UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Planar Maximal Covering with Ellipses

Danilo Françoso Tedeschi

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: ______________________

Danilo Françoso Tedeschi

Planar Maximal Covering with Ellipses

Dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Marina Andretta

USP – São Carlos
March 2019

Danilo Françoso Tedeschi

Cobertura Planar Maximal por Elipses

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Marina Andretta

USP – São Carlos
Março de 2019

ABSTRACT

TEDESCHI, D. F. **Planar Maximal Covering with Ellipses**. 2019. 57 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Planar maximal covering with ellipses is an optimization problem where one wants to place ellipses on the plane to cover demand points, such that a function depending on the value of the covered points and on the cost of the ellipses that have been used is maximized. Initially, we developed an algorithm for the version of the problem where the ellipses are parallel to the coordinate axis. For the future, we intend to adapt an approximation algorithm developed for the planar maximal covering by disks and develop a method for the variant of the problem where the ellipses can be freely rotated.

Keywords: Optimization, Planar Maximal Covering Location Problem, maximal covering of points using ellipses.

LIST OF FIGURES

LIST OF ALGORITHMS

CONTENTS

CHAPTER

1

INTRODUCTION

Two main types of optimal covering problems can be found in the literature: the Minimum Cover Problem, also known as just Set Cover Problem, and the Maximal Covering Problem (KARATAS; RAZI; TOZAN, 2016).

One of the 21 Karp's NP-Complete problems[1] (KARP, 1972), the Minimum Cover Problem is very well explored and considered to be a classic. Given a demand set and a collection of subsets of the demand set, the problem asks what is the minimum number of elements from the collection of subsets needed to cover the whole demand set. One of its most famous examples is the Minimum Vertex Cover defined over graphs, where the vertex set has to be covered by a subset of edges.

The second type of covering problems arose from the fact that covering almost all the demand set can be a lot cheaper than having to cover it all (QUILES; MARíN, 2015). This second type is known as Maximal Covering Location Problem (MCLP) and was introduced in (CHURCH; VELLE, 1974). In this first study, it is defined on a network with demand nodes, a facility set is also given and a solution maximizes the demand coverage satisfying the constraint that only a subset of the facilities is used. Just like the Minimum Cover, MCLP is a NP-Hard problem (HATTA *et al.*, 2013) and both deterministic, using integer programming (CHURCH; VELLE, 1974), and heuristic methods (REVELLE; SCHOLSSBERG; WILLIAMS, 2008) have been proposed to solve it.

In (CHURCH, 1984) a new kind of MCLP named Planar Maximal Covering Location Problem (PMCLP) was introduced. This version of the problem was not defined on a network, instead the demand set and the facilities are located in $\mathbb{R}^2$ having the coverage area of a facility be defined by a distance function. PMCLP is said to have been studied under Euclidean and rectilinear distance functions (YOUNIES; ZEIDAN, 2019). The Euclidean norm PMCLP, which has a lot of results that can be applied for the elliptical PMCLP, is also found in the literature

[1] The decision version, which asks if there is a cover of size k, is NP-Complete.

as the problem of maximization of points covered by a fixed number of unit disks (BERG; CABELLO; HAR-PELED, 2006). Early works only tackled the one-disk version of the problem, in (CHAZELLE; LEE, 1986) a $\mathcal{O}(n^2)$ algorithm, which still stands as the best in terms of run-time complexity, was proposed beating the prior $\mathcal{O}(n^2 \log n)$ algorithm created by (DREZNER, 1981). The m unit disks maximal covering was studied in (BERG; CABELLO; HAR-PELED, 2006) which had as its most important result a $(1-\varepsilon)$-approximation algorithm which runs in $\mathcal{O}(n \log n)$. To achieve its main goal, however, they developed a deterministic $\mathcal{O}(n^{2m-1} \log n)$ algorithm which gets employed into their approximation scheme. Additionally, in (ARONOV; HAR-PELED, 2008) one-disk maximal covering is proven to be 3SUM-HARD. This means that maximizing the amount of points covered by a disk is as hard as finding three real numbers that sum to zero among n given real numbers.

Planar maximal covering with ellipses differs from its disks counterpart only in the shape of the facility's coverage area. The main motivation to study this modified version is that cellphone towers can have elliptical shaped coverage area, so in order to determine what are the best locations to place m cellphone towers to maximize the amount of the population covered by its signal, an elliptical PMCLP is better suited (CANBOLAT; MASSOW, 2009). Only two articles have been found published in the literature that study this problem. In (CANBOLAT; MASSOW, 2009), a mixed-integer non-linear programming method was proposed as a first approach to the problem. For some instances the method took too long and did not find an optimal solution. For this reason a heuristic method was developed using a technique called Simulated Annealing. Solutions for the instances that timed-out with the first method were then obtained. The problem was further explored in (ANDRETTA; BIRGIN, 2013) which proposed a deterministic method that showed better performance obtaining optimal solutions for the instances which the first method could not. Also, in (ANDRETTA; BIRGIN, 2013), a version of the problem where every ellipse can be freely rotated was introduced and an exact method, which could not find optimal solutions for large instances, and a heuristic method were proposed for it. Despite the similarities, none of the works cited above base their development on the maximal covering with disks algorithms found in the literature.

This work is structured in the following way: Chapter 2 introduces some definitions and results that are used throughout the next chapters; in Chapter 3, the maximal covering by disks problem is studied and a $\mathcal{O}(n^{2m})$ algorithm is proposed; in Chapter 4, the maximal covering by ellipses is introduced and the algorithm for the disks case is adapted for it; finally, Chapter 7 presents what is left as future work. Also, Appendix A determines with detail the intersection of two ellipses, which is used in the algorithm developed in Chapter 4.

CHAPTER 2

NOTATION AND PRELIMINARIES

Some definitions and results that are used throughout the text are given in this chapter.

2.1 Elliptical and Euclidean norm functions

A norm in $\mathbb{R}^2$ is a function that maps every vector onto a non-negative real number satisfying homogeneity and the triangle inequality.

Let $u \in \mathbb{R}^2$ be a vector, the Euclidean norm of u is defined as

$$||u||_2 = \sqrt{u^T u}. \tag{2.1}$$

The elliptical norm, also known as weighted norm, takes a 2 by 2 positive definite matrix as its parameter. This matrix can be seen as a linear transformation of the Euclidean norm. The elliptical norm of $u \in \mathbb{R}^2$ is defined as

$$||u||_Q = \sqrt{u^T Q u}, \tag{2.2}$$

where Q is a 2 by 2 positive definite matrix.

The elliptical norm, when taking Q to be the identity matrix, becomes the Euclidean norm.

Determining the distance between two points, given a norm function, is done by calculating the norm of the vector defined by the difference between the two points. For example, the elliptical distance between the points $p, q \in \mathbb{R}^2$ is given by $||p - q||_Q$.

2.2 Disk

A circle (or circumference) is a set of points in $\mathbb{R}^2$ that have the same Euclidean distance, also known as radius, to another point, also referred to as the center of the circle. A unit circle is a circle with radius equal to 1.

A disk is the set of points bounded by a circle. In other words let $c \in \mathbb{R}^2$. A unit disk with center c is the set of every point $p \in \mathbb{R}^2$ which satisfies

$$||p-c||_2^2 \leq 1. \tag{2.3}$$

2.3 Ellipse

An ellipse is a curve which is categorized, along with the parabola and the hyperbola, as a conic section. As the name suggests, conic sections are curves resulted from the intersection of a right circular cone in $\mathbb{R}^3$ with a plane (BRANNAN; ESPLEN; GRAY, 1999). From that definition, an equation which describes any conic section is given as follows

$$Ax^2+Bxy+Cy^2+Dx+Ey+F=0, \tag{2.4}$$

where $A,B,C,D,E,F \in \mathbb{R}$ are fixed and $x,y \in \mathbb{R}$.

To distinguish an ellipse from the other conic sections given an instance of Equation 2.4, the condition $4AC-B^2>0$ must be verified (AYOUB, 1993).

Assuming the center of an ellipse is $c \in \mathbb{R}^2$, then Equation 2.4 can be rewritten as a quadratic form as follows

$$(p-c)^T Q(p-c)=1, \tag{2.5}$$

with $p \in \mathbb{R}^2$ and Q being a 2 by 2 positive definite matrix which carries the parameters of the ellipse. From Equation 2.4, Q can be defined as follows

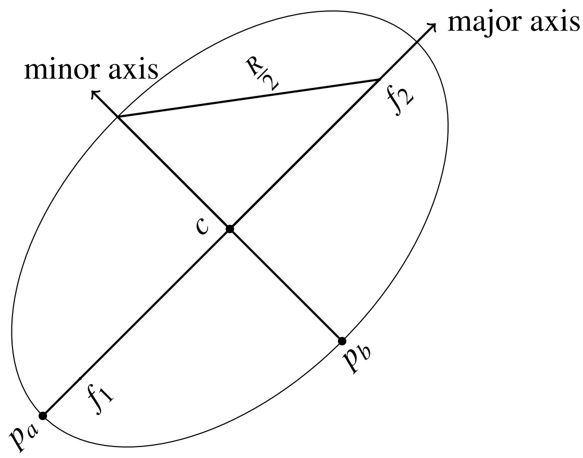
$$Q=\begin{pmatrix} A & \frac{B}{2} \\ \frac{B}{2} & C \end{pmatrix}.$$

Note that asking Q to be positive definite is the same as asking $4AC-B^2$ to be positive. This makes us arrive at the following definition of the ellipse.

Definition 1. Let $c \in \mathbb{R}^2$ be the center of an ellipse and Q be a 2 by 2 positive definite matrix. An ellipse is the set of every point $p \in \mathbb{R}^2$ such that $||p-c||_Q^2=(p-c)^T Q(p-c)=1$. Also, a point p is considered covered by an ellipse if $||p-c||_Q^2=(p-c)^T Q(p-c)\leq 1$.

An alternative way to define an ellipse, which can be seen as just a property derived from the definition above, is to begin its construction with two points called foci and a constant $R \in \mathbb{R}$, with R being greater than the Euclidean distance between the two foci points (see Figure 1). The ellipse is, then, defined as the set of points whose distance to the foci is equal to R. In other words, let $f_1, f_2 \in \mathbb{R}^2$ be the two foci points, the ellipse is the set of every point $p \in \mathbb{R}^2$, such that $||p - f_1||_2 + ||p - f_2||_2 = R$. It can be shown that this definition is equivalent to Definition 1, with the coverage of a point p being equivalent to $||p - f_1||_2 + ||p - f_2||_2 \leq R$.

Figure 1 – A non-axis-parallel ellipse and its foci points.



Source: Elaborated by the author.

Also, in Figure 1, the distance $a = ||p_a - c||_2$, where p_a is one of the intersection points of the ellipse with the major axis, is called the semi-major, and the distance $b = ||p_b - c||_2$, where p_b is one of the intersection points of the ellipse with the minor axis, is called the semi-minor. These two values are also referred to as the shape parameters of an ellipse. Let $d = ||c - f_1||_2$, then it is easy to see that $a = R - d$ and $b = \sqrt{\frac{R^2}{4} - d^2}$.

Finally, an ellipse is said to be axis-parallel if its major-axis (see Figure 1), which is the line that passes through its two foci points, is parallel to the x-axis.

2.3.1 Axis-parallel

An axis-parallel ellipse centered at $c = (c_x, c_y)$ can be described using Definition 1 with Q being a diagonal matrix [1]. This can be understood as a scaling transformation applied to the Euclidean norm.

Defining the matrix Q as

[1] The only non-zero terms are in the main diagonal.

$$Q = \begin{pmatrix} \frac{1}{a^2} & 0 \\ 0 & \frac{1}{b^2} \end{pmatrix},$$

then, starting from Definition 1, we can obtain the following equation

$$\begin{aligned} (p-c)^T Q(p-c) = 1 \Rightarrow \\ \left(\frac{p_x - c_x}{a^2}, \frac{p_y - c_y}{b^2}\right)^T (p_x - c_x, p_y - c_y) = 1 \Rightarrow \end{aligned}$$

$$\frac{(p_x - c_x)^2}{a^2} + \frac{(p_y - c_y)^2}{b^2} = 1, \tag{2.6}$$

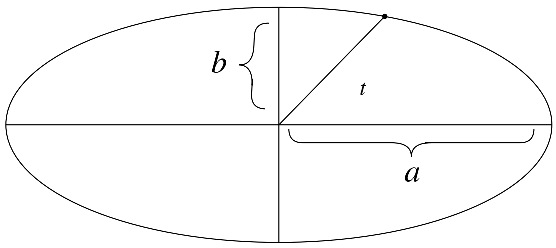
where a and b are the semi-major and semi-minor shape parameters, respectively.

Also, the coverage region is determined by just changing the equality to a inequality as follows

$$\frac{(p_x - c_x)^2}{a^2} + \frac{(p_y - c_y)^2}{b^2} \leq 1. \tag{2.7}$$

Another way to represent ellipses, which will be useful in some occasions, is through writing it as a curve, function of the angle with its major-axis (see Figure 2).

Figure 2 – The ellipse as a parametric curve.



Source: Elaborated by the author.

Let $c \in \mathbb{R}^2$ be the center of an ellipse with shape parameters $(a,b) \in \mathbb{R}^2_{>0}$. Then $\gamma : [0, 2\pi] \mapsto \mathbb{R}^2$ defines a curve which maps every angle onto a point on the ellipse and it is defined as follows

$$\gamma(t) = \begin{cases} x(t) = a\cos t + c_x, \\ y(t) = b\sin t + c_y. \end{cases} \tag{2.8}$$

No equivalent disk-circle wording exists for ellipses, this could be a source of ambiguity in the text, that is why a note for the reader was judged to be necessary. Throughout this work an ellipse will represent the set of points that satisfy Definition 1. In some places, though, with prior clarification, we will denote as an ellipse the set of points that are covered by the ellipse itself. For example, when we define $\mathcal{P} \cap E$ as the set of points in $\mathcal{P}$ that are covered by E, we are implicitly calling E the set of points that are covered by the ellipse itself as it is defined by Definition 1.

2.3.2 Non-axis-parallel

A non-axis-parallel ellipse centered at $c \in \mathbb{R}^2$ can also be described by Definition 1. Nonetheless, a simpler equation is going to be used here. Besides the center, the shape parameters $(a,b) \in \mathbb{R}^2_{>0}$, with $a > b$, an angle of rotation $\theta \in [0,\pi]$ is given. This angle represents the angle between the x-axis and the major-axis of the ellipse.

A rotated ellipse by θ can be transformed into a axis-parallel centered at the origin by applying two linear transformations. Reversing these transformations produces the following equation for a non-axis-parallel ellipse:

$$\frac{((x-c_x)\cos\theta + (y-c_y)\sin\theta)^2}{a^2} + \frac{((x-c_x)\sin\theta - (y-c_y)\cos\theta)^2}{b^2} = 1. \tag{2.9}$$

Also, the coverage region of an ellipse centered at $c \in \mathbb{R}^2$ with shape parameters $(a,b) \in \mathbb{R}^2_{>0}$ and rotation angle $\theta \in [0,\pi)$ is represented by every point $(x,y) \in \mathbb{R}^2$ that satisfies Equation 2.10.

$$\frac{((x-c_x)\cos\theta + (y-c_y)\sin\theta)^2}{a^2} + \frac{((x-c_x)\sin\theta - (y-c_y)\cos\theta)^2}{b^2} \leq 1. \tag{2.10}$$

2.4 Complex numbers

The set of complex numbers $\mathbb{C}$ is an extension of the set of real numbers $\mathbb{R}$ and can be very useful depending on the problem at hand. A thorough introduction on this topic is out of the scope and we just go through some basic properties that are going to be used later on Chapter 6.

A complex number is composed of a real part added to an imaginary part which is a multiple of the imaginary unit $i = \sqrt{-1}$. This is expressed as $a + bi$, with $a, b \in \mathbb{R}$. Because z is composed of two real numbers, the complex number system can be visualized on $\mathbb{R}^2$ as shown on Figure 3. This is also a good way to visualize Euler's Formula, as it can be also seen on Figure 3, which says that any complex number can be written in terms of its radius and polar angle as follows:

$$z = re^{i\theta} = r(\cos\theta + i\sin\theta), \tag{2.11}$$

with r being the length of the vector determined by the point z on the complex plane and $\theta = angle(z)$ being its polar angle. Note that $angle$ is a function from $\mathbb{C}$ to $[0, 2\pi)$.

The complex conjugate is another important operator that is utilized later. Let $z = a + bi \in \mathbb{C}$, then we refer to $\bar{z}$ as the complex conjugate of z and it is defined as:

$$\bar{z} = a - bi. \tag{2.12}$$

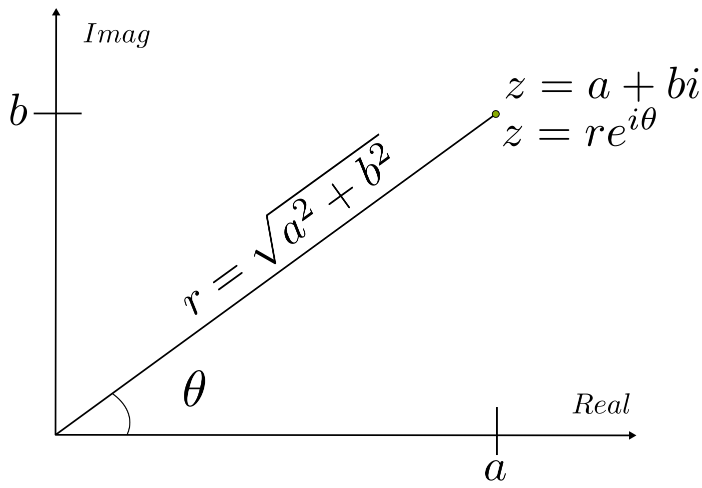


Figure 3 – The representation of a complex number on two dimensions.

Finally, we need to state two observations about the angles of $-z$ and $\bar{z}$ with respect to the angle of z, this will be useful later on Chapter 6.

$$angle(\bar{z}) = 2\pi - angle(z) \tag{2.13}$$

$$angle(-z) = \pi + angle(z), \tag{2.14}$$

checking the validity of these two equalities can be done by just observing the symmetry between the points defined by $z, \bar{z}$, and $-z$ on the plane.

2.5 Polynomials and their roots

In this work, we are mostly interested in univariate polynomials defined over the complex numbers.

A function $p_n : \mathbb{C} \mapsto \mathbb{C}$ is a n-degree polynomial if it can be written as:

$$p_n(z) = \sum_{k=0}^{n} a_k z^k, \tag{2.15}$$

with $a_0, \ldots, a_n \in \mathbb{C}$. The famous Abel-Ruffini Theorem (a proof can be seen in Skopenkov (2015)) states that for polynomials of degree higher than four, there is no closed formula[2] to determine their roots. Fortunately, a numerical approach exists for higher-degree polynomials which in practice works really well.

In Horn and Johnson (1986, p. 195) a theorem is presented which says that for every univariate polynomial, there exists a companion matrix, such that its eigenvalues are the zeros

[2] A finite number of $+, -, \times, \div, \sqrt{\cdot}$.

of that polynomial. For example, the companion matrix of a degree-5 polynomial written as Equation 2.15 is given by:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\frac{a_0}{a_5} & -\frac{a_1}{a_5} & -\frac{a_2}{a_5} & -\frac{a_3}{a_5} & -\frac{a_4}{a_5} \end{bmatrix}. \tag{2.16}$$

Finding every eigenvalue of a matrix can be done using the QR algorithm, which runs in $\mathcal{O}(n^3)$ and uses $\mathcal{O}(n^2)$ memory. A very complete introduction to this algorithm can be found on Watkins (2008). The idea is to convert the input matrix to the Hessenberg form, after that, the matrix preserves its form through the iterations and under some assumptions, convergence to the eigenvalues is achieved. Also, in Barel *et al.* (2010) it is pointed that companion matrices are already in the Hessenberg form, making the Hessenberg conversion step unnecessary. This and other properties are used in Barel *et al.* (2010) on the development of a $\mathcal{O}(n^2)$ algorithm for the specific case of companion matrices.

In practice, LAPACK's ZGEEV routine is utilized–the user guide can be found in Anderson *et al.* (1999)–which is an implementation of the QR algorithm that returns every eigenvalue of a complex matrix.

2.6 Real trigonometric polynomial

The same definition found in Powell (1981, p. 150) for real trigonometric polynomials is given here. They are also referred to as truncated Fourier Series in Boyd (2006) and are given by:

$$p_n(\theta) = \sum_{k=0}^{n} a_k \cos(k\theta) + \sum_{k=1}^{n} b_k \sin(k\theta). \tag{2.17}$$

We say that $p_n : \mathbb{R} \mapsto \mathbb{R}$ as defined by Equation 2.17 is a n-degree real trigonometric polynomial. An important property is stated in Powell (1981, p. 150), it says that a n-degree polynomial can have up to $2n$ distinct roots on the interval $[0, 2\pi)$. It also says that a function written in the format:

$$\cos^j\theta \sin^k\theta \qquad j,k \in \mathbb{Z}_+ \ ,$$

can be transformed into a real trigonometric polynomial of degree $j+k$. Therefore, the following expression also represents a n-degree real trigonometric polynomial:

$$\sum_{0 \le j+k \le n} c_{j,k} \cos^j\theta \sin^k\theta, \tag{2.18}$$

for some $\{c_{j,k} \in \mathbb{R} : 0 \le j+k \le n\}$.

CHAPTER

3

MAXIMAL COVERING BY DISKS

In this chapter, the classical version of PMCLP using disks will be defined and a version of the method will be proposed with the intention of later being used to solve the axis-parallel ellipses version of PMCLP. Throughout the course of this work, the maximal covering by disks problem is going to be referred to as $MCD(\mathcal{P}, m)$, where $\mathcal{P}$ is a set of points and m is the number of unit disks.

3.1 One disk, $MCD(\mathcal{P}, 1)$

Two exact methods for the $MCD(\mathcal{P}, 1)$ have been found in the literature. A $\mathcal{O}(n^2)$ algorithm is proposed by (CHAZELLE; LEE, 1986) which improved the previously $\mathcal{O}(n^2 \log n)$ one proposed by (DREZNER, 1981). As it has been mentioned, $MCD(\mathcal{P}, 1)$ is a 3SUM-HARD problem, which means that it is as hard as the 3SUM problem (the problem of finding 3 real numbers that sum to 0, given n real numbers). Initially the lower bound of the 3SUM problem was conjectured to be $\Omega(n^2)$, matching the best algorithm for $MCD(\mathcal{P}, 1)$, which meant that no better time-complexity could be achieved. Since then, however, better algorithms for 3SUM have been developed with a $\mathcal{O}(\frac{n^2}{poly(n)})$ run time complexity (KOPELOWITZ; PETTIE; PORAT, 2014).

The $m = 1$ version is treated here before the general case because it will be shown that, using the algorithm here proposed for $MCD(\mathcal{P}, 1)$, an optimal solution can be obtained for the $MCD(\mathcal{P}, m)$ as well as for the axis-parallel ellipse version of the problem.

3.1.1 Notation and definition of the problem

Initially, the input of the problem defines a unit disk with its center point undefined, a solution for the problem will then choose a point to be the center of the unit disk. In other words, a solution places the disk somewhere in the plane. We refer to the unit disk with undefined center as D. If it is placed at a center $q \in \mathbb{R}^2$, we call it $D(q)$.

Definition 2. Let $\mathcal{P} = \{p_1, \dots, p_n\}$ be a set of n points in $\mathbb{R}^2$ and $w(p) > 0, p \in \mathcal{P}$, the weight of every point in $\mathcal{P}$. We denote $w(A)$, with $A \subset \mathcal{P}$, as the sum of weights of every point in A. Finally, let D be a unit disk, we define an optimal solution of $MCD(\mathcal{P}, 1)$ as

$$\max_q w(\mathcal{P} \cap D(q)). \tag{3.1}$$

Therefore, an optimal solution for an instance of $MCD(\mathcal{P}, 1)$ will be a point in which a unit disk located at, covers points whose weights, when summed, is maximal.

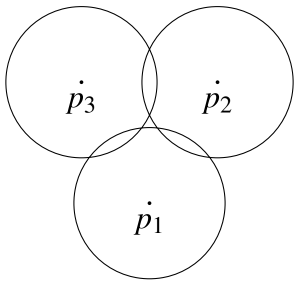
In (DREZNER, 1981), the main idea used to develop the $\mathcal{O}(n^2 \log n)$ algorithm is that, even though there are infinitely many points where the disk could be placed, only a few of them, a finite amount of $\mathcal{O}(n^2)$, needs to be considered for the method to find an optimal one. The algorithm, for every point, sorts the other points with respect to the angle they form with the first one. After that, the first point is placed on the border of the disk and, going through the sorted list, the algorithm inserts and removes points from the disk coverage. Also, when inserting and removing a point from the coverage, it only checks the disk centers that make the entering/leaving point to be on the border. Because the algorithm only checks the centers that make the disk have two points on its border, the number of centers it goes through is bounded by the number of pairs of points, which is $\binom{n}{2} = \mathcal{O}(n^2)$.

In (CHAZELLE; LEE, 1986) and (BERG; CABELLO; HAR-PELED, 2006), on the other hand, instead of working directly with $MCD(P, 1)$, an equivalent problem called Maximum Weight Clique was introduced. The algorithm that we later describe in this work also uses this equivalence. That is why it has been taken to be fundamental to introduce the Maximum Weight Clique Problem and discuss the equivalence.

3.2 Maximum Weight Clique Problem

Let $\mathcal{P} = \{p_1, \dots, p_n\}$, with $p_i \in \mathbb{R}^2$, be a set of points, $\mathcal{D} = \{D_1, \dots, D_n\}$ a set of unit disks, such that D_i is centered at p_i, $i = 1, \dots, n$, with every disk having a weight $w_i > 0$, $i = 1, \dots, n$. A clique, in this context, is a non-empty intersection area of a subset of disks. Note that this is different than the clique problem on a intersection graph (a graph where the vertices are the disks and an edge exists if there is an intersection between two disks). As shown in Figure 4, three disks could have non-empty pairwise intersection (which qualifies them as a clique), but the intersection of all the three together is empty. That is why the clique problem for unit disks is also referred to as the Maximum Geometric Clique Problem when the condition of common intersection exists and as the Maximum Graphical Clique Problem when there is only the pairwise intersection condition (DE; NANDY; ROY, 2014). The graphical version of the problem was studied by (CLARK; COLBOURN; JOHNSON, 1990), where a $\mathcal{O}(n^{4.5})$ algorithm was proposed. Also, in (DE; NANDY; ROY, 2014), a $\mathcal{O}(n^2 \log n)$ time in-place algorithm[1] for arbitrary radii disks was proposed. In (CHAZELLE; LEE, 1986), the method for the Maximum Geometric Weight Clique Problem consists on building a planar graph on which the vertices were the $\mathcal{O}(n^2)$ pairwise intersection of the circumferences and the edges were the arcs of the circumferences connecting the intersections. With the graph constructed, a traversal is done to obtain the answer, thus the time complexity of $\mathcal{O}(n^2)$.

Figure 4 – Three disks that have non-empty pairwise intersection among them, but no common intersection.



Source: Elaborated by the author.

A solution for the Maximum Weight Clique is a set of points Q, such that the sum of weights of all disks that cover it is maximized. Even though there could be a use for the whole set Q, as this problem is used as a tool to solve another problem, only finding a point from the Maximum Weight Clique is enough. This will become clear when the equivalence is stated. With everything in hands, we can define the Maximum Weight Clique Problem as follows.

Definition 3. Let $\mathcal{D}$ be a set of unit disks and $\mathcal{P}$ be a set of points as defined before. An optimal solution for the Maximum Weight Clique Problem is given by

$$\max_{q} \sum_{D_k \cap q \neq \emptyset} w_k. \tag{3.2}$$

As it has been proposed, with the equivalence of the two problems, an optimal solution of the Maximum Weight Clique Problem is also an optimal solution of the $MCD(\mathcal{P}, 1)$, which means that a disk centered at q, defined in Definition 3, will have a maximal weight covering of the set $\mathcal{P}$.

Given an instance of $MCD(\mathcal{P}, 1)$, the equivalent Maximum Weight Clique Problem is obtained by defining set $\mathcal{D}$ to contain the disks centered at $\mathcal{P}$ and setting the weight of every disk to be the weight of its corresponding point in $\mathcal{P}$. A disk D_i will represent the area where a disk can be placed in order to cover p_i. This means that a intersection between some disks is an area where a disk could be placed to cover the corresponding points.

[1] An in-place algorithm is an algorithm that needs $\mathcal{O}(1)$ extra space.

In Figure 4, it can be seen that there is no point where a disk could be placed such that it would cover p_1, p_2 and p_3, nonetheless, in any of the pairwise intersections, a disk could be placed to cover the two corresponding points.

Formally, in the Maximum Weight Clique Problem, if a point q lies inside $\bigcap_{k \in I} D_k$, with $I \subset \{1, \ldots, n\}$, then a disk centered at q will cover the points p_k, with $k \in I$ in the $MCD(P,1)$ problem. Conversely, the same applies for a disk placed at q that covers points p_k, with $k \in I$ in the one disk maximal covering problem. It means that q will lie inside region $\bigcap_{k \in I} D_k$.

3.2.1 An algorithm for the Maximum Weight Clique Problem

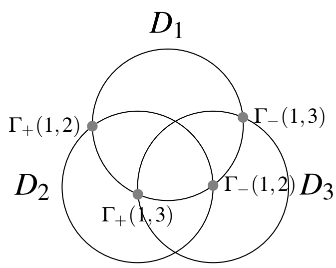
The algorithm described here is based on the one in (DREZNER, 1981), also with some ideas from (DE; NANDY; ROY, 2014) and (BERG; CABELLO; HAR-PELED, 2006). It has a run time complexity of $\mathcal{O}(n^2 \log n)$ and uses $\mathcal{O}(n)$ of extra space. It is worth noting, however, that a $\mathcal{O}((n+K)\log n)$ run time, with K being the number of intersections, can be obtained by using the algorithm in (BENTLEY; OTTMANN, 1979) to find all the intersections among the n circumferences.

Without loss of generality, the weights will be ignored, and the method will be described for the Maximum Clique Problem, assuming that every disk has unit weight. Also, it will be assumed that no pair of disks are placed at the same center.

Let $\mathcal{D}$ be a set of n unit disks, a non-empty intersection area of a subset of disks is convex and bounded by the arcs of the disks that are intersecting (DE; NANDY; ROY, 2014). With this observation, in order to find an optimal solution, it is sufficient to check, for every disk D, every intersection area that is bounded by its arc.

Definition 4. Let D_i and D_j be two unit disks that intersect (at least at one point). Also let $(\theta_1, \theta_2) \in [0, 2\pi]^2$ be the two angles that the circumferences induced by D_i and D_j intersect, with the condition that (θ_1, θ_2) defines an arc (counter-clockwise order) of D_i that is the border of $D_i \cap D_j$. If D_i is tangent to D_j, then $\theta_1 = \theta_2$. Then, define $\Gamma_+(i, j) = \theta_1$ and $\Gamma_-(i, j) = \theta_2$, also we refer to them as opening and closing intersection angles respectively.

Figure 5 – Three disks and their intersection points.

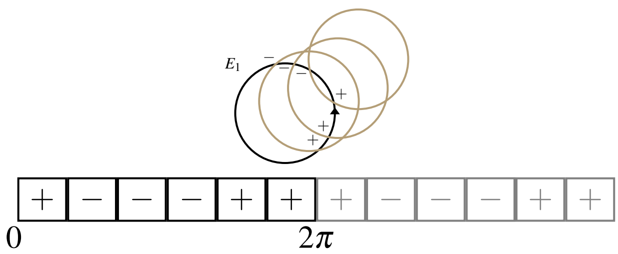


Source: Elaborated by the author.

In Figure 5, it is shown all the intersection points between D_1 with D_2 and D_3. Also, they are labeled according to Definition 4. Note that $\Gamma_+(1,3) > \Gamma_-(1,3)$ (the angles should be in the $[0,2\pi]$ interval).

With Definition 4 in hand, we can establish the basis of the algorithm to find the maximum clique in which a disk D_i participates: a traversal going through every point of intersection with D_i, in counter-clockwise order, keeping a set of active disks. When an opening intersection angle is reached, the corresponding disk is added to the active set; when a closing one is reached, the corresponding disk is removed from the active set. This simple traversal, however, would not handle the special case with $\Gamma_+(i,j) > \Gamma_-(i,j)$ (see Figure 5). If the traversal begins at the point with smallest angle, the algorithm would remove D_3 from the active disks without first adding it. If there was another disk starting before $\Gamma_-(1,3)$, the algorithm would not have both of them in the active set at the same time, and an optimal solution could end up not being found. This can be worked around repeating the traversal once without resetting the active disks set, that way, in the beginning of the second traversal, the active set would contain the disks that have $\Gamma_+(i,j) > \Gamma_-(i,j)$. This is shown in Figure 6, where the intersections list is duplicated, simulating the traversal repetition (note the indication to where the traversal starts as well as the positive and negative signs representing when a intersection with another disk opens and closes, respectively). It can be seen that without the repetition, the algorithm would find 2 as the value of an optimal solution, which is wrong.

Figure 6 – The intersections list of a disk with three other disks.



Source: Elaborated by the author.

Lemma 1. Algorithm 1 for solving the Maximum Clique Problem has a $\mathcal{O}((n+K)\log n)$ run time complexity, where K is the number of intersections of the n disks.

Proof. Finding every intersection can be done in $\mathcal{O}((n+K)\log n)$ by a plane sweep, the method is described in (BENTLEY; OTTMANN, 1979). Because the traversal is made in counter-clockwise order, the intersection points have to be sorted by their intersection angles, so an additional $\mathcal{O}(K\log K)$ pre-processing is needed. All the other operations can be done in constant time. Therefore, the final algorithm complexity is $\mathcal{O}((n+K)\log n)$. □

Procedure 1 – Algorithm for $MCD(\mathcal{P},1)$ with unit weights

Input: A set of points $\mathcal{P} = \{p_1,\dots,p_n\}$.
Output: The maximum number of points that can be covered by a unit disk.

```
procedure MCD_1(P)
    Q_best ← {}
    ans ← center of D_1
    for all p_i ∈ P do
        Let D_i be the disk with center at p_i
        Let I_i be the set of disks that intersect with D_i
        A ← {}                                  ▷ The multiset of intersection angles with D_i.
        for all j ∈ I_i do
            A ← A ∪ {Γ_+(i, j) ∪ Γ_−(i, j)}
        end for
        Q ← {D_i}                               ▷ The set of active disks.
        for cnt = 1..2 do                       ▷ Do it twice.
            for a ∈ A do                        ▷ Assuming A is sorted.
                Let D_a be the disk that intersects D_i at angle a.
                if a is a starting angle then
                    Q ← Q ∪ {D_a}
                else
                    Q ← Q \ {D_a}
                end if
                if |Q_best| < |Q| then
                    Q_best ← Q
                    ans ← point corresponding to the intersection angle a
                end if
            end for
        end for
    end for
    return ans
end procedure
```

If a simpler implementation is desired, or the number of intersections is large, determining the set I_i (the set of disks that intersect with D_i, defined in Algorithm 1) can be simply done in $\mathcal{O}(n^2)$, making the algorithm have a worst-case complexity of $\mathcal{O}(n^2 \log n)$.

3.3 Multiple disks, $MCD(\mathcal{P},m)$

In (BERG; CABELLO; HAR-PELED, 2006), a $\mathcal{O}(n^{2m-1} \log n)$ algorithm for $MCD(\mathcal{P},m)$ is developed as a sub-routine for its $(1-\varepsilon)$-approximation algorithm. Firstly, they solve the sub-problem $MCD(\mathcal{P},2)$ in $\mathcal{O}(n^3 \log n)$. Then for the rest of the points that are not in that solution, it uses the algorithm developed in (CHAZELLE; LEE, 1986) for the one-disk case, checking every possible solution for every one of the disks left.

Also, in (HE *et al.*, 2015) an heuristic method for large-scale *MCD* is proposed. It uses a probabilistic algorithm called mean-shift which is a gradient ascent method proven to converge to a local density maxima of any probability distribution. The mean-shift is utilized to find good candidates of centers for the unit disks, then the method backtracks to find the best assignment. The results showed that the greedy algorithm achieves an optimal coverage in some instances, but for some other ones it has a 15 percent worse coverage ratio.

It can be seen that in any solution of $MCD(\mathcal{P}, m)$, a disk placed at a point q that covers at least one point $p \in \mathcal{P}$ has a correspondence to the Maximum Weight Clique Problem: the point q is inside an intersection area of at least one disk and that area is bounded by some disk, which means it will be checked by Algorithm 1 as a candidate to be an optimal solution. The number of points Algorithm 1 goes through is $\mathcal{O}(n^2)$, then checking every possible center for every ellipse yields a $\mathcal{O}(n^{2m})$ run-time complexity. This algorithm is described in Chapter 4 for the ellipses case.

The choice of developing a different method for the problem, instead of using the one from (BERG; CABELLO; HAR-PELED, 2006), is taken for the sake of simplicity, considering both algorithms achieve similar bounds.

CHAPTER 4

MAXIMAL COVERING BY ELLIPSES

In this chapter, the version of the Planar Maximal Covering by Ellipses Problem where every ellipse has to be parallel to the coordinate axis will be introduced and a method for it will be developed.

4.1 Axis-Parallel

The maximal planar covering using axis-parallel ellipses was first introduced in (CANBOLAT; MASSOW, 2009) which proposed a mixed integer non-linear programming method for the problem. This first approach showed to be not that efficient as it could not find an optimal solution for some instances within a timeout defined by them. To obtain solutions, not necessarily optimal ones, for the instances which the exact method showed inefficiency, a heuristic technique called Simulated Annealing was used to develop another method. Comparisons were made, which showed that the second approach was able to obtain good solutions, compared to the optimal ones found for some of the instances, within a good run-time.

The second work found in the literature was (ANDRETTA; BIRGIN, 2013), which developed a method that breaks the problem into smaller ones fixing the set of points an ellipse is going to cover. For each set of points fixed as the points an ellipse is going to cover, a small optimization problem is solved to find out if there is a location where the ellipse can be placed, so to cover the set of fixed points. To enumerate the possible solutions and then find an optimal one, the method defined a data structure that stores every set of points an ellipse can cover. This method showed better results and was able to find optimal solutions for the instances that the first method could not get as well as for new created instances.

4.1.1 Notation and definition of the problem

Axis-parallel ellipses are defined as the set of points that satisfy Equation 2.6. Therefore, all it takes to describe one is a pair of positive real numbers $(a,b) \in \mathbb{R}^2_{>0}$, also called the shape parameters, and a center point $q \in \mathbb{R}^2$.

Firstly, the case with only one ellipse is considered, an instance of the problem is denoted as $MCE(\mathcal{P},a,b)$ where $\mathcal{P}$ is a set of points and $(a,b) \in \mathbb{R}^2_{>0}$ is a pair of real numbers called the shape parameters of an ellipse. In the general case, every point has weights, but firstly this detail will be ignored and the weights are assumed to be unitary. The notation used here is similar to the one introduced on Chapter 3, the ellipse with an undefined center is referred to as E. To denote the ellipse with center set to be at point q, $E(q)$ is used. Also, the set of points covered by $E(q)$ is denoted by $E(q) \cap \mathcal{P}$, which indirectly defines $E(q)$ to be the set of points that satisfy Equation 2.7 (in other words $E(q)$ is the coverage region defined by the ellipse with shape parameters (a,b), located at center q). Also, to denote the points on the border of $E(q)$ we use the notation $\tilde{E}(q)$, therefore $p \in \tilde{E}(q)$ translates to p being on the border of $E(q)$. Hence, the problem can be defined as follows.

Definition 5. Let $MCE(\mathcal{P},a,b)$ be an instance of the maximal covering by one ellipse, with E being an ellipse with shape parameters $(a,b) \in \mathbb{R}^2_{>0}$, an optimal solution of $MCE(\mathcal{P},a,b)$ is given by

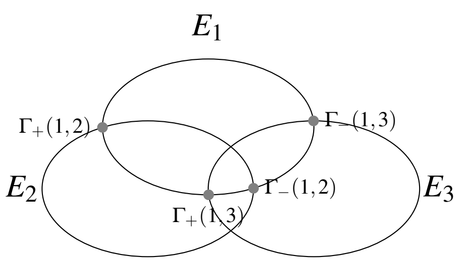
$$\max_q |\mathcal{P} \cap E(q)|. \tag{4.1}$$

4.1.2 One Disk algorithm adaptation

An adaptation of Algorithm 1 is obtained by just replacing the function that finds the intersection points between two disks by a function that finds the intersection points between two ellipses E_i and E_j. It can be seen in Figure 7 that the intersection points and their correspondents $\Gamma_-(i,j)$ and $\Gamma_+(i,j)$ functions behave the same way as in the disks case.

The intersection of two ellipses as well as determining $\Gamma_-(i,j)$ and $\Gamma_+(i,j)$ is described thoroughly in Appendix A.

Figure 7 – Three ellipses and their intersection points



Source: Elaborated by the author.

4.1.3 Multiple ellipses

The multiple ellipses case is handled using the same idea of the multiple disks case. The only difference is that an instance of the multiple ellipses may contain ellipses of different shapes, which does not happen for the disks case as every disk has the same radius. For this reason, a different pre-processing has to be done for every one of them.

An instance of the multiple ellipses case is denoted as $MCE(\mathcal{P}, \mathcal{E})$, with $\mathcal{P}$ being a set of n points and $\mathcal{E} = \{E_1, \dots, E_m\}$ being a set of m ellipses, each one with shape parameters $(a_i, b_i) \in \mathbb{R}^2_{>0}, i = 1 \dots m$. Also, without loss of generality, the weight of every point is assumed to be unitary.

Definition 6. Let $MCE(\mathcal{P}, \mathcal{E})$ be an instance of the maximal covering by ellipses, an optimal solution is given by

$$\max_{q_1, \dots, q_m} \left| \bigcup_{i=1}^{m} \mathcal{P} \cap E_i(q_i) \right|. \tag{4.2}$$

Algorithm 3 describes the adapted version of the maximal disk covering algorithm for the ellipses case. In Algorithm 2, the MCE_1 procedure returns every possible set of points that an ellipse with shape parameters (a, b) can cover. With that, procedure MCE does the backtracking process, assigning every possible cover to every ellipse.

As stated in Lemma 1, MCE_1 runs in $\mathcal{O}(n^2 \log n)$, where n is the number of points. The number of sets of points an ellipse can cover, however, is $\mathcal{O}(n^2)$. Note that the $\log n$ being part of the complexity is due to sorting the set A. If MCE_1 is called only in a pre-process phase storing its return for every ellipse, a $\mathcal{O}(n^{2m})$ run-time complexity can be achieved.

Also, it can be seen that the unitary weights assumption can be easily removed through replacing the way the answer is updated: the weights of the covered points should be added to the answer instead of the number of covered points, this could be done by keeping an extra variable along with every possible set of points an ellipse can cover that is returned by MCE_1.

It is worth noting that some easy improvements, which do not change the algorithm's overall complexity, can be made in the implementation. For example, if an ellipse covers two sets of points X and Y, with $X \subset Y$, then set X can be ignored by the algorithm because of the positive weights constraint. Also, if two ellipses have their centers with Euclidean distance greater than their semi-major parameter, they for sure do not intersect. Depending on the input, this observation can make the algorithm not go through the whole list of ellipses every time it needs to determine the ellipses pairwise intersections.

Procedure 2 – Algorithm for $MCE(\mathcal{P}, a, b)$ with unit weights

Input: A set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$, and the shape parameters (a, b) of an ellipse.
Output: A collection of subsets of $\mathcal{P}$ that the ellipse is able to cover.

procedure $MCE_1(\mathcal{P}, a, b)$
$Z \leftarrow \{\}$ ▷ A collection of subsets of $\mathcal{P}$, each being a possible coverage.
for all $p_i \in \mathcal{P}$ **do**
Let E_i be the ellipse with center at p_i and parameters (a, b)
Let I_i be the set of ellipses that intersect with E_i
$A \leftarrow \bigcup_{j \in I_i} \{\Gamma_+(i, j) \cup \Gamma_-(i, j)\}$ ▷ The multiset of intersection angles with E_i.
$Z \leftarrow Z \cup \{\{p_i\}\}$
$Cov \leftarrow \{p_i\}$ ▷ The set of active disks.
for $cnt = 1..2$ **do** ▷ Do it twice.
for $a \in A$ **do** ▷ Assuming A is sorted.
Let p_a be the point represented by the ellipse that intersects E_i at angle a.
if a is a starting angle **then**
$Cov \leftarrow Cov \cup \{p_a\}$
else
$Cov \leftarrow Cov \setminus \{p_a\}$
end if
$Z \leftarrow Z \cup \{Cov\}$
end for
end for
end for
return Z
end procedure

Procedure 3 – Algorithm for $MCE(\mathcal{P}, \mathcal{E})$ with unit weights

Input: A set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$, and a set of ellipses $\mathcal{E} = \{E_1, \ldots, E_m\}$.
Output: The maximum number of points that can be covered by the set of ellipses $\mathcal{E}$.

procedure $MCE(\mathcal{P}, \mathcal{E}, j = 1)$
if $j = m + 1$ **then**
return 0
end if
$ans \leftarrow 0$
for $E \in \mathcal{E}$ **do**
Let (a, b) be the shape parameters of E
$Q \leftarrow MCE_1(\mathcal{P}, a, b)$
for $Cov \in Q$ **do**
$ans \leftarrow \max\{ans, |Cov| + MCE(\mathcal{P} \setminus Cov, \mathcal{E}, j + 1)\}$ ▷ Calls the procedure for the next ellipse.
end for
end for
return ans
end procedure

CHAPTER 5

MAXIMAL COVERING BY ELLIPSES WITH ROTATION

This section introduces the covering problem that does not have any axis-parallel constraint for the ellipses, we denote this version of the problem as Maximal Covering by Ellipses with Rotation (MCER). The removal of this constraint introduces a new variable which is responsible for determining the rotation angle of every ellipse.

An instance of the non-axis-parallel is defined exactly like the axis-parallel one on Chapter 4. It is given by a set of demand points $\mathcal{P} = \{p_1, \ldots, p_n\}$, with every point having a unitary weight, and a set of ellipses $\mathcal{E} = \{E_1, \ldots, E_m\}$, with fixed shape parameters $(a_i, b_i) \in \mathbb{R}^2_{>0}$, $i \in \{1, \ldots, n\}$ with the additional condition that $a_i > b_i$. Given an instance of *MCER*, we define $Q := (q_1, \ldots, q_m) \in \mathbb{R}^{2m}$ to be the centers of each ellipse, $\Theta := (\theta_1, \ldots, \theta_m) \in [0, \pi)^m$ to be the angle of rotation of each ellipse and $E_i(q_i, \theta_i)$ to be the coverage region of ellipse E_i with its center at point q_i rotated by angle θ_i, which is given by Equation 2.10. Also, $\mathcal{P} \cap E_i(q_i, \theta_i)$ is denoted as the set of points covered by E_i on this configuration. Therefore MCER is defined as the problem of determining Q and Θ (placing and rotating each ellipse) to maximize the number of points covered by the m ellipses, which is given by Equation 5.1.

$$\max_{Q,\Theta} \left| \bigcup_{i=1}^{m} \mathcal{P} \cap E_i(q_i, \theta_i) \right|. \tag{5.1}$$

An additional notation is used on this chapter, $\tilde{E}_i(q_i, \theta_i)$ is defined to be the set of points on the border of $E_i(q_i, \theta_i)$, specially, the operation $\mathcal{P} \cap \tilde{E}_i(q_i, \theta_i)$ is used to refer to the set of points from $\mathcal{P}$ that lie on the border of $E_i(q_i, \theta_i)$.

Proposition 1. Let $(\mathcal{P}, \mathcal{E})$ be an instance of MCER. In an optimal solution of MCER, for any $E_j \in \mathcal{E}$, such that $|\mathcal{P} \cap E_j(q_j, \theta_j)| \geq 2$, there is q'_j such that $\mathcal{P} \cap E_j(q'_j, \theta_j) = \mathcal{P} \cap E_j(q_j, \theta_j)$ and $\mathcal{P} \cap \tilde{E}_j(q'_j, \theta_j) \geq 2$.

Demonstration. First, the angle of rotation can be ignored as it does not change.

Let $A = \mathcal{P} \cap E_j(q_j, \theta_j)$ be the set of points covered by E_j and $X = \cap_{p \in A} E_j(p, \theta_j)$ be the region of intersection of ellipses centered at each point from A.

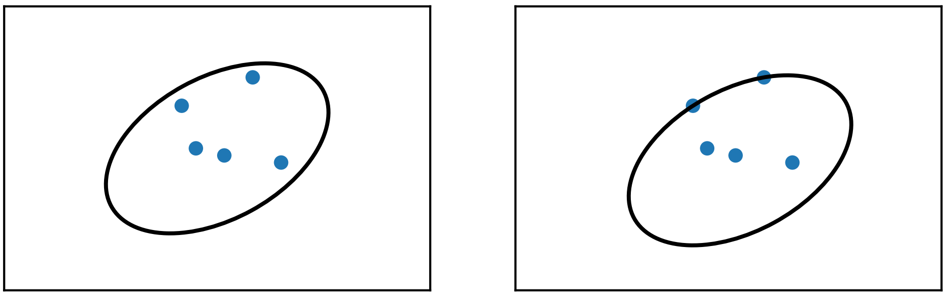
As it was shown on Chapter 4, X is a region that is limited by arcs of ellipses. As this region is the non-empty intersection of more than one ellipse, there are at least two of these arcs that encounter at one point creating a vertex. Selecting any of these vertices as q'_j will make $|\mathcal{P} \cap \tilde{E}_j(q'_j, \theta_j)| \geq 2$.

□

What Proposition 1 is saying is that any optimal solution for MCER can be transformed into another optimal solution where every ellipse covers the same set of points and those which cover more than one point has two points on their border. Also, this equivalent optimal solution can be always achieved by just translating the ellipses. An example is shown on Figure 8 for one ellipse.

A lot of the ideas developed in this chapter are based on fixing two points on the border of an ellipse, which is why Definition 7 introduces a new notation for angles that an ellipse rotated by it can be translated into a center, such that it contains two fixed points.

Figure 8 – An optimal solution before and after applying Proposition 1.



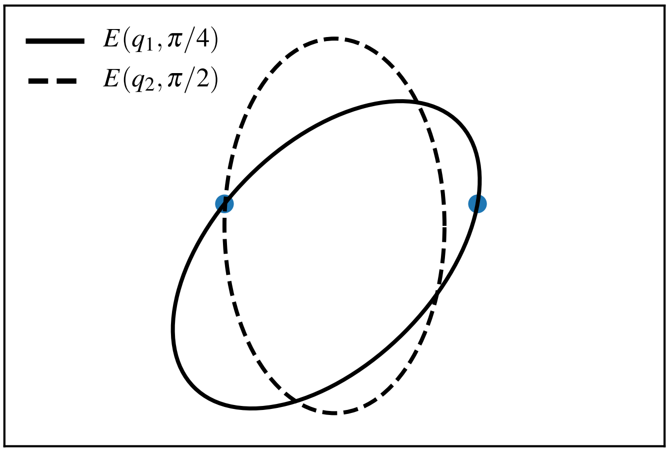
Source: Elaborated by the author.

Definition 7. Let E be an ellipse and $u, v \in \mathbb{R}^2$. An angle $\theta \in [0, \pi)$ is said to be (E, u, v)-feasible if there is $q \in \mathbb{R}^2$ such that $\{u, v\} \subset \tilde{E}(q, \theta)$.

On Figure 9 two examples for Definition 7 are shown. On one of them, an ellipse is rotated by $\pi/4$ and it is located such that it contains the two fixed points on its border. That means $\pi/4$ is a (E, u, v)-feasible angle. On the other example, the ellipse is rotated by $\pi/2$ and there is not a center where the ellipse can be placed so it contains the two fixed points–they are too far apart. This makes $\pi/2$ a not (E, u, v)-feasible angle.

Lemma 2. Let $(\mathcal{P}, \mathcal{E})$ be an instance of MCER, in an optimal solution, for any $E_j \in \mathcal{E}$, such that $|\mathcal{P} \cap E_j(q_j, \theta_j)| > 2$, at least one of the two cases is true:

Figure 9 – A (E,u,v)-feasible angle and a not (E,u,v)-feasible angle.



Source: Elaborated by the author.

1. There is q', θ', and $\{u,v,w\} \subset \mathcal{P} \cap E_j(q_j,\theta_j)$, such that $\{u,v,w\} \subset \tilde{E}(q',\theta')$.

2. Let $A = \mathcal{P} \cap E_j(q_j,\theta_j)$, and $u,v \in A$ such that there exists $\hat{q}_j$ such that $\{u,v\} \subset \tilde{E}_j(\hat{q}_j,\theta_j)$ and $A \subset E_j(\hat{q}_j,\theta_j)$. Then for any (E_j,u,v)-feasible angle $\theta \in [0,2\pi]$, there exists $\bar{q}_j$ such that $\{u,v\} \subset \tilde{E}_j(\bar{q}_j,\theta)$ and $A \subset E_j(\bar{q}_j,\theta)$.

The first case of Lemma 2 is saying that there is another optimal solution which has E_j covering the same set of points, but with three points on its border.

The second case of Lemma 2 says that after fixing a pair of points on the border of E_j maintaining the covered set, for any angle that allows the two points to stay on the border of E_j, there is a center that maintains the covered set the same.

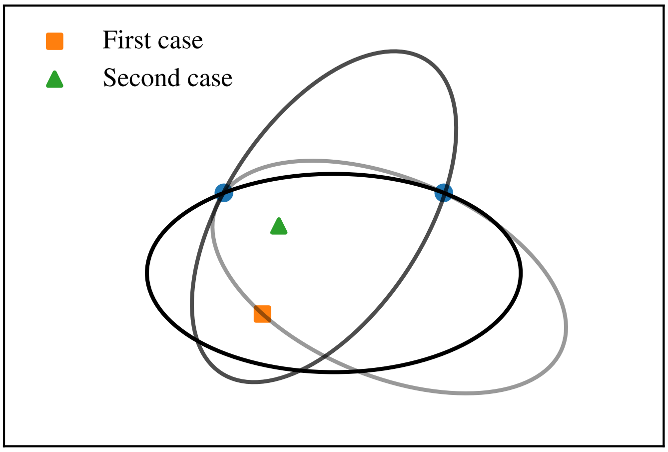
On Figure 10 both cases of Lemma 2 are shown. There, it can be seen that for the second case, it does not matter which feasible angle by which the ellipse is rotated, the third point will always be inside the coverage area. Also, an example of the first case is shown where there are three points lie exactly on the border of the ellipse.

The idea to prove Lemma 2 is that after fixing u,v on the border of E_j, which is possible by Proposition 1, the movement of rotation and translation while keeping u,v on the border is continuous. Because of that, the negation of case two implies case one and vice versa.

If we define an equivalence relation between optimal solutions as: S_1 is equivalent to S_2 if they both cover the same set of points, we can use Lemma 2 and Proposition 1 to identify the equivalence classes. Let S be any optimal solution, in $[S]$ (its equivalence class) there is another solution where any ellipse $E_j \in \mathcal{E}$ falls in at least one of the cases below:

- E_j covers only one point.

Figure 10 – An example of Lemma 2.



Source: Elaborated by the author.

- E_j covers more than one point with u and v being on the border of E_j. Note that there could be infinitely many of solutions like that, however, Lemma 2 guarantees that any (E_j, u, v)-feasible angle yields an equivalent optimal solution.
- E_j covers more than two points with three of them on its border.

The following two sections will treat the second and third cases (the first case is trivial). Going through every possibility of an ellipse falling in any of the three cases guarantees that an optimal solution is found.

5.1 Ellipse by two points

Let E be an ellipse with shape parameters $(a, b) \in \mathbb{R}^2_{>0}$ and $u, v \in \mathbb{R}^2$, one wants to find a (E, u, v)-feasible angle $\theta \in [0, \pi]$ and every center $q \in \mathbb{R}^2$ such that $\{u, v\} \subset \tilde{E}(q, \theta)$.

For a fixed angle, finding every center such that two points are on the border of the ellipse is done on Appendix A, from there we know that there could be at most 2 of such centers. The only thing left to be done is finding a feasible angle. It turns out that the angle that makes the major-axis of the ellipse to be aligned with the line that passes through u and v will be a feasible angle if, and only if the set of feasible angles is not empty. This can be seen geometrically as other angles achieve a lesser maximum distance between the two points on the border.

5.2 Ellipse by three points

CHAPTER

6

ELLIPSE BY THREE POINTS

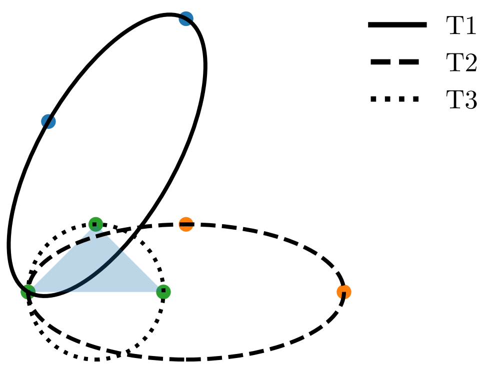
The problem of finding every center and angle of rotation of a fixed shape ellipse which makes it have three points on its border is presented in this section. Even though its simple statement–it is short and uses only basic mathematical concepts–we were not able to find any work on it, or even on related problems. As a result, starting from scratch, we ended up trying a handful of approaches with most of them failing on the way. We try to give a review of some of those going through the issues with the failing approaches, at the end we propose a $\mathcal{O}(1)$ algorithm that finds every solution of the problem and run some numerical experiments to show the precision of the found solutions.

We refer to this problem as Ellipse by Three points (E3PNT), and an instance of it is given by three points $u, v, w \in \mathbb{R}^2$ and E, an ellipse with shape parameters $(a, b) \in \mathbb{R}^2_{>0}$, with $a > b$. A solution of E3PNT is a pair $(q, \theta) \in \mathbb{R}^2 \times [0, \pi)$, such that $\{u, v, w\} \subset \tilde{E}(q, \theta)$. The goal is to develop a method to find every solution of E3PNT.

6.1 Transforming the problem

To make it simpler, let us translate the system, so the point u is at $(0, 0)$. Then, we assume that the ellipse is actually axis-parallel and the points are the ones rotating. When an angle is found such that the three points lie on the border of the axis-parallel ellipse, a linear transformation can be applied to compress the x-axis by $\frac{b}{a}$, transforming the ellipse into a circle of radius b. This transformation can be seen on Figure 11 where a solution of the E3PNT is transformed into a solution of the problem of finding a circumscribed circle of a triangle. This process can be parametrized by the angle of rotation of the points, as described by Equation 6.1, and because of the invertibility of linear transformations, solutions for E3PNT can be obtained by reversing these transformations.

Figure 11 – Transforming an ellipse into a circle. T1, T2, and T3 represent the steps of the transformation.



Source: Elaborated by the author.

$$\varphi(p,\theta) = \begin{bmatrix} \frac{b}{a} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}. \tag{6.1}$$

After that transformation, the problem to be solved is finding a circumscribed circle of the triangle formed by the points $\{(0,0),\varphi(v,\theta),\varphi(w,\theta)\}$, such that the circle has radius b. Therefore, the only unknown of this problem is the angle of rotation. This idea reduces the number of variables from 3 to only 1, however, this would only be useful if checking the existence of a circumscribed circle with radius b is a convenient problem.

It turns out that for three non-colinear fixed points, there is always an unique circumscribed circle. Also, finding this unique circle can be done analytically, hence its radius can be easily computed and compared to b when an angle is given. Let $A(\theta)$ be the area of the triangle formed by the points $\{(0,0),\varphi(v,\theta),\varphi(w,\theta)\}$–note that the transformation does not preserve distance or area. Then, the radius R of the circumscribed circle is given by Equation 6.2 (JOHNSON; YOUNG, 1960, p. 189).

$$R = \frac{\|\varphi(v,\theta)\|_2 \, \|\varphi(w,\theta)\|_2 \, \|\varphi(v,\theta) - \varphi(w,\theta)\|_2}{4A(\theta)}. \tag{6.2}$$

Imposing the radius to be equal b and squaring to eliminate the square roots present in the Euclidean distance, a function $\xi : [0,2\pi) \mapsto \mathbb{R}_{>0}$ is defined by Equation 6.3 in such a way that its zeros determine solutions to the E3PNT's instance. Two questions about $\xi(\theta)$ that arise are: is its set of roots finite? And, can they be found analytically?

$$\xi(\theta) = 16b^2 A(\theta)^2 - \|\varphi(v,\theta)\|_2^2 \, \|\varphi(w,\theta)\|_2^2 \, \|\varphi(v,\theta) - \varphi(w,\theta)\|_2^2. \tag{6.3}$$

6.1.1 The number of solutions is limited

One of the steps of the method developed on Chapter 5 is to iterate over every solution of E3PNT for every triplet of points. Of course doing that is only possible for a finite number of solutions. Also, discovering a bound for that is essential for determining the efficiency of the algorithm.

Back on Chapter 2 real trigonometric polynomials were introduced. It was stated that a n-degree polynomial can have up to $2n$ distinct roots. It turns out that ξ is a real trigonometric polynomial of degree 6 and it can be written in the format given by Equation 2.18 which implies that it can have up to 12 distinct roots. To show that, just note that it is possible to write $\|\varphi(v,\theta)\|_2^2$ and $A(\theta)^2$ in that form, as it can be seen on Equation 6.4 and Equation 6.5. It is also possible to see that the term of higher the degree of ξ is the multiplication of the three squared distances. As $\|\varphi(v,\theta)\|_2^2$ has degree 2, the degree of ξ is 6.

$$\|\varphi(v,\theta)\|_2^2 = (v_x\frac{b}{a}\cos\theta + v_y\frac{b}{a}\sin\theta)^2 + (v_y\cos\theta - v_x\sin\theta)^2 \tag{6.4}$$

$$A(\theta)^2 = \frac{1}{4}\det\begin{pmatrix} v_x\frac{b}{a}\cos\theta + v_y\frac{b}{a}\sin\theta & v_y\cos\theta - v_x\sin\theta \\ w_x\frac{b}{a}\cos\theta + w_y\frac{b}{a}\sin\theta & w_y\cos\theta - w_x\sin\theta \end{pmatrix}^2 \tag{6.5}$$

Because ellipses are symmetrical with respect to their major-axis, and any rotation in the interval $[0,\pi)$ is identical to a rotation in $[\pi,2\pi)$, the number of different solutions is cut in half. Therefore, the number of angles of rotation and centers that an ellipse of fixed shape can be placed, so it has three fixed points on its border is limited to 6.

6.2 An attempt using the conic general equation

The idea of this approach was to use the six-parameter conic equation to represent an ellipse. This equation is given by Equation 6.6.

$$Ax^2 + Bxy + Cy^2 + Dy + Ex + F = 0. \tag{6.6}$$

This equation actually represents any conic, for it to be an ellipse the condition $B^2 - 4AC < 0$ must be satisfied.

Setting the first point to be the origin, we get $F = 0$, using the other two points, it is possible to write D and E in terms of A,B,C. As any multiple of Equation 6.6 represents the same conic, we can set B to be equal 1. Then, we end up with two variables, A and C, and still need to impose that the final equation represents an ellipse and its major-axis and minor-axis have the predefined value. Let $\Delta = 4AC - B^2 = 4AC - 1$, Equation 6.7 and Equation 6.8 for both major-axis and minor-axis respectively, assuming $F = 0$.

$$a^2 = \frac{2\dfrac{AE^2 - BDE + CD^2}{\Delta}}{A + C - \sqrt{1 + (A - C)^2}} \tag{6.7}$$

$$b^2 = \frac{2\dfrac{AE^2 - BDE + CD^2}{\Delta}}{A + C + \sqrt{1 + (A - C)^2}} \tag{6.8}$$

These two equations define two curves in $\mathbb{R}^2$ with A and C being the chosen variables. The solutions lie in the set of intersection of these curves. Finding this set was judged to be non-trivial and probably could be approximated numerically, however, we decided not to further pursue this approach.

Another idea which has been explored was working with the ratio $\frac{a^2}{b^2}$ which becomes an expression that allows A to be written as a function of C. This function appeared, at first we thought, to be monotonic, we tried to develop a method based on that, however, cases where the function does not behave as nicely were found. It is likely that developing a method to approximate solutions working with this function is possible, but we decided not to continue on this track.

6.3 An approximation method

One of the most useful techniques when dealing with complicated functions is approximation. They appear in various methods whenever a derivative or integral needs to be calculated or for example, in our case, when the roots of a function need to be determined. In general, one has a function f that is part of a family of functions $\mathscr{A}$ and wants to select a simpler function f^* from a set of functions $\mathscr{A}^*$, such that f^* is close enough to f (POWELL, 1981, p. 3). For this problem, the approximation of $\xi(\theta)$ on the interval $[0, \pi)$ is considered. The approximation set of functions is going to be the set of n-degree Chebyshev polynomials which the roots can be found through determining the eigenvalues of a n by n matrix.

6.3.1 Chebyshev interpolation

Chebyshev polynomials are widely used in Numerical Analysis in areas like numerical integration, polynomial approximation, and ordinary and partial differential equations. They are also very useful in practice and are present in extension libraries in Python and MATLAB.

Because of the scope of this work, only a brief introduction of Chebyshev polynomials of the first kind and its usage in polynomial interpolation is given. For a more thorough work on the subject, please check the book by Mason and Handscomb (2003).

We refer to $T_n : [-1,1] \mapsto [-1,1]$ as the n-degree Chebyshev polynomial of the first kind, and it is defined as follows:

$$T_n(x) = \cos(n \arccos x) \tag{6.9}$$

It is important to mention that this definition can be extended to the whole real line. Using some trigonometric identities, T_n can also be expressed as a recurrence relation:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x). \tag{6.10}$$

An important property worth bringing up is that Chebyshev polynomials are orthogonal and form a basis for the polynomial space. This implies that any p_n of degree up to n can be expressed as a truncated Chebyshev series:

$$p_n(x) = \sum_{j=0}^{n} a_j T_j(x). \tag{6.11}$$

One of the greatest qualities of Chebyshev polynomials is its numerical stability. Gautschi (1979) showed that the matrix that maps polynomials onto its coefficients written in the power form[1] has a condition number that grows exponentially with n. On the other hand, the matrix that converts polynomials to the Chebyshev basis as Equation 6.11, has a linear condition number bounded by $\sqrt{2}n$.

Polynomial interpolation is a form of approximating a function by a polynomial of degree n that passes through $n+1$ chosen points. In fact, this polynomial is unique and it is determined by Lagrange's formula:

$$f_n(x) = \sum_{j=0}^{n} f(x_j) \frac{\prod_{k \neq j}^{n+1}(x - x_k)}{\prod_{k \neq j}^{n+1}(x_j - x_k)}, \tag{6.12}$$

with f being the function to be approximated, and f_n the unique n-degree polynomial that passes through $\{(x_j, f(x_j)) : j = 0, 1, \dots n\}$. Because of the uniqueness of interpolant polynomials, there is a direct link between the quality of an approximation and the points chosen to interpolate. As a matter of fact, depending on the points one chooses, even increasing the degree of the interpolation makes the approximation worsen. This is known as Runge's phenomenon and an example can be seen in Powell (1981, p. 37) where uniformly spaced points are chosen to interpolate the function $f(x) = (1+x^2)^{-1}$ on the interval $[-5, 5]$.

That is where Chebyshev interpolation comes in. Instead of choosing $n+1$ arbitrary points, the $n+1$ roots of T_{n+1}, which are also known as Chebyshev Nodes, are chosen as the

[1] A polynomial is in the power form or the monomial form if it can be written as $\sum_{j=0}^{n} a_j x^j$

interpolation points:

$$x_j = \cos\left(\frac{\pi(k-\frac{1}{2})}{n+1}\right), \tag{6.13}$$

for $j = 1,\dots,n+1$. This particular choice defeats Runge's phenomenon and provides a convergent approximation.

Note that, if the domain of the function to be interpolated is defined on a range other than $[-1,1]$, let us say $[a,b]$, then a transformation can be done to map it to the Chebyshev Nodes' domain:

$$\hat{x}_j = \frac{a+b}{2} + \frac{b-a}{2}x_j. \tag{6.14}$$

Then, the Chebyshev interpolation of a function $f : [a,b] \mapsto \mathbb{R}$ can be determined using Lagrange's formula and the points $\hat{x}_1,\dots,\hat{x}_n$. As it was mentioned, finding the roots of a polynomial written in the monomial form can be done by determining the eigenvalues of a so-called Frobenius companion matrix. For small n this works fine, however, converting the polynomial obtained by Equation 6.12 to the power form, as n grows, becomes a very ill-conditioned problem. An alternative method can be found in Boyd (2013) where the Chebyshev interpolation is calculated directly as a truncated Chebyshev series, as in Equation 6.11, in $\mathcal{O}(n^2)$. Also, given a polynomial written in the Chebyshev basis, a $n \times n$ matrix can be constructed, such that its eigenvalues are the roots of that polynomial. Boyd (2013) refers to this matrix as the Chebyshev-Frobenius companion matrix.

Therefore, the whole process of interpolating and finding the roots can be done using only Chebyshev polynomials, which have great numerical stability. Also, Chebyshev-Frobenius matrices have the same property as companion matrices which allows their eigenvalues to be found by a QR decomposition. Summing the two steps, a $\mathcal{O}(n^2)$ algorithm can be achieved.

The last question that needs to be addressed is how close the roots of the Chebyshev interpolant f_n are to the roots of ξ?

Even though ξ is complicated enough, in a sense that finding its roots directly is no trivial task, it is a very well-behaved function: it is analytic and has infinitely many continuous and integrable derivatives. This satisfy all the requirements of the result in Gottlieb and Orszag (1977, p. 28) which says that if a function has m continuous and integrable derivatives on a closed interval, then its absolute difference between the Chebyshev truncate series is $\mathcal{O}(n^{-m})$. Also, in Battles and Trefethen (2004) a theorem is presented stating that if a function is analytic on a neighborhood of $[-1,1]$, then the convergence is $\mathcal{O}(C^n)$, for some $C < 1$.

To choose the degree of the interpolation we use the last coefficient rule-of-thumb introduced by Boyd (2001, p. 50). There is no guarantee that this method will choose n, such that f_n is close enough to ξ everywhere on $[0,\pi)$, nonetheless, in practice, it is taken to be a good estimate for the error r_n:

$$r_n = \max_{0 \le \theta \le \pi} |f_n(\theta) - \xi(\theta)|. \tag{6.15}$$

6.4 Converting ξ into a polynomial

On Chapter 2 a brief introduction is given on how to get the roots of a polynomial. For that reason, we discuss two ways of converting ξ into a polynomial in this section. Symbolic computing was used to compute the polynomials, the Python external library called SymPy was utilized (see Meurer *et al.* (2017) for more details).

The first attempt was using the identity $x = \tan\frac{\theta}{2}$ from which it is possible to construct a 12-degree polynomial. At first, the root-finding algorithm described on Chapter 2 seemed to work fine and return every solution of E3PNT, however, we later found out that for some instances, priorly known roots were not being found. The cause was not for sure identified, but a good guess would be that for angles which are greater than $\frac{\pi}{4}$, x starts growing too rapidly which could lead to numerical instability.

The second approach is based on a idea published on Boyd (2006) which uses the identities on Equation 6.16 to convert real trigonometric polynomials into univariate complex polynomials in order to obtain its roots using the companion matrix scheme. This approach is preferable as it preserves the numerical stability of the original real trigonometric polynomial– more details about this can be found in Weidner (1988) where it is said that computing the roots of a real trigonometric polynomial through this transformation does not yield loss of accuracy.

$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2} \tag{6.16}$$

$$\sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}. \tag{6.17}$$

It is possible to show that with that substitution and changing the variable to $z = e^{i\theta}$, we obtain the following function $g : \mathbb{S} \mapsto \mathbb{C}$, with $\mathbb{S}$ being the unit complex circle ($\mathbb{S} = \{z \in \mathbb{C} : |z| = 1\}$):

$$g(z) = \sum_{k=0}^{12} c_k z^{k-6}, \tag{6.18}$$

for some $c_0, \ldots, c_{12} \in \mathbb{C}$. As the equalities on Equation 6.16 are valid for any $\theta \in \mathbb{R}$, the function $g'(\theta) = g(e^{i\theta})$ and the function ξ are equivalent since $g(e^{i\theta}) = \xi(\theta)$ for any $\theta \in [0, 2\pi)$.

Now there is only two things preventing g from being a polynomial: firstly, z shows up with negative exponents up to -6, secondly its domain is the unit circle, not the whole complex set $\mathbb{C}$. The first issue can be fixed by multiplying g by z^6, note that this does not create further problems as $0 \notin \mathbb{S}$. The second issue is removed by extending the domain to $\mathbb{C}$, and considering only roots that lie on $\mathbb{S}$, notice also that no roots can go missing as $\mathbb{S} \subset \mathbb{C}$. Finally, the polynomial $h : \mathbb{C} \mapsto \mathbb{C}$ is defined as follows:

$$h(z) = z^6 g(z) = \sum_{k=0}^{12} c_k z^k. \tag{6.19}$$

By its definition is possible to see that every root of g is also a root of h, and conversely, every root $\hat{z}$ of h which is in $\mathbb{S}$ is also a root of g. Naturally, given a root $\hat{z}$ of g, the angle of $\hat{z}$ is a root of ξ by Euler's Formula used in the substitution $z = e^{i\theta}$.

It is possible to make another reduction to obtain a degree-6 polynomial, lowering in half the number of rows of the companion matrix utilized on obtaining the roots of h. As it has been mentioned before, an ellipse is symmetric with respect to its own axis. This means that θ and $\pi + \theta$ are equivalent angles of rotation for any ellipse, thus $\xi(\theta) = \xi(\pi + \theta)$. On Chapter 2, it was stated that the angle of z and $-z$ has the same symmetry with each other as an ellipse's angle of rotation:

$$angle(-z) = \pi + angle(z).$$

From that, as $g(e^{i\theta}) = \xi(\theta)$ for every $\theta \in [0, 2\pi)$, we conclude that $g(-z) = g(z)$. This implies that h is, in fact an even polynomial, or that $h(-z) = h(z)$ is true for every $z \in \mathbb{C}$:

$$h(-z) = (-z)^6 g(-z) = z^6 g(z). \tag{6.20}$$

Therefore, all the odd degree coefficients of h are 0 and we can define the 6-degree polynomial $f : \mathbb{C} \mapsto \mathbb{C}$ with the substitution $y = z^2$:

$$f(y) = \sum_{k=0}^{6} c_{2k} y^k. \tag{6.21}$$

Then from every root $\hat{y}$ of f, two roots of h can be obtained: $\sqrt{\hat{y}}$ and $-\sqrt{\hat{y}}$. As the angle of $-\sqrt{\hat{y}}$ is not between $[0, pi)$ we can ignore it. Note that the the square root of $\hat{y}$ does not need to be calculated, as only the angles are needed and they can be obtained by:

$$angle(\sqrt{z}) = angle(z)/2. \tag{6.22}$$

Finally, using the QR algorithm mentioned on Chapter 2 a $\mathcal{O}(n^3)$ algorithm, with $n = 6$, can be constructed for E3P.

It is also worth mentioning that a pattern on the coefficients of f was identified, and maybe for future work it can be used for further improvements. Analyzing the polynomials produced for several instances, the following seems to be true:

$$c_k = \overline{c_{6-k}}, \tag{6.23}$$

for $k = 0, \ldots, 6$. For now, we do not have any ideas on how it could be proved. Nevertheless, this seems to lead somewhere interesting because this condition guarantees that f is a self-reciprocal polynomial which implies that its roots will always come in pairs $(\hat{z}, 1/\hat{z})$.

6.5 Numerical Experiments

In this section we run some experiments to verify how big the interpolation degree must be for a good precision to be achieved. Let f_n be the n-degree Chebyshev interpolation of ξ, we define the interpolation error r_n as:

Then, we want to determine the smallest n, such that $r_n \leq \varepsilon$, for a predefined ε. Unfortunately, calculating r_n involves taking samples from both functions on the whole interval, which is not viable computationally. In Boyd (2001, p. 50) a rule-thumb for estimating the interpolation error is given. It is worth mentioning that this rule

It uses a theorem that states that r_n is limited by the sum of the coefficients of the Chebyshev series that were removed by the truncation. The rule-of-thumb for estimating r_n is given by:

$$r_n \approx |a_n|. \tag{6.24}$$

Note that

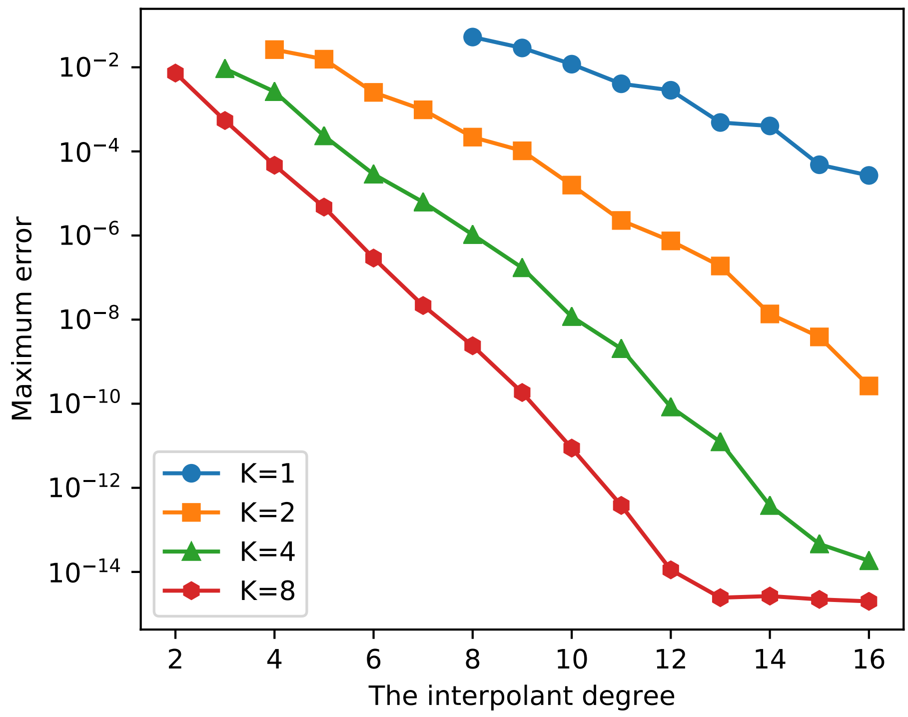
Obtaining an expression for it is not trivial, and sampling the whole interval is not viable computationally.

Therefore, an estimation of r_n is used to measure how good is the approximation.

Also we adopt two suggestions from (BOYD, 2013). The first is to divide the interval into K subintervals to achieve a precision without having to increase the degree of the interpolant too much. The second is to use a couple of Newton's iterations to refine the roots found.

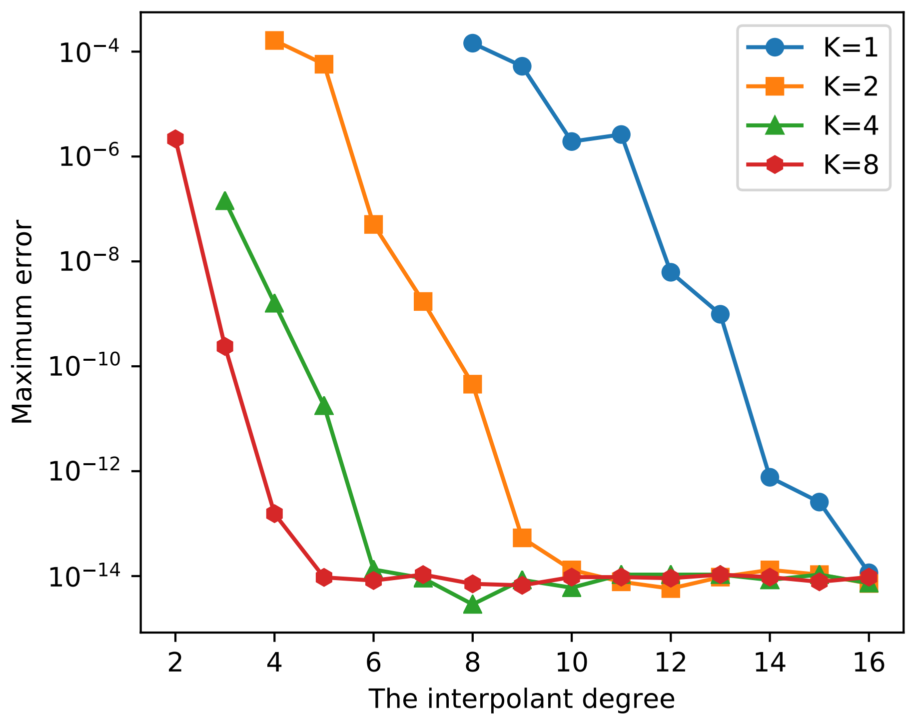
Let θ^* be a root of f_n, then we measure the error associated with that root as $|\xi(\theta^*)|$. For the numerical experiments, we considered every triplet of points from an instance with 25 points. Then for some values of K and n we define the error as the maximum error for every root that were found.

Figure 12 – The interpolation error measured on roots that were found.



Source: Elaborated by the author.

Figure 13 – The interpolation error measured on roots that were found after three Newton's iterations.



Source: Elaborated by the author.

CHAPTER

7

FUTURE WORK

To take advantage of the great amount of work found in the literature, we decided to first introduce the planar maximal covering by disks problem, develop a method for it, and just then adapt it for the ellipses case. It turned out that because of the similarities between the two problems, adapting was possible and actually very simple. This made the method developed by us have a very different approach than the ones in (ANDRETTA; BIRGIN, 2013) and (CANBOLAT; MASSOW, 2009). The next step is to implement it and compare the results that (ANDRETTA; BIRGIN, 2013) obtained.

For the next step of our master's research we set the following objectives as primary:

- Study the $(1-\varepsilon)$-approximation method for the planar covering with disks in (BERG; CABELLO; HAR-PELED, 2006) and develop an adapted version of the algorithm for ellipses with the same time complexity of $\mathcal{O}(n \log n)$.
- Develop an exact method for the version of the problem introduced in (ANDRETTA; BIRGIN, 2013) where the ellipses can be freely rotated.

The following goals are set as secondary:

- Develop a probabilistic approximation algorithm based on (ARONOV; HAR-PELED, 2008) which proposed a Monte Carlo approximation for the problem of finding the deepest point in a arrangement of regions. The method runs in $\mathcal{O}(n\varepsilon^2 \log n)$ and can be applied to solve the case with one ellipse. The case with more than one ellipse is left as a challenge for us for the next steps of our research.
- In (HE *et al.*, 2015), the task of finding every center candidate, after eliminating all the non-essential ones, is done in $\mathcal{O}(n^5)$ run-time complexity. We want to generalize this for the elliptical distance function and achieve a better run-time complexity. We also intend to use the mean-shift algorithm to try to develop a greedy version for the ellipses version.

BIBLIOGRAPHY

ANDERSON, E.; BAI, Z.; BISCHOF, C.; BLACKFORD, S.; DEMMEL, J.; DONGARRA, J.; CROZ, J. D.; GREENBAUM, A.; HAMMARLING, S.; MCKENNEY, A.; SORENSEN, D. **LAPACK Users' Guide**. Third. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999. ISBN 0-89871-447-8 (paperback). Citation on page 21.

ANDRETTA, M.; BIRGIN, E. Deterministic and stochastic global optimization techniques for planar covering with ellipses problems. **European Journal of Operational Research**, v. 224, n. 1, p. 23 – 40, 2013. ISSN 0377-2217. Available: <http://www.sciencedirect.com/science/article/pii/S0377221712005619>. Citations on pages 14, 31, and 49.

ARONOV, B.; HAR-PELED, S. On approximating the depth and related problems. **SIAM J. Comput.**, v. 38, n. 3, p. 899–921, 2008. Available: <https://doi.org/10.1137/060669474>. Citations on pages 14 and 49.

AYOUB, A. B. The central conic sections revisited. **Mathematics Magazine**, Mathematical Association of America, v. 66, n. 5, p. 322–325, 1993. ISSN 0025570X, 19300980. Available: <http://www.jstor.org/stable/2690513>. Citation on page 16.

BAREL, M. V.; VANDEBRIL, R.; DOOREN, P. V.; FREDERIX, K. Implicit double shift qr-algorithm for companion matrices. **Numerische Mathematik**, v. 116, n. 2, p. 177–212, Aug 2010. ISSN 0945-3245. Available: <https://doi.org/10.1007/s00211-010-0302-y>. Citation on page 21.

BATTLES, Z.; TREFETHEN, L. N. An extension of MATLAB to continuous functions and operators. **SIAM Journal on Scientific Computing**, SIAM, v. 25, n. 5, p. 1743–1770, 2004. Citation on page 44.

BENTLEY, J. L.; OTTMANN, T. A. Algorithms for reporting and counting geometric intersections. **Computers, IEEE Transactions on**, C-28, p. 643 – 647, 10 1979. Citations on pages 26 and 27.

BERG, M. de; CABELLO, S.; HAR-PELED, S. Covering many or few points with unit disks. In: . [S.l.: s.n.], 2006. v. 45, p. 55–68. Citations on pages 14, 24, 26, 28, 29, and 49.

BOYD, J. Finding the zeros of a univariate equation: Proxy rootfinders, chebyshev interpolation, and the companion matrix. **SIAM Review**, v. 55, 01 2013. Citations on pages 44 and 47.

BOYD, J. P. **Chebyshev and Fourier Spectral Methods**. Second. Mineola, NY: Dover Publications, 2001. (Dover Books on Mathematics). ISBN 0486411834 9780486411835. Citations on pages 44 and 47.

____. Computing the zeros, maxima and inflection points of chebyshev, legendre and fourier series: solving transcendental equations by spectral interpolation and polynomial rootfinding. **Journal of Engineering Mathematics**, v. 56, n. 3, p. 203–219, Nov 2006. ISSN 1573-2703. Available: <https://doi.org/10.1007/s10665-006-9087-5>. Citations on pages 21 and 45.

BRANNAN, D.; ESPLEN, M.; GRAY, J. **Geometry**. Cambridge University Press, 1999. ISBN 9781107393639. Available: <https://books.google.co.id/books?id=HbytAQAAQBAJ>. Citation on page 16.

CANBOLAT, M. S.; MASSOW, M. von. Planar maximal covering with ellipses. **Computers and Industrial Engineering**, v. 57, p. 201–208, 2009. Citations on pages 14, 31, and 49.

CHAZELLE, B. M.; LEE, D. On a circle placement problem. **Computing**, v. 36, p. 1–16, 03 1986. Citations on pages 14, 23, 24, 25, and 28.

CHURCH, R. The planar maximal covering location problem. **Journal of Regional Science**, v. 24, p. 185–201, 1984. Citation on page 13.

CHURCH, R.; VELLE, C. R. The maximal covering location problem. **Papers in Regional Science**, v. 32, n. 1, p. 101–118, 1974. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1435-5597.1974.tb00902.x>. Citation on page 13.

CLARK, B.; COLBOURN, C.; JOHNSON, D. Unit disk graphs. **Discrete Mathematics**, Elsevier, v. 86, n. 1-3, p. 165–177, 12 1990. ISSN 0012-365X. Citation on page 24.

DE, M.; NANDY, S. C.; ROY, S. In-place algorithms for computing a largest clique in geometric intersection graphs. **Discrete Applied Mathematics**, v. 178, p. 58 – 70, 2014. ISSN 0166-218X. Available: <http://www.sciencedirect.com/science/article/pii/S0166218X1400300X>. Citations on pages 24, 25, and 26.

DREZNER, Z. Note—on a modified one-center model. **Management Science**, v. 27, p. 848–851, 07 1981. Citations on pages 14, 23, 24, and 26.

GAUTSCHI, W. The condition of polynomials in power form. **Mathematics of Computation - Math. Comput.**, v. 33, 01 1979. Citation on page 43.

GOTTLIEB, D.; ORSZAG, S. **Numerical Analysis of Spectral Methods: Theory and Applications**. Society for Industrial and Applied Mathematics, 1977. (CBMS-NSF Regional Conference Series in Applied Mathematics). ISBN 9781611970425. Available: <https://books.google.com.br/books?id=7afHrqGFjSoC>. Citation on page 44.

HATTA, W.; LIM, C. S.; ABIDIN, A. F. Z.; AZIZAN, M.; TEOH, S. S. Solving maximal covering location with particle swarm optimization. **International Journal of Engineering and Technology**, v. 5, p. 3301–3306, 08 2013. Citation on page 13.

HE, Z.; FAN, B.; CHENG, T. C. E.; WANG, S.-Y.; TAN, C.-H. A mean-shift algorithm for large-scale planar maximal covering location problems. **European Journal of Operational Research**, v. 250, 09 2015. Citations on pages 29 and 49.

HORN, R. A.; JOHNSON, C. R. (Ed.). **Matrix Analysis**. New York, NY, USA: Cambridge University Press, 1986. ISBN 0-521-30586-1. Citation on page 20.

JOHNSON, R.; YOUNG, Y. **Advance Euclidean Geometry (modern Geometry): An Elementary Treatise on the Geometry of the Triangle and the Circle**. Dover, 1960. (Dover books on advanced mathematics). Available: <https://books.google.com.br/books?id=HdCjnQEACAAJ>. Citation on page 40.

KARATAS, M.; RAZI, N.; TOZAN, H. A comparison of p-median and maximal coverage location models with q–coverage requirement. **Procedia Engineering**, v. 149, p. 169–176, 12 2016. Citation on page 13.

KARP, R. Reducibility among combinatorial problems. In: MILLER, R.; THATCHER, J. (Ed.). **Complexity of Computer Computations**. [S.l.]: Plenum Press, 1972. p. 85–103. Citation on page 13.

KOPELOWITZ, T.; PETTIE, S.; PORAT, E. **Higher Lower Bounds from the 3SUM Conjecture**. 2014. Citation on page 23.

MASON, J. C.; HANDSCOMB, D. C. **Chebyshev Polynomials**. Boca Raton, FL: Chapman & Hall/CRC, 2003. xiv+341 p. ISBN 0-8493-0355-9. Citation on page 42.

MEURER, A.; SMITH, C. P.; PAPROCKI, M.; ČERTÍK, O.; KIRPICHEV, S. B.; ROCKLIN, M.; KUMAR, A.; IVANOV, S.; MOORE, J. K.; SINGH, S.; RATHNAYAKE, T.; VIG, S.; GRANGER, B. E.; MULLER, R. P.; BONAZZI, F.; GUPTA, H.; VATS, S.; JOHANSSON, F.; PEDREGOSA, F.; CURRY, M. J.; TERREL, A. R.; ROUČKA, Š.; SABOO, A.; FERNANDO, I.; KULAL, S.; CIMRMAN, R.; SCOPATZ, A. Sympy: symbolic computing in python. **PeerJ Computer Science**, v. 3, p. e103, Jan. 2017. ISSN 2376-5992. Available: <https://doi.org/10.7717/peerj-cs.103>. Citation on page 45.

POWELL, M. J. D. M. J. D. Book; Book/Illustrated. **Approximation theory and methods**. [S.l.]: Cambridge [England] ; New York : Cambridge University Press, 1981. Includes index. ISBN 0521295149. Citations on pages 21, 42, and 43.

QUILES, S. G.; MARíN, A. Covering location problems. In: ____. [S.l.: s.n.], 2015. p. 93–114. ISBN 978-3-319-13110-8. Citation on page 13.

REVELLE, C.; SCHOLSSBERG, M.; WILLIAMS, J. Solving the maximal covering location problem with heuristic concentration. **Comput. Oper. Res.**, Elsevier Science Ltd., Oxford, UK, UK, v. 35, n. 2, p. 427–435, Feb. 2008. ISSN 0305-0548. Available: <http://dx.doi.org/10.1016/j.cor.2006.03.007>. Citation on page 13.

SKOPENKOV, A. A short elementary proof of the insolvability of the equation of degree 5. **arXiv preprint arXiv:1508.03317**, 2015. Citation on page 20.

WATKINS, D. S. The qr algorithm revisited. **SIAM Rev.**, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 50, n. 1, p. 133–145, Feb. 2008. ISSN 0036-1445. Available: <http://dx.doi.org/10.1137/060659454>. Citation on page 21.

WEIDNER, P. The durand-kerner method for trigonometric and exponential polynomials. **Computing**, v. 40, n. 2, p. 175–179, Jun 1988. ISSN 1436-5057. Available: <https://doi.org/10.1007/BF02247945>. Citation on page 45.

YOUNIES, H.; ZEIDAN, H. Planar maximal covering location problem under different block norms [microform]. 02 2019. Citation on page 13.

APPENDIX

A

INTERSECTIONS OF TWO ELLIPSES

In this appendix the intersection of two ellipses with the same shape parameters $(a,b) \in \mathbb{R}^2_{>0}$ is described with more detail, as well as determining the functions $\Gamma_+(i,j)$ and $\Gamma_-(i,j)$ for two ellipses that intersect.

A.1 Intersection

Let E_1 and E_2 be two ellipses that the intersection will be determined here. Without loss of generality, let us assume that E_1 is at the origin and E_2 is located at the center $(h,k) \in \mathbb{R}^2$. Their equations are given by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \qquad (E_1)$$

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1 \qquad (E_2)$$

As they are both equal to one we can get the following

$$\begin{aligned} b^2x^2 + a^2y^2 &= b^2(x-h)^2 + a^2(y-k)^2 \\ b^2(-2xh+h^2) + a^2(-2yk+k^2) &= 0 \\ x(2hb^2) &= b^2h^2 + a^2(-2yk+k^2) \\ x &= y\frac{-2ka^2}{2hb^2} + \frac{b^2h^2+a^2k^2}{2hb^2} \end{aligned}$$

Which can be rewritten as

$$x = y\alpha + \beta$$

with the constants α and β being

$$\alpha = \frac{-2ka^2}{2hb^2}$$
$$\beta = \frac{b^2h^2 + a^2k^2}{2hb^2}$$

Then replacing it back to the equation of E_1 we get

$$\frac{(y\alpha + \beta)^2}{a^2} + \frac{y^2}{b^2} = 1$$
$$b^2(y\alpha + \beta)^2 + y^2a^2 - a^2b^2 = 0$$
$$y^2(b^2\alpha^2 + a^2) + y(2\beta\alpha b^2) + b^2\beta^2 - a^2b^2 = 0$$

Which is a second degree polynomial, therefore, E_1 and E_2 intersect if, and only if the roots of the polynomial are real. The intersection points itself can be obtained by solving the polynomial for y and applying its value onto the $x = y\alpha + \beta$ equation.

A.1.1 Determining $\Gamma_+(i,j)$ and $\Gamma_-(i,j)$

Let us assume that E_1 and E_2, each one with shape parameters $(a,b) \in \mathbb{R}^2_{>0}$, intersect at p_1 and p_2. Then, to determine $\Gamma_+(1,2)$ and $\Gamma_-(1,2)$, we need to first determine the angles of intersection of p_1 and p_2 on E_1. For that, we will use the curve defined in Equation 2.8 because it is easier to work with angles here.

Given a point (x,y), to find the angle it makes with the major axis, from Equation 2.8, we can get that

$$\frac{y - q_y}{x - q_x} = \frac{b}{a}\tan t$$
$$t = \arctan\left(\frac{a}{b}\frac{y - q_y}{x - q_x}\right)$$

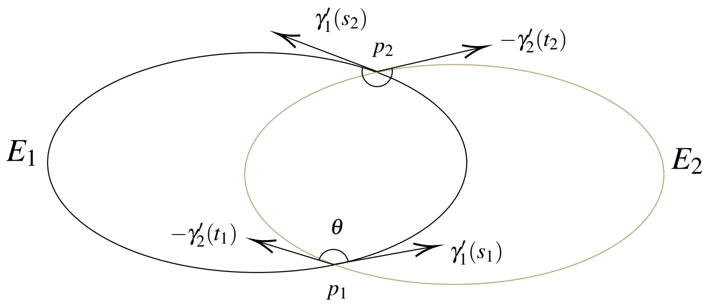
As the image of arctan is $[-\frac{\pi}{2}, \frac{\pi}{2}]$, we need to check the sign of $x - q_x$ to determine the angle in $[0, 2\pi]$. After that, we can get the two angles that represent the intersection points p_1 and p_2 on E_1.

To find out which one of the angles are $\Gamma_+(1,2)$, we need to go further and determine the derivative of $\gamma(t)$ which is going to be used to determine the vectors tangent to the ellipses at the intersection points.

$$\gamma'(t) = \begin{cases} x'(t) = -a\sin t \\ y'(t) = b\cos t \end{cases} \tag{A.1}$$

Let γ_1 and γ_2 be the curves describing E_1 and E_2 respectively. Also, let s_1 be the angle, such that $\gamma_1(s_1) = p_1$, and t_1 be the angle, such that $\gamma_2(t_1) = p_1$. Then, the tangent vectors to the E_1 and E_2 at p_1 are $\gamma_1'(s_1)$ and $\gamma_2'(t_1)$ respectively.

Figure 14 – Determining $\Gamma_+(1,2)$



Source: Elaborated by the author.

The following lemma states a relation between s_1 and $\Gamma_+(1,2)$

Lemma 3. Let θ be the angle between $\gamma_1'(s_1)$ and $-\gamma_2'(t_1)$. Then, $\theta \leq \pi$ if, and only if $\Gamma_+(1,2) = s_1$.

Instead of a formal proof of Lemma 3, a graphical explanation using Figure 14 is provided.

First, let us state some facts that can also be seen in Figure 14

- $E_1 \cap E_2$ is convex and bounded by two arcs, one from each ellipse.
- Starting at any of the intersection points, one of the $E_1 \cap E_2$ arcs will be clockwise-oriented and the other, counter-clockwise-oriented. In Figure 14, for example, it is clear that only the E_1 arc starting at p_1, ending at p_2, is counter-clockwise-oriented.
- The counter-clockwise-oriented arc starting at $\Gamma_+(1,2)$ is from the ellipse E_1.

Let us assume that p_1 is the intersection point which is the opening angle $\Gamma_+(1,2)$. Then, the vectors $\gamma_1'(s_1)$ and $-\gamma_2'(t_1)$ are tangent to the $E_1 \cap E_2$ area at point p_1. Because of the convexity of $E_1 \cap E_2$, the angle between $\gamma_1'(s_1)$ and $-\gamma_2'(t_1)$ has to be less than or equal to π (see Figure 14), which is what Lemma 3 says. It is easy to prove the converse by proving the contra-positive assuming that p_1 is the point which determines the angle $\Gamma_-(1,2)$.

Lastly, in Figure 14, it can be seen that if one the intersection points is classified as $\Gamma_+(1,2)$ the other will necessarily be classified as $\Gamma_-(1,2)$. This gives us all we need to implement Algorithm 3.

ICMC USP
SÃO CARLOS