: 2019-12-18.

[27] Dalmau, R., Prats, X., *Fuel and time savings by flying continuous cruise climbs: Estimating the benefit pools for maximum range operations*, Transportation Research Part D: Transport and Environment 35, 62–71, 2015.

[28] dal Sasso, V., Djeumou, F., Lulli, G., *Zografos, K.G. Planning efficient 4D trajectories in Air Traffic Flow Management*, European Journal of Operational Research 276:2, 676–687, 2019.

[29] De la Hoz, E., Gimenez-Guzman, J. M., Marsa-Maestre, I., Orden, D., *Automated negotiation for resource assignment in wireless surveillance sensor networks*, Sensors 15(11), 29547-29568, 2015.

[30] de la Hoz, E., Marsa-Maestre, I., Gimenez-Guzman, J.M., Orden, D., Klein, M., *Multi-agent nonlinear negotiation for Wi-Fi channel assignment*, Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, 1035–1043, 2017.

[31] Dey, T., *Improved bounds for planar k-sets and related problems*, Discrete and Computational Geometry 19, 373–382, 1989.

[32] Edelsbrunner, H., Hasan, N., Seidel, R., Shen, X., *Circles through two points that always enclose many points*, Geometriae Dedicata 32, 1–12, 1989.

[33] Erdős, P., and Guy, R., *Crossing number problems*, The American Mathematical Monthly 8, 52–58, 1973.

[34] Erdős, P., Szekeres, G., *A combinatorial problem in geometry*, Compositio Mathematica 2, 463–470, 1935.

[35] Escoufier, Y., *Le traitement des variables vectorielles.* Biometrics, 29(4), 751—760, 1973.

[36] Escofier, B., Pagès, J., *Analyses factorielles simples et multiples; objectifs, méthodes et interprétation*, Dunod, Paris, 2008.

[37] Faye, P., Brémaud, D., Teillet, E., Courcoux, P., Giboreau, A., Nicod, H., *An alternative to external preference mapping based on consumer perceptive mapping.* Food Quality and Preference, 17(7–8), 604—614. https://doi.org/10.1016/j.foodqual.2006.05.006, 2006.

[38] Fink, E., Wood, D., *Fundamentals of restricted-orientation convexity*, Information Sciences, 92(1–4), 175–196, 1996.

[39] Gabriel, K. R., Sokal, R. R., *A new statistical approach to geographic variation analysis.* Systematic Biology, 18(3), 259—278. https://doi.org/10.2307/2412323, 1969.

[40] Geem, Z.W., Kim, J.H., Loganathan, G.V, *A new heuristic optimization algorithm: Harmony search*, Simulation 76, 60–68, 2001.

[41] Gewali, L.P., *Recognizing s-star polygons*, Pattern Recognition 28(7), 1019–1032, 1995.

[42] Gilbert, J. A., Miller, D., Olson, S., St-Pierre, S., *After-school snack intake among Canadian children and adolescents*. Canadian Journal of Public Health, 103(6), e448-–e452. https://doi.org/10.1007/BF03405636, 2012.

[43] Gimenez-Guzman, J.M., Marsa-Maestre, I., Orden, D., de la Hoz, E., Ito, T., *On the goodness of using orthogonal channels in WLAN IEEE 802.11 in realistic scenarios*, Wireless Communications and Mobile Computing, 5742712, 2018.

[44] Goodman, J.E., O'Rourke, J., Tóth, C.D., *Handbook of discrete and computational geometry*, CRC Press LLC, Boca Raton, 1997.

[45] Graham, R.L., *An efficient algorithm for determining the convex hull of a finite planar set*, Information Processing Letters 1(4), 132–133, 1972.

[46] Gross, J. L., Yellen, J., Zhang, P., *Handbook of graph theory*. Chapman and Hall/ CRC ISBN 9781439880180, 2013.

[47] Güting, R.H., *Stabbing C-oriented polygons*, Information Processing Letters 16, 35–40, 1983.

[48] Härdle, W., Simar, L., *Applied multivariate statistical analysis*. Berlin: Springer., 1051-–8215 https://doi.org/10.1007/978-3-662-45171-7, 2007.

[49] Hayward, R., *A note on the circle containment problem*, Discrete and Computational Geometry 4, 263–264, 1989.

[50] Hayward, R., Rappaport, D., Wenger, R., *Some extremal results on circles containing points*, Discrete and Computational Geometry 4, 253–258, 1989.

[51] Hershberger, J., *Finding the upper envelope of n line segments in $O(n \log n)$ time* Information Processing Letters 33(4), 169–174, 1989.

[52] Huskić, G., Buck, S., Zell, A., *GeRoNa: Generic robot navigation*, Journal of Intelligent & Robotic Systems 95(2), 419–442, 2019.

[53] Icking, C., Klein, R., *Searching for the kernel of a polygon: A competitive strategy*, Proceedings of the 11th Annual Symposium on Computational Geometry, 258–266, 1995.

[54] Jaeger, F., *A survey on the cycle double cover conjecture*, Annals of Discrete Mathematics 27, North-Holland, 1–12, 1985.

[55] Jain, S., Biyani, P., *Improved real time ride sharing via graph coloring*, 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019.

[56] Jansen, P.W., Perez, R.E., *Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach*, Computers & Structures 89(13-14), 1352–1366, 2011.

[57] Jarrar, A., Bakouki, Y., *Formal modeling of a complex adaptive air traffic control system*, Complex Adaptive Systems Modeling volume 6, 6, 2018.

[58] Jarvis, R.A., *On the identification of the convex hull of a finite set of points in the plane*, Information Processing Letters 2, 18–21, 1973.

[59] Jian, B., Sum, Z., Anderson, B.D.O., Lageman, C., *Higher order mobile coverage control with applications to clustering of discrete sets*, Automatica 102, 27–32, 2019.

[60] Josse, J., Holmes, S., *Measuring multivariate association and beyond.* Statistics surveys, 10, 132. https://doi.org/10.1214/16-SS116, 2016.

[61] Kamada, T., Kawai, S., *An algorithm for drawing general undirected graphs.* Information Processing Letters, 31(1), 7—15. https://doi.org/10.1016/0020-0190(89)90102-6, 1989.

[62] Ke, Y., O'Rourke, J., *Computing the kernel of a point set in a polygon*, Lecture Notes in Computer Science 382, 135–146, 1989.

[63] Kirkpatrick, D.G., Snoeyink, J., *Computing common tangents without a separating line*, Proceedings of the 4th International Workshop on Algorithms and Data Structures, 183-193, 1995.

[64] Klein, M., Faratin, P., Sayama, H., Bar-Yam, Y., *Negotiating complex contracts*, Group Decision and Negotiation 12, 111–125, 2003.

[65] Krussell, J.W., Schaudt, B.F., *A note on higher order Voronoi diagrams*, Nordic Journal of Computing 1, 268–272, 1994.

[66] Lahne, J., *Sorting Backbone Analysis: A network-based method of extracting key actionable information from free-sorting task results.* Food Quality and Preference, 103870. https://doi.org/10.1016/j.foodqual.2020.103870, 2020.

[67] Lang, F., Fink, A., *Learning from the metaheuristics: Protocols for automated negotiations.* Group Decision and Negotiation 24, 299–332, 2015.

[68] Lawless, H. T., Heymann, H., *Sensory evaluation of food: Principles and practices.* Springer Science and Business Media. https://doi.org/10.1007/978-1-4419-6488-5, 2010.

[69] Lê, S., Husson, F., *SensoMineR: A package for sensory data analysis.* Journal of Sensory Studies, 23(1), 14—25. https://doi.org/10.1111/j.1745-459X.2007.00137.x, 2008.

[70] Lê, S., Josse, J., Husson, F., *FactoMineR: An R package for multivariate analysis.* Journal of Statistical Software, 25(1), 1—18. https://doi.org/10.18637/jss.v025.i01, 2008.

[71] Lee, D.T., *On k-nearest neighbor Voronoi diagrams in the plane*, IEEE Transactions on Computer Science 31, 478–487, 1982.

[72] Lindenbergh, R.C., *A Voronoi poset*, Journal for Geometry and Graphs 7, 41–52, 2003.

[73] Lovegren, J.A., Hansman, R.J., *Estimation of potential aircraft fuel burn reduction in cruise via speed and altitude optimization strategies*, MIT International Center for Air Transportation, Cambridge, Massachussetts, USA, 2011, accessed April 2020.

[74] Malaguti, E., Toth, P., *A survey on vertex coloring problems*, International Transactions on Operational Research 17, 1–34, 2010.

[75] Mantel, N., *The detection of disease clustering and a generalized regression approach.* Cancer Research, 27, 209–220, 1967.

[76] Marchette, D. J., *cccd, Class Cover Catch Digraphs.* https://cran.r-project.org/package=cccd, 2015. Accessed: 2019-12-18.

[77] Marsa-Maestre, I., de la Hoz, E., Gimenez-Guzman, J.M., Orden, D., Klein, M., *Nonlinear negotiation approaches for complex-network optimization: A study inspired by Wi-Fi channel assignment*, Group Decision and Negotiation 28, 175–196, 2019.

[78] Marsa-Maestre, I., Gimenez-Guzman, J.M., de la Hoz, E., Orden, D., *competitive belief propagation to efficiently solve complex multi-agent negotiations with network structure*, International Conference on Autonomous Agents and Multiagent Systems, 1–16, 2017.

[79] Marsa-Maestre, I., Gimenez-Guzman, J.M., Orden, D., de la Hoz, E., Klein, *M. REACT: REActive resilience for critical infrastructures using graph-coloring techniques*, Journal of Network and Computer Applications 145, 102402, 2019.

[80] Martínez-Moraian, A., Orden, D., *Book of Abstracts of the XIX Spanish Meeting on Computational Geometry*, 31–35, 2021.

[81] Martínez-Moraian, A., Orden, D., Gimenez-Guzman, J.M., Marsa-Maestre, I., *Using spectrum graph coloring in air traffic management*, XVIII Spanish Meeting on Computational Geometry, 2019.

[82] McCallum, D., Avis, D., *A linear algorithm for finding the convex hull of a simple polygon*, Information Processing Letters 9(5), 201-–206, 1979.

[83] Melkman, A., *On-line construction of the convex hull of a simple polygon*, Information Processing Letters 25, 11–12, 1987.

[84] Ministerio de Agricultura, Pesca y Alimentación, *Informe del consumo alimentario en España 2018.* https://www.mapa.gob.es/gl/alimentacion/temas/consumo-y-comercializacion-y-distribucion-alimentaria/panel-de-consumo-alimentario/ultimos-datos/, 2019. Accessed: 2019-12-02.

[85] Mirosavljevic, P., Gvozdenovic, S., Vasov, L., *The transport aircraft minimum pollution climb schedule*, FME Transactions 41, 11–24, 2013.

[86] Moelich, E. I., Muller, M., Joubert, E., Næs, T., Kidd, M., *Validation of projective mapping as potential sensory screening tool for application by the honeybush herbal tea industry.* Food Research International,

99, 275—286. https://doi.org/10.1016/j.foodres.2017.05.014, 2017.

[87] Moussaoui, K. A., Varela, P., *Exploring consumer product profiling techniques and their linkage to a quantitative descriptive analysis*. Food Quality and Preference, 21(8), 1088—1099. https://doi.org/10.1016/j.foodqual.2010.09.005, 2010.

[88] Næs, T., Berget, I., Liland, K. H., Ares, G., Varela, P., *Estimating and interpreting more than two consensus components in projective mapping: INDSCAL vs. multiple factor analysis (MFA)*. Food Quality and Preference, 58, 45—60. https://doi.org/10.1016/j.foodqual.2016.11.012, 2017.

[89] Nestrud, M. A., Lawless, H. T., *Perceptual mapping of citrus juices using-projective mapping and profiling data from culinary professionals and consumers*. Food Quality and Preference, 19(4), 431-–438. https://doi.org/10.1016/j.foodqual.2008.01.001, 2008.

[90] Nestrud, M. A., Lawless, H. T., *Perceptual mapping of apples and chesses using projective mapping and sorting*. Journal of Sensory Studies, 25, 390—405. https://doi.org/10.1111/j.1745-459X.2009.00266.x, 2010.

[91] NetworkX – Network Analysis in Python, https://networkx.github.io/, accessed August 2020.

[92] Neumann-Lara, V., Urrutia, J., *A combinatorial result on points and circles on the plane*, Discrete Mathematics 69, 173–178, 1988.

[93] OEIS Foundation Inc. (2021), The On-Line Encyclopedia of Integer Sequences, http://oeis.org/A096338, accessed: January 2021.

[94] Okabe, A., Boots, B.,. Sugihara, K., *Spatial tessellations: concepts and applications of Voronoi diagrams*, Wiley, 1992.

[95] Orden, D., Fernández-Fernández, E., *Data for: Geometric and statistical techniques for projective mapping of chocolate chip cookies with a large number of consumers*, Mendeley Data, v1. http://dx.doi.org/10.17632/msm2j99rcm.1, 2020.

[96] Orden, D., Tejedor-Romero, M., *SensoGraph (under registration process)*. https://sensograph.it, 2019. Accessed: 2019-12-17.

[97] Orden, D., Fernández-Fernández, E., Rodríguez-Nogales, J. M., Vila-Crespo, J., *Testing SensoGraph, a geometric approach for fast sensory evaluation*, Food Quality and Preference 72, 1–9, 2019.

[98] Orden, D., Gimenez-Guzman, J.M., Marsa-Maestre, I., de la Hoz, E., *Spectrum graph coloring and applications to WiFi channel assignment*, Symmetry 10(3), 65, 2018.

[99] Pagès, J., *Collection and analysis of perceived product inter-distances using multiple factor analysis: Application to the study of 10 white wines from the Loire Valley*. Food Quality and Preference, 16(7), 642—649. https://doi.org/10.1016/j.foodqual.2005.01.006, 2005.

[100] Palios, L., *An output-sensitive algorithm for computing the s-kernel*, 27th Canadian Conference on Computational Geometry, 199–204, 2015.

[101] Palios, L., *A new competitive strategy for reaching the kernel of an unknown polygon*, Lecture Notes in Computer Science 1851, 367–382, 2000.

[102] Patokallio, J., OpenFlights Data, 2014. http://openflights.org/data.html, accessed November 2019.

[103] Peck, G.W., *On k-sets in the plane*, Discrete Mathematics 56, 73–74, 1985.

[104] Poli, R., Kennedy, J., Blackwell, T., *Particle swarm optimization*, Swarm intelligence 1(1), 33–57, 2007.

[105] Preparata, F., Lee, D.T., *An optimal algorithm for finding the kernel of a polygon*, Journal of the ACM 26(3), 415–421, 1979.

[106] Preparata, F., Shamos, M.I., *Computational geometry: an introduction*, Springer, 1985.

[107] Prodromou, M.N., *A combinatorial property of points and balls, a colored version*, Discrete and Computational Geometry 38, 641–650, 2007.

[108] Qiu, C., Shen, H., Chen, K., *An energy-efficient and distributed cooperation mechanism for k-coverage hole detection and healing in WSNs*, IEEE Transactions on Mobile Computing 17(6), 1247–1259, 2018.

[109] R Development Core Team, *R: A language and environment for statistical computing.* Available at http://www.R-project.org, 2007.

[110] Ramos, P.A., Viaña, R., *Depth of segments and circles through points enclosing many points: a note*, Computational Geometry Theory and Applications 42, 338–341, 2009.

[111] Rawlins, G., *Explorations in restricted-orientation geometry*, PhD thesis, University of Waterloo, Canada, 1987.

[112] Rawlins, G., Wood, D., *Computational geometry with restricted orientations*, System Modelling and Optimization, 375–384, 2006.

[113] Risvik, E., McEwan, J. A., Colwill, J. S., Rogers, R., Lyon, D. H., *Projective mapping: A tool for sensory analysis and consumer research*, Food Quality and Preference 5(4), 263–269, 1994.

[114] Rosenfeld, A., *Picture processing by computers*, Academic Press, New York, 1969.

[115] Salcedo-Sanz, S., Del Ser, J., Landa-Torres, I., Gil-López, S., Portilla-Figueras, J.A., *The coral reefs optimization algorithm: a novel metaheuristic for efficiently solving optimization problems*, The Scientific World Journal, 739768, 2014.

[116] Savidan, C. H., Morris, C., *Panelists' Performances and Strategies in Paper-Based and Computer-Based Projective Mapping.* Journal of Sensory Studies, 30(2), 145—155. https://doi.org/10.1111/joss.12146, 2015.

[117] Schuierer, S., Rawlins, G.J.E., Wood, D., *A generalization of staircase visibility*, 3rd Canadian Conference on Computational Geometry, 96–99, 1991

[118] Schuierer, S., Wood, D., *Generalized kernels of polygons with holes*, 5th Canadian Conference on Computational Geometry, 222–227, 1993.

[119] Schuierer, S., Wood, D., *Multiple-guard kernels of simple polygons*, Theoretical Computer Science Center Research Report HKUST-TCSC-98-06, 1998.

[120] Shamos, M.I., Hoey, D., *Geometric intersection problems*, 17th Annual Symposium on Foundations of Computer Science, 208–215, 1976.

[121] Shao, J., Tu, D., *The jackknife and bootstrap*, New York: Springer-Verlag. https://doi.org/10.1007/978-1-4612-0795-5, 1995.

[122] Smorodinsky, S., Sulovský, M., Wagner, U., *On center regions and balls containing many points*, 14th Annual International Computing and Combinatorics Conference, Lecture Notes on Computer Science 5092, Springer, 363–373, 2008.

[123] Society of Sensory Professionals, *Projective Mapping*, https://www.sensorysociety.org/knowledge/sspwiki/Pages/Projective%20Mapping.aspx, accessed March 2022.

[124] Steinegger, R., *Fuel economy as function of weight and distance*, https://digitalcollection.zhaw.ch/bitstream/11475/1896/2/Steinegger_Fuel_Economy_for_Aircraft_Operation_as_a_Function_of_Weight_and_Distance_v1-0.pdf, accessed April 2020.

[125] Sugihara, K., Smith, J., *Genetic algorithms for adaptive motion planning of an autonomous mobile robot*, Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, 138–143, 1997.

[126] Tan, Q., Wang, Z., Ong, Y., *Low, K.H. evolutionary optimization-based mission planning for UAS traffic management (UTM)*, International Conference on Unmanned Aircraft Systems (ICUAS), 952-958, 2019.

[127] Tarrega, A., Marcano, J., Fiszman, S., *Consumer perceptions of indulgence: A case study with cookies*. Food Quality and Preference, 62, 80—89. https://doi.org/10.1016/j.foodqual.2017.07.001, 2017.

[128] Teillet, E., Schlich, P., Urbano, C., Cordelle, S., Guichard, E., *Sensory methodologies and the taste of water*. Food Quality and Preference, 21(8), 967—976. https://doi.org/10.1016/j.foodqual.2010.04.012, 2010.

[129] The Pandas Development Team, *pandas-dev/pandas: Pandas*, https://doi.org/10.5281/zenodo.3509134, 2020.

[130] The scikit-bio development team, *scikit-bio: A Bioinformatics Library for Data Scientists, Students, and Developers*, http://scikit-bio.org, 2020.

[131] Tomic, O., Berget, I., Næs, T., *A comparison of generalised procrustes analysis and multiple factor analysis for projective mapping data*. Food Quality and Preference, 43, 34—46. https://doi.org/10.1016/j.foodqual.2015.02.004, 2015.

[132] Turgut, E.T., Usanmaz, O., Cavcar, M., Dogeroglu, T.; Armutlu, K. *Effects of descent flight-path angle on fuel consumption of commercial aircraft*, Journal of Aircraft 56(1), 313–323, 2018.

[133] Tuza, Z., *Handbook of Graph Theory*, Discrete Mathematics and Its Applications 25, CRC Press, 408–438, 2003.

[134] Urrutia, J., *Open problems in computational geometry*, Latin American Symposium on Theoretical Informatics LATIN 2002, Lecture Notes on Computer Science 2286, Springer, Berlin, Heidelberg, 4–11, 2002.

[135] Valentin, D., Cholet, S., Nestrud, M., Abdi, H., *Projective mapping and sorting tasks*. In J. Hort, S. Kemp, & T. Hollowood (Eds.). Descriptive Analysis in Sensory Evaluation, Wiley-Blackwell, 1—19 2016.

[136] Varela, P., Ares, G., *Sensory profiling, the blurred line between sensory and consumer science. A review of novel methods for product characterization*. Food Research International, 48(2), 893—908. https://doi.org/10.1016/j.foodres.2012.06.037, 2012.

[137] Varela, P., Ares, G., *Novel techniques in sensory characterization and consumer profiling*. CRC Press, http://refhub.elsevier.com/S0950-3293(20)30337-2/h0255, 2014.

[138] Veinand, B., Godefoy, C., Adam, C., Delarue, J., *Highlight of important product characteristics for consumers. Comparison of three sensory descriptive methods performed by consumers*. Food Quality and Preference, 22(5), 474—485. https://doi.org/10.1016/j.foodqual.2011.02.011, 2011.

[139] Vidal, L., Cadena, R. S., Antúnez, L., Giménez, A., Varela, P., Ares, G., *Stability of sample configurations from projective mapping: How many consumers are necessary?* Food Quality and Preference, 34, 79—87. https://doi.org/10.1016/j.foodqual.2013.12.006, 2014.

[140] Vidal, L., Jaeger, S. R., Antúnez, L., Giménez, A., Ares, G., *Product spaces derived from projective mapping and CATA questions: Influence*

of replicated assessments and increased number of study participants. Journal of Sensory Studies, 31, 373—381. https://doi.org/10.1111/joss.12220, 2016.

[141] Vogtenhuber, B., *Combinatorial aspects of [colored] point sets in the plane*, PhD thesis, Institute for Software Technology, Graz University of Technology, Graz, Austria, 2011.

[142] Vossen, T.W.M., Hoffman, R., Mukherjee, A., *Air traffic flow management*, Quantitative Problem Solving Methods in The Airline Industry: A Modeling Methodology Handbook, Springer Science + Business Media, Berlin/Heidelberg, Germany, 2012.

[143] YouGov, *Italian cuisine is world's most popular*. https://yougov.co.uk/topics/food/articles-reports/2019/03/12/italian-cuisine-worlds-most-popular, 2019. Accessed: 2019-12-18.

Appendix A

Further lines of research

In this appendix we describe our current lines of research in the fields of restricted orientation convexity and higher order voronoi diagrams, respectively. Our current line of research in restricted orientation convexity focuses on orthoconvexity. In particular, we formulate and partially solve an Erdős-Szekeres problem; see Section A.1. Our current line of research on the topic of higher order Voronoi diagrams is based on a labeling of their edges. Using that labeling, we make a review of known properties of these diagrams with new proofs, and we extend this list of properties with new results; see Section A.2.

A.1 An orthogonal Erdős-Szekeres problem

The work that we describe in this section can be found in the Book of Abstracts of the XIX Spanish Meeting on Computational Geometry and was written in collaboration with David Orden [80].

Classic Erdős-Szekeres problems deal with the existence and number of convex k-gons determined by a set of points in the plane. The original problem, formulated by Klein, and answered positively by Erdős and Szekeres, states: *For every $k \in \mathbb{N}$, can we find a number $N(k)$ such that every set of $N(k)$ points in the plane contains at least one k-gon?* [34]. Provided the existence of convex k-gons in a sufficiently large point set, the following question was asked by Erdős and Guy: *What is the smallest number of convex k-gons determined by any set of n points in the plane?* [33].

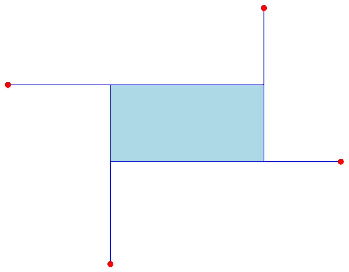


Figure A.1: A connected rectilinear convex 4-gon.

In [80], we formulate an Erdős-Szekeres problem for sets of points in the plane replacing usual convexity by rectilinear convexity.

We observe that the minimum number of (connected) rectilinear convex k-gons that can be found in a set of n points in the plane is zero. Therefore, we address the problem of maximizing the number of connected rectilinear convex k-gons instead. In particular, we deal with the case $k = 4$; see Figure A.1. We provide a lower bound for this problem by counting the exact number of connected rectilinear 4-gons determined by the vertices of a (carefully positioned) regular polygon.

Precisely, we prove that the maximum number of such 4-gons in a set of n points is at least $A(n-3)$, where $A(n)$ is the sequence A096338 in the OEIS [93]. We conjecture that this is the exact maximum number of such 4-gons in a set of n points based on computer simulation.

A.2 Properties of higher order Voronoi diagrams

The work that we describe in this section is available as the arXiv preprint arXiv:2109.13002, and was writen in colaboration with Mercè Claverol, Andrea de las Heras Parrilla, and Clemens Huemer [24]. It has also been sent to a journal, where it is currently under review.

In this work, we define an edge labeling of order-k Voronoi diagrams of sets of n points in the plane. From this labeling, we derive new properties about these diagrams.

Let $V_k(S)$ be the order-k Voronoi diagram of the point set S. Among the new properties of $V_k(S)$ that we obtain in [24], we highlight the property that $V_k(S)$ admits a small orientable cycle and path double cover of its edges.

A cycle double cover of a graph G is a collection of cycles $\mathcal{C}$ with the property that every edge of G belongs to exactly two cycles in $\mathcal{C}$. The *small double cover conjecture* states that any bridgeless graph on n vertices has a cycle double cover $\mathcal{C}$ with $|\mathcal{C}| \leq n-1$ [18]. For the case of order-k Voronoi diagrams, we construct a cycle and path double cover of the edges of $V_k(S)$ with a much smaller number of cycles compared to the number of vertices of $V_k(S)$. Note that paths are necessary in this double cover because $V_k(S)$ contains unbounded edges. The double cover $\mathcal{C}$ that we construct is also orientable, which means that we can assign an orientation to every element in $\mathcal{C}$ in such a way that for every edge e of $V_k(S)$, the two cycles or paths that cover e have opposite orientations [54].

Among the new properties that we obtain in [24], we also highlight the properties determining forbidden configurations of vertices and edges in the faces of $V_k(S)$ for small values of k. In addition, this work contains a review of well known properties of higher order Voronoi diagrams. For those properties, we obtain new proofs that only use simple arguments from planar geometry.

In the future, we aim to use this work to make progress in the monochromatic combinatorial problem described in Chapter 3.

Appendix B

Information about the articles

In this appendix we include the front page of the articles that form this thesis. All the articles have been published in peer-reviewed journals that appear in the Journal Citation Reports[1]: Journal of Global Optimization (Q1-T1), Discrete Mathematics (Q3-T2), Food Quality and Preference (Q1-T1), and Applied Sciences (Q2-T2).

[1] https://jcr.clarivate.com

Journal of Global Optimization
https://doi.org/10.1007/s10898-021-01020-3

Optimizing generalized kernels of polygons

Alejandra Martinez-Moraian[1] · David Orden[1] · Leonidas Palios[2] · Carlos Seara[3] · Paweł Żyliński[4]

Received: 31 July 2020 / Accepted: 27 March 2021

Abstract

Let $\mathcal{O}$ be a set of k orientations in the plane, and let P be a simple polygon in the plane. Given two points p, q inside P, we say that p $\mathcal{O}$-*sees* q if there is an $\mathcal{O}$-*staircase* contained in P that connects p and q. The $\mathcal{O}$-*Kernel* of the polygon P, denoted by $\mathcal{O}\text{-Kernel}(P)$, is the subset of points of P which $\mathcal{O}$-see all the other points in P. This work initiates the study of the computation and maintenance of $\mathcal{O}\text{-Kernel}(P)$ as we rotate the set $\mathcal{O}$ by an angle θ, denoted by $\mathcal{O}\text{-Kernel}_\theta(P)$. In particular, we consider the case when the set $\mathcal{O}$ is formed by either one or two orthogonal orientations, $\mathcal{O} = \{0°\}$ or $\mathcal{O} = \{0°, 90°\}$. For these cases and P being a simple polygon, we design efficient algorithms for computing the $\mathcal{O}\text{-Kernel}_\theta(P)$ while θ varies in $[-\frac{\pi}{2}, \frac{\pi}{2})$, obtaining: (i) the intervals of angle θ where $\mathcal{O}\text{-Kernel}_\theta(P)$ is not empty, (ii) a value of angle θ where $\mathcal{O}\text{-Kernel}_\theta(P)$ optimizes area or perimeter. Further, we show how the algorithms can be improved when P is a simple orthogonal polygon. In addition, our results are extended to the case of a set $\mathcal{O} = \{\alpha_1, \dots, \alpha_k\}$.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 734922.

✉ David Orden
david.orden@uah.es

Alejandra Martinez-Moraian
alejandra.martinezm@uah.es

Leonidas Palios
palios@cs.uoi.gr

Carlos Seara
carlos.seara@upc.edu

Paweł Żyliński
zylinski@inf.ug.edu.pl

[1] Departamento de Física y Matemáticas, Universidad de Alcalá, Madrid, Spain

[2] Deparment of Computer Science and Engineering, University of Ioannina, Ioannina, Greece

[3] Mathematics Department, Universitat Politècnica de Catalunya, Barcelona, Spain

[4] Institute of Informatics, University of Gdańsk, Gdańsk, Poland

Published online: 09 April 2021

Discrete Mathematics 344 (2021) 112541

Contents lists available at ScienceDirect

Discrete Mathematics

www.elsevier.com/locate/disc

Note

On circles enclosing many points

Mercè Claverol [a], Clemens Huemer [a,*], Alejandra Martínez-Moraian [b]

[a] Departament de Matemàtiques, Universitat Politècnica de Catalunya, Jordi Girona, 1-3, 08034 Barcelona, Spain
[b] Departamento de Física y Matemáticas, Universidad de Alcalá, Ctra. Madrid-Barcelona, Km. 33,600, 28871 Alcalá de Henares, Spain

ARTICLE INFO

Article history:
Received 19 August 2019
Received in revised form 30 June 2021
Accepted 1 July 2021
Available online 16 July 2021

Keywords:
Discrete geometry
Point set
Circle containment
Voronoi diagram

ABSTRACT

We prove that every set of n red and n blue points in the plane contains a red and a blue point such that every circle through them encloses at least $n(1-\frac{1}{\sqrt{2}})-o(n)$ points of the set. This is a two-colored version of a problem posed by Neumann-Lara and Urrutia. We also show that every set S of n points contains two points such that every circle passing through them encloses at most $\lfloor\frac{2n-1}{3}\rfloor$ points of S. The proofs make use of properties of higher order Voronoi diagrams, in the spirit of the work of Edelsbrunner, Hasan, Seidel and Shen on this topic. Closely related, we also study the number of collinear edges in higher order Voronoi diagrams and present several constructions.

1. Introduction

Neumann-Lara and Urrutia [14] posed the following problem: Prove that every set S of n points in the plane contains two points p and q such that any circle which passes through p and q encloses “many” points of S. The question is to quantify this number of enclosed points which can always be guaranteed. We only consider point sets without three collinear points and without four cocircular points. We say that such a point set is in *general position*. Almost all the results on this question date back to the late 1980’s. The first bound $\lceil\frac{n-2}{60}\rceil$, by Neumann-Lara and Urrutia [14], was improved in a series of papers [5,9,10]. The best bound was obtained by Edelsbrunner et al. [8] who proved that any set of n points in the plane contains two points such that any circle through those two points encloses at least $n(\frac{1}{2}-\frac{1}{\sqrt{12}})+O(1)\approx\frac{n}{4.7}$ points. Their proof makes use of properties of *higher order Voronoi diagrams*. The order-k Voronoi diagram of a point set S is a subdivision of the plane into convex regions, such that a region consists of those points of the plane which have the same k nearest points of S [4]. Twenty years later, Ramos and Viaña [17] made progress on that problem and proved a stronger statement: There is always a pair of points such that any circle through them has, both inside and outside, at least $n(\frac{1}{2}-\frac{1}{\sqrt{12}})+O(1)\approx\frac{n}{4.7}$ points. To prove their result, they transformed the problem from circles in the plane to planes in $\mathbb{R}^3$ and used results on the number of j-facets of point sets in $\mathbb{R}^3$.
The known upper bound on the number of enclosed points is $\lceil\frac{n}{4}\rceil-1$, due to a construction by Hayward et al. [10]. Urrutia [19] conjectured that $\frac{n}{4}\pm c$, for some constant c, is the tight bound. For sets of n points in convex position, a tight bound of $\lceil\frac{n}{3}\rceil-1$ is known [10].
Our contributions to this problem are the following ones:

- In Section 2 we present a modified and shorter version of the proof of Edelsbrunner et al. [8] which also leads to the result of Ramos and Viaña [17]. As in [8], the proof makes use of properties of higher order Voronoi diagrams.

[*] Corresponding author.
E-mail addresses: merce.claverol@upc.edu (M. Claverol), clemens.huemer@upc.edu (C. Huemer), alejandra.martinezm@uah.es (A. Martínez-Moraian).

https://doi.org/10.1016/j.disc.2021.112541
0012-365X/© 2021 Elsevier B.V. All rights reserved.

Food Quality and Preference 87 (2021) 104068

Contents lists available at ScienceDirect

Food Quality and Preference

journal homepage: www.elsevier.com/locate/foodqual

Geometric and statistical techniques for projective mapping of chocolate chip cookies with a large number of consumers

David Orden[a,*], Encarnación Fernández-Fernández[b], Marino Tejedor-Romero[a], Alejandra Martínez-Moraian[a]

[a] Departamento de Física y Matemáticas, Universidad de Alcalá, Campus Universitario, Ctra. Madrid-Barcelona, Km. 33.600, 28805 Alcalá de Henares, Spain

[b] Área de Tecnología de Alimentos, E.T.S. de Ingenierías Agrarias, Universidad de Valladolid, Campus La Yutera, Avda. Madrid 50, 34004 Palencia, Spain

ARTICLE INFO

Keywords:
Projective mapping
Chocolate chip cookies
Consumers
Multiple factor analysis
SensoGraph
Stability

ABSTRACT

The so-called rapid sensory methods have proved to be useful for the sensory study of foods by different types of panels, from trained assessors to unexperienced consumers. Data from these methods have been traditionally analyzed using statistical techniques, with some recent works proposing the use of geometric techniques and graph theory. The present work aims to deepen this line of research introducing a new method, mixing tools from statistics and graph theory, for the analysis of data from Projective Mapping. In addition, a large number of n = 349 unexperienced consumers is considered for the first time in Projective Mapping, evaluating nine commercial chocolate chips cookies which include a blind duplicate of a multinational best-selling brand and seven private labels. The data obtained are processed using the standard statistical technique Multiple Factor Analysis (MFA), the recently appeared geometric method SensoGraph using Gabriel clustering, and the novel variant introduced here which is based on the pairwise distances between samples. All methods provide the same groups of samples, with the blind duplicates appearing close together. Finally, the stability of the results is studied using bootstrapping and the RV and Mantel coefficients. The results suggest that, even for unexperienced consumers, highly stable results can be achieved for MFA and SensoGraph when considering a large enough number of assessors, around 200 for the consensus map of MFA or the global similarity matrix of SensoGraph.

1. Introduction

Traditionally, the sensory evaluation of foods has been carried out using generic descriptive analysis with trained panels (Lawless & Heymann, 2010) and consumer sensory analysis with a 9-point hedonic scale, with different purposes and objectives (Varela & Ares, 2012). As an alternative to QDA and consumer sensory analysis with hedonic scale, a number of new sensory techniques have arisen in the last couple of decades (Varela & Ares, 2012, 2014; Valentin, Cholet, Nestrud, & Abdi, 2016). These methods, sometimes called rapid sensory methods as opposed to the fact that generic descriptive analysis is quite time consuming, have proved to be useful in order to obtain accurate and reliable information from consumers.

Among these rapid methods, the present work focuses on Projective Mapping (Risvik, McEwan, Colwill, Rogers, & Lyon, 1994), later used in Napping (Pagès, 2005). Some of the interesting characteristics of this method are being a holistic methodology, based on consumers' perception of the global similarities and differences among a set of samples, as well as allowing its use both with trained panellists and with unexperienced consumers. Data from rapid methods are traditionally analyzed using statistical techniques, with multiple factor analysis (MFA) (Pagès, 2005) being the most common choice for Projective Mapping. Very recently, some works have also proposed the use of geometric techniques from graph theory (Orden, Fernández-Fernández, Rodríguez-Nogales, & Vila-Crespo, 2019; Lahne, 2020). A first contribution of the present work is to deepen in this recent line of research, introducing a substantial modification of the SensoGraph method proposed in (Orden et al., 2019) and further comparing the results and their stability between this new variant, the original SensoGraph, and MFA. Our hypothesis is that mixing geometric techniques from graph theory with common tools in statistics, like distances between samples or dendrograms, would provide further insight in sensory studies.

The number of assessors required to perform Projective Mapping differs depending upon the study, the products, and the level of expertise of the participants. Previous research has shown that the minimum number of consumers needed to obtain stable maps in Projective Mapping using MFA strongly depends on the number of samples and their degree of difference (Vidal et al., 2014; Vidal, Jaeger,

[*] Corresponding author.
E-mail address: david.orden@uah.es (D. Orden).

https://doi.org/10.1016/j.foodqual.2020.104068
Received 18 December 2019; Received in revised form 21 August 2020; Accepted 23 August 2020
Available online 27 August 2020

applied sciences

MDPI

Article

Flight Level Assignment Using Graph Coloring

Jose Manuel Gimenez-Guzman [1,*,†], Alejandra Martínez-Moraian [2,†], Rene D. Reyes-Bardales [3,†], David Orden [2,†] and Ivan Marsa-Maestre [1,†]

[1] Departamento de Automática, Universidad de Alcalá, 28805 Alcalá de Henares, Spain; ivan.marsa@uah.es
[2] Departamento de Física y Matemáticas, Universidad de Alcalá, 28805 Alcalá de Henares, Spain; alejandra.martinezm@uah.es (A.M.-M.); david.orden@uah.es (D.O.)
[3] Massachusetts Institute of Technology, Cambridge, MA 02139, USA; rdreyes@mit.edu
* Correspondence: josem.gimenez@uah.es; Tel.: +34-918856953
† These authors contributed equally to this work.

Received: 6 August 2020; Accepted: 2 September 2020; Published: 4 September 2020

check for updates

Abstract: This paper models an air traffic optimization problem where, on the one hand, flight operators seek to minimize fuel consumption flying at optimal cruise levels and, on the other hand, air traffic managers aim to keep intersecting airways at as distant as possible flight levels. We study such a problem as a factorized optimization, which is addressed through a spectrum graph coloring model, evaluating the effect that safety constraints have on fuel consumption, and comparing different heuristic approaches for allocation.

Keywords: flight level assignment; graph coloring; simulated annealing

1. Introduction

The increasing congestion of the airspace is widely accepted to be one of the outstanding problems in today's society. Researchers agree in considering the state-of-the-art methods for air traffic management (ATM) to be insufficient to handle the expected growth, see the survey by Vossen et al. [1]. Barnier and Brisset [2] describe how the airspace is divided into smaller regions, each one handled by air traffic control (ATC). Therefore, the assignment of flight levels (FLs) is carried out locally within each region, resulting in a suboptimal solution both for individual routes and for the overall network. This is mainly due to the fact that ATM often requires level changes, which are costly in terms of fuel and emissions, as well as deviations from the optimal path from origin to destination, which are also costly in terms of time and fuel. This is why allocation of flight levels is the goal in [2], where a graph coloring problem is solved by a combination of a greedy algorithm and constraint programming.

Being a well-established tool coming from discrete mathematics, graph coloring has proved to be useful in many different applications. There are a number of different types of graph coloring problems. For an exhaustive description of most of them, the reader is kindly redirected to the surveys in [3,4]. For the purposes of this paper, we will discuss the classic graph coloring problem, which is the one used by Barnier and Brisset in [2], and the spectrum coloring problem, which we proposed in [5]. The classic graph coloring problem assigns to each vertex in the graph a label or color, with the constraint that two vertices of the graph cannot have the same color if there is an edge connecting them—these kinds of edges are called monochromatic edges, and the colorings which avoid them, proper colorings. If we envision a graph where aircraft routes are vertices, edges connect intersecting routes, and colors represent FLs, this problem can be used to ensure that no two crossing routes are assigned the same FL [2]. A limitation of this approach is that routes may cross at adjacent FLs. Safety considerations, however, should favor those colorings where crossing routes use further separated FLs, so that height changes due to flight conditions are less likely to result in accidents. Taking this into account, we consider here the general spectrum graph coloring problem [5], where there is a