

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Planar Maximal Covering with Ellipses

Danilo França Tedeschi

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

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Danilo Franoso 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 Franoso Tedeschi

Cobertura Planar Maximal por Elipses

Dissertao apresentada ao Instituto de Cincias Matemticas e de Computao – ICMC-USP, como parte dos requisitos para obteno do ttulo de Mestre em Cincias – Cincias de Computao e Matemtica Computacional. *VERSO REVISADA*

rea de Concentrao: Cincias de Computao e Matemtica Computacional

Orientadora: Profa. Dra. Marina Andretta

USP – So Carlos
Maro de 2019

ABSTRACT

TEDESCHI, D. F. **Planar Maximal Covering with Ellipses**. 2019. 73 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

Figure 1 – A non-axis-parallel ellipse and its foci points.	19
Figure 2 – The ellipse as a parametric curve.	20
Figure 3 – Plot of function L in the interval $[-7, 7]$	21
Figure 4 – The rotated ellipse.	22
Figure 5 – The representation of a complex number on two dimensions.	23
Figure 6 – Three disks that have non-empty pairwise intersection among them, but no common intersection.	31
Figure 7 – Three disks and their intersection points.	32
Figure 8 – The intersection list of a disk with three other disks.	33
Figure 9 – Three ellipses and their intersection points	39
Figure 10 – Determining $\Gamma_+(1, 2)$ and $\Gamma_-(1, 2)$	40
Figure 11 – Transforming an ellipse into a circle. T1, T2, and T3 represent the steps of the transformation.	44
Figure 12 – The interpolation error measured on roots that were found.	50
Figure 13 – An example of E3P with six solutions.	54
Figure 14 – An example of E3P with four solutions.	54
Figure 15 – The error measured on solutions found by Algorithm 5.	55
Figure 16 – An optimal solution before and after applying Proposition 6.1.	59
Figure 17 – A (E, u, v) -feasible angle and a not (E, u, v) -feasible angle.	60
Figure 18 – A visualization of Lemma 6.1.	62

LIST OF ALGORITHMS

Algorithm 1 – Algorithm for MWC.	34
Algorithm 2 – Algorithm for MCD1 that returns a CLS.	35
Algorithm 3 – Algorithm for MCE	42
Algorithm 4 – Algorithm for MCE- k	42
Algorithm 5 – The algorithm for E3P.	53
Algorithm 6 – An algorithm that constructs a CLS for a given ellipse.	64
Algorithm 7 – Algorithm for MCER	65
Algorithm 8 – Algorithm for MCER- k	66

CONTENTS

1	INTRODUCTION	13
2	NOTATION AND PRELIMINARIES	17
2.1	Elliptical and Euclidean norm functions	17
2.2	Disk	17
2.3	Ellipse	18
2.3.1	<i>Axis-parallel ellipse</i>	19
2.3.2	<i>The distance between points of an ellipse-line intersection</i>	20
2.3.3	<i>Non-axis-parallel ellipse</i>	21
2.3.4	<i>Notation</i>	22
2.4	Complex numbers	23
2.5	Polynomials and their roots	24
2.6	Real trigonometric polynomial	25
3	MAXIMUM COVERING BY DISKS	27
3.1	Definition	27
3.1.1	<i>CLS and CIPS</i>	28
3.2	Related Work	28
3.3	One disk version	28
3.4	Maximum Weight Clique	29
3.4.1	<i>An algorithm for the Maximum Weight Clique Problem</i>	31
3.5	An algorithm for MCD	34
4	MAXIMUM COVERING BY ELLIPSES	37
4.1	Definition	37
4.2	Related work	38
4.3	One Ellipse Version	38
4.4	Determining $\Gamma_+(i, j)$ and $\Gamma_-(i, j)$	39
4.5	An algorithm for MCE	40
4.6	Adding facility cost	41
5	ELLIPSE BY THREE POINTS	43
5.1	Definition	43
5.2	Transforming the problem	43

5.2.1	<i>A circumradius problem</i>	44
5.2.2	<i>The number of solutions of E3PNT</i>	45
5.3	<i>An attempt using the conic general equation</i>	46
5.4	<i>An approximation method</i>	47
5.4.1	<i>Chebyshev polynomial</i>	47
5.4.2	<i>Chebyshev interpolation</i>	48
5.4.3	<i>Numerical Experiments</i>	49
5.5	<i>Converting ξ into a polynomial</i>	50
5.5.1	<i>Real polynomial</i>	50
5.5.2	<i>Complex polynomial</i>	51
5.5.2.1	<i>Further improvements</i>	52
5.6	<i>An algorithm for E3PNT</i>	52
5.6.1	<i>Solutions for some instances</i>	54
5.6.2	<i>Numerical Stability</i>	54
6	MAXIMAL COVERING BY ELLIPSES WITH ROTATION	57
6.1	<i>Definition</i>	57
6.2	<i>Constructing a CLS</i>	58
6.2.1	<i>First case</i>	63
6.2.2	<i>Second case</i>	63
6.2.3	<i>Third case</i>	63
6.3	<i>An algorithm for MCER</i>	64
6.3.1	<i>Adding facility cost</i>	66
6.3.2	<i>Improvements and implementation details</i>	66
7	FUTURE WORK	69
	BIBLIOGRAPHY	71

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 problems¹ ([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 scale² 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 (PM CER). 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 ?? 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 ??, 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, ?? determines with detail the intersection of two ellipses, which is used in the algorithm developed in ??.

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

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

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

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

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

$$\|p - c\|_2^2 \leq 1. \quad (2.1)$$

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; ESPLIN; GRAY, 1999). From that definition, an equation which describes any conic section is given as follows

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \quad (2.2)$$

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 can be done using the condition

$$4AC - B^2 > 0.$$

More details about conic sections can be found in Ayoub (1993).

Assuming the center of an ellipse is $c \in \mathbb{R}^2$, then Equation 2.2 can be rewritten as a quadratic form as follows

$$(p - c)^T Q (p - c) = 1,$$

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 2.2, Q can be defined as follows

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

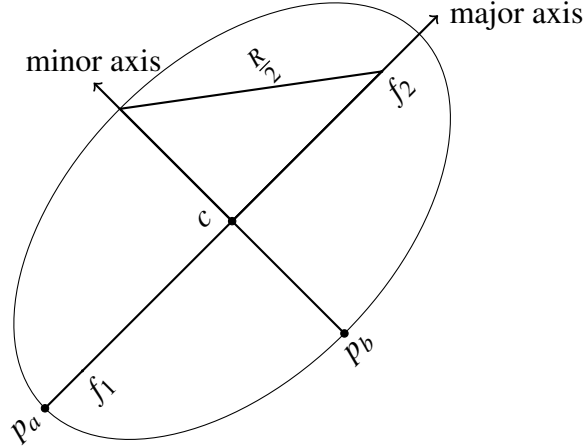
Note that asking Q to be positive definite is the same as asking $4AC - B^2$ to be positive. This makes us arrive at the following definition of the ellipse.

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. 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 ellipse

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.

$$\begin{aligned}
(p - c)^T Q(p - c) &= 1 && \Rightarrow \\
\left(\frac{p_x - c_x}{a^2}, \frac{p_y - c_y}{b^2}\right)^T (p_x - c_x, p_y - c_y) &= 1 && \Rightarrow \\
\frac{(p_x - c_x)^2}{a^2} + \frac{(p_y - c_y)^2}{b^2} &= 1, && (2.3)
\end{aligned}$$

where $(a, b) \in \mathbb{R}_{>0}^2$, with $a > b$, are ellipse's shape parameters. Also, the coverage region is determined by just changing the equality to an inequality as follows

$$\frac{(p_x - c_x)^2}{a^2} + \frac{(p_y - c_y)^2}{b^2} \leq 1. \quad (2.4)$$

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 $(a, b) \in \mathbb{R}_{>0}^2$, with $a > b$. 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

$$\gamma(t) = \begin{cases} x(t) = a \cos t + c_x, \\ y(t) = b \sin t + c_y. \end{cases} \quad (2.5)$$

Also, its derivative with respect to t is given as follows

$$\gamma'(t) = \begin{cases} x'(t) = -a \sin t, \\ y'(t) = b \cos t. \end{cases} \quad (2.6)$$

2.3.2 The distance between points of an ellipse-line intersection

Consider an ellipse with shape parameters $(a, b) \in \mathbb{R}_{>0}^2$, 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 least at one point. Plugging the line's equation into Equation 2.3, it is possible to obtain the distance between the intersection points. The final expression is given by

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

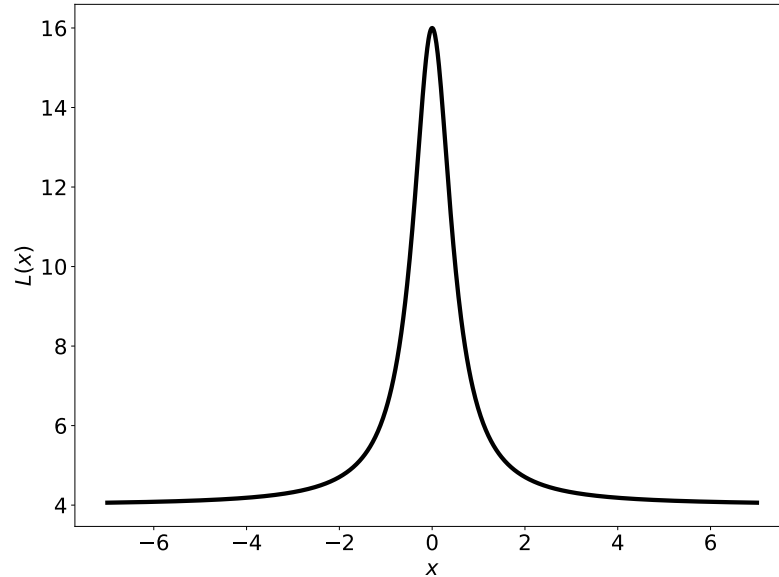
with $D : \mathbb{R}^2 \mapsto \mathbb{R}_{\geq 0}$ being a function of the line parameters (m, c) . It is also possible to see that, when m is fixed, $D(m, c)^2$ is a parabola, and that $D(m, c)$ is maximized at $c = 0$. From that, we can conclude that if m is fixed, the line that has the most distant 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 $L : \mathbb{R} \mapsto \mathbb{R}_{>0}$ as

$$L(m) := D(m, 0)^2 = \frac{(a^2 m^2 + b^2)(4a^2 b^2 (1 + m^2))}{(a^2 m^2 + b^2)^2}, \quad (2.7)$$

which describes the maximum distance between points of an ellipse-line intersection considering all lines with m angular coefficient.

It is possible, by calculating the derivatives, to conclude that L has its maximum at $m = 0$, 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.

2.3.3 Non-axis-parallel ellipse

A non-axis-parallel ellipse centered at $(c_x, c_y) \in \mathbb{R}^2$ can also be described by Definition 2.1, nonetheless, in this work, a simpler equation is used instead. Besides the center and the

shape parameters $(a, b) \in \mathbb{R}_{>0}^2$, with $a > b$; an angle of rotation $\theta \in \mathbb{R}$ is given representing 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.

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

$$\frac{((x - c_x) \cos \theta + (y - c_y) \sin \theta)^2}{a^2} + \frac{((x - c_x) \sin \theta - (y - c_y) \cos \theta)^2}{b^2} = 1. \quad (2.8)$$

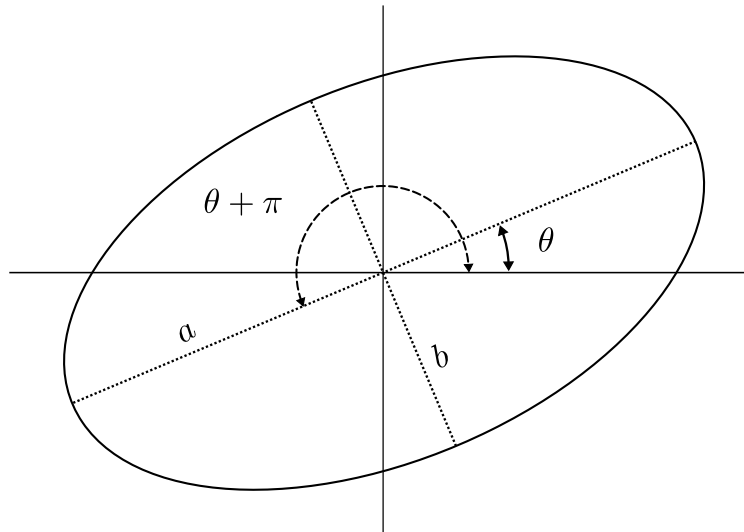
The coverage region of that same ellipse is given by every point $(x, y) \in \mathbb{R}^2$ that satisfies the following equation

$$\frac{((x - c_x) \cos \theta + (y - c_y) \sin \theta)^2}{a^2} + \frac{((x - c_x) \sin \theta - (y - c_y) \cos \theta)^2}{b^2} \leq 1, \quad (2.9)$$

which is the same as [Equation 2.8](#) with the equality sign ($=$) replaced by (\leq).

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 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, to represent any ellipse, it is enough to specify the domain of θ as $[0, \pi)$.

Figure 4 – The rotated ellipse.



Source: Elaborated by the author.

2.3.4 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} can be seen as just an extension of the set of real numbers \mathbb{R} . 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 in [Chapter 5](#).

Any complex number $z \in \mathbb{C}$ is composed of a real part $a \in \mathbb{R}$, and an imaginary part $b \in \mathbb{R}$, multiple of the imaginary unit $i = \sqrt{-1}$. This is expressed as $z = a + ib$. Because complex numbers are composed of two real numbers, mapping \mathbb{C} to \mathbb{R}^2 , as shown in [Figure 5](#), provides a good way to visualize the set of complex numbers. This is also a good way to visualize Euler's Formula. As it can be seen on [Figure 5](#), any complex number can be written in terms of its radius and polar angle as

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

with r being the length of the vector determined by the point z on the complex plane and $\theta = \text{angle}(z)$ 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

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

Lastly, two observations that are very important for the developments of [Chapter 5](#) need to be stated. Let $z \in \mathbb{C}$, then we have

$$\begin{aligned} \text{angle}(\bar{z}) &= 2\pi - \text{angle}(z), \\ \text{angle}(-z) &= \pi + \text{angle}(z). \end{aligned}$$

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

$$p_n(z) = \sum_{k=0}^n a_k z^k, \quad (2.10)$$

with $a_0, \dots, a_n \in \mathbb{C}$. In this work, when a polynomial is written in the form of [Equation 2.10](#) 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 formula² 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 2.10](#) is given by

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

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 it can be found in [Watkins \(2008\)](#)). The first step of the QR algorithm is to convert the input matrix into the Hessenberg form. This is done because matrices in the Hessenberg form maintain its form under the iterations of the algorithm. After that, the algorithm, under some assumptions, converges to the matrix's eigenvalues after $\mathcal{O}(n)$ iterations, each one taking $\mathcal{O}(n^2)$ computations. Another method specific

² A formula with a finite number of $+$, $-$, \times , \div , $\sqrt{\cdot}$.

to companion matrices can be found in [Barel *et al.* \(2010\)](#). It uses the fact that companion matrices are already in the Hessenberg form to propose a $\mathcal{O}(n^2)$ algorithm to find the roots of a n -degree polynomial.

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

$$p_n(\theta) = \sum_{k=0}^n a_k \cos(k\theta) + \sum_{k=1}^n b_k \sin(k\theta). \quad (2.11)$$

We say that $p_n : \mathbb{R} \mapsto \mathbb{R}$ as defined by [Equation 2.11](#) is a n -degree real trigonometric polynomial. An important property is stated in [Powell \(1981, p. 150\)](#), which 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

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

can be transformed into a real trigonometric polynomial of degree $j + k$. Therefore, for some $\{c_{j,k} \in \mathbb{R} : 0 \leq j + k \leq n\}$, the expression

$$\sum_{0 \leq j+k \leq n} c_{j,k} \cos^j \theta \sin^k \theta, \quad (2.12)$$

also represents a n -degree real trigonometric polynomial.

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 an algorithm for it to later adapt it for the elliptical PMCLP in the next chapter.

3.1 Definition

An instance of MCD is given by a set of n demand points $\mathcal{P} := \{p_1, \dots, p_n\}$, with $p_j \in \mathbb{R}^2$; a set of weights $\mathcal{W} := \{w_1, \dots, w_n\}$, with $w_j \in \mathbb{R}_{\geq 0}$ being the weight of point p_j ; and m disks given by their radii $\mathcal{R} := \{r_1, \dots, r_m\}$, with $r_j \in \mathbb{R}_{> 0}$. Additionally, to make the text more clear, we define a set of m disks as $\mathcal{D} := \{D_1, \dots, D_m\}$, 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 2.1](#).

A solution for an instance of MCD is determined by $Q := (q_1, \dots, q_m) \in \mathbb{R}^{2m}$, which specifies the center of every disk in \mathcal{D} . Let $w : 2^{\mathcal{P}} \mapsto \mathbb{R}_{\geq 0}$ be a function, which takes a subset of \mathcal{P} and returns the sum of the weights of every point in it, defined as

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

Then an optimal solution of MCD is formulated as a solution of

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

It is worth mentioning 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 point is considered covered if a facility is located within its radius.

3.1.1 CLS and CIPS

The method proposed in Church (1984) involves the construction of a finite set of locations where each facility can be placed as a way of transforming a problem, where every point in \mathbb{R}^2 is a possible solution, into one where only a finite number of possibilities need to be considered. This set of possible locations is called Candidate List Set (CLS).

In Church (1984), a CLS is set to be the circle intersection point set (CIPS), which contains the intersection of circles with fixed radii centered at every demand point. This approach provides an optimal solution if a complete search on the CLS for every facility is done.

We introduce, in this chapter, a $\mathcal{O}(n^2 \lg n)$ algorithm that returns a CLS for each facility, which is based on the idea presented in Church (1984) and in the works for the one disk case by Chazelle and Lee (1986) and Berg, Cabello and Har-Peled (2006). We refer to the CLS for the j -th facility as S_j and prove that, indeed, an optimal solution can be found by just considering the centers in S_j for each facility.

3.2 Related Work

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.

3.3 One disk version

This version of the problem will be referred to as Maximum Covering by One Unit Disk (*MCD1*) and it is just a specific case of *MCD* with only one disk with radius one ($m = 1$ and $r_1 = 1$). We refer to an instance of *MCD1* as the tuple $(\mathcal{P}, \mathcal{W})$. We later use the algorithm for

MCD1 here described to construct a CLS which is guaranteed to contain an optimal solution for MCD.

Two exact methods for MCD1 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, MCD1 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 MCD1, which meant that no better time-complexity could be achieved. Since then, however, better algorithms for 3SUM have been developed with a $\mathcal{O}(\frac{n^2}{\text{poly}(n)})$ 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 $\binom{n}{2} = \mathcal{O}(n^2)$.

The algorithm for MCD1, developed in this chapter, can be seen as a parallel version of the algorithm developed by [Drezner \(1981\)](#). We, however, based on [Chazelle and Lee \(1986\)](#) and [Berg, Cabello and Har-Peled \(2006\)](#), decided to work with an equivalent problem called Maximum Weight Clique (MWC) which is introduced in the next section.

3.4 Maximum Weight Clique

An instance of the Maximum Weight Clique (MWC) is given by a list of points $\mathcal{P} := \{p_1, \dots, p_n\}$, with $p_i \in \mathbb{R}^2$ representing the center of a unit disk; and $\mathcal{W} := \{w_1, \dots, w_n\}$, with $w_i \in \mathbb{R}_{>0}$ being the weight of the i -th unit disk. We also define a list $\mathcal{D} = \{D_1, \dots, D_n\}$, such that $D_i, i \in \{1, \dots, n\}$, is a unit disk centered at p_i having weight w_i assigned to it.

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 $q \in \bigcup_{j=1}^n D_j$, 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 as follows

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

With a geometric observation, though, the number of possible values for the solution can be reduced. Let $\partial\mathcal{D} = \{\partial D_1, \dots, \partial D_n\}$ be the set of circles corresponding to each disk in \mathcal{D} . Unless a clique is formed by only one disk, its boundary contains at least two points, which are the intersection of two circles that are part of the clique. Because of that, q can be limited to the set of pairwise intersections of $\partial\mathcal{D}$ as well as the set of centers of each disk \mathcal{P} , which considers the case where the optimal clique is composed of only one disk. With this observation, an optimal solution of MWC can be defined by the optimization problem

$$\max_q \sum_{D_k \cap q \neq \emptyset} w_k,$$

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, given an instance of MWC, an optimal solution can be found by going through $\binom{n}{2} + n$ points, that is, $\mathcal{O}(n^2)$ points.

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 are the $\binom{n}{2}$ pairwise intersection of the circumferences and the edges are 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 $\mathcal{O}(n^2)$.

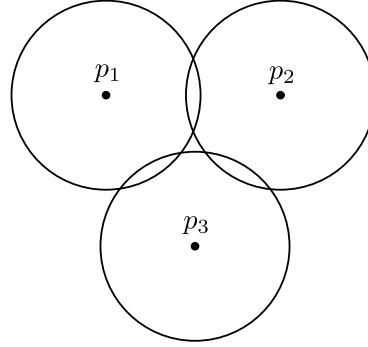
As it has been mentioned, with the equivalence of the two problems, an optimal solution of the Maximum Weight Clique Problem is also an optimal solution of MCD1, which means that a disk centered at q^* which is an optimal solution of MWC, will have a maximal weight covering of the demand set \mathcal{P} .

Given an instance of MCD1, the equivalent MWC instance 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 an 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.

Figure 6 – Three disks that have non-empty pairwise intersection among them, but no common intersection.



Source: Elaborated by the author.

Formally, in MWC, if a point q lies inside $\bigcap_{k \in I} D_k$, with $I \subset \{1, \dots, n\}$, then a disk centered at q will cover the points p_k , with $k \in I$ in the equivalent MCD1 instance. Conversely, the same applies for a disk placed at q that covers points p_k , with $k \in I$ in the MCD1 instance. It means that q will lie inside region $\bigcap_{k \in I} D_k$ in MWC.

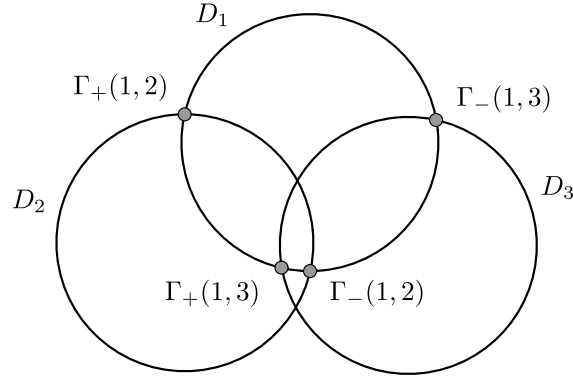
3.4.1 An algorithm for the Maximum Weight Clique Problem

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, though, 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 $\Gamma_-(i, j) = 2\pi$.

Figure 7 – Three disks and their intersection points.



Source: Elaborated by the author.

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 circular list that contains the intersection angles of ∂D_i with every circle in $\partial \mathcal{D}$ defined as

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

Assume also that A_i is sorted in ascending order by the angle values with ties being broken by prioritizing opening angles.

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. Therefore, the list B_i is defined here as a list that contains the elements of A_i and a copy of it shifted to the interval $[2\pi, 4\pi]$. It is defined as

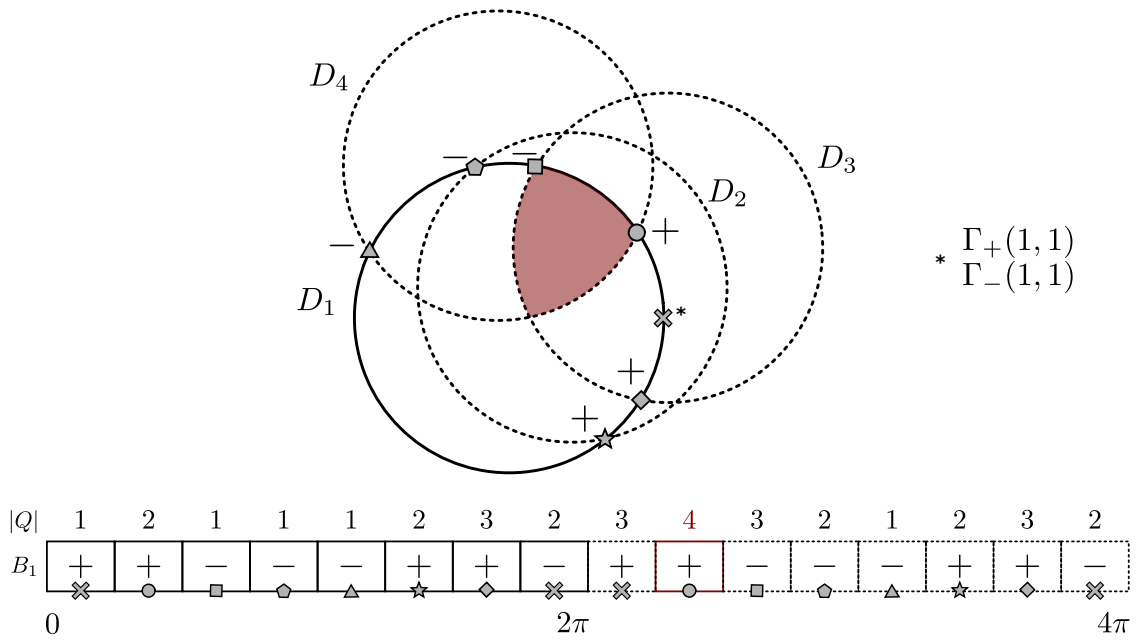
$$B_i = A_i \cup \bigcup_{j=1}^n \{2\pi + \Gamma_-(i, j), 2\pi + \Gamma_+(i, j)\}. \quad (3.2)$$

Assuming B_i is sorted with the same criteria as A_i , a simple traversal, starting at the first element and going until the last one, simulates a traversal on the circular list A_i . This works because for

any pair of disks D_i, D_j ; B_i contains $\Gamma_+(i, j) < \Gamma_-(i, j) + 2\pi$. That is, the algorithm encounters an opening angle before reaching a closing one for any circle.

In Figure 8, the intersection points between ∂D_1 (solid border) with $\partial D_2, \partial D_3$, and ∂D_4 (dashed border) are shown with a plus or minus sign indicating opening or closing intersection angles. The intersection list B_1 is also displayed in Figure 8 along with the size of the set of active disks Q after processing a point in B_1 – this is exactly what Algorithm 1 for MWC does for every disk. It is possible to see that the optimal clique highlighted in Figure 8 is enclosed by the arcs defined by $\Gamma_+(1, 4)$ and $\Gamma_-(1, 4)$, and can also be identified by following B_1 while keeping track of Q . The special intersection point of ∂D_1 with itself can also be seen in Figure 8. Its usage is very convenient as with $\Gamma_+(1, 1)$ and $\Gamma_-(1, 1)$ in B_1 , the algorithm inserts D_1 in the set of active disks before processing any point, and removes D_1 only after every point has been processed.

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



Source: Elaborated by the author.

Finally, we define Algorithm 1 for MWC. Given an instance $(\mathcal{P}, \mathcal{W})$ of MWC, the algorithm uses the approach described here of keeping a set of active disks while traversing the list B_i . It returns a point that is inside an optimal clique, this way Algorithm 1 can also be used to get an optimal solution for an instance $(\mathcal{P}, \mathcal{W})$ of MCD1.

Algorithm 1 – Algorithm for MWC.**Input:** A set of points $\mathcal{P} = \{p_1, \dots, p_n\}$, and a set of weights $\mathcal{W} = \{w_1, \dots, w_n\}$.**Output:** A point that is inside the maximum weight clique of unit disks.

```

1: procedure  $MWC(\mathcal{P}, \mathcal{W})$ 
2:   Let  $\mathcal{D} = \{D_1, \dots, D_n\}$  be a set of unit disks, with centers in  $\mathcal{P}$  and weights in  $\mathcal{W}$ 
3:    $Q_{best} \leftarrow \{\}$ 
4:    $q^* \leftarrow p_1$ 
5:   for all  $D_i \in \mathcal{D}$  do
6:     Let  $B_i$  be the list of intersection angles of  $D_i$  as defined by Equation 3.2
7:      $Q \leftarrow \{\}$  ▷ The set of active disks.
8:     for  $\theta \in B_i$  do ▷ Assuming  $B_i$  is sorted.
9:       Let  $D_j$  be the disk, such that  $\theta \in D_j \cap D_i$ 
10:      if  $\theta$  is a opening angle then
11:         $Q \leftarrow Q \cup \{D_j\}$ 
12:      else
13:         $Q \leftarrow Q \setminus \{D_j\}$ 
14:      end if
15:      if  $w(Q_{best}) < w(Q)$  then
16:         $Q_{best} \leftarrow Q$ 
17:         $q^* \leftarrow$  point corresponding to the intersection angle  $\theta$ 
18:      end if
19:    end for
20:  end for
21:  return  $q^*$ 
22: end procedure

```

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 $\mathcal{O}((n + K) \log n)$. \square

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

3.5 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 MCD1 is defined only for unit disks, however, this constraint 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 [Theorem 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 $j \in \{1, \dots, m\}$ 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^*, \dots, 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 $|S_j| = \mathcal{O}(n^2)$. \square

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 ?? 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.

Algorithm 2 – Algorithm for MCD1 that returns a CLS.

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

```

1: procedure CLS-MCD( $\mathcal{P}, \mathcal{W}, r$ )
2:    $S \leftarrow \{\}$ 
3:   for all  $p_i \in \mathcal{P}$  do
4:     Let  $B_i$  be the list of intersection angles of  $\partial D_i(p_i)$  as defined by Equation 3.2
5:     for  $\theta \in B_i$  do ▷ Assuming  $B_i$  is sorted.
6:       if  $\theta$  is a opening angle then
7:         Let  $q_\theta$  be the intersection point correspondent to angle  $\theta$ 
8:          $S \leftarrow S \cup \{q_\theta\}$ 
9:       end if
10:    end for
11:  end for
12:  return  $S$ 
13: end procedure

```

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 one developed for MCD in [Chapter 3](#).

4.1 Definition

Axis-parallel ellipses are defined as the set of points that satisfy [Equation 2.3](#). All it takes to describe one is a pair of positive real numbers $(a, b) \in \mathbb{R}_{>0}^2$, with $a > b$, also called its shape parameters, and a center $q \in \mathbb{R}^2$.

An instance of MCE is given by a set of n demand points $\mathcal{P} = \{p_1, \dots, p_n\}$, with $p_j \in \mathbb{R}^2$; a set of weights $\mathcal{W} := \{w_1, \dots, w_n\}$, 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 $\mathcal{R} := \{(a_1, b_1), \dots, (a_m, b_m)\}$, with $(a_j, b_j) \in \mathbb{R}_{>0}^2$ and $a_j > b_j$. Additionally, to make the text more clear, we define a set $\mathcal{E} = \{E_1, \dots, E_m\}$, 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 2.4](#).

A solution for MCE is given by $Q := (q_1, \dots, q_m) \in \mathbb{R}^{2m}$, with q_j being the center of j -th ellipse. 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

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

In the next sections, we first study the specific case of MCE with only one facility and discuss that the results of [Chapter 3](#) still apply when ellipses are used instead of disks. After that,

we use the approach mentioned in [Chapter 3](#) of constructing a CLS for each facility and propose an algorithm for MCE.

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 (MCE1).

An instance of MCE1 has $m = 1$, and we set $(a, b) := (a_1, b_1)$, and $\mathcal{E} := \{E\}$. Therefore, an instance of MCE1 is described by the tuple $(\mathcal{P}; \mathcal{W}; (a, b))$. A solution of MCE1 is then given by a point $q \in \mathbb{R}^2$, and an optimal solution is given by the optimization problem

$$\max_q w(\mathcal{P} \cap E(q)).$$

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 in the next section.

Figure 9 – Three ellipses and their intersection points



Source: Elaborated by the author.

4.4 Determining $\Gamma_+(i, j)$ and $\Gamma_-(i, j)$

Let $E_1(q_1)$, and $E_2(q_2)$ be two coverage region of ellipses centered at $q_1, q_2 \in \mathbb{R}^2$ respectively, with shape parameters $(a, b) \in \mathbb{R}_{>0}^2$. After changing the coordinates to make the center of the first ellipse be at the origin, the intersection points between the two ellipses are defined by

$$\begin{aligned} \frac{x^2}{a^2} + \frac{y^2}{b^2} &= 1 & (E_1) \\ \frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} &= 1 & (E_2), \end{aligned} \quad (4.1)$$

where $(h, k) \in \mathbb{R}^2$ is the center of the second ellipse after the coordinates were translated by $-q_1$. As both equations are equal to 1, we have

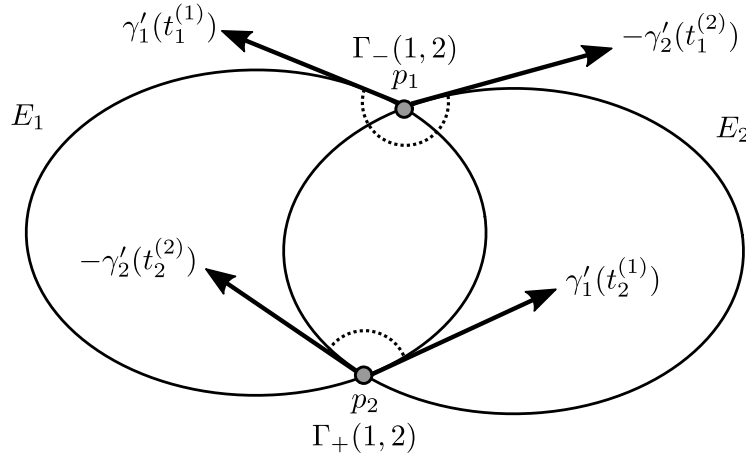
$$\begin{aligned} b^2x^2 + a^2y^2 &= b^2(x-h)^2 + a^2(y-k)^2 \\ x &= y \frac{-2ka^2}{2hb^2} + \frac{b^2h^2 + a^2k^2}{2hb^2} \\ x &= y\alpha + \beta. \end{aligned} \quad (4.2)$$

Replacing Equation 4.2 into Equation 4.1, we get

$$y^2(b^2\alpha^2 + a^2) + y(2\beta\alpha b^2) + b^2\beta^2 - a^2b^2 = 0, \quad (4.3)$$

which is a second degree polynomial. Then, $\partial E_1(q_1) \cap \partial E_2(q_2) \neq \{\}$ if, and only if the roots of Equation 4.3 are real. The intersection points itself can be obtained by solving the polynomial for y and plugging its value back into the $x = y\alpha + \beta$ equation.

Suppose that $\partial E_1(q_1) \cap \partial E_2(q_2) = \{p_1, p_2\}$, with $p_1 \neq p_2$. To determine $\Gamma_+(1, 2)$ and $\Gamma_-(1, 2)$, we need to first determine the intersection angles corresponding to p_1 and p_2 on $E_1(q_1)$.

Figure 10 – Determining $\Gamma_+(1, 2)$ and $\Gamma_-(1, 2)$.

Source: Elaborated by the author.

Let γ_1 and γ_2 be two curves defined as Equation 2.5 for $E_1(q_1)$ and $E_2(q_2)$ respectively. The intersection angle of p_i in $E_j(q_j)$ is defined as $t_i^{(j)} \in [0, 2\pi)$, such that $\gamma_j(t_i^{(j)}) = p_i$, for $i, j \in \{1, 2\}$. Obtaining $t_i^{(j)}$ can be done analytically solving the equation

$$\frac{a p_{iy} - q_{jy}}{b p_{ix} - q_{jx}} = \tan(t_i^{(j)}).$$

Let γ'_1 and γ'_2 be the derivatives of γ_1 and γ_2 respectively as defined by Equation 2.6. Then, considering the tangent vectors $\gamma'_1(t_i^{(1)})$ and $-\gamma'_2(t_i^{(2)})$, $i \in \{1, 2\}$, as shown in Figure 10, we can define a function $\phi_{1,2}: \{p_1, p_2\} \rightarrow \{\Gamma_+(1, 2), \Gamma_-(1, 2)\}$ that takes an intersection point and returns its corresponding closing/opening intersection angle as

$$\phi_{1,2}(p_i) = \begin{cases} \Gamma_+(1, 2) & \text{angle}(\gamma'_1(t_i^{(1)}), -\gamma'_2(t_i^{(2)})) > \pi \\ \Gamma_-(1, 2) & \text{angle}(\gamma'_1(t_i^{(1)}), -\gamma'_2(t_i^{(2)})) < \pi. \end{cases}$$

Lastly, the case where that angle is equal to π happens only when both ellipses intersect at only one point. This case has to be treated separately as, by Definition 3.1, we need to have two equal intersection points: one as $\Gamma_+(1, 2)$ and the other as $\Gamma_-(1, 2)$.

4.5 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 CLS-MCE (we do not define it in this chapter because it would look the same as CLS-MCD defined in Algorithm 2, 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 CLS-MCE 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.

4.6 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 $\mathcal{C} = \{c_1, \dots, c_m\}$, with $c_j \in \mathbb{R}_{\geq 0}$ being the j -th ellipse's cost, and $k \in \mathbb{N}$, $k \leq m$.

A solution for MCE- k , however, when compared to MCE's, has a bit more cluttered description. We define it as a set $I := \{i_1, \dots, i_k\} \subset \{1, \dots, m\}$, such that $|I| = k$; and a tuple $Q := (q_1, \dots, q_k)$, 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

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

Finally, [Algorithm 3](#) can serve as basis for MCE- k 's [Algorithm 4](#). Firstly, for every subset $I \subset \{1, \dots, m\}$, 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 $I \subset \{1, \dots, m\}$, an optimal solution can be obtained. This simple adjustment achieves a run-time complexity of $\mathcal{O}(\binom{m}{k} \times n^{2k}) = \mathcal{O}(n^{3m})$.

Algorithm 3 – Algorithm for MCE

Input: A set of points $\mathcal{P} = \{p_1, \dots, p_n\}$, a list of weights $\mathcal{W} = \{w_1, \dots, w_n\}$, and a list of shape parameters $\mathcal{R} = \{(a_1, b_1), \dots, (a_m, b_m)\}$.

Output: An optimal solution for MCE.

```

1: procedure  $MCE(\mathcal{P}, \mathcal{W}, \mathcal{R})$ 
2:   return  $MCE_{bt}(\mathcal{P}, \mathcal{W}, \mathcal{R}, 1)$ 
3: end procedure
4:
5: procedure  $MCE_{bt}(Z, \mathcal{W}, \mathcal{R}, j)$ 
6:   if  $j = m + 1$  then
7:     return 0
8:   end if
9:    $(q_j^*, \dots, q_m^*) \leftarrow (0, \dots, 0)$ 
10:   $S_j \leftarrow \text{CLS-MCE}(Z, a_j, b_j)$ 
11:  for  $q_j \in S_j$  do
12:     $Cov \leftarrow \mathcal{P} \cap E_j(q_j)$ 
13:     $(q_{j+1}, \dots, q_m) \leftarrow MCE_{bt}(Z \setminus Cov, \mathcal{W}, \mathcal{R}, j + 1)$ 
14:    if  $w(\cup_{k=j}^m Z \cap E_k(q_k)) > w(\cup_{k=j}^m Z \cap E_k(q_k^*))$  then
15:       $(q_j^*, \dots, q_m^*) \leftarrow (q_j, \dots, q_m)$ 
16:    end if
17:  end for
18:  return  $(q_j^*, \dots, q_m^*)$ 
19: end procedure

```

Algorithm 4 – Algorithm for MCE- k

Input: A set of points $\mathcal{P} = \{p_1, \dots, p_n\}$, a list of weights $\mathcal{W} = \{w_1, \dots, w_n\}$, a list of shape parameters $\mathcal{R} = \{(a_1, b_1), \dots, (a_m, b_m)\}$, a list of costs $\mathcal{C} = \{c_1, \dots, c_m\}$, and $k \in \mathbb{N}$.

Output: An optimal solution for MCE- k .

```

1: procedure  $MCE-k(\mathcal{P}, \mathcal{W}, \mathcal{R}, \mathcal{C}, k)$ 
2:    $I^* = \{i_1^*, \dots, i_k^*\} \leftarrow \{1, \dots, k\}$ 
3:    $Q^* = (q_1^*, \dots, q_k^*) \leftarrow (0, \dots, 0)$ 
4:   for all  $I = \{i_1, \dots, i_k\} \subset \{1, \dots, m\}$  do
5:      $\mathcal{R}' \leftarrow \{(a_j, b_j) \in \mathcal{R} : j \in I\}$ 
6:      $(q_1, \dots, q_k) \leftarrow MCE(\mathcal{P}, \mathcal{W}, \mathcal{R}')$ 
7:     if  $w(\cup_{j=1}^k \mathcal{P} \cap E_{i_j}(q_j)) - \sum_{j \in I} c_j > w(\cup_{j=1}^k \mathcal{P} \cap E_{i_j^*}(q_j^*)) - \sum_{j \in I^*} c_j$  then
8:        $Q^* \leftarrow (q_1, \dots, q_k)$ 
9:        $I^* \leftarrow I$ 
10:    end if
11:  end for
12:  return  $I^*, Q^*$ 
13: end procedure

```

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

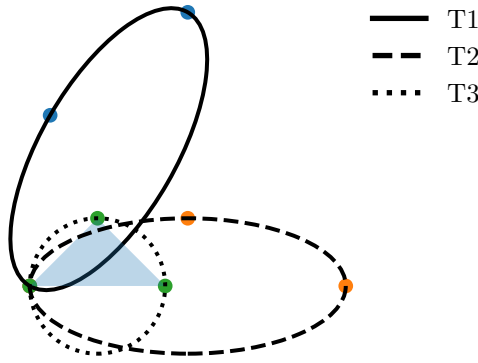
We call the problem introduced in this section Ellipse by Three Points (E3PNT). An instance of it is given by three points $u, v, w \in \mathbb{R}^2$ and the shape parameters of an ellipse $(a, b) \in \mathbb{R}_{>0}^2$, with $a > b$.

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 11 – 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 11 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

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

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 11, 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

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

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

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

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. Any instance of 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 2.12. 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 2.12:

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

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

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. \square

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

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0. \quad (5.6)$$

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

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

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

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 \mathcal{A} and wants to select a simpler function f^* from a set of functions \mathcal{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

$$T_n(x) = \cos(n \arccos x). \quad (5.9)$$

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

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

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

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

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 form¹ 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

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

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

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

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

for $j = 1, \dots, n + 1$. This particular choice defeats Runge's phenomenon and provides a convergent approximation. Note that, if the domain of the function to be interpolated is defined on a range other than $[-1, 1]$, let us say $[a, b]$, then the transformation

$$\hat{x}_j = \frac{a + b}{2} + \frac{b - a}{2} x_j \quad (5.14)$$

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 $\hat{x}_1, \dots, \hat{x}_n$. 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

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

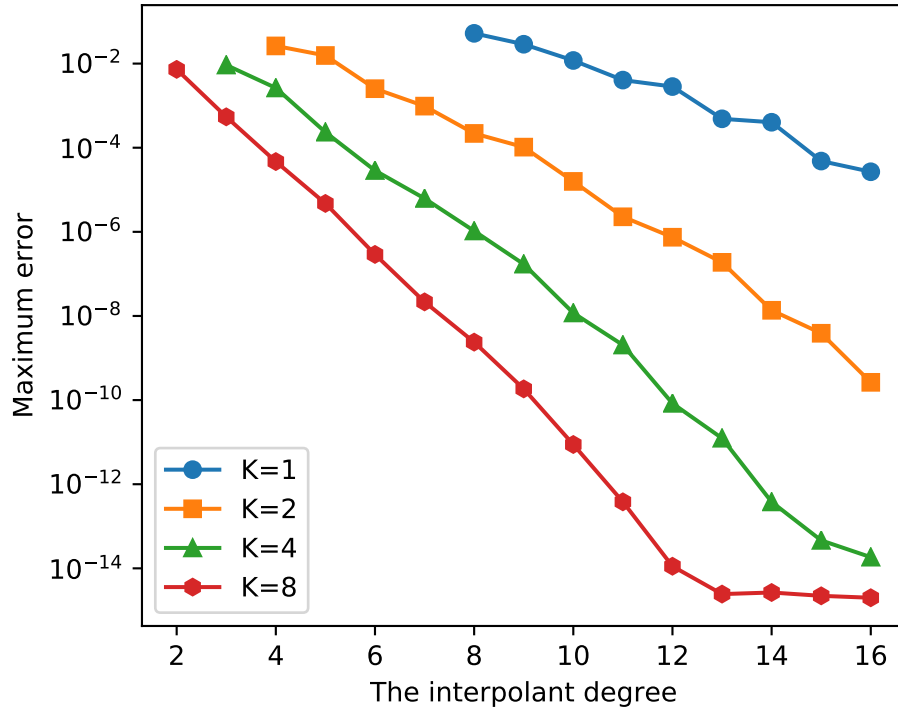
which measures how far the interpolation is at the point it worst approximates.

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 12 where the worst error found among every root found

Figure 12 – 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 12](#) 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.

$$\cos \theta = \frac{e^{i\theta} + e^{-i\theta}}{2} \quad (5.16)$$

$$\sin \theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}. \quad (5.17)$$

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

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

for some $c_0, \dots, c_{12} \in \mathbb{C}$. 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

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

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:

$$\text{angle}(-z) = \pi + \text{angle}(z).$$

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 $z \in \mathbb{C}$:

$$h(-z) = (-z)^6 g(-z) = z^6 g(z). \quad (5.20)$$

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

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

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 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

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

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:

$$c_k = \overline{c_{6-k}}, \quad (5.23)$$

for $k = 0, \dots, 6$. 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 5](#).

Details about getting the eigenvalues of a companion matrix, and determining the center of a circumscribed circle of a triangle are omitted from [Algorithm 5](#) 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, \dots, c_{12} can be written as functions of a E3PNT's instance.

Algorithm 5 – 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.

```

1: procedure E3PNT( $u, v, w, a, b$ )
2:    $\hat{u} \leftarrow (0, 0)$  ▷ Translate the system, so  $u$  is at the origin.
3:    $\hat{v} \leftarrow v - u$ 
4:    $\hat{w} \leftarrow w - u$ 
5:   Let  $c_0, \dots, c_{12}$  be the coefficients of polynomial  $h$  as Equation 5.19.
6:   Let  $A$  be a  $6 \times 6$  zero matrix.
7:   for  $i \in \{1, \dots, 6\}$  do ▷ Constructing the companion matrix.
8:      $A_{i,i+1} \leftarrow 1$ 
9:      $A_{6,i} \leftarrow -\frac{c_{2(i-1)}}{c_{12}}$ 
10:  end for
11:   $Q \leftarrow \{\}$ 
12:  for all  $q \in \text{eig}(A)$  do ▷  $\text{eig}(A)$  returns every eigenvalue of  $A$ .
13:     $\theta \leftarrow \min\{\text{angle}(-q)/2, \text{angle}(q)/2\}$ 
14:    if  $|\theta| = 1$  then
15:      Let  $c$  be the center of the circumscribe circle of  $\Lambda(\theta)$ .
16:       $Q \leftarrow Q \cup \{(\varphi^{-1}(c, \theta) + u, \theta)\}$ 
17:    end if
18:  end for
19:  return  $Q$ 
20: end procedure

```

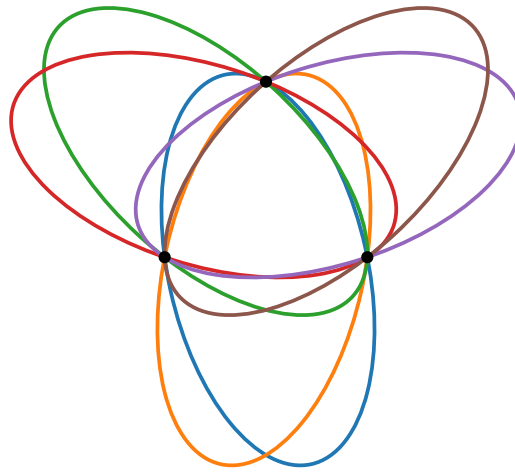
Theorem 1. The [Algorithm 5](#) computes every solution of E3PNT in $\mathcal{O}(1)$ operations.

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. \square

5.6.1 Solutions for some instances

From [Lemma 5.1](#), we know that any instance of E3P has at most six solutions. At first, though, it seemed like that bound was loose as we could not find any instances that made [Algorithm 5](#) return six solutions. After some investigation, this turned out not to be the case. An example with six solutions is presented in [Figure 13](#). It is worth mentioning that the only instances that we found which produce six solutions have their three points form an equilateral triangle.

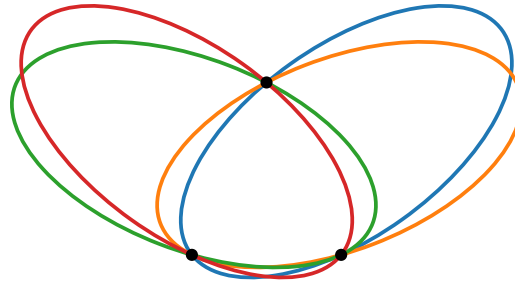
Figure 13 – An instance of E3P with six solutions.



Source: Elaborated by the author.

Another example is shown in [Figure 14](#) where an instance of E3P is displayed along with its four solutions.

Figure 14 – An instance of E3P with four solutions.



Source: Elaborated by the author.

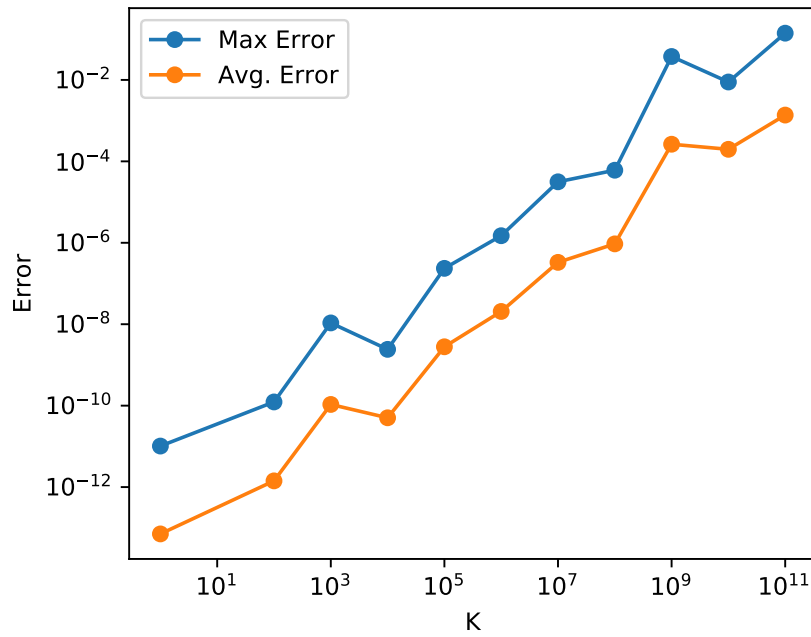
5.6.2 Numerical Stability

In this section we show the results of an experiment that is developed to study the numerical stability of [Algorithm 5](#).

We define $K \in \mathbb{R}_{>0}$ as an experiment parameter and consider instances that the ellipse has shape parameters $(K, \frac{K}{2})$. Then, random instances are created by setting the three points to be $(K \cos(t_j), \frac{K \sin(t_j)}{2})$, $j = 1..3$, with t_j being uniformly distributed in the interval $[0, 2\pi]$. To evaluate the stability of [Algorithm 5](#), for any solution $(\hat{q}, \hat{\theta})$ found, we adopt its error measure to be $|\xi(\hat{\theta})|$.

The experiment was run for various values of K . For each one of them, 100 randomly generated instances were considered. The results are shown in [Figure 15](#), where the average and the maximum error are plotted.

Figure 15 – The error measured on solutions found by [Algorithm 5](#).



Source: Elaborated by the author.

From this experiment, it is possible to conclude that the accuracy of [Algorithm 5](#) 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.

MAXIMAL COVERING BY ELLIPSES WITH ROTATION

This chapter 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 $\mathcal{P} = \{p_1, \dots, p_n\}$, $p_j \in \mathbb{R}^2$; a list of weights $\mathcal{W} := \{w_1, \dots, w_n\}$, with $w_j \in \mathbb{R}_{\geq 0}$ being the weight of point p_j ; and m ellipses given by their shape parameters $\mathcal{R} := \{(a_1, b_1), \dots, (a_m, b_m)\}$, with $(a_j, b_j) \in \mathbb{R}_{>0}^2$ and $a_j > b_j$. Additionally, to make the text more clear, we define a set of m ellipses as $\mathcal{E} = \{E_1, \dots, E_m\}$, with $E_j: \mathbb{R}^2 \times \mathbb{R} \mapsto \mathbb{R}^2$ 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 2.9](#). Lastly, an instance of MCER is defined as a tuple $(\mathcal{P}, \mathcal{W}, \mathcal{R})$.

Given an instance of *MCER*, we define $Q := (q_1, \dots, q_m) \in \mathbb{R}^{2m}$ to be the centers of each ellipse, $\Theta := (\theta_1, \dots, \theta_m) \in [0, \pi)^m$ 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 2.9](#). 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

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

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 $j \in \{1, \dots, m\}$ 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.

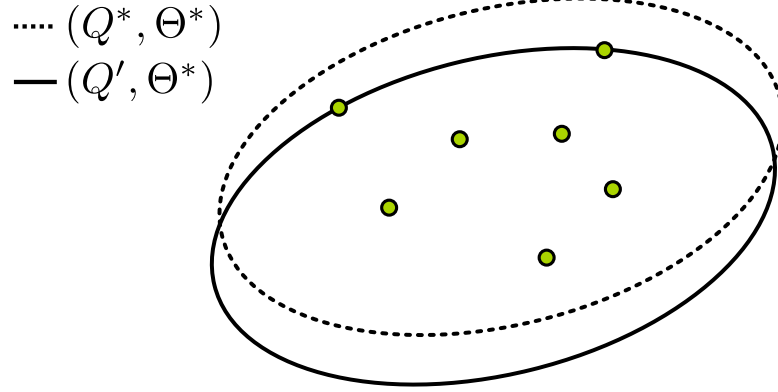
Let $A = \mathcal{P} \cap E_j(q_j^*, \theta_j^*)$ be the set of points covered by the j -th ellipse and $X = \bigcap_{p \in A} E_j(p, \theta_j)$ be the region of intersection of ellipses centered at each point in A .

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 16 – 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 16.

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.

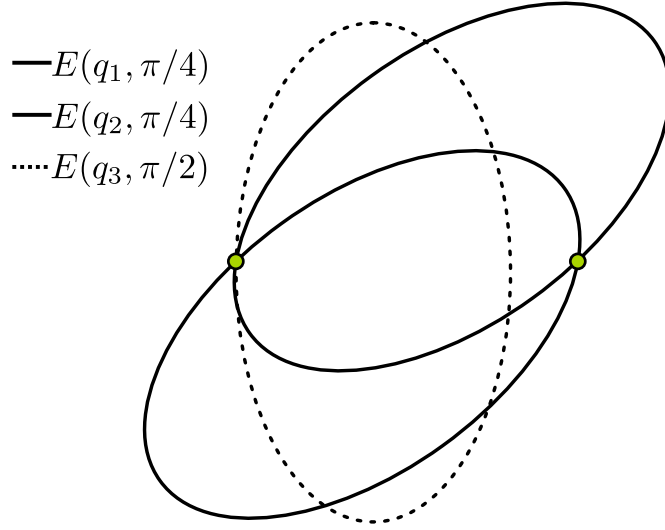
Definition 6.1. Let (Q^*, Θ^*) be an optimal solution for an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER. We define $\Pi(Q^*, \Theta^*)$ as the set of every equivalent solution of (Q^*, Θ^*) , such that for any $(Q, \Theta) \in \Pi(Q^*, \Theta^*)$, for $j \in \{1, \dots, m\}$ with $|\mathcal{P}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

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

In Figure 17 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 17 – 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; $j \in \{1, \dots, m\}$, 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 $L: \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ was defined in Equation 2.7. 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 17](#) for an example), however, as $\delta(\theta'_j) = q'_j$, δ is well-defined. This is shown in [Figure 18](#) 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 2.8](#).

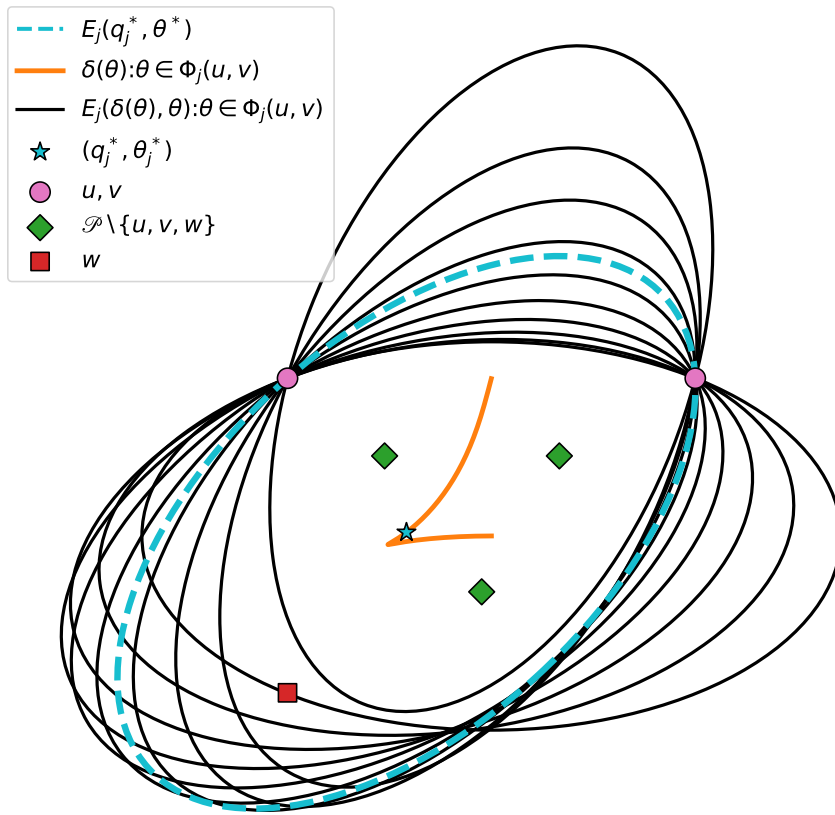
For all $w \in \mathcal{P} \cap E_j(q_j^*, \theta_j^*) \setminus \{u, v\}$, we have that $f_w(q'_j, \theta'_j) < 1$. We then define a function $g_w : \Phi_j(u, v) \mapsto \mathbb{R}_{\geq 0}$ as $g_w(\theta) = f_w(\delta(\theta), \theta)$. As g_w is a composition of continuous functions, it is also continuous.

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 $E_j(\delta(\theta), \theta)$. \square

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 exists, 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 18 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 18, 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 18 – 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 $j \in \{1, \dots, m\}$, 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

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

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

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

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 [section 4.4](#).

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

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

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

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

Theorem 2. 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 $j \in \{1, \dots, m\}$, 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 section 4.4, 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 $|\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})| = |S_1| \times |S_2| \times \dots \times |S_m|$, we have $|\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})| = \mathcal{O}(n^{3m})$. \square

6.3 An algorithm for MCER

In this section we describe an algorithm for MCER that does a complete search on the CLS of each ellipse.

Firstly, we describe an algorithm that constructs a CLS for a given ellipse. Let E be the coverage region of an ellipse with shape parameters (a, b) . We assume that in Algorithm 6, the procedure $e2p(u, v, a, b)$ returns every $(q, \theta) \in \mathbb{R}^2 \times \mathbb{R}$, such that, $\{u, v\} \subset \partial E(q, \theta)$, and θ is an rotation angle that makes E 's major-axis parallel to the line defined by u and v . Also, the procedure $e3p$ called in Algorithm 6 is defined in Algorithm 5.

Algorithm 6 – An algorithm that constructs a CLS for a given ellipse.

Input: A set of points $\mathcal{P} = \{p_1, \dots, p_n\}$, and an ellipse's shape parameters (a, b) .

Output: A CLS for the ellipse with shape parameters (a, b) considering the demand set \mathcal{P} .

```

1: procedure CLS-MCER( $\mathcal{P}, a, b$ )
2:    $S \leftarrow \{\}$ 
3:   for  $u \in \mathcal{P}$  do
4:      $S \leftarrow S \cup \{(u, 0)\}$ 
5:   end for
6:   for  $\{u, v\} \in \mathcal{P}$  do
7:      $S \leftarrow S \cup e2p(u, v, a, b)$ 
8:   end for
9:   for  $\{u, v, w\} \in \mathcal{P}$  do
10:     $S \leftarrow S \cup e3p(u, v, a, b)$  ▷ Defined in Algorithm 5.
11:   end for
12:   return  $S$ 
13: end procedure

```

After that, we define Algorithm 7 which takes an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER, and returns an optimal solution for it. Even though it is based on Theorem 2, the set of solutions

$\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ is not explicitly built in Algorithm 7. Instead, a complete search is done by backtracking the CLS of every ellipse returned by procedure CLS-MCER.

Two procedures are defined in Algorithm 7. The first one, called *MCER*, returns an optimal solution for an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ using the second procedure *MCER_{bt}*. This second procedure is responsible for the backtracking and takes two additional parameters $j \in \{1, \dots, m\}$, which represents the index of the ellipse that *MCER_{bt}* is currently processing, and $Z \subset \mathcal{P}$, which represents the set of points that have not been covered by the ellipses already processed by *MCER_{bt}*.

Algorithm 7 – Algorithm for MCER

Input: A set of points $\mathcal{P} = \{p_1, \dots, p_n\}$, a list of weights $\mathcal{W} = \{w_1, \dots, w_n\}$, and a list of shape parameters $\mathcal{R} = \{(a_1, b_1), \dots, (a_m, b_m)\}$.

Output: An optimal solution of MCER.

```

1: procedure MCER( $\mathcal{P}, \mathcal{W}, \mathcal{R}$ )
2:   return MCERbt( $\mathcal{P}, \mathcal{W}, \mathcal{R}, 1$ )
3: end procedure

4: procedure MCERbt( $Z, \mathcal{W}, \mathcal{R}, j$ )
5:   if  $j = m + 1$  then
6:     return 0
7:   end if
8:    $ans \leftarrow 0$ 
9:    $(q_j^*, \dots, q_m^*); (\theta_j^*, \dots, \theta_m^*) \leftarrow (0, \dots, 0); (0, \dots, 0)$   $\triangleright$  Setting to 0 as a default value.
10:   $S_j \leftarrow \text{CLS-MCER}(Z, a_j, b_j)$ 
11:  for  $(q_j, \theta_j) \in S_j$  do
12:     $Cov \leftarrow \mathcal{P} \cap E_j(q_j, \theta_j)$ 
13:     $(q_{j+1}, \dots, q_m); (\theta_{j+1}, \dots, \theta_m) \leftarrow \text{MCER}_{bt}(Z \setminus Cov, \mathcal{W}, \mathcal{R}, j + 1)$ 
14:     $Cov \leftarrow \bigcup_{k=j}^m \mathcal{P} \cap E_k(q_k, \theta_k)$ 
15:    if  $w(Cov) > ans$  then
16:       $ans \leftarrow w(Cov)$ 
17:       $(q_j^*, \dots, q_m^*); (\theta_j^*, \dots, \theta_m^*) \leftarrow (q_j, \dots, q_m); (\theta_j, \dots, \theta_m)$ 
18:    end if
19:  end for
20:  return  $(q_j^*, \dots, q_m^*); (\theta_j^*, \dots, \theta_m^*)$ 
21: end procedure

```

Corollary 6.1. Algorithm 7 takes $\mathcal{O}(n^{3m})$ computations and returns an optimal solution for an instance of MCER.

Proof. For every $j \in \{1, \dots, m\}$, unless $Z = \{\}$, when choosing the center and angle of rotation for the j -th ellipse, Algorithm 7 does not consider any $(q_j, \theta_j) \in \mathbb{R}^2 \times \mathbb{R}$, such that $Z \cap E_j(q_j, \theta_j) = \{\}$. Apart from those solutions, which are non-optimal, Algorithm 7 considers every solution in $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$. \square

6.3.1 Adding facility cost

In this section, we consider the extended version of MCER where each facility has a cost assigned to it and exactly k of them must be used in a solution. We refer to this version of the problem as Maximum Covering by Ellipses with Rotation and a k -constraint (MCER- k).

An instance of MCER- k has the same parameters as MCER plus a list of costs $\mathcal{C} := \{c_1, \dots, c_m\}$, with $c_j \in \mathbb{R}_{>0}$ being the cost of the j -th facility; and $k \in \mathbb{N}$, $k \leq m$.

A solution for MCER- k is given by (I, Q, Θ) , with $I := \{i_1, \dots, i_k\} \subset \{1, \dots, m\}$; $Q := (q_1, \dots, q_k) \in \mathbb{R}^{2k}$, with q_j being the center of the i_j -th ellipse; and $\Theta := (\theta_1, \dots, \theta_k) \in [0, \pi)^k$, with θ_j being the angle of rotation of the i_j -th ellipse. An optimal solution of MCER- k is given by the optimization problem

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

Then, in the same way that it is done in [Chapter 4](#), we introduce [Algorithm 8](#) for MCER- k that uses [Algorithm 7](#) for every $I := \{i_1, \dots, i_k\} \subset \{1, \dots, m\}$. Therefore, [Algorithm 8](#) returns an optimal solution for MCER- k in $\mathcal{O}(\binom{m}{k} n^{3k}) = \mathcal{O}(n^{4m})$ time.

Algorithm 8 – Algorithm for MCER- k

Input: A set of points $\mathcal{P} = \{p_1, \dots, p_n\}$, a list of weights $\mathcal{W} = \{w_1, \dots, w_n\}$, a list of shape parameters $\mathcal{R} = \{(a_1, b_1), \dots, (a_m, b_m)\}$, a list of costs $\mathcal{C} = \{c_1, \dots, c_m\}$, and $k \in \mathbb{N}$.

Output: An optimal solution for MCER- k .

```

1: procedure MCE- $k(\mathcal{P}, \mathcal{W}, \mathcal{R}, \mathcal{C}, k)$ 
2:    $I^* = \{i_1^*, \dots, i_k^*\} \leftarrow \{1, \dots, k\}$ 
3:    $Q^* = (q_1^*, \dots, q_k^*) \leftarrow (0, \dots, 0)$ 
4:    $\Theta^* = (\theta_1^*, \dots, \theta_k^*) \leftarrow (0, \dots, 0)$ 
5:   for all  $I = \{i_1, \dots, i_k\} \subset \{1, \dots, m\}$  do
6:      $\mathcal{R}' \leftarrow \{(a_j, b_j) \in \mathcal{R} : j \in I\}$ 
7:      $(q_1, \dots, q_k); (\theta_1, \dots, \theta_k) \leftarrow \text{MCER}(\mathcal{P}, \mathcal{W}, \mathcal{R}')$ 
8:     if  $w(\bigcup_{j=1}^k \mathcal{P} \cap E_{i_j}(q_j, \theta_j)) - \sum_{j \in I} c_j > w(\bigcup_{j=1}^k \mathcal{P} \cap E_{i_j^*}(q_j^*, \theta_j^*)) - \sum_{j \in I^*} c_j$  then
9:        $Q^* \leftarrow (q_1, \dots, q_k)$ 
10:       $\Theta^* \leftarrow (\theta_1, \dots, \theta_k)$ 
11:       $I^* \leftarrow I$ 
12:   end if
13: end for
14:   return  $I^*, Q^*, \Theta^*$ 
15: end procedure

```

6.3.2 Improvements and implementation details

Some improvements can be made in the implementation of [Algorithm 7](#). Instead of calling procedure CLS-MCER every time in the backtracking routine, it is suggested that it is

done outside in a pre-processing phase.

Let (Q, Θ) and (Q', Θ') be two solutions of MCER. If, for any $j \in \{1, \dots, m\}$, we have $\mathcal{P} \cap E_j(q'_j, \theta'_j) \subset \mathcal{P} \cap E(q_j, \theta_j)$, solution (Q', Θ') can be dismissed. This can be done in the CLS-MCER procedure in [Algorithm 6](#).

In our implementation we use the same approach used in [Andretta and Birgin \(2013\)](#). When constructing the CLS for the j -th ellipse, a tree-like data struct is utilized to verify if a solution (q, θ) covers a subset of another solution or not.

Keeping an upper-bound for the solution during the backtracking in [Algorithm 7](#) can also provide a good improvement in practice. Let $z_j = \max_{(q, \theta) \in S_j} w(\mathcal{P} \cap E_j(q, \theta))$, and OPT the value of an optimal solution. When the algorithm chooses $(q_j, \theta_j) \in S_j$ as the solution for the j -th ellipse, we have

$$OPT \leq w\left(\bigcup_{k=1}^j \mathcal{P} \cap E_k(q_k, \theta_k)\right) + \sum_{k=j+1}^m z_k. \quad (6.7)$$

This upper-bound is easy to compute as $\{z_1, \dots, z_m\}$ can be pre-processed, and the weight of every point covered so far in the backtrack can be obtained by adding another parameter to the MCER procedure.

In the next section we present the results of numerical experiments for MCER and MCER- k and compare them with the results published in [Andretta and Birgin \(2013\)](#).

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.; SORESENSEN, D. **LA-PACK 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, 41, 67, and 69.

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 69.

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>. Citation on page 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 49.

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 34.

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, 28, 29, 31, 35, and 69.

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 49 and 50.

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

_____. 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 51.

BRANNAN, D.; ESPLIN, 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, 41, 49, and 69.

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

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, 27, 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, 29, and 31.

GAUTSCHI, W. The condition of polynomials in power form. **Mathematics of Computation - Math. Comput.**, v. 33, 01 1979. Citation on page 48.

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 49.

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 28 and 69.

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 45.

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 29.

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 47.

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 50 and 53.

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, 47, and 48.

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 unsolvability 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 51.

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.

