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. 69 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

The Minimum Cover Problem –also known as just the Set Cover Problem–, and the Maximal Covering Problem are the two main types of optimal covering problems found in the literature (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 along with a collection of subsets of the demand set, the problem is to determine 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 and Velle (1974). In this first study, the author defined the problem on graphs, and the objective was to maximize the coverage of a demand set, which was a subset of the graph's vertices, by choosing the location of a facility set, which covered any vertex within a given coverage radius.

Just like the Minimum Cover problem, MCLP is NP-Hard (HATTA *et al.*, 2013) and both deterministic, using integer programming, and heuristic methods have been proposed to solve it. A very complete survey of developments in this area can be found in ReVelle, Eiselt and Daskin (2008).

In Church (1984) a new kind of MCLP named Planar Maximal Covering Location Problem (PMCLP) was introduced. Unlike its predecessor, this version of the problem was not defined on graphs. Instead, on PMCLP the demand set and the facilities are located in $\mathbb{R}^2$ and a facility's coverage area is determined by a distance function. Initially, the Euclidean distance was

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

considered and the idea behind the method proposed in Church (1984) was to convert PMCLP into an instance of MCLP and then utilize any of the previous developed exact methods to obtain a solution for PMCLP. This reduction was done by identifying a Candidate Locations Set (CLS) which represented the possible locations that needed to be evaluated for every facility, such that the optimal solution could be found. From the CLS, a network was built on which MCLP could be applied. Generating the CLS specifically for the case of Euclidean distance will be described here on Chapter 3.

Furthermore, some variations of PMCLP can also be found in the literature: in Younies and Wesolowsky (2007) PMCLP was studied under the block norm distance, in Craparo *et al.* (2019) a mean-shift algorithm for large scale[2] PMCLP was proposed, and in Bansal and Kianfar (2017) a version with partial coverage and rectangular demand and facility zones was introduced.

PMCLP under Euclidean distance is also found in the literature as the Maximum Covering by Disks (MCD) problem. Early works only tackled the one-disk version of it. In Chazelle and 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 disks version of MCD was studied in Berg, Cabello and 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 and Har-Peled (2008) one-disk maximum covering is proven to be 3SUM-HARD. This means that maximizing the number of points covered by a disk is as hard as finding three real numbers that sum to zero among n given real numbers.

Two versions of PMCLP are studied in this work, both of them related to ellipses. The first one was introduced in Canbolat and Massow (2009) and will be referred to here as Planar Maximal Covering by Ellipses (PMCE). This version does not allow the ellipses to rotate, so basically it only differs from its disks counterpart in the shape of the facility's coverage area. The second version was introduced in Andretta and Birgin (2013) and will be referred to here as Planar Maximal Covering by Ellipses with Rotation (PMCER). This version of the problem had no restriction on the rotation of ellipses–note that this is a slightly bigger modification of PMCLP compared to the first one, as not only the location for the facilities must be determined, but also their angle of rotation. The main motivation to study these two versions of PMCLP is that cellphone towers can have elliptically shaped coverage area, therefore, to determine what are the best locations to place m cellphone towers to maximize the amount of the population covered by their 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 and 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

[2] Numerical experiments were done for up to 3000 points.

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 and 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 and Birgin (2013), the version of the problem where the ellipses could 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 maximum covering by disks algorithms found in the literature.

The main results of this work are found in Chapter 6 where a new algorithm for the version of the elliptical PMCLP where the ellipses can be freely rotated is developed. The proposed algorithm has a runtime complexity of $\mathcal{O}(n^{3m})$ and one of its most important steps is finding every solution of a not-previously-known problem. In Chapter 5, this problem is introduced and an algorithm for it is presented. The rest of the 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 maximum covering by disks problem is studied and a $\mathcal{O}(n^{2m})$ algorithm is proposed; in Chapter 4, the maximum covering by ellipses is introduced and the algorithm for the disks case is adapted for it; finally, Chapter 7 presents a conclusion as well as 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. Note that 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. They get this name because 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 $x, y \in \mathbb{R}$. Distinguishing an ellipse from the other conic sections ca be done using the condition given by Equation 2.5. More details about conic sections can be found in Ayoub (1993).

$$4AC - B^2 > 0. \tag{2.5}$$

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

$$(p-c)^T Q(p-c) = 1, \tag{2.6}$$

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

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

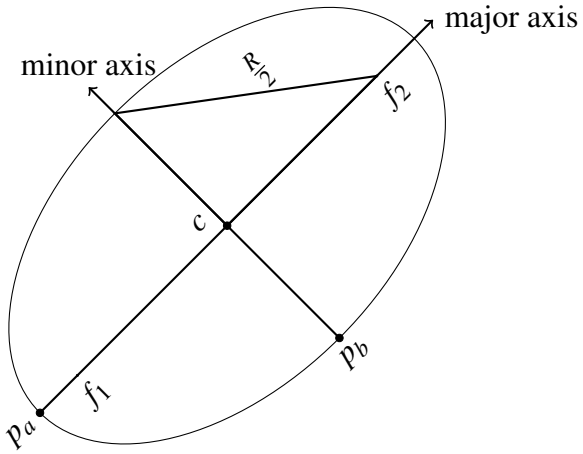
Note that asking Q to be positive definite is the same as asking $4AC - B^2$ to be positive. This makes us arrive at the following definition of the ellipse.

Definition 2.1. 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 2.1, 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 2.1 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

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

then, starting from Definition 2.1, we can obtain the following equation

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

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

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

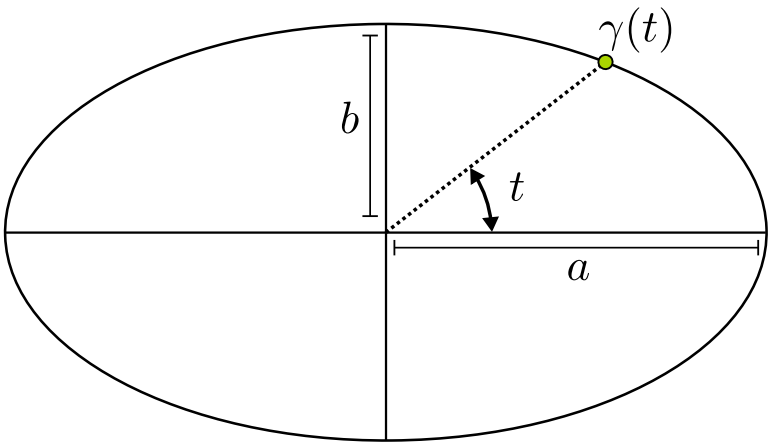
where a and b are the semi-major and semi-minor shape parameters, respectively.

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

$$\frac{(p_x - c_x)^2}{a^2} + \frac{(p_y - c_y)^2}{b^2} \leq 1. \tag{2.8}$$

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

$$\gamma(t) = \begin{cases} x(t) = a\cos t + c_x, \\ y(t) = b\sin t + c_y. \end{cases} \tag{2.9}$$

Consider an ellipse with shape parameters $(a,b) \in \mathbb{R}^2_{>0}$ centered at the origin, and a line represented by the equation $y = mx + c$, with $m, c \in \mathbb{R}$. Suppose that this line intersects the ellipse at a least one point. Plugging the line's equation into Equation 2.7, it is possible to obtain the distance between the intersection points. The final expression is given by

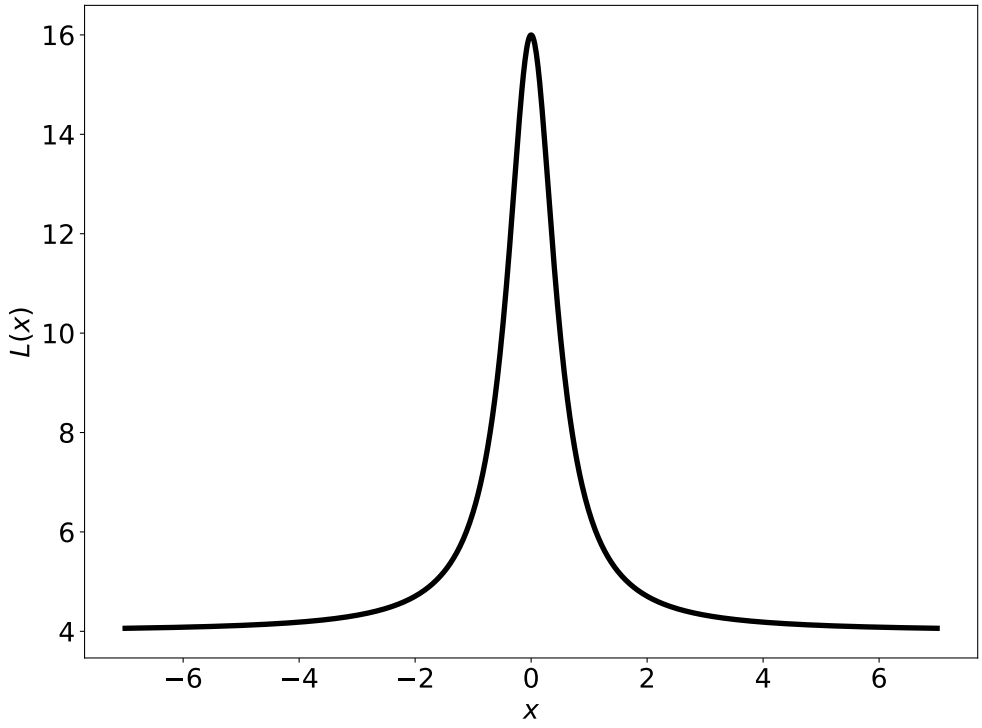
$$D(m,c) = \frac{\sqrt{(a^2m^2 + b^2 - c^2)(4a^2b^2(1+m^2))}}{(a^2m^2 + b^2)}, \tag{2.10}$$

with $D : \mathbb{R}^2 \mapsto \mathbb{R}_{\geq 0}$ being a function of the line parameters (m, c). It is also possible to see that, when m is fixed, $D(m,c)^2$ is a parabola, and that $D(m,c)$ is maximized at $c = 0$. From that, we can conclude that if m is fixed, the line that has the most distant intersection points with an ellipse is the one that passes through the origin; and also, that $D(m,c)$ attains every value in the range $[0, D(m,0)]$. Following that, we define a function $L : m \mapsto \mathbb{R}_{>0}$ that describes the maximum squared distance between the points of the intersection of an ellipse with a line with a fixed angular coefficient:

$$L(m) := D(m,0)^2 = \frac{(a^2m^2 + b^2)(4a^2b^2(1+m^2))}{(a^2m^2 + b^2)^2}. \tag{2.11}$$

It is possible, by calculating the derivatives, to conclude that L has its maximum at $x = 0$, is increasing in $[0, \infty)$, is decreasing in $(-\infty 0]$, and attains every value in the interval $(4b^2, 4a^2]$. Notice that L never hits $4b^2$ because that is the distance between the intersection of the ellipse with a vertical line. In Figure 3, an example of function L is shown with $(a,b) = (2,1)$.

Figure 3 – Plot of function L in the interval $[-7,7]$.



Source: Elaborated by the author.

2.3.2 Non-axis-parallel

A non-axis-parallel ellipse centered at $(c_x, c_y) \in \mathbb{R}^2$ can also be described by Definition 2.1. Nonetheless, a simpler equation is going to be used here. Besides the center, and the shape parameters $(a,b) \in \mathbb{R}^2_{>0}$, with $a > b$, an angle of rotation $\theta \in \mathbb{R}$ is given. This new parameter represents the angle between the x-axis and the major-axis of the ellipse, this can be

seen on Figure 4 where the dashed lines represent the ellipse's axes and the angle between the major-axis and the x-axis is displayed.

A rotated ellipse by θ can be transformed into a axis-parallel centered at the origin by applying two linear transformations: translation, so its center is at $(0,0)$, then rotation so its major-axis is parallel to the x-axis. Reversing these transformations produces the following equation for a non-axis-parallel ellipse

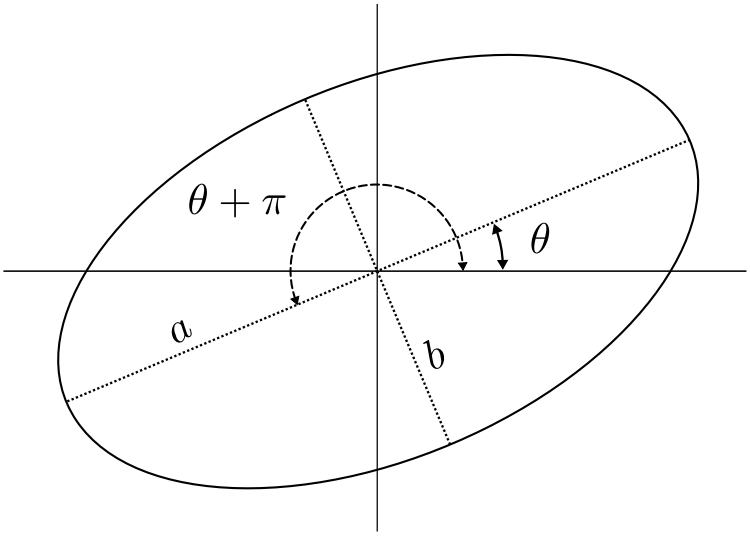
$$\frac{((x-c_x)\cos\theta+(y-c_y)\sin\theta)^2}{a^2}+\frac{((x-c_x)\sin\theta-(y-c_y)\cos\theta)^2}{b^2}=1. \tag{2.12}$$

The coverage region of that same ellipse is given by every point $(x,y)\in\mathbb{R}^2$ that satisfies the following equation, which is the same as Equation 2.12 with the equality sign $(=)$ replaced by $(\leq)$

$$\frac{((x-c_x)\cos\theta+(y-c_y)\sin\theta)^2}{a^2}+\frac{((x-c_x)\sin\theta-(y-c_y)\cos\theta)^2}{b^2}\leq 1. \tag{2.13}$$

Another important property of ellipses is shown on Figure 4. The two angles of rotation between the major-axis and the x-axis (θ and $\theta+\pi$) are actually equivalent–they produce the same ellipse. This symmetry is true for any angle of rotation, which means that θ is equivalent to $\theta+k\pi$, $k\in\mathbb{Z}$. Therefore, it is possible to conclude that to represent any ellipse, it is enough to specify the domain of θ to be $[0,\pi)$.

Figure 4 – The rotated ellipse.



Source: Elaborated by the author.

2.3.3 Notation

As stated by Definition 2.1, the word ellipse is used to refer to the set of points that satisfies the equality equation, which can be seen as the border of an ellipse's coverage area. For

this work, however, it is more convenient to refer directly to the coverage area of an ellipse and add a notation to express its border. For example, let E be an ellipse's coverage area, and $\mathcal{P} \subset \mathbb{R}^2$ a set of points, then $E \cap \mathcal{P}$ denotes the set of points from $\mathcal{P}$ inside the coverage area of that ellipse. When we need to refer specifically to the border of E, we use the boundary operator: ∂E.

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 5.

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 5. This is also a good way to visualize Euler's Formula, as it can be also seen on Figure 5, which says that any complex number can be written in terms of its radius and polar angle as follows:

$$z = re^{i\theta} = r(\cos\theta + i\sin\theta), \tag{2.14}$$

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 $[0, 2\pi)$.

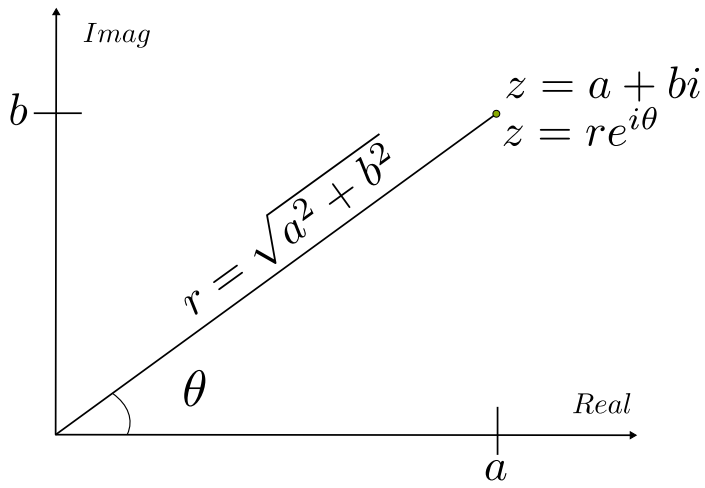


Figure 5 – The representation of a complex number on two dimensions.

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:

$$\bar{z} = a - bi. \tag{2.15}$$

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 5.

$$angle(\bar{z}) = 2\pi - angle(z) \tag{2.16}$$

$$angle(-z) = \pi + angle(z), \tag{2.17}$$

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

$$p_n(z) = \sum_{k=0}^{n} a_k z^k, \tag{2.18}$$

with $a_0, \ldots, a_n \in \mathbb{C}$. In this work, when a polynomial is written in the form of Equation 2.18 we say that it is in the power form or in the monomial form.

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 works really well in practice.

In Horn and Johnson (1986, p. 195) a theorem is presented which says that for every univariate polynomial of degree n, there exists a companion matrix which is a $n \times n$ matrix, such that its eigenvalues are the zeros of that polynomial. For example, the companion matrix of a degree-5 polynomial written as Equation 2.18 is given by

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\frac{a_0}{a_5} & -\frac{a_1}{a_5} & -\frac{a_2}{a_5} & -\frac{a_3}{a_5} & -\frac{a_4}{a_5} \end{bmatrix}. \tag{2.19}$$

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

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

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

$$p_n(\theta) = \sum_{k=0}^{n} a_k \cos(k\theta) + \sum_{k=1}^{n} b_k \sin(k\theta). \tag{2.20}$$

We say that $p_n : \mathbb{R} \mapsto \mathbb{R}$ as defined by Equation 2.20 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

$$\sum_{0 \leq j+k \leq n} c_{j,k} \cos^j \theta \sin^k \theta, \tag{2.21}$$

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

CHAPTER

3

MAXIMUM COVERING BY DISKS

In this chapter, we introduce a version of the classical Euclidean norm PMCLP where each facility has a given coverage radius. We refer to this problem as Maximum Covering by Disks (MCD). We also propose a method for it intending to adapt it for the elliptical PMCLP that will be introduced in the next chapter.

3.1 Definition

An instance of MCD is given by a set of n demand points $\mathcal{P} := \{p_1, \ldots, p_n\}$, with $p_j \in \mathbb{R}^2$; a set of weights $\mathcal{W} := \{w_1, \ldots, w_n\}$, with $w_j \in \mathbb{R}_{\geq 0}$ being the weight of point p_j; and m disks given by their radii $\mathcal{R} := \{r_1, \ldots, r_m\}$, with $r_j \in \mathbb{R}_{>0}$. Additionally, to make the text more clear, we define a set of m disks as $\mathcal{D} = \{D_1, \ldots, D_m\}$, with $D_j : \mathbb{R}^2 \mapsto \mathbb{R}^2$ being a function that takes the center where the j-th disk is located as input, and returns its coverage region as defined by Equation 2.3.

A solution for an instance of MCD is determined by $Q := (q_1, \ldots, q_m) \in \mathbb{R}^{2m}$ which specifies the center of every disk in $\mathcal{D}$. The coverage region of the j-th disk centered at q_j is denoted by $D_j(q_j)$. Finally, let $w : 2^{\mathcal{P}} \mapsto \mathbb{R}_{\geq 0}$ be the function

$$w(A) = \sum_{j : p_j \in A} w_j, \tag{3.1}$$

which takes a subset of $\mathcal{P}$ and returns the sum of the weights of every point in it; then an optimal solution of MCD is given by the optmization problem:

$$\max_{Q} w\left(\bigcup_{j=1}^{m} \mathcal{P} \cap D_j(q_j) \right). \tag{3.2}$$

As a remark, it is important to note that MCD is a slightly different PMCLP than the

one introduced in the first study on the subject in Church (1984). There, a coverage radius is given for each demand, rather than for each facility, and a demand is only considered covered if a facility is located within its radius.

3.1.1 CLS and CIPS

The method proposed in Church (1984) involves the creation of a Candidate List Set (CLS), which is a finite set of possible solutions, for each facility. After a CLS is constructed, a complete search can be executed to find the best solution.

As this work is developed towards the creation of an exact method for MCD, the focus is on constructing a CLS that contains an optimal solution. This is done in Church (1984) where a circle is defined for every demand point, and a circle intersection point set (CIPS) is constructed containing the demand points and every pairwise circle intersection. Here, we introduce a similar approach based on works found on the case for only one disk and describe a $\mathcal{O}(n^2 \lg n)$ algorithm for it.

3.2 One disk version

This version of the problem will be refered to as Maximum Covering by One Unit Disk (MCD-1) and it is just a specific case of MCD with only one disk with radius one ($m = 1$ and $r_1 = 1$). We later use the algorithm for MCD-1 here described to construct a CLS which is guaranteed to contain an optimal solution for MCD.

Two exact methods for MCD-1 have been found in the literature. A $\mathcal{O}(n^2)$ algorithm is proposed by Chazelle and Lee (1986) which improved the previously $\mathcal{O}(n^2 \log n)$ one proposed by Drezner (1981). As it has been mentioned, MCD-1 is a 3SUM-HARD problem, which means that it is as hard as the 3SUM problem (the problem of finding three real numbers that sum to zero, 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-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).

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 solution. 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 and Lee (1986) and Berg, Cabello and Har-Peled (2006), on the other hand, instead of working directly with MCD-1, an equivalent problem called Maximum Weight Clique (MWC) was introduced. The algorithm for MCD that is developed in this chapter also uses that equivalence. Because of that, MWC is introduced in the next section.

3.3 Maximum Weight Clique Problem

An instance of MWC is given by a set of points $\mathcal{P} := \{p_1, \ldots, p_n\}$, with $p_i \in \mathbb{R}^2$; a set of unit disks $\mathcal{D} := \{D_1(p_1), \ldots, D_n(p_n)\}$; and a set of weights $\mathcal{W} = \{w_1, \ldots, w_n\}$ with the i-th disk having a weight $w_i \in \mathbb{R}_{>0}$ assigned to it. To make the notation less cluttered, we use, in this problem, $D_j := D_j(p_j)$, for $j \in \{1, \ldots, n\}$.

A clique, in this context, is a non-empty intersection region of one or more disks, and its weight is the sum of the weights of those disks in the intersection. Following this, a solution for MWC can be defined as just a point $q \in \cup_{j=1}^{n} D_j$ which is inside any of the given disks in $\mathcal{D}$. From a solution q, the corresponding clique S can be obtained by intersecting every disk that contains q:

$$S = \bigcap_{j : q \in D_j} D_j. \tag{3.3}$$

With a geometric observation, though, the number of possible values for the solution can be reduced. Let $\partial\mathcal{D} = \{\partial D_1, \ldots, \partial D_n\}$ be the set of circles corresponding to each disk in $\mathcal{D}$, it is possible to see that any clique has at least one point on its border that is in the intersection of two circles in ∂D or, in the case where the clique is composed of only one disk, it contains the center of its only disk. Because of that, q can be actually limited to the set of pairwise intersections of $\partial\mathcal{D}$ as well as the set of centers of each disk $\mathcal{P}$. Then, with this observation, an optimal solution of MWC can be defined by the optimization problem

$$\max_{q} \sum_{D_k \cap q \neq \emptyset} w_k, \tag{3.4}$$

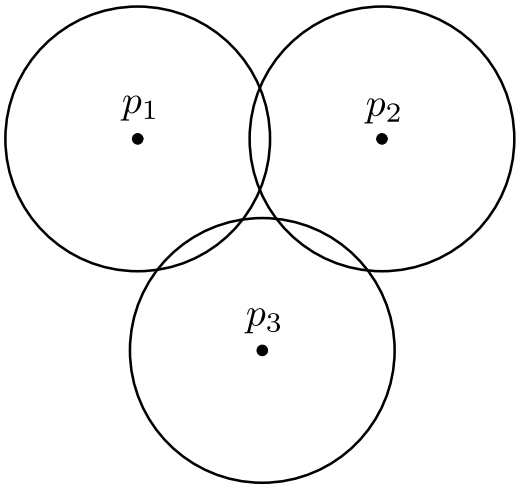
with $q \in \{\partial D_i \cap \partial D_j : 1 \leq i < j \leq n\} \cup \mathcal{P}$. With this new specification of the solution's search space, it can be shown that given an instance of MWC, an optimal solution for it can be found by going through $\binom{n}{2} + n = \mathcal{O}(n^2)$ possible ones.

It is worth pointing out that MWC is a different problem than the maximum clique 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 6, three disks could have non-empty pairwise intersection and still have an empty intersection of all of them together. That is why

MWC is also referred to as the Maximum Geometric Clique Problem and the other version, when there is only the pairwise intersection condition, is referred to as the Maximum Graphical Clique Problem (DE; NANDY; ROY, 2014).

In Chazelle and Lee (1986), the method for MWC consists on building a planar graph on which the vertices were the $\binom{n}{2}$ 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 6 – Three disks that have non-empty pairwise intersection among them, but no common intersection.



Source: Elaborated by the author.

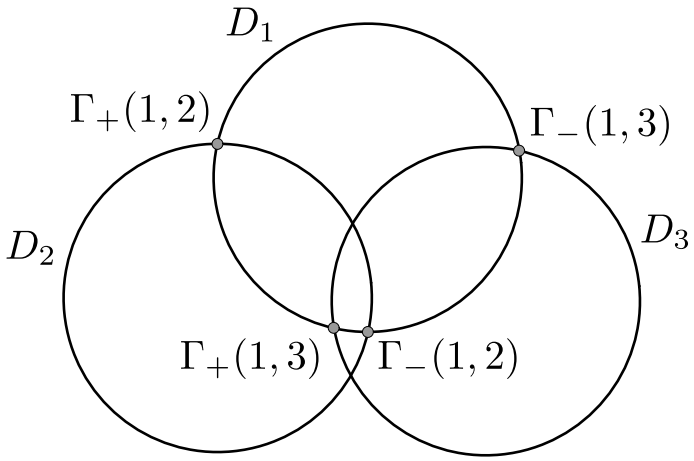
As it has been mentioned, with the equivalence of the two problems, an optimal solution of the Maximum Weight Clique Problem is also an optimal solution of MCD-1, which means that a disk centered at q^* which is an optimal solution of MWC, will have a maximal weight covering of the demand set $\mathcal{P}$.

Given an instance of MCD-1, the equivalent MWC intance is obtained by defining the 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 a region where a disk could be placed to cover the corresponding points.

In Figure 6, 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 MWC, 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 equivalent MCD-1 instance. Conversely, the same applies for a disk placed at q that covers points p_k, with $k \in I$ in the MCD-1 instance. It means that q will lie inside region $\bigcap_{k \in I} D_k$ in MWC.

Figure 7 – Three disks and their intersection points.



Source: Elaborated by the author.

3.3.1 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 and Roy (2014) and Berg, Cabello and 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 and Ottmann (1979) to find all the intersections among the n circumferences.

In De, Nandy and Roy (2014) an important observation is made about the intersection regions of disks. Given an instance of MWC, any clique formed by a subset of $\mathcal{D}$ is bounded by the arcs of circles that intersect with it. Also, those arcs have the intersection of circles as their end-points. This can be seen on Figure 7 where the cliques that D_1 is part of are bounded by D_1's arcs which have its intersections with the other circles as end-points. Following this, a definition is presented to characterize the end-points of an arc bordering a clique.

Definition 3.1. Let D_i and D_j be two unit disks with non-empty intersection, and $(\theta_1, \theta_2) \in [0, 2\pi)^2$ be the two angles that ∂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 $D_i \cap D_j$. Then, define $\Gamma_+(i,j) = \theta_1$ and $\Gamma_-(i,j) = \theta_2$. For convenience, if D_i is tangent to D_j, then $\theta_1 = \theta_2$; and if $i = j$, then $\Gamma_-(i,j) = 0$ and $\Gamma_+(i,j) = 2\pi$.

Also, we refer to $\Gamma_+(i,j)$ as an opening angle, and to $\Gamma_-(i,j)$ as a closing angle. In Figure 7, it is shown all the intersection points between D_1 with D_2 and D_3. Also, they are labeled according to Definition 3.1. Note that $\Gamma_+(1,3) > \Gamma_-(1,3)$ (the angles should be in the $[0, 2\pi]$ interval).

With Definition 3.1 in hand, we can establish the basis of the algorithm for MWC. For every disk D_i, let us describe an algorithm that gets the best clique which D_i is part of. This

way, an algorithm for MWC just uses that method for every disk and returns the best solution found. Firstly, let A_i be a list that contains the intersection angles of ∂D_i with every circle in $\partial\mathcal{D}$. Assume also that A_i is actually a circular list sorted in ascending order and it is defined as

$$A_i = \bigcup_{j=1}^{n} \{\Gamma_-(i,j), \Gamma_+(i,j)\}. \tag{3.5}$$

Finding the best solution which D_i is part of can be done by traversing A_i while keeping a set of active disks. When an opening intersection angle is reached, the corresponding disk is added to the active set; and when a closing one is seen, the corresponding disk is removed from the active set. This way, finding an optimal solution can be achieved by keeping the weight of the active disks as well as the best clique found so far. Notice also that because $\Gamma_+(i,i) = 0$ and $\Gamma_-(i,i) = 2\pi$, any clique found by the traversal will also contain D_i.

In practice, traversing a circular list can be emulated by traversing a regular list that has a copy of the original circular list added to its end. This is done here by traversing the regular list B_i defined as

$$B_i = A_i \cup \bigcup_{j=1}^{n} \{2\pi + \Gamma_-(i,j), 2\pi + \Gamma_+(i,j)\}. \tag{3.6}$$

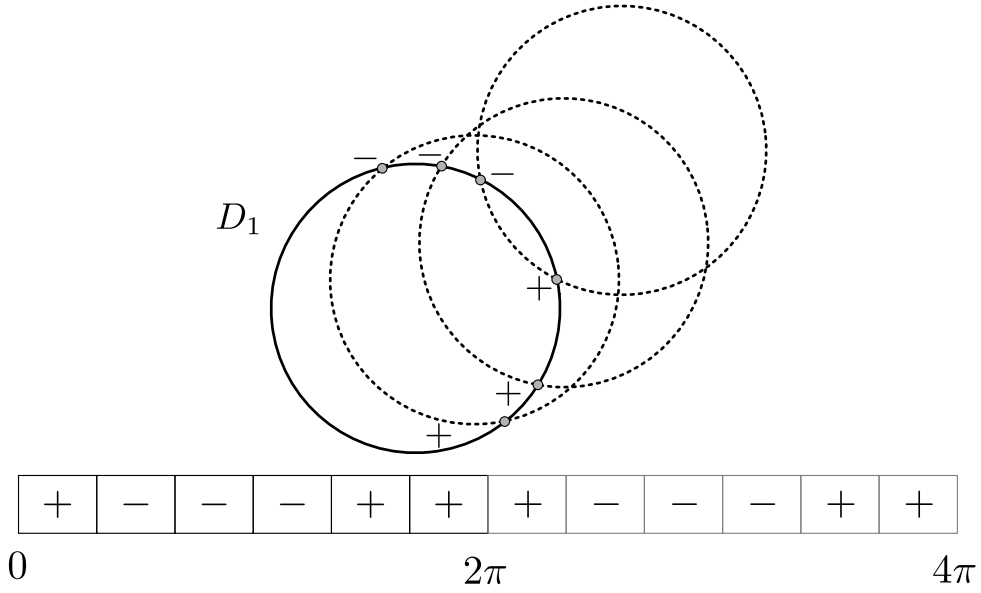
Assuming B_i is sorted in ascending order, a copy of the elements of A_i shifted to the interval $[2\pi, 4\pi]$ is added to its end and a simple traversal, going through the first element until the last one, simulates a traversal on the circular list A_i. This works because for any pair of disks, such that $\Gamma_+(i,j) > \Gamma_-(i,j)$, B_i will contain $\Gamma_-(i,j) < \Gamma_+(i,j) + 2\pi$. This is shown in Figure 8, where the intersections list is duplicated simulating the traversal in B_i (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).

Theorem 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 and Ottmann (1979). Because sorting the intersection angles needs to be done, an additional $\mathcal{O}(K \log K)$ pre-processing is added. All the other operations can be done in constant time. Therefore, the final algorithm complexity is $\mathcal{O}((n+K)\log n)$. □

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 $\mathcal{O}(n^2 \log n)$.

Figure 8 – The intersections list of a disk with three other disks.



Source: Elaborated by the author.

3.4 Multiple disks

In Berg, Cabello and Har-Peled (2006), a $\mathcal{O}(n^{2m-1}\log n)$ algorithm for *MCD* is developed as a sub-routine for its $(1-\varepsilon)$-approximation algorithm. Firstly, they solve a sub-problem for two disks 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 and 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.

3.4.1 An algorithm for MCD

A simple adaptation can be done on Algorithm 1 to make it return a CLS that contains an optimal solution of MCD for that disk. This is shown in Algorithm 2. Notice also that MCD-1 is defined only for unit disks, however, this constrained can be dropped, as it is introduced just for the sake of keeping the text more simple and Algorithm 1 works for any radius. A result about the runtime complexity of Algorithm 2 has already been given by 1, the following result states about the adaption of it to be used in an algorithm for MCD.

Lemma 3.1. Suppose that an instance of MCD and an index $j \in \{1,\ldots,m\}$ are given. Then

Algorithm 2, when given the instance $(\mathcal{P}, \mathcal{W}, r_j)$ as input, returns a CLS S_j of size $\mathcal{O}(n^2)$, such that $q_j^* \in S_j$, with $(q_1^*, \ldots, q_m^*)$ being an optimal solution of the given MCD's instance.

Proof. It can be seen that in any solution of MCD, 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 2 as a candidate to be an optimal solution. The number of points Algorithm 2 goes through is $\mathcal{O}(n^2)$, then obviously $|S_j| = \mathcal{O}(n^2)$. □

Then, with Algorithm 2, an algorithm for MCD that checks every possible center for every disk yields a $\mathcal{O}(n^{2m})$ run-time complexity. This algorithm is described in Chapter 4 for the axis-parallel ellipses case.

It is worth mentioning that the choice of developing a different method for the problem, instead of using the one from Berg, Cabello and Har-Peled (2006), is taken for the sake of simplicity, considering both algorithms achieve similar bounds.

Procedure 1 – Algorithm for MCD-1.

Input: A set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$ with weights $\mathcal{W} = \{w_1, \ldots, w_n\}$.
Output: The maximum cover by a unit disk.

```
1:  procedure MCD_1(P)
2:      Q_best ← {}
3:      ans ← center of D_1
4:      for all p_i ∈ P do
5:          Let D_i be the disk with center at p_i
6:          Let B_i be the list of intersection angles of D_i as defined by Equation 3.6.
7:          Q ← {}                                          ▷ The set of active disks.
8:          for a ∈ B_i do                                  ▷ Assuming B_i is sorted.
9:              Let D_a be the disk that intersects D_i at angle a.
10:             if a is a starting angle then
11:                 Q ← Q ∪ {D_a}
12:             else
13:                 Q ← Q \ {D_a}
14:             end if
15:             if w(Q_best) < w(Q) then        ▷ w(Q) returns the weights of the disks in Q.
16:                 Q_best ← Q
17:                 ans ← point corresponding to the intersection angle a
18:             end if
19:         end for
20:     end for
21:     return ans
22: end procedure
```

Procedure 2 – Algorithm for MCD-1 that returns a CLS.

Input: A set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$ with weights $\mathcal{W} = \{w_1, \ldots, w_n\}$, and a radius $r \in \mathbb{R}_{>0}$.
Output: A CLS for the disk given by radius r.

```
1:  procedure MCD_1(P, W, r)
2:      S ← {}
3:      for all p_i ∈ P do
4:          Let D_i be the disk with center at p_i
5:          Let B_i be the list of intersection angles of D_i as defined by Equation 3.6.
6:          Cov ← {}
7:          for a ∈ B_i do                                  ▷ Assuming B_i is sorted.
8:              Let q_a be the intersection point correspondent to angle a.
9:              S ← S ∪ {q_a}
10:         end for
11:     end for
12:     return S
13: end procedure
```

CHAPTER

4

MAXIMUM COVERING BY ELLIPSES

In this chapter, we introduce the version of PMCLP where every facility has an axis-parallel ellipse as its coverage area. We refer to this problem as Maximum Covering by Ellipses (MCE). We also present an algorithm for it which is an adaptation of the algorithm developed for MCD in Chapter 3.

4.1 Definition

Axis-parallel ellipses are defined as the set of points that satisfy Equation 2.7. Note that all it takes to describe an axis-parallel ellipse is a pair of positive real numbers $(a,b) \in \mathbb{R}^2_{>0}$, also called its shape parameters, and a center point $q \in \mathbb{R}^2$.

An instance of MCE is given by a set of n demand points $\mathcal{P} = \{p_1, \ldots, p_n\}$, with $p_j \in \mathbb{R}^2$; a set of weights $\mathcal{W} := \{w_1, \ldots, w_n\}$, with $w_j \in \mathbb{R}_{\geq 0}$ being the weight of point p_j; and a set of m axis-parallel ellipses given by their shape parameters $\mathcal{R} := \{(a_1,b_1), \ldots, (a_m,b_m)\}$, with $(a_j,b_j) \in \mathbb{R}^2_{>0}$ and $a_j > b_j$. Additionally, to make the text more clear, we define a set $\mathcal{E} = \{E_1, \ldots, E_m\}$, with $E_j : \mathbb{R}^2 \mapsto \mathbb{R}^2$ being a function that takes the center where the j-th ellipse is located as input, and returns its coverage region as defined by Equation 2.8.

Then, a solution for MCE is given by $Q := (q_1, \ldots, q_m) \in \mathbb{R}^{2m}$, with q_j being the center of j-th ellipse. Finally, let $w : 2^{\mathcal{P}} \mapsto \mathbb{R}_{\geq 0}$ as defined by Equation 3.1, then an optimal solution of MCE is given by the optimization problem

$$\max_q w\left(\bigcup_{j=1}^{m} \mathcal{P} \cap E_j(q_j)\right). \tag{4.1}$$

In the next sections, we first describe a method for the case with only one ellipse, and then use it in the algorithm for multiple ellipses to construct a CLS containing an optimal solution.

4.2 Related work

The maximal planar covering using axis-parallel ellipses was first introduced in Canbolat and 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 and 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.

4.3 One Ellipse Version

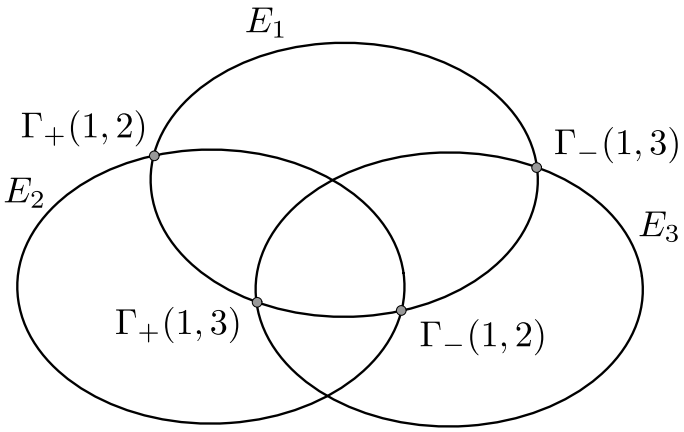
The case with only one ellipse is considered first because it will be adapted to become the basis of the algorithm for more than one ellipse. We refer to this version as Maximum Cover by One Ellipse (MCE-1).

An instance of MCE-1 has $m = 1$, and we set $(a,b) := (a_1,b_1)$, and $\mathcal{E} := \{E\}$. Therefore, an instance of MCE-1 is described by the tuple $(\mathcal{P};\mathcal{W};(a,b))$. A solution of MCE-1 is then given by a point $q \in \mathbb{R}^2$, and an optimal solution is given by the optimization problem

$$\max_{q} w(\mathcal{P} \cap E(q)). \tag{4.2}$$

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 9 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 $\Gamma_+(i,j)$ are described thoroughly in Appendix A.

Figure 9 – Three ellipses and their intersection points



Source: Elaborated by the author.

4.4 An algorithm for MCE

The same procedure defined in Algorithm 2 can be used to get a CLS for every ellipse in MCE. We refer to the elliptical version of that procedure as MCE_1 (we do not define it in this chapter because it would look the same as MCD_1, apart from the name, of course).

Then, with the algorithm to construct a CLS for every ellipse in hands, an algorithm for MCE naturally comes into existence. In Algorithm 3, a complete search is done backtracking every possibility in the CLS of every ellipse. This strategy is backed-up by Lemma 3.1, which says that there is an optimal solution in the CLS of each ellipse. Following this, counting every possibility that the algorithm goes through, we arrive at the run-time complexity of $\mathcal{O}(n^{2m})$.

It is worth mentioning that, even though we call MCE_1 every time in the recursion, in practice, it is probably best to pre-process this step, and only call it m times for the whole set of points. Some other easy improvements can also 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 non-negative 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.

4.5 Adding facility cost

Additionally, in Andretta and Birgin (2013) and Canbolat and Massow (2009), two other parameters are present in the definition of the problem. This extension is the result of having costs associated with every facility. In MCE, though, the total cost, which is the sum of costs of every used facility, is constant; hence, to create a decision about which ones are utilized, a new

parameter $k \in \mathbb{N}$ is given, along with a constraint on the number of used ellipses.

We refer to this version of the problem as Maximum Covering by Ellipses with a k-constraint (MCE-k). An instance of it is given by the same parameters as MCE, plus a list of costs $\mathcal{C} = \{c_1, \ldots, c_m\}$, with $c_j \in \mathbb{R}_{\geq 0}$ being the j-th ellipse's cost, and $k \in \mathbb{N}$.

A solution for MCE-k, however, when compared to MCE's, has a bit more cluttered description. We define it as a set $I := \{i_1, \ldots, i_k\} \subset \{1, \ldots, m\}$, such that $|I| = k$; and a tuple $Q := (q_1, \ldots, q_k)$, with $q_j \in \mathbb{R}^2$ being the center of the j-th ellipse in I. An optimal solution of MCE-k is given by the optimization problem

$$\max_{I,Q} w\left(\bigcup_{j=1}^{k} \mathcal{P} \cap E_{i_j}(q_j)\right). \tag{4.3}$$

Finally, Algorithm 3 can serve as basis for an algorithm for MCE-k. Firstly, for every subset $I \subset \{1, \ldots, m\}$, such that $|I| = k$, the algorithm for MCE is invoked for the instance $(\mathcal{P}, \mathcal{W}, \{(a_j, b_j) : j \in I\})$; that is, an instance where only the ellipses in I are present. After that, by keeping track of the utilized ellipses' costs for every $I \subset \{1, \ldots, m\}$, an optimal solution can be obtained. This simple adjustment achieves a run-time complexity of $\mathcal{O}(\binom{m}{k} \times n^{2k})$.

Procedure 3 – Algorithm for MCE

Input: A set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$, a list of weights $\mathcal{W} = \{w_1, \ldots, w_n\}$, and a list of shape parameters $\mathcal{R} = \{(a_1, b_1), \ldots, (a_m, b_m)\}$.
Output: The value of an optimal solution of MCE.

```
1:  procedure MCE(P, W, R, j = 1)
2:      if j = m + 1 then
3:          return 0
4:      end if
5:      ans ← 0
6:      S_j ← MCE_1(P, a_j, b_j)
7:      for q ∈ S_j do
8:          Cov ← P ∩ q
9:          ans ← max{ans, w(Cov) + MCE(P \ Cov, W, R, j + 1)}    ▷ Calls the procedure for the next ellipse.
10:     end for
11:     return ans
12: end procedure
```

CHAPTER

5

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 ones. At the end we propose an algorithm that takes around 6^3 operations to compute every solution of the problem. Also, we run some experiments to analyze its efficiency in terms of numerical precision and CPU time.

5.1 Definition

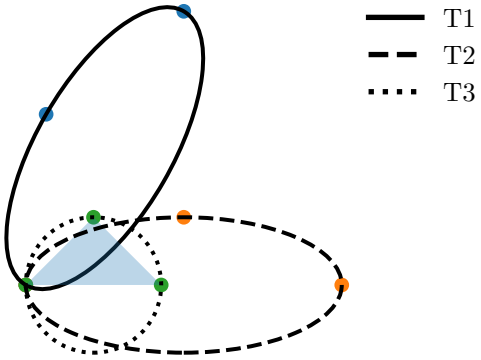
We call the problem introduced in this section Ellipse by Three Points (E3PNT). An instance of it is given by three points $u, v, w \in \mathbb{R}^2$ and the shape parameters of an ellipse $(a, b) \in \mathbb{R}^2_{>0}$, with $a > b$.

Let E represent the coverage area of an ellipse with the given shape parameters (a, b), then a solution of E3PNT can be defined as a pair $(q, \theta) \in \mathbb{R}^2 \times [0, \pi)$, such that $\{u, v, w\} \subset \partial E(q, \theta)$. As a last remark, because of its application on Chapter 6, a method would only be useful for our case if it can encounter every solution of E3PNT. This requirement, of course, makes the problem more challenging.

5.2 Transforming the problem

Initially, E3PNT is a problem with three unknown variables: the two coordinates of the center point q_x, q_y, and the angle of rotation θ. In this section, we transform E3PNT into the problem of finding the roots of an univariate function using the known problem of determining the circumcircle of a triangle. We also show E3PNT has at most 6 solutions.

Figure 10 – Transforming an ellipse into a circle. T1, T2, and T3 represent the steps of the transformation.



Source: Elaborated by the author.

To make the problem simpler, let us translate the system, so the point u is at the origin; that is, assume that the three points given by the E3PNT instance are $(0,0); v; w$. After that, 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 10 where a solution of the E3PNT is transformed firstly into an axis-parallel ellipse, and then into a circle of radius b. This process can be parameterized by the angle of rotation of the points, as described by

$$\varphi(p,\theta) = \begin{bmatrix} \frac{b}{a} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}. \tag{5.1}$$

Also, to make the notation more clear, we denote by $\Lambda(\theta)$ the triangle formed by the points $(0,0); \varphi(v,\theta); \varphi(w,\theta)$, as long as they are not collinear. This being said, we go further to introduce another problem which is equivalent to E3PNT.

5.2.1 A circumradius problem

The term circumradius, in this work, is used to describe the radius of a triangle's circumscribed circle, which is a circle that has the triangle's vertices on its border. Given an instance of E3PNT, we define the circumradius problem as the problem of determining an angle of rotation θ, such that the circumradius of $\Lambda(\theta) \in [0,\pi)$ is equal to b.

As it can be seen on Figure 10, given an instance of E3PNT and an angle of rotation $\theta \in [0,\pi)$ that makes $\Lambda(\theta)$ have a circumradius b; a solution for E3PNT can be obtained using the inverse transformation φ^{-1}. This is possible because for any fixed θ, φ is just a linear

function which are known to be invertible. With that in mind, it is possible to conclude that both problems are equivalent because, from a solution of one, an unique solution of the other can be obtained.

The main reason to work with this problem is the reduction in the number of unknown variables from three to just one. This idea, however, 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-collinear points, there is always an unique circumscribed circle. Also, finding this unique circle can be done analytically, and its radius R can be computed through the expression

$$R = \frac{\|\varphi(v,\theta)\|_2 \, \|\varphi(w,\theta)\|_2 \, \|\varphi(v,\theta) - \varphi(w,\theta)\|_2}{4A(\theta)}, \tag{5.2}$$

with $A(\theta)$ being the area of the triangle $\Lambda(\theta)$ (for more details about Equation 5.2, or on how to determine the center of a circumscribed circle see Johnson and Young (1960, p. 189)). It should be pointed out that the transformation does not preserve distance or area; if that was true, the radius defined by Equation 5.2 would be constant.

With the formula for the circumradius in hands, a function can be defined, such that its roots provide solutions for the circumradius problem. Imposing the radius R 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 as

$$\xi(\theta) = 16b^2 A(\theta)^2 - \|\varphi(v,\theta)\|_2^2 \, \|\varphi(w,\theta)\|_2^2 \, \|\varphi(v,\theta) - \varphi(w,\theta)\|_2^2 . \tag{5.3}$$

Any root of ξ produces a triangle whose circumradius is b and subsequently provides a solution for E3PNT.

The next step is then to develop an algorithm to find every root of ξ. Before continuing with this approach though, it is important to address the question about the number of roots of ξ in the interval $[0, \pi)$.

5.2.2 The number of solutions of E3PNT

One of the steps of the method developed in Chapter 6 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. Moreover, even if the number of solutions is finite, discovering a bound for it is essential for determining the algorithm's efficiency.

Lemma 5.1. E3PNT has at most 6 solutions.

Proof. 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 2.21. This implies that ξ 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 the same form as given by Equation 2.21:

$$\|\varphi(v,\theta)\|_2^2 = (v_x\frac{b}{a}\cos\theta + v_y\frac{b}{a}\sin\theta)^2 + (v_y\cos\theta - v_x\sin\theta)^2 \tag{5.4}$$

$$A(\theta)^2 = \frac{1}{4}\det\begin{pmatrix} v_x\frac{b}{a}\cos\theta + v_y\frac{b}{a}\sin\theta & v_y\cos\theta - v_x\sin\theta \\ w_x\frac{b}{a}\cos\theta + w_y\frac{b}{a}\sin\theta & w_y\cos\theta - w_x\sin\theta \end{pmatrix}^2. \tag{5.5}$$

It is also possible to see that the term of higher the degree of ξ is the multiplication of the three squared lengths of the triangle formed by the points $\{(0,0),\varphi(v,\theta),\varphi(w,\theta)\}$. This multiplication has the same degree of $(\|\varphi(v,\theta)\|_2^2)^3$, and because $\|\varphi(v,\theta)\|_2^2$ has degree 2, the degree of $\|\varphi(v,\theta)\|_2^2$ is 6 which consequently is the degree of ξ. Going from 12 solutions to 6 is done by using the symmetry of ellipses. In Chapter 2, it was stated that any rotation in the interval $[0,\pi)$ is identical to a rotation in $[\pi,2\pi)$. Therefore, half of the roots of ξ are in $[\pi,2\pi)$ and can be dismissed. □

5.3 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

$$Ax^2 + Bxy + Cy^2 + Dy + Ex + F = 0. \tag{5.6}$$

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 5.6 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 $\Delta = 4AC - B^2 = 4AC - 1$, and assume $F = 0$, then the expressions for both major-axis and minor-axis respectively are

$$a^2 = \frac{2\dfrac{AE^2 - BDE + CD^2}{\Delta}}{A + C - \sqrt{1 + (A-C)^2}} \tag{5.7}$$

$$b^2 = \frac{2\dfrac{AE^2 - BDE + CD^2}{\Delta}}{A + C + \sqrt{1 + (A-C)^2}}. \tag{5.8}$$

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.

5.4 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 ξ 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 $n \times n$ matrix.

5.4.1 Chebyshev polynomial

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, MATLAB and C.

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

$$T_n(x) = \cos(n \arccos x). \tag{5.9}$$

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

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x). \tag{5.10}$$

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

$$p_n(x) = \sum_{j=0}^{n} a_j T_j(x). \tag{5.11}$$

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 5.11, has a linear condition number bounded by $\sqrt{2}n$.

5.4.2 Chebyshev interpolation

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

$$f_n(x) = \sum_{j=0}^{n} f(x_j) \frac{\prod_{k\neq j}^{n+1}(x - x_k)}{\prod_{k\neq j}^{n+1}(x_j - x_k)}, \tag{5.12}$$

with f being the function to be approximated, and f_n the unique n-degree polynomial that passes through $\{(x_j, f(x_j)) : j = 0, 1, \ldots n\}$. 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 interpolation points. The $n+1$ Chebyshev nodes are given by

$$x_j = \cos\left(\frac{\pi(j - \frac{1}{2})}{n+1}\right), \tag{5.13}$$

for $j = 1, \ldots, n+1$. 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 the transformation

$$\hat{x_j} = \frac{a+b}{2} + \frac{b-a}{2} x_j \tag{5.14}$$

can be done to map it to the Chebyshev Nodes' domain $[-1, 1]$.

Then, the Chebyshev interpolation of a function $f : [a, b] \mapsto \mathbb{R}$ can be determined using Lagrange's formula and the points $\hat{x}_1, \ldots, \hat{x}_n$. As it was mentioned in Chapter 2, finding the

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

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 5.12 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 5.11, 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 algorithm. Summing the two steps, a $\mathcal{O}(n^3)$ algorithm can be achieved, with n being the degree of the interpolation.

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

$$r_n = \max_{0 \le \theta < \pi} |f_n(\theta) - \xi(\theta)|, \tag{5.15}$$

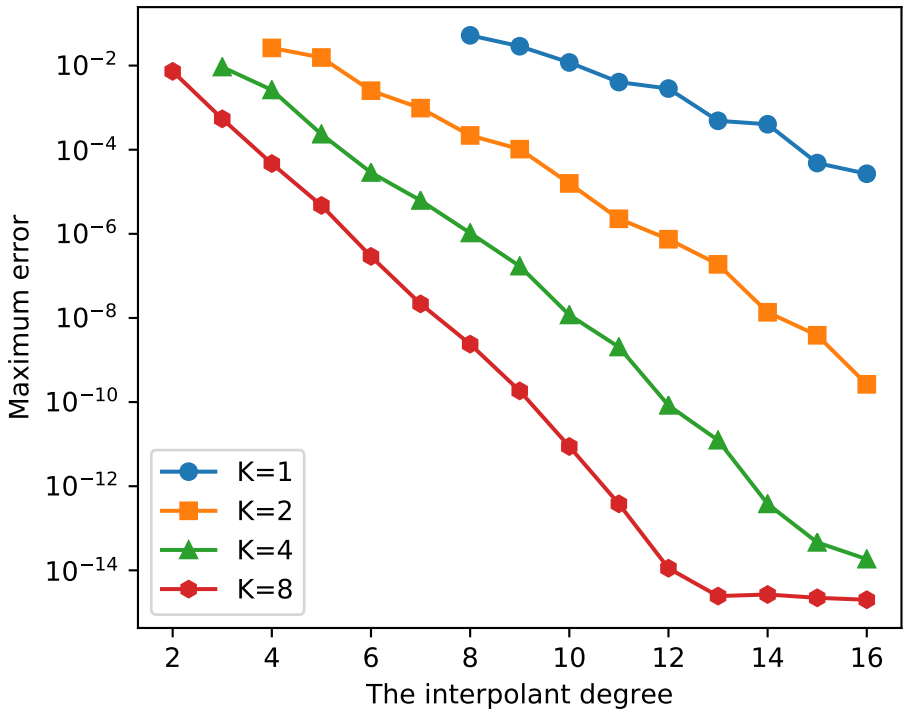
which measures how far the interpolation is at the point it worst approximates.

5.4.3 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. The interpolation was run for several degrees for every triplet of points present in an instance from Canbolat and Massow (2009). We considered the error to be $|\xi(\hat{\theta})|$ for every root $\hat{\theta}$ found; simply put we measure how close the roots are to zero.

The results are shown in Figure 11 where the worst error found among every root found

Figure 11 – The interpolation error measured on roots that were found.



Source: Elaborated by the author.

is plotted against the interpolant degree used. A parameter K is also used in the experiments. This is an idea suggested in Boyd (2013) to improve accuracy, this parameter represents the number of sub-intervals that $[0, \pi)$ is divided into. This way, K interpolations are used to obtain every root in the whole interval. It can be seen in Figure 11 that this strategy helps a lot in improving the accuracy even when not so large degrees are used.

5.5 Converting ξ into a polynomial

In 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).

5.5.1 Real polynomial

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. This issue made us abandon this approach and pursue a different way to convert ξ into a polynomial.

5.5.2 Complex polynomial

The second approach is based on a idea published in Boyd (2006) which uses the identities on Equation 5.16 to convert real trigonometric polynomials, which is the case of ξ, into univariate complex polynomials. 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.

$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2} \tag{5.16}$$

$$\sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}. \tag{5.17}$$

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 ($\mathbb{S} = \{z \in \mathbb{C} : |z| = 1\}$):

$$g(z) = \sum_{k=0}^{12} c_k z^{k-6}, \tag{5.18}$$

for some $c_0, \ldots, c_{12} \in \mathbb{C}$. As the equalities on Equation 5.16 are valid for any $\theta \in \mathbb{R}$, the function $g(e^{i\theta})$ and the function ξ are equivalent since $g(e^{i\theta}) = \xi(\theta)$ for any $\theta \in [0, 2\pi)$.

The function g is still not a polynomial as z shows up with negative exponents (up to -6), and its domain is the unit circle $\mathbb{S}$, 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 simply extending the domain from $\mathbb{S}$ to $\mathbb{C}$, as $\mathbb{S} \subset \mathbb{C}$, roots outside the unit circle could appear which requires a further check for every root that is found. Finally, from g, the polynomial $h : \mathbb{C} \mapsto \mathbb{C}$ is defined as

$$h(z) = z^6 g(z) = \sum_{k=0}^{12} c_k z^k. \tag{5.19}$$

By its definition it is possible to see that every root of g is also a root of h, and conversely, every root of h which is in $\mathbb{S}$, is also a root of g. Lastly, every root of g will correspond to a root of ξ through their angles on the unit circle.

5.5.2.1 Further improvements

It is possible to make another reduction to obtain a degree-6 polynomial. As it has been mentioned before, an ellipse is symmetric with respect to its own major or minor axis. In other words, rotating an ellipse by $\theta \in [0,\pi)$ is the same as rotating the same ellipse by $\pi+\theta$. On the other hand, in Chapter 2, it was stated that the angle of the complex numbers z and $-z$ has the same symmetry with each other as the aforementioned symmetry of ellipses:

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

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 $z \in \mathbb{C}$:

$$h(-z) = (-z)^6 g(-z) = z^6 g(z). \tag{5.20}$$

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

$$f(y) = \sum_{k=0}^{6} c_{2k} y^k. \tag{5.21}$$

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 one of the roots will not be between $[0,\pi)$ we can ignore one of them. 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 the identity

$$angle(\sqrt{z}) = angle(z)/2. \tag{5.22}$$

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:

$$c_k = \overline{c_{6-k}}, \tag{5.23}$$

for $k = 0,\ldots,6$. For now, we neither have any ideas on how Equation 5.23 could be proved nor how it could be used to find the roots of f.

Finally, in the next section we use this approach of converting ξ into a complex polynomial to develop an algorithm for E3PNT.

5.6 An algorithm for E3PNT

Among the methods that have been described here, converting ξ into a complex polynomial was the chosen one as the basis of the algorithm for E3PNT. The other approach that showed good results was the numerical approximation one that uses Chebyshev interpolation.

Ultimately, this method requires finding the eigenvalues of a companion matrix, which is usually going to be larger than the 6×6, which is the size of the companion matrix needed to find the roots of the complex polynomial in Algorithm 4.

Details about getting the eigenvalues of a companion matrix, and determining the center of a circumscribed circle of a triangle are omitted from Algorithm 4 for the sake of clarity. Computing the coefficients of the complex polynomial h is done in our implementation using a symbolic computation library for Python called Sympy (for more information about it see Meurer *et al.* (2017)). This way, $c_0, \ldots, c_{12}$ can be written as functions of a E3PNT's instance.

Procedure 4 – The algorithm for E3P.

Input: $u, v, w \in \mathbb{R}^2$, and $a, b \in \mathbb{R}_{>0}$, with $a > b$.
Output: Every solution of E3P.

procedure $E3PNT(u, v, w, a, b)$
  $\hat{u} \leftarrow (0,0)$ ▷ Translate the system, so u is at the origin.
  $\hat{v} \leftarrow v - u$
  $\hat{w} \leftarrow w - u$
  Let $c_0, \ldots, c_{12}$ be the coefficients of polynomial h as Equation 5.19.
  Let A be a 6×6 zero matrix.
  for $i \in \{1, \ldots, 6\}$ **do** ▷ Constructing the companion matrix.
    $A_{i,i+1} \leftarrow 1$
    $A_{6,i} \leftarrow -\dfrac{c_{2(i-1)}}{c_{12}}$
  end for
  $Q \leftarrow \{\}$
  for all $q \in eig(A)$ **do** ▷ $eig(A)$ returns every eigenvalue of A.
    $\theta \leftarrow \min\{angle(-q)/2, angle(q)/2\}$
    if $|\theta| = 1$ **then**
      Let c be the center of the circumscribe circle of $\Lambda(\theta)$.
      $Q \leftarrow Q \cup \{(\varphi^{-1}(c, \theta) + u, \theta)\}$
    end if
  end for
  return Q
end procedure

Theorem 2. The Algorithm 4 computes every solution of E3PNT in $\mathcal{O}(1)$ operations.

Proof. It has already been shown that computing every root of ξ through the complex polynomial yields every solution of E3PNT, the only thing left to prove is the running time of the algorithm. Computing every eigenvalue of a matrix can be done in $\mathcal{O}(n^3)$, but as for our case n is fixed at 6, it can be stated that computing the eigenvalues for the companion matrix of f can be done in $\mathcal{O}(1)$. Therefore, the whole algorithm runs in constant time. □

5.6.1 Numerical Experiments

In this section we show the results of an experiment we made to study the numerical stability of Algorithm 4.

We generate random instances of E3PNT according to

$$u, v, w \sim \mathcal{N}(0, K^2), \tag{5.24}$$

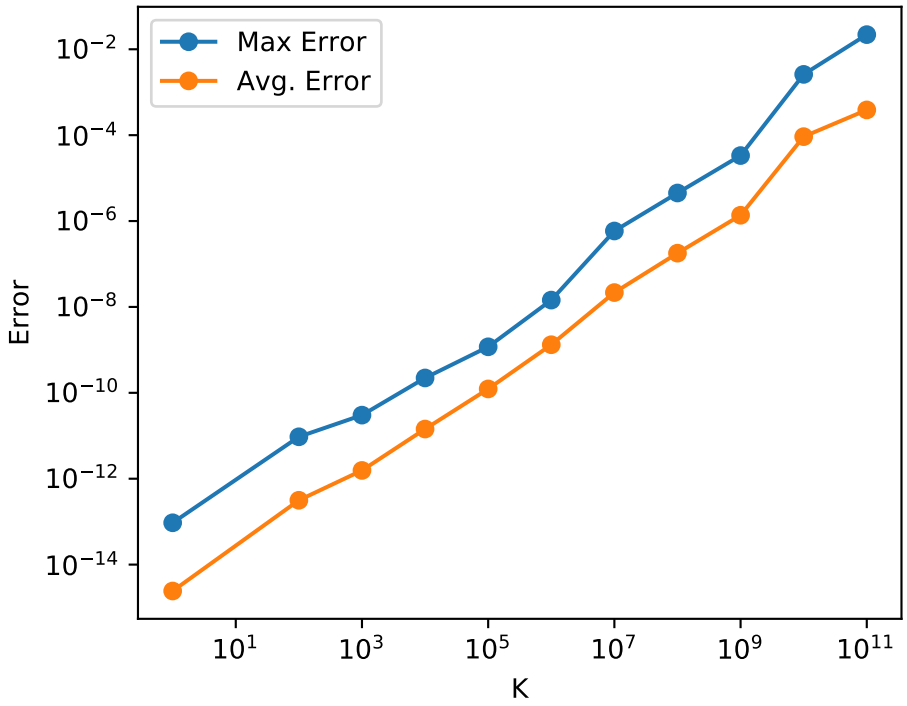
with $K \in \mathbb{R}_{>0}$ being a parameter of the experiment. We also define the ellipse's shape parameters ass

$$a = K \tag{5.25}$$

$$b = \frac{K}{2}. \tag{5.26}$$

For the experiment, we tried several values for K and adopted the error measure to be $|\xi(\hat{\theta})|$, with $\hat{\theta}$ being a root of the complex polynomial h. For each different value of K, we ran the algorithm for 1000 random instances. The results are shown in Figure 12 where the maximum and the average error is plotted for each value of K for which the experiment was run.

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



Source: Elaborated by the author.

From this experiment, it is possible to conclude that the accuray of Algorithm 4 is acceptable when the instance of E3PNT has numerical values within the box $[-10^5, 10^5]^2$. A more rigorous study on the numerical stability of E3PNT is left as future work.

CHAPTER

6

MAXIMAL COVERING BY ELLIPSES WITH ROTATION

This section introduces the elliptical PMCLP where there is no axis-parallel constraint, and the ellipses can be freely rotated. We refer to this problem as Maximal Covering by Ellipses with Rotation (MCER). In comparison with MCE, this problem introduces a new variable that is responsible for determining the rotation angle of every ellipse, making MCER a more challenging problem.

6.1 Definition

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 $\mathcal{P} = \{p_1, \ldots, p_n\}$, $p_j \in \mathbb{R}^2$; a list of weights $\mathcal{W} := \{w_1, \ldots, w_n\}$, with $w_j \in \mathbb{R}_{\geq 0}$ being the weight of point p_j; and m ellipses given by their shape parameters $\mathcal{R} := \{(a_1, b_1), \ldots, (a_m, b_m)\}$, with $(a_j, b_j) \in \mathbb{R}^2_{>0}$ and $a_j > b_j$. Additionally, to make the text more clear, we define a set of m ellipses as $\mathcal{E} = \{E_1, \ldots, E_m\}$, with $E_j \colon \mathbb{R}^2 \times \mathbb{R} \mapsto \mathbb{R}^2$ being a function that takes the center and angle of rotation where the j-th ellipse is located as input, and returns its coverage region as defined by Equation 2.13. Lastly, an instance of MCER is defined as a tuple $(\mathcal{P}, \mathcal{W}, \mathcal{R})$.

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 2.13. Therefore MCER is defined as the problem of determining Q and Θ (placing and rotating each ellipse) to maximize the weight of points covered by the m ellipses, which is given

by

$$\max_{Q,\Theta} w\left(\bigcup_{i=1}^{m} \mathcal{P} \cap E_i(q_i, \theta_i)\right). \tag{6.1}$$

In addition to that, we define an equivalence relation between solutions of MCER. We say that two solutions are equivalent if the set of points covered by them is the same. In other words, two solutions of MCER (Q, Θ) and (Q', Θ') are said to be equivalent if, and only if

$$\bigcup_{j=1}^{m} \mathcal{P} \cap E_j(q'_j, \theta'_j) = \bigcup_{j=1}^{m} \mathcal{P} \cap E_j(q_j, \theta_j).$$

In the next section we present some results which ultimately lead up to the construction of a CLS that contains an optimal solution for MCER.

6.2 Constructing a CLS

The goal of this section is to construct a finite set of solutions, also referred to as CLS, for MCER that contains at least one optimal solution. The main idea for doing that is to take advantage of Lemma 5.1 and use the fact that there are at most six solutions for any instance of E3P. This set of solutions is then used in the next section on the development of a $\mathcal{O}(n^{3m})$ runtime algorithm for MCER.

To begin with, a proposition is introduced which says that given an optimal solution of MCER, it is possible to find another optimal solution with two points on the border of every ellipse that is already covering at least two points.

Proposition 6.1. Let (Q^*, Θ^*) be an optimal solution of an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER. Then, for any $j \in \{1, \ldots, m\}$ with $|\mathcal{P} \cap E_j(q_j^*, \theta_j^*)| \geq 2$, an equivalent solution (Q', Θ^*) exists, such that $|\mathcal{P} \cap \partial E_j(q'_j, \theta_j^*)| \geq 2$.

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

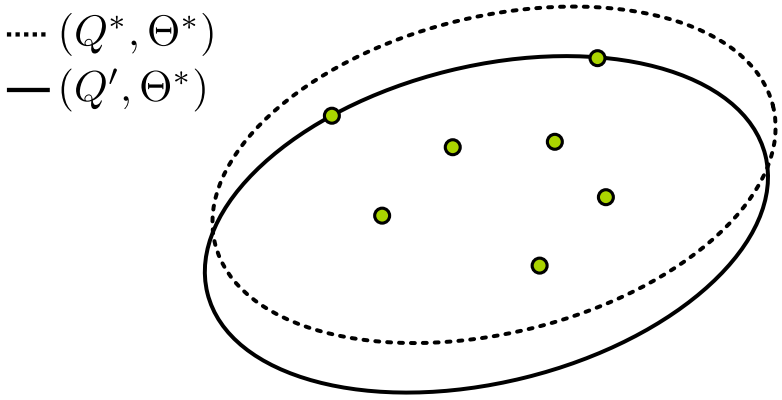
Let $A = \mathcal{P} \cap E_j(q_j^*, \theta_j)$ be the set of points covered by the j-th ellipse and $X = \cap_{p \in A} E_j(p, \theta_j)$ be the region of intersection of ellipses centered at each point in A.

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 $|\mathcal{P} \cap \partial E_j(q'_j, \theta_j)| \geq 2$.

□

What Proposition 6.1 is saying is that any optimal solution for MCER can be transformed into an equivalent optimal solution where every ellipse that covers more than one point has two

Figure 13 – An optimal solution before and after applying Proposition 6.1.



Source: Elaborated by the author.

points on their border. Also, this equivalent optimal solution can be always achieved by just translating the ellipses. An example of that can be seen in Figure 13.

Next, a notation is defined for the set of equivalent solutions of an optimal solution that has at least two points on the border of every ellipse. This helps us make the text less cluttered.

Definition 6.1. Let (Q^*, Θ^*) be an optimal solution for an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER. We define $\Pi(Q^*, \Theta^*)$ as the set of every equivalent solution of (Q^*, Θ^*), such that for any $(Q, \Theta) \in \Pi(Q^*, \Theta^*)$, for $j \in \{1, \ldots, m\}$ with $|\mathcal{P} \cap E_j(q_j^*, \theta_j^*)| \geq 2$, we have $|\mathcal{P} \cap \partial E_j(q_j', \theta_j)| \geq 2$.

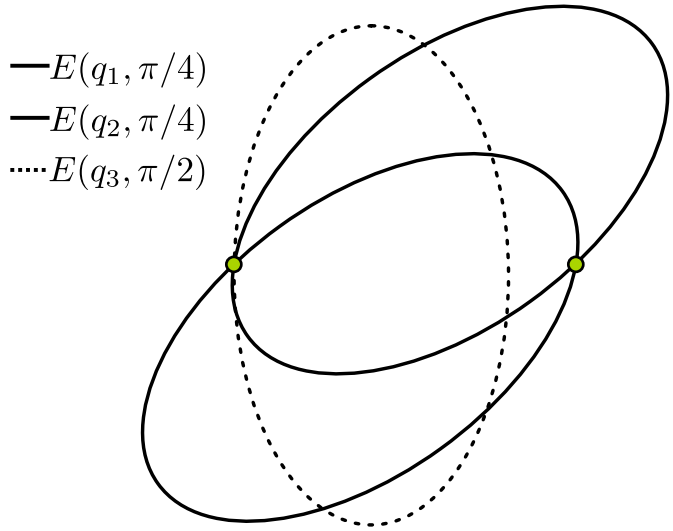
A lot of the ideas developed in this chapter are based on fixing two points on the border of an ellipse. Based on that, Definition 6.2 introduces a new notation for angles that, given two points and an ellipse, it is possible to find a center where the ellipse rotated by that angle is placed, such that the two given points lie on its border.

Definition 6.2. Let E be the coverage region of 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 $\{u, v\} \subset \partial E(q, \theta)$. In addition to that, given an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER, the set of (E_j, u, v)-feasible angles is referred to as

$$\Phi_j(u, v) := \{\theta \in [0, \pi] : \theta \text{ is a } (E_j, u, v)\text{-feasible angle}\}. \tag{6.2}$$

In Figure 14 two examples for Definition 6.2 are shown. The example with a solid contour shows two ellipses rotated by $\pi/4$ with the two given points on its border, making $\pi/4$ a (E, u, v)-feasible angle. The other example, with a dashed contour, presents a case where an ellipse rotated by $\pi/2$ is not able to have the two points on its border, no matter where it is placed; because of that, $\pi/2$ is said to be a not (E, u, v)-feasible angle.

Figure 14 – A (E,u,v)-feasible angle and a not (E,u,v)-feasible angle.



Source: Elaborated by the author.

Following that, we present a lemma that, given an optimal solution, it says that for any ellipse that covers more than two points, an equivalent solution exists with at least one of the two properties:

- Three points lie on the ellipse's border.
- Two points lie on the ellipse's border, for any feasible angle.

Lemma 6.1. Let (Q^*,Θ^*) be an optimal solution of an instance $(\mathcal{P},\mathcal{W},\mathcal{R})$ of MCER; $j \in \{1,\ldots,m\}$, such that $|\mathcal{P}\cap E_j(q_j^*,\theta_j^*)|>2$; $(Q',\Theta')\in\Pi(Q^*,\Theta^*)$; and $\{u,v\}\subset\partial E_j(q_j',\theta_j')$.

If, for every equivalent solution $(\hat{Q},\hat{\theta})$, $|\mathcal{P}\cap\partial E_j(\hat{q}_j,\hat{\theta}_j')|<3$, then for all $\theta\in\Phi_j(u,v)$, there exists $q\in\mathbb{R}^2$, such that $\{u,v\}\subset\partial E_j(q,\theta)$ and $\mathcal{P}\cap E_j(q_j^*,\theta_j^*)=\mathcal{P}\cap E_j(q,\theta)$.

Proof. According to Proposition 6.1, there exists $\{u,v\}\subset\mathcal{P}\cap E_j(q_j^*,\theta_j^*)$, such that an equivalent optimal solution (Q',Θ') exists with u and v on the border of $E_j(q_j',\theta_j^*)$. Therefore, $\theta_j^*\in\Phi_j(u,v)$.

Now suppose that u and v have the same y-coordinate, that is, the angle between them is 0. If they do not, a rotation can be applied to make them have the same y-coordinate. Then, the first thing we are proving is that $\Phi_j(u,v)=[0,2\alpha]$ for a specific case that any instance can be transformed into using translation and rotation on every element of $\mathcal{P}$.

In Chapter 2, a function $L\colon\mathbb{R}\to\mathbb{R}_{\geq 0}$ was defined in Equation 2.11. This function takes the angular coefficient $m\in\mathbb{R}$ and, considering the family of lines parallel to the one described by $y=mx$, returns the maximum squared distance between two intersection points of a line in that family and an axis-parallel ellipse centered at the origin.

To use those results here, we need to consider the ellipse to be fixed at the origin and axis-parallel, and instead, rotate the points in $\mathcal{P}$.

Let $\theta \in [0,\pi] \setminus \{\pi/2\}$, and u', v' be the points u, v after a rotation by θ. Then, if $L(\tan\theta) \geq ||v-u||_2^2$, it is possible to apply a translation to u', v', such that they end up in the border of the fixed ellipse. This means that, it is possible to find an angle of rotation and a center to place E_j, such that it has u, v on its border.

Now we use some properties of function L whose details are given in Chapter 2. Defining $l(\theta) = L(\tan\theta)$, with $l : [0,\pi] \setminus \{\pi/2\}$, we can say that

- l is decreasing in $[0,\pi/2)$ because L is decreasing in $[0,\infty)$. Therefore, if there is $\alpha \in [0,\pi/2)$, such that $l(\alpha) = ||v-u||_2^2$, then $l(\theta) > ||v-u||_2^2$, for $\theta \in (\alpha,\pi/2)$. That implies $[0,\alpha] \subset \Phi_j(u,v)$.
- $l(\theta) = l(\pi-\theta)$ because L is an even function. Therefore, if there is $\alpha \in [0,\pi/2)$, such that $l(\alpha) = ||v-u||_2^2$, then $l(\theta) > ||v-u||_2^2$, for $\theta \in (\pi/2,\pi-\alpha)$. That implies $[\pi-\alpha,\pi] \subset \Phi_j(u,v)$.

We then conclude that $\Phi_j(u,v) = [0,\alpha] \cup [\pi-\alpha,\pi]$, and, of course, in the case that there is no $\alpha \in [0,\pi/2)$, such that $l(\alpha) = ||v-u||_2^2$, we have $\Phi_j(u,v) = [0,\pi]$. From that, if we rotate every point in $\mathcal{P}$ by $\pi-\alpha$, we obtain $\Phi_j(u,v) = [0,2\alpha]$.

With this result in hands, we can use a continuity argument to complete our proof as follows. Let $\delta : \Phi_j(u,v) \mapsto \mathbb{R}^2$ be a continuous function which takes an angle $\theta \in \Phi_j(u,v)$ and returns a center, such that $\{u,v\} \subset \partial E_j(\delta(\theta),\theta)$, and, from solution (Q',Θ'), $\delta(\theta_j') = q_j'$. In general, for any angle in $\Phi_j(u,v)$, there are two possible centers that make $\{u,v\} \subset \partial E_j(\delta(\theta),\theta)$ (see Figure 14 for an example), however, as $\delta(\theta_j') = q_j'$, δ is well-defined. This is shown in Figure 15 where δ is plotted for the whole interval $\Phi_j(u,v)$.

Let $w \in \mathcal{P} \cap E_j(q_j^*,\theta_j^*) \setminus \{u,v\}$, then we define $f_w : \mathbb{R}^3 \mapsto \mathbb{R}_{\geq 0}$ to be a function that takes a center point q and an angle of rotation θ, and returns the elliptical distance between w and $E_j(q,\theta)$ as defined by the left-and-side of Equation 2.12.

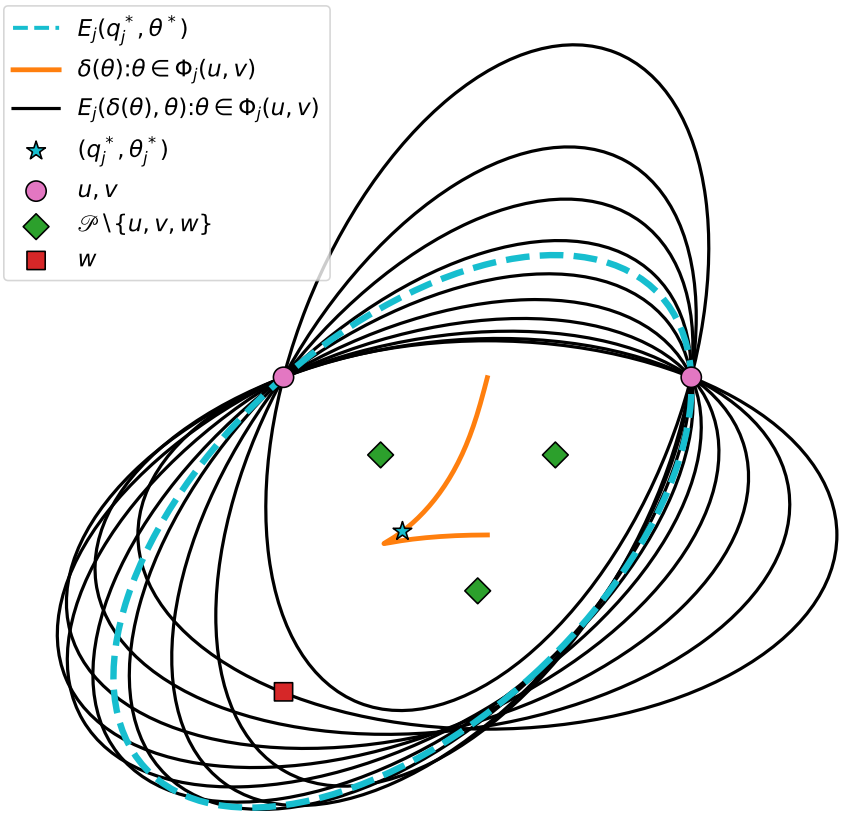
For all $w \in \mathcal{P} \cap E_j(q_j^*,\theta_j^*) \setminus \{u,v\}$, we have that $f_w(q_j',\theta_j^*) < 1$. We then define a function $g_w : \Phi_j(u,v) \mapsto \mathbb{R}_{\geq 0}$ as $g_w(\theta) = f_w(\delta(\theta),\theta)$. As g_w is a composition of continuous functions, it is also continuous.

Therefore, for any $\theta \in \Phi_j(u,v)$, if a point $w \in \mathcal{P} \cap E_j(q_j^*,\theta_j^*) \setminus \{u,v\}$ is not covered by $E_j(\delta(\theta),\theta)$, it must have $g_w(\theta) > 1$. Then, by continuity, another angle $\hat{\theta} \in (\theta_j^*,\theta)$ must exist, such that $g_w(\hat{\theta}) = 1$, which means that $w \in \partial E_j(\delta(\hat{\theta}),\hat{\theta})$, contradicting the hypothesis. The same can be said about a point that enters the coverage of $E_j(\delta(\theta),\theta)$. □

What Lemma 6.1 is saying is that, for every ellipse in an instance of MCER, unless an equivalent optimal solution with three points on its border exits, the angle of rotation can practically be ignored. Because of that, it will be shown that we can construct a CLS which is finite and also contains an optimal solution.

In Figure 15 a visualization of Lemma 6.1 is presented. An initial optimal solution is given by the dashed-border ellipse, and its center, represented by a star point. From it, the continuous function δ is defined by moving the ellipse through the rotation angles in $\Phi_j(u,v)$ while maintaining u, v on its border. Ten angles were chosen from $\Phi_j(u,v)$ to be shown in Figure 15, among those were 0 and $\max\{\Phi_j(u,v)\}$; their corresponding ellipses are displayed with solid-line borders. Consistently with Lemma 6.1, the points in $\mathcal{P} \setminus \{u,v,w\}$ stay within the ellipse's cover for any angle of rotation, and, for point w, there exists an angle, such that it is on the ellipse's border.

Figure 15 – A visualization of Lemma 6.1.



Source: Elaborated by the author.

.

Let (Q^*, Θ^*) be an optimal solution of an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER. We will define a set $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ and show that there exists an equivalent solution (Q', Θ') to (Q^*, Θ^*), such that $(Q', \Theta') \in \Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$. This is the same thing as showing that $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ contains an optimal solution for the instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$. The whole set $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ is constructed from a set of solutions for every ellipse. Let $j \in \{1, \ldots, m\}$, we denote by S_j the CLS for the j-th ellipse; its construction is broken into three separated cases denoted by $S_j^{(1)}$, $S_j^{(2)}$, and $S_j^{(3)}$. Next, we define

them precisely.

6.2.1 First case

This is the case where $|E_j(q_j^*, \theta_j^*)| = 1$, that is, the j-th ellipse only covers one point. This case is not treated by any of the previous results presented in this chapter. Nevertheless, taking this possibility into account can be done by including solutions in S_j that are guaranteed to cover at least one point. This is represented by

$$S_j^{(1)} = \bigcup_{u \in \mathcal{P}} \{(u, 0)\}. \tag{6.3}$$

6.2.2 Second case

The second and third cases are addressed by Lemma 6.1. Consider the second case to be the one where the j-ellipse covers more than one point and there is no equivalent solution with three points on the its border. By Lemma 6.1, (Q^*, Θ^*) can be represented by a solution that has some pair of points u, v on the ellipse's border, and is rotated by some angle in $\Phi_j(u, v)$. As only one angle is needed, we define θ_{uv} as the angle that makes the major-axis of the ellipse to be aligned with the line that passes through u and v; this angle surely is in $\Phi_j(u, v)$ as the longest segment that crosses an ellipse is its major-axis. Having said that, we refer to $S_j^{(2)}$ the set of possible solutions that takes this case into account; we define it as

$$S_j^{(2)} = \bigcup_{\{u,v\} \subset \mathcal{P}} \{(q, \theta_{uv}) \in \mathbb{R}^2 \times \mathbb{R} : \{u, v\} \subset \partial E_j(q, \theta_{uv})\}. \tag{6.4}$$

For a pair of points u, v, determining every ellipse with a fixed rotation angle that has u and v on its border is equivalent to determining the intersection points of two axis-parallel ellipses with the same shape parameters. This problem is discussed with great detail in Appendix A.

6.2.3 Third case

The only case left is the one where there is an equivalent solution with three points on the j-th ellipse's border. We call this set of possible solutions $S_j^{(3)}$, and define it as

$$S_j^{(3)} = \bigcup_{\{u,v,w\} \subset \mathcal{P}} \{(q, \theta) \in \mathbb{R}^2 \times \mathbb{R} : \{u, v, w\} \subset \partial E_j(q, \theta)\}. \tag{6.5}$$

Determining the set $S_j^{(3)}$ can be done using the results from Chapter 5 for every triplet of points.

Finally, we define the CLS for the j-th ellipse as $S_j = S_j^{(1)} \cup S_j^{(2)} \cup S_j^{(3)}$, and the set of possible solutions of an instance of MCER as

$$\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R}) = \{(Q, \Theta) \in \mathbb{R}^{2m} \times \mathbb{R}^m : (q_j, \theta_j) \in S_j \text{ for all } j \in \{1, \ldots, m\}\}. \tag{6.6}$$

Theorem 3. Let $(\mathcal{P},\mathcal{W},\mathcal{R})$ be an instance of MCER, and $\Omega(\mathcal{P},\mathcal{W},\mathcal{R})$ be a set of solutions defined by Equation 6.6. Then there exists an optimal solution $(Q^*,\Theta^*) \in \Omega(\mathcal{P},\mathcal{W},\mathcal{R})$, and $|\Omega(\mathcal{P},\mathcal{W},\mathcal{R})| = \mathcal{O}(n^{3m})$.

Proof. For every $j \in \{1,\ldots,m\}$, by Lemma 6.1, any optimal solution that covers more than one point can be transformed into one that is in $S_j^{(2)}$ or in $S_j^{(3)}$. If an optimal solution covers only one point, it can be transformed into another one that is in $S_j^{(1)}$.

From Appendix A, we can conclude that $|S_j^{(2)}| \leq 2\binom{n}{2} = \mathcal{O}(n^2)$. By Lemma 5.1, we have that $|S_j^{(3)}| \leq 6\binom{n}{3} = \mathcal{O}(n^3)$. Therefore, $|S_j| = \mathcal{O}(n^3)$, and as $|\Omega(\mathcal{P},\mathcal{W},\mathcal{R})| = |S_1| \times |S_2| \times \cdots \times |S_m|$, we have $|\Omega(\mathcal{P},\mathcal{W},\mathcal{R})| = \mathcal{O}(n^{3m})$. □

6.3 An algorithm for MCER

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 25.

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, 15, 38, 39, and 61.

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 61.

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 18.

BANSAL, M.; KIANFAR, K. Planar maximum coverage location problem with partial coverage and rectangular demand and service zones. **INFORMS J. on Computing**, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 29, n. 1, p. 152–169, Feb. 2017. ISSN 1526-5528. Available: <https://doi.org/10.1287/ijoc.2016.0722>. Citation on page 14.

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>. Citations on pages 24 and 25.

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 47.

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 31 and 32.

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, 29, 31, 33, 34, and 61.

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 47 and 48.

BOYD, J. P. **Chebyshev and Fourier Spectral Methods**. Second. Mineola, NY: Dover Publications, 2001. (Dover Books on Mathematics). ISBN 0486411834 9780486411835. Citation on page 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 25 and 49.

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 18.

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, 38, 39, 47, and 61.

CHAZELLE, B. M.; LEE, D. On a circle placement problem. **Computing**, v. 36, p. 1–16, 03 1986. Citations on pages 14, 28, 29, 30, and 33.

CHURCH, R. The planar maximal covering location problem. (symposium on location problems: in memory of leon cooper). **Journal of regional science Philadelphia**, 1984. Citations on pages 13, 14, and 28.

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.

CRAPARO, E. M.; FüGENSCHUH, A.; HOF, C.; KARATAS, M. Optimizing source and receiver placement in multistatic sonar networks to monitor fixed targets. **European Journal of Operational Research**, v. 272, n. 3, p. 816–831, 2019. Available: <https://ideas.repec.org/a/eee/ejores/v272y2019i3p816-831.html>. Citation on page 14.

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 30 and 31.

DREZNER, Z. Note—on a modified one-center model. **Management Science**, v. 27, p. 848–851, 07 1981. Citations on pages 14, 28, and 31.

GAUTSCHI, W. The condition of polynomials in power form. **Mathematics of Computation - Math. Comput.**, v. 33, 01 1979. Citation on page 46.

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 47.

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 33 and 61.

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 24.

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 43.

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 28.

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 45.

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>. Citations on pages 48 and 51.

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 25, 45, and 46.

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.; EISELT, H.; DASKIN, M. A bibliography for some fundamental problem categories in discrete location science. **European Journal of Operational Research**, v. 184, n. 3, p. 817 – 848, 2008. ISSN 0377-2217. Available: <http://www.sciencedirect.com/science/article/pii/S037722170700080X>. 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 24.

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 24.

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 49.

YOUNIES, H.; WESOLOWSKY, G. O. Planar maximal covering location problem under block norm distance measure. **The Journal of the Operational Research Society**, Palgrave Macmillan Journals, v. 58, n. 6, p. 740–750, 2007. ISSN 01605682, 14769360. Available: <http://www.jstor.org/stable/4622758>. Citation on page 14.

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 2.9 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 2.9, we can get that

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

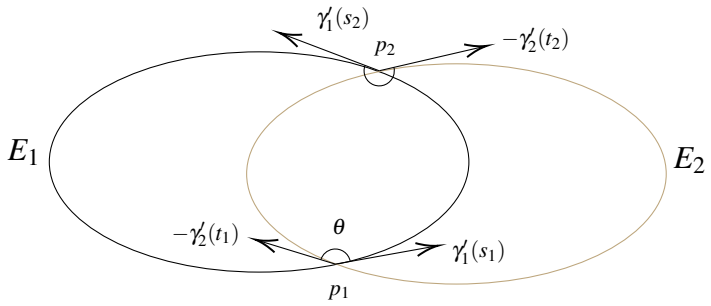
As the image of arctan is $[-\frac{\pi}{2}, \frac{\pi}{2}]$, we need to check the sign of $x-q_x$ to determine the angle in $[0, 2\pi]$. After that, we can get the two angles that represent the intersection points p_1 and p_2 on E_1.

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 16 – Determining $\Gamma_+(1,2)$



Source: Elaborated by the author.

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

Lemma A.1. 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 A.1, a graphical explanation using Figure 16 is provided.

First, let us state some facts that can also be seen in Figure 16

- $E_1 \cap E_2$ 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 16, 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 16), which is what Lemma A.1 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 16, 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