UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Algorithms for Maximum Covering by Ellipses Problems

Danilo Françoso Tedeschi

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

ICMC USP
SÃO CARLOS

Danilo Françoso Tedeschi

Algorithms for Maximum Covering by Ellipses Problems

Thesis 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 Doctor in Science. *EXAMINATION BOARD PRESENTATION COPY*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Marina Andretta

USP – São Carlos
March 2020

Danilo Françoso Tedeschi

Algoritmos para Problemas de Cobertura Máxima por Elipses

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *EXEMPLAR DE DEFESA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Marina Andretta

USP – São Carlos
Março de 2020

ABSTRACT

Maximum covering by ellipses is an optimization problem where one wants to place fixed shape ellipses on the plane to cover demand points maximizing a function depending on the value of covered points. We propose new exact algorithms for two versions of this problem, one where the ellipses have to be parallel to the coordinate axis, and another where they can be freely rotated. Besides finding optimal solutions for previously published instances, including the ones where no optimal solution was known, both algorithms proposed by us were able to obtain optimal solutions for some new larger instances having with up to seven hundred demand points and five ellipses.

Keywords: Optimization, Planar Maximal Covering Location Problem, Maximum Covering by Ellipses.

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF TABLES

CONTENTS

CHAPTER

1

INTRODUCTION

The Minimum Covering Problem – also known as just the Set Cover Problem –, and the Maximal Covering Problem are the two main types of optimal covering problems found in the literature (KARATAS; RAZI; TOZAN, 2016).

One of the 21 Karp's NP-Complete problems[1] (KARP, 1972), the Minimum Covering 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 Covering 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 an instance of 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 **??**.

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

PMCLP under Euclidean distance is also found in the literature as Maximum Covering by Disks (MCD). 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.

We study two versions of PMCLP with elliptical coverage facilities in this work. For both of them, each ellipse is defined to have a fixed shape and an undefined location, which is part of the solution. In the first version, introduced in Canbolat and Massow (2009), all the ellipses are restricted to be axis-parallel, while in the second version, introduced in Andretta and Birgin (2013), this constraint is dropped, and all the ellipses can be freely rotated. The first version will be referred to as Maximum Cover by Ellipses Problem (MCEP) and the second one as Maximum Covering by Ellipses with Rotation Problem (MCERP).

The main practical motivation to study these two versions of PMCLP is that cellphone towers can have an elliptically shaped coverage area. Then, 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).

It is fair to say that PMCLP with elliptical coverage has not been vastly studied as only two articles have been found on it. 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

---

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

method took too long and did not find an optimal solution. Because of that, a heuristic method was developed using a technique called Simulated Annealing, which was able to obtain solutions for the instances proposed in that study. The problem was further explored in Andretta and Birgin (2013), which introduced the version where the ellipses can be freely rotated, to which an exact and a heuristic method was proposed, and developed a new method for the axis-parallel version of the problem, which was able to obtain optimal solutions for instances that the method proposed by Canbolat and Massow (2009) could not. The exact method for the version with rotation could not obtain optimal solutions within a predefined time limit for several instances, the heuristic method though returned solutions for every instance, and impressively enough, obtained optimal solutions for every verifiable instance.

Our work is structured in the following way: in Chapter 2, some definitions and results that are used throughout the next chapters are introduced; in **??**, the maximum covering by disks problem is studied, and an algorithm is proposed for it; in **??**, the maximum covering by ellipses is introduced, and the algorithm for the disks case is adapted for it; in Chapter 4 and Chapter 5 the main results of our work are presented; in Chapter 4, we introduce a new geometry problem and propose an algorithm for it; in Chapter 5, we use the algorithm developed in Chapter 4 to propose a new algorithm for MCERP; in Chapter 6, numerical experiments are analyzed, and implementation details are given; finally, a conclusion is presented Chapter 7, along with some suggestions of what can be done in future works on this subject; finally, a conclusion is presented Chapter 7, along with some suggestions of what can be done in future works on this subject.

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 one.

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

$$||p - c||_2^2 \leq 1. \tag{2.1}$$

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

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \tag{2.2}$$

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.
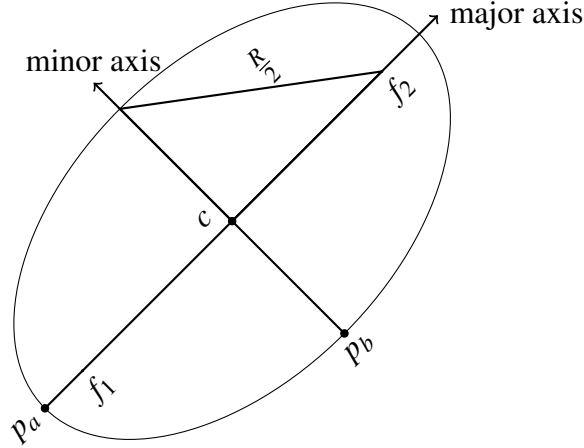
Source: Elaborated by the author.

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.

$$(p-c)^T Q(p-c) = 1 \qquad\qquad \Rightarrow$$

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

$$\frac{(p_x - c_x)^2}{a^2} + \frac{(p_y - c_y)^2}{b^2} = 1, \qquad\qquad (2.3)$$

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

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

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.
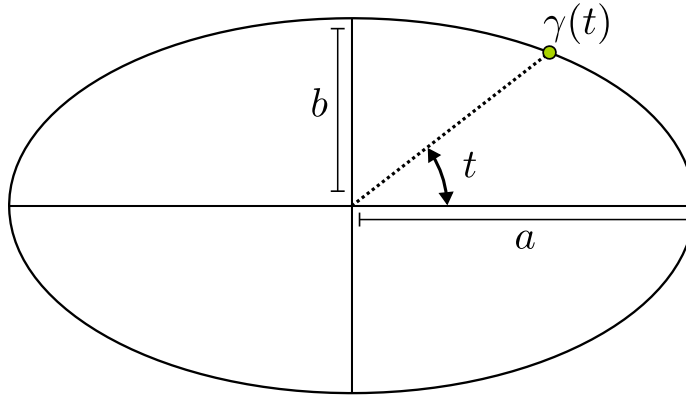
Source: Elaborated by the author.

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

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

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

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

2.3.2   *The distance between points of an ellipse-line intersection*

Consider an ellipse with shape parameters $(a,b) \in \mathbb{R}^2_{>0}$, centered at the origin, and a line represented by the equation $y = mx + c$, with $m, c \in \mathbb{R}$. Suppose that this line intersects the

ellipse at 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

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

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

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

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 $[-7,7]$.
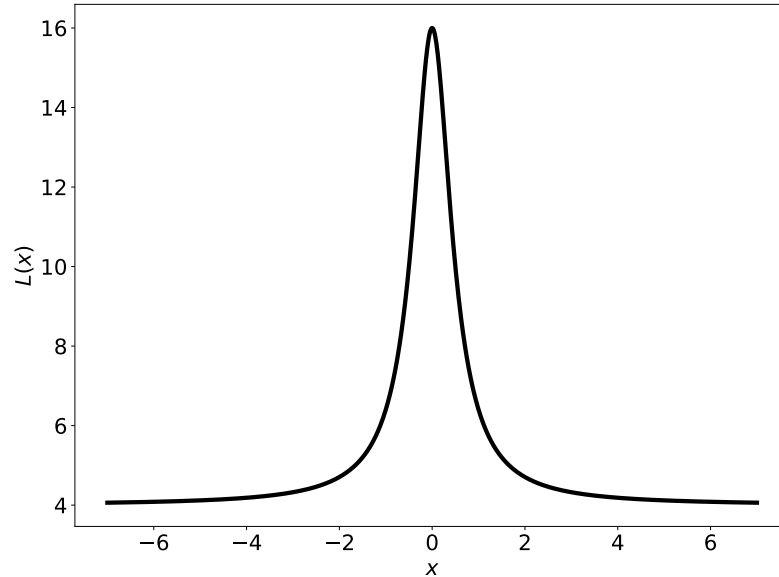
Source: Elaborated by the author.

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

$$\frac{((x - c_x)\cos\theta + (y - c_y)\sin\theta)^2}{a^2} + \frac{((x - c_x)\sin\theta - (y - c_y)\cos\theta)^2}{b^2} - 1 = 0. \qquad (2.8)$$

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

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

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.
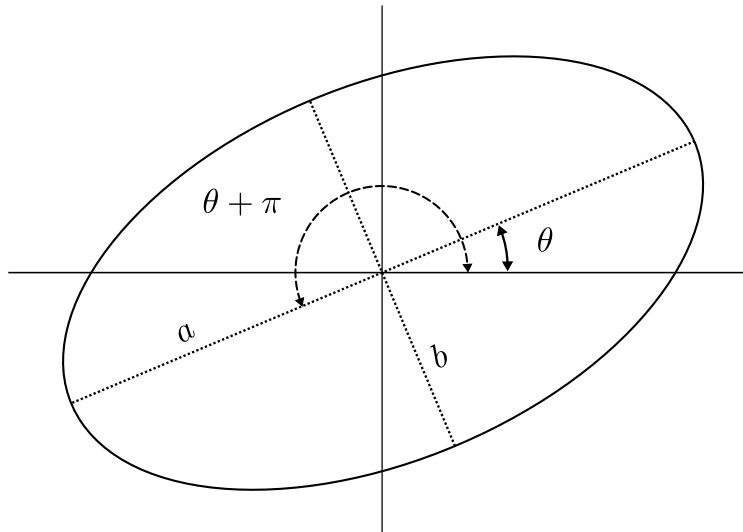
Source: Elaborated by the author.

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 4.

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 $\theta = angle(z)$ being its polar angle. Note that *angle* is a function from $\mathbb{C}$ to $[0, 2\pi)$.
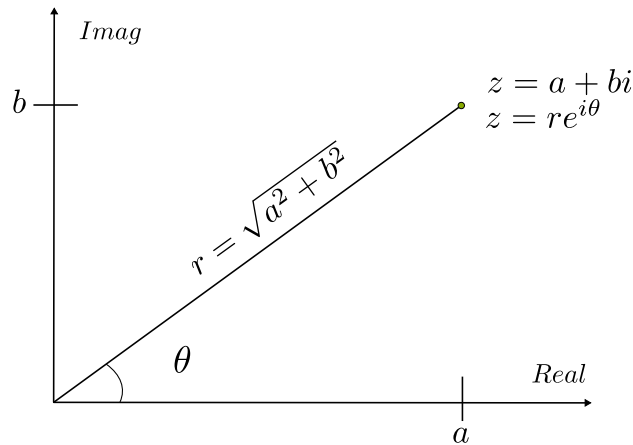
Figure 5 – The representation of a complex number on two dimensions.

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 4 need to be stated. Let $z \in \mathbb{C}$, then we have

$$angle(\bar{z}) = 2\pi - angle(z),$$
$$angle(-z) = \pi + angle(z). \tag{2.10}$$

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

2.5   Polynomials and their roots

In this work, we are mostly interested in univariate polynomials defined over the complex numbers. A function $p_n : \mathbb{C} \mapsto \mathbb{C}$ is a n-degree polynomial if it can be written as

$$p_n(z) = \sum_{k=0}^{n} a_k z^k, \tag{2.11}$$

with $a_0, \ldots, a_n \in \mathbb{C}$. In this work, when a polynomial is written in the form of Equation 2.11 we say that it is in the power form or in the monomial form.

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

In Horn and Johnson (1986, p. 195) a theorem is presented which says that for every univariate polynomial of degree n, there exists a companion matrix which is a $n \times n$ matrix, such that its eigenvalues are the zeros of that polynomial. For example, the companion matrix of a degree-5 polynomial written as Equation 2.11 is given by

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

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 $+, -, \times, \div, \sqrt{\ }$.

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 proposes 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

$$p_n(\theta) = \sum_{k=0}^{n} a_k \cos(k\theta) + \sum_{k=1}^{n} b_k \sin(k\theta). \tag{2.12}$$

We say that $p_n : \mathbb{R} \mapsto \mathbb{R}$ as defined by Equation 2.12 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

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

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

$$\sum_{0 \le j+k \le n} c_{j,k} \cos^j \theta \sin^k \theta, \tag{2.13}$$

also represents a *n*-degree real trigonometric polynomial.

CHAPTER

3

MAXIMUM COVERING BY ELLIPSES

In this chapter, we consider the problem, which we refer to as Maximum Covering by Ellipses (MCE). We introduce an algorithm for it, which in fact, works not only for ellipses but for any disk in a strictly convex normed plane.

3.1 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 $(a, b) \in \mathbb{R}^2_{>0}$, 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 $(a_j, b_j) \in \mathbb{R}^2_{>0}$ 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 $w : 2^{\mathcal{P}} \mapsto \mathbb{R}_{\geq 0}$ be a function, which takes a subset of $\mathcal{P}$ and returns the sum of the weights of every point in it, defined as

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

then an optimal solution of MCE is given by the optimization problem

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

3.2   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.

3.3   Strictly Convex Normed Plane

Based on the results of Martín and Martini (2015), instead of developing an algorithm for MCE, we are going to consider a more general case and develop an algorithm that works for any set of facilities with a strictly convex coverage region.

A strictly convex normed space is defined in Wada (1958) as a space whose every chord of its unit sphere has its midpoint in the interior of its unit ball. Thus, $(\mathbb{R}^2, ||\cdot||)$ is a strictly convex normed plane if the unit disk defined by $||\cdot||$ is a strictly convex body. For more details about convexity and strictly convexity see MANI-LEVITSKA (1993).

For example, the norms $||\cdot||_p$, $p \in \mathbb{N}, p > 1$, define strictly convex unit disks, while $||\cdot||_1$ and $||\cdot||_\infty$ do not. Ellipses are also strictly convex bodies, given an axis-parallel ellipse with shape parameters (a, b), a norm $||\cdot||_{a,b}$ can be defined, such that its unit disk is the coverage region of that ellipse.

From now on, since ellipses are just unit circles in a strictly convex normed space $(\mathbb{R}^2, ||\cdot||_{a,b})$, with $||\cdot||_{a,b}$ being an elliptical norm whose unit disk is an ellipse with shape parameters (a, b), we will develop this work for any set of strictly convex unit disks.

3.4   Maximum Weight Clique

We introduce in this section a problem equivalent to MCE with only one facility. We refer to this equivalent problem as Maximum Weight Clique (MWC). This equivalence is also used in the works for MCD in Chazelle and Lee (1986) and Berg, Cabello and Har-Peled (2006).

An instance of the Maximum Weight Clique (MWC) is given by a list of points $\mathcal{P} := \{p_1, \dots, p_n\}$, with $p_i \in \mathbb{R}^2$; a norm $||\cdot||$, such that $(\mathbb{R}^2, ||\cdot||)$ is a strictly convex normed plane; and a set of weights $\mathcal{W} := \{w_1, \dots, w_n\}$, with $w_i \in \mathbb{R}_{>0}$. Then, we define a set of unit disks $\mathcal{D} := \{D_1, \dots, D_n\}$, with $D_i := \{x \in \mathbb{R}^2 : ||x - p_i|| \leq 1\}$ being a unit disk centered at p_i; we also denote by ∂D_i as the circle correspondent to the i-th unit disk, and that w_i is the weight of D_i. Having said all that, we define an instance of MWC as a tuple $(\mathcal{P}, \mathcal{W}, ||\cdot||)$.

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

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

Therefore, an optimal solution of MWC is defined as $\max_q \sum_{D_k \cap q \neq \emptyset} w_k$.

Let D be a unit disk in a strictly convex normed plane $(\mathbb{R}^2, ||\cdot||)$, and $p, q \in \partial D$. Then we denote by $\widehat{p,q}$ as the counter-clockwise arc of D starting at p and ending at q. This is represented in Figure 6 where the arc $\widehat{p,q}$ is shown with a solid line, and the arc $\widehat{q,p}$ is shown by a dashed line.

Figure 6 – The definition of arcs of a circle.
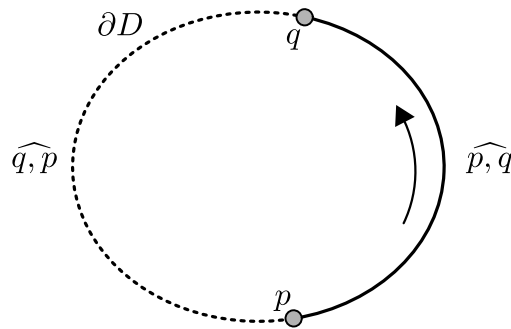
Source: Elaborated by the author.

Let $(\mathcal{P}, \mathcal{W}, \{(a,b)\})$ be an instance of MCE with only one facility, and $(\mathcal{P}, \mathcal{W}, ||\cdot||_{a,b})$ an instance of MWC, with the unit circle defined by $||\cdot||_{a,b}$ being an axis-parallel ellipse with shape parameters (a,b). If q_1 be a solution for the MCE's instance. Then, the disks with centers in $\mathcal{P} \cap E_1(q_1)$ have a non-empty intersection. Also, suppose that q is a solution of MWC. Then, the ellipse with shape parameters (a,b) centered at q covers the points, which are the centers of disks, such that $q \in D_j$. Therefore, both problems are equivalent.

Let us consider the intersection set of k unit disks $\cap_{j=1}^{k} D_j(x_j)$, for any strictly convex normed space, with $x_j \in \mathbb{R}^2$, $j \in \{1, \ldots, k\}$ being all distinct. Then, in Martín and Martini (2015), the following results are stated about that set: its boundary is formed by arcs of unit circles whose centers are in $\{x_1, \ldots, x_k\}$, its vertices are in the set $\partial D_i(x_i) \cap \partial D_j(x_j)$, for any $i \neq j$; and $|\partial D_i(x_i) \cap \partial D_j(x_j)| \leq 2$, for any $i \neq j$. Based on that, we introduce the next definition.

Definition 3.1. Let D_i and D_j be two unit disk in a strictly convex normed space, and $\{\alpha_{ij}^+, \alpha_{ij}^-\} = \partial D_i(p_i) \cap \partial D_j(p_j)$, we denote by $\overset{\frown}{\alpha_{ij}^+, \alpha_{ij}^-}$ the minimal counter-clockwise arc of ∂D_i starting in α_{ij}^+ and ending in α_{ij}^-. We also refer to α_{ij}^+ as an opening intersection point, and to α_{ij}^- as a closing intersection point. If ∂D_i and ∂D_j intersect at exactly one point, we define $\alpha_{ij}^+ = \alpha_{ij}^- = \partial D_i \cap \partial D_j$.

Let $\overset{\frown}{\alpha_{ij}^+, \alpha_{ij}^-}$ and $\overset{\frown}{\alpha_{ij}^-, \alpha_{ij}^+}$ be the two arcs of ∂D_i with respect to the endpoints $\alpha_{ij}^-, \alpha_{ij}^+$. From Martín and Martini (2015, Lemma 2), we can state that, $\overset{\frown}{\alpha_{ij}^+, \alpha_{ij}^-} \subset D_j$, and $\overset{\frown}{\alpha_{ij}^-, \alpha_{ij}^+} \cap D_j = \{\alpha_{ij}^-, \alpha_{ij}^+\}$. That is, only the minimal arc is contained in the interior of D_j. In Figure 7, the minimal arcs are highlighted with a solid border, and it is possible to see that they are the only ones that are in the intersection's interior.

Figure 7 – The intersection between two circles, and the arcs .
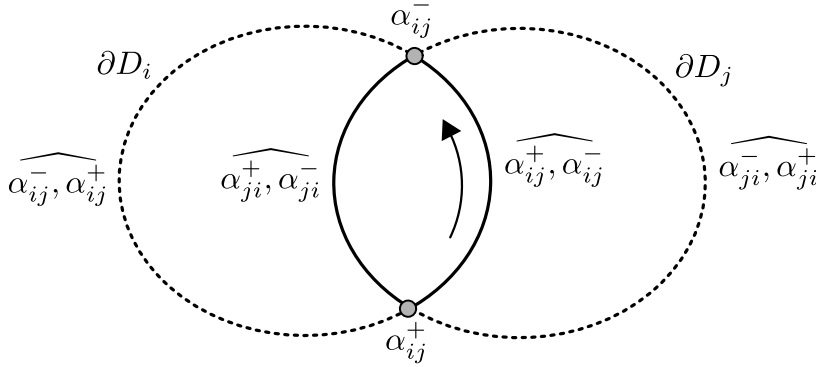
Source: Elaborated by the author.

Based on that, we are going to develop an algorithm that finds the best clique that ∂D_i is part of, for each $i = 1, \ldots, n$, and then combine the solutions to get an overall optimal one. Notice that this is enough because, by Martín and Martini (2015), the arcs of the circles in $\partial \mathcal{D}$ form the boundary of any clique. For each $i \in \{1, \ldots, m\}$, let q_i be an optimal solution of MWC with an additional constraint that $q_i \in \partial D_i$. Then, an optimal solution of MWC can be obtained by just taking a solution of $\max_{i=1}^{n} \max_{q \in \partial D_i} \sum_{D_k \cap q} w_k$.

Lemma 3.1. Let $(\mathcal{P}, \mathcal{W}, ||\cdot||)$ be an instance of MWC. For each $i \in \{1, \ldots, n\}$, let q_i be any solution, such that $q_i \in \partial D_i$; and $J = \{j : q_i \in D_j\}$. If there is $j \in \{1, \ldots, n\}$, $j \neq i$, such that $D_i \cap D_j \neq \emptyset$, then $q_i \in \cap_{j \in J, j \neq i} \{\overset{\frown}{\alpha_{ij}^+, \alpha_{ij}^-}\}$.

Proof. Suppose that $q_i \notin \widehat{\alpha_{ij}^+, \alpha_{ij}^-}$, for some $j \in J \setminus \{i\}$. By definition $q_i \in \partial D_i$, and by Martín and Martini (2015, Lemma 2), we have that $q_i \in \widehat{\alpha_{ij}^-, \alpha_{ij}^+}$, which implies that $q_i \in \{\alpha_{ij}^-, \alpha_{ij}^+\}$, which would imply that $q_i \in \widehat{\alpha_{ij}^+, \alpha_{ij}^-}$, contradicting the assumption we made. $\qquad\square$

Given the set of indexes $J = \{j\colon q_i \in D_j\}$ of disks that contain a solution $q_i \in \partial D_i$, we have that there exists $l, k \in J \setminus \{i\}$, such that $\cap_{j \in J, j \neq i}\{\widehat{\alpha_{ij}^+, \alpha_{ij}^-}\} = \widehat{\alpha_{il}^+, \alpha_{ik}^-}$. Therefore, by Lemma 3.1, we can just look for an index $l \in \{1, \ldots, n\} \setminus \{i\}$, such that it is the solution of

$$\max_{l \neq i} \sum_{D_k \cap \alpha_{il}^+} w_k.$$

Then an optimal solution $q_i \in \partial D_i$ can be obtained by just setting $q_i = \alpha_{il}^+$. From that, a $\mathcal{O}(n^3)$ algorithm for MWC can be developed by just evaluating, for every pair $i, j \in \{1, \ldots, n\}$, the solution defined by α_{ij}^+. Based on the algorithms for MCD proposed by Drezner (1981) and Berg, Cabello and Har-Peled (2006) though, we can develop a $\mathcal{O}(n^2 \lg n)$ algorithm for MWC. Let A_i be a circular defined as

$$A_i = \{\alpha_{ij}^+\colon j \neq i, D_i \cap D_j \neq \emptyset\} \cup \{\alpha_{ij}^-\colon j \neq i, D_i \cap D_j \neq \emptyset\},$$

which contains all the intersection points of ∂D_i with all the other disks. Assume that this list is sorted by the angles on D_i in the interval $[0, 2\pi)$ corresponding to the intersection points, breaking ties prioritizing opening intersection points. 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.

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

$$B_i = A_i \cup \bigcup_{j \neq i} \{2\pi + \alpha_{ij}^+\colon j \neq i, D_i \cap D_j \neq \emptyset\} \cup \{2\pi + \alpha_{ij}^-\colon j \neq i, D_i \cap D_j \neq \emptyset\}. \qquad (3.1)$$

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 $\alpha_{ij}^+ < \alpha_{ij}^- + 2\pi$. That is, the algorithm encounters an opening intersection point before reaching a closing one for any circle.

If it is possible to find the intersection between two unit disks (it will be shown that it is possible for the case where the unit disk is an axis-parallel ellipse), for each disk, the runtime complexity of processing every intersection point and sorting the list B_i is $\mathcal{O}(n \lg n)$. Therefore, the overall runtime complexity of this algorithm for MWC is $\mathcal{O}(n^2 \lg n)$.

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 points. 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. It is possible to see that the optimal clique highlighted in Figure 8 is enclosed by the arcs defined by α_{14}^+ and α_{14}^-, 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 α_{11}^+ and α_{11}^- 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.
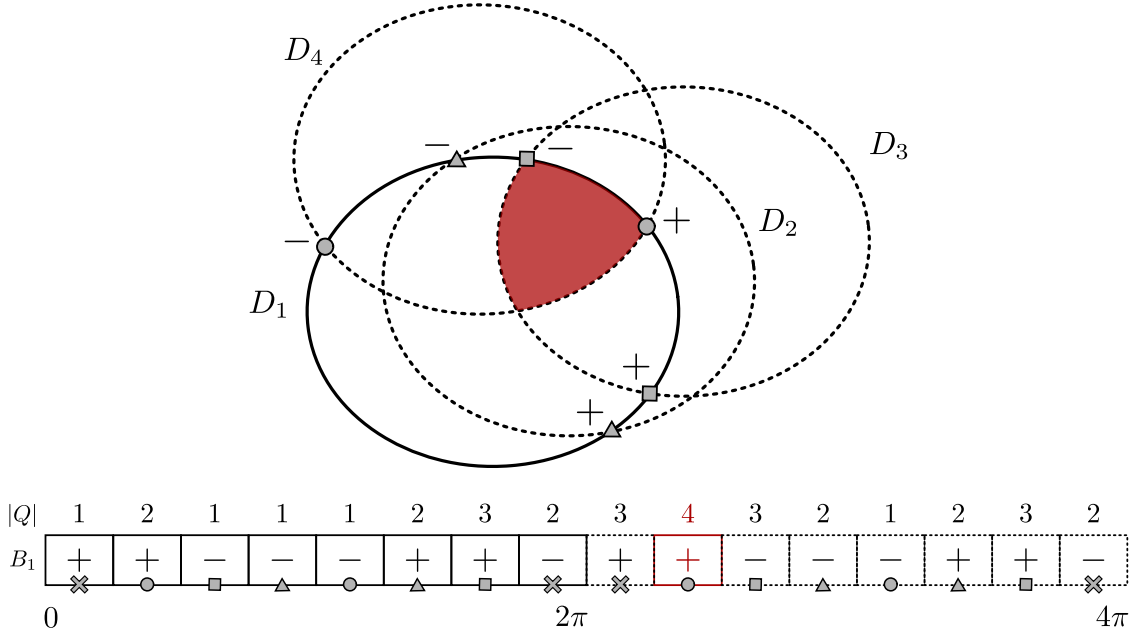
Source: Elaborated by the author.

3.5   From MWC to MCE

Based on the ideas developed for MWC, we are going to propose an algorithm for MCE in this section. Suppose that an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCE is given. For each ellipse with shape parameters (a_j, b_j), we have the equivalent instance $(\mathcal{P}, \mathcal{W}, ||\cdot||_{a_j, b_j})$ of MWC.

Definition 3.2. Let $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ be an instance of MCE. For each $j \in \{1, \ldots, m\}$, let $(\mathcal{P}, \mathcal{W}, ||\cdot||_{a_j, b_j})$ be the equivalent instance of MWC using the j-th ellipse as their unit disk, we define as the Candidate List Set (CLS) S_j for j-th ellipse as

$$S_j = \bigcup_{i=1}^{n} \{\alpha_{ik}^+ : k \neq i, D_i \cap D_k \neq \emptyset\} \cup \{p_i\}.$$

Based on this, we introduce a theorem that allows us to develop an algorithm for MCE based on the developments we made for MWC.

Theorem 1. Let $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ be an instance of MCE, and $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ be a set of solutions defined as

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

Then there exists an optimal solution $Q^* \in \Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$, and $|\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})| \leq n^{2m}$.

Proof. Notice that $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ is defined as the combination of every possible solution from each CLS. To prove that $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ contains an optimal solution, we just need to show that for any optimal solution Q', for every $j \in \{1, \ldots, m\}$, there exists $q_j \in S_j$, such that $\mathcal{P} \cap E_j(q'_j) \subset \mathcal{P} \cap E_j(q_j)$. That is, we only need to show that the CLS of every ellipse contains a center that makes the ellipse cover the same points (possibly some additional ones) that it covers in an optimal solution. We ignore the case where an ellipse does not cover any points.

First case: $|\mathcal{P} \cap E_j(q'_j)| = 1$. This case is included in S_j as it contains the possible solutions where the center of the ellipse is the actual points in $\mathcal{P}$.

Second case: $|\mathcal{P} \cap E_j(q'_j)| > 1$. Let $X = \{i : p_i \in \mathcal{P} \cap E_j(q'_j)\}$. In the equivalent instance of MWC, we have that $\cap_{i \in X} D_i \neq \emptyset$ is a region bounded by arcs of circles with centers in $\mathcal{P} \cap E_j(q'_j)$ with vertices being pairwise intersections of $\partial \mathcal{D}$. By [Lemma 3.1](), we have that at least one of the vertices of $\cap_{i \in X} D_i$ is an opening intersection point.

Lastly, we have that $|S_j| \leq \binom{n}{2} + n = \frac{n(n+1)}{2} \leq n^2$. Therefore, $|\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})| \leq |S_1| \times \cdots \times |S_m| \leq n^{2m}$. $\qquad \square$

3.6 An algorithm for MCE

First, we describe, based on [Theorem 1](), an algorithm to return the CLS for every ellipse. This algorithm runs in $\mathcal{O}(n^2)$. Then we define [Algorithm 2](), which just backtracks the CLS of every ellipse to find an optimal solution. This algorithm can be implemented to have a $\mathcal{O}(mn^{2m+1})$ runtime complexity if the CLS construction is done in a preprocessing phase.

3.7 Determining α_{ij}^+ and α_{ij}^- for axis-parallel ellipses

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

---

Algorithm 1 – Algorithm that returns a CLS for a disk.

Input: A set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$ with weights $\mathcal{W} = \{w_1, \ldots, w_n\}$, and the shape parameters $(a,b) \in \mathbb{R}^2_{>0}$.
Output: A CLS for the axis-parallel ellipse.

 1: **procedure** CLS-MCE($\mathcal{P}, \mathcal{W}, (a,b)$)
 2:     Let $\mathcal{D}$ be a set of unit disks on the strictly convex normed space $(\mathbb{R}^2, ||.||_{a,b})$.
 3:     $S \leftarrow \{\}$
 4:     **for all** $p_i \in \mathcal{P}$ **do**
 5:         $S \leftarrow S \cup \{p_i\}$
 6:         **for all** $j \neq i \colon D_j \cap D_i \neq \emptyset$ **do**
 7:             $S \leftarrow S \cup \{\alpha_{ij}^+\}$
 8:         **end for**
 9:     **end for**
10:     **return** S
11: **end procedure**

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \qquad\qquad (E_1) \qquad\qquad (3.2)$$

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

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

$$b^2 x^2 + a^2 y^2 = b^2(x-h)^2 + a^2(y-k)^2$$

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

$$x = y\alpha + \beta. \qquad\qquad (3.3)$$

Replacing Equation 3.3 into Equation 3.2, we get

$$y^2(b^2\alpha^2 + a^2) + y(2\beta\alpha b^2) + b^2\beta^2 - a^2b^2 = 0, \qquad\qquad (3.4)$$

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 3.4 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 $E_1(q_1)$.

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

$$\frac{a}{b} \frac{p_{iy} - q_{jy}}{p_{ix} - q_{jx}} = \tan\left(t_i^{(j)}\right). \tag{3.5}$$

Determining which one of the points is α_{ij}^+, we just need to use the assumption that $\widehat{\alpha_{ij}^+, \alpha_{ij}^-}$ is minimal. This can be done using the angle of the intersection points obtained from Equation 3.5.

Algorithm 2 – Algorithm for MCE

Input: A set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$, a list of weights $\mathcal{W} = \{w_1, \ldots, w_n\}$, and a list of shape parameters $\mathcal{R} = \{(a_1, b_1), \ldots, (a_m, b_m)\}$.
Output: 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:     $(q_j^*, \ldots, q_m^*) \leftarrow (0, \ldots, 0)$
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:         $(q_{j+1}, \ldots, q_m) \leftarrow MCE_{bt}(Z \setminus Cov, \mathcal{W}, \mathcal{R}, j+1)\}$
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:             $(q_j^*, \ldots, q_m^*) \leftarrow (q_j, \ldots, q_m)$
16:         **end if**
17:     **end for**
18:     **return** $(q_j^*, \ldots, q_m^*)$
19: **end procedure**

DETERMINING EVERY LOCATION OF AN ELLIPSE GIVEN ITS SHAPE AND THREE POINTS

In this chapter, we introduce the problem of determining every location, here defined as the center and angle of rotation, of an ellipse with fixed shape parameters, such that it contains three given points. This problem comes up in the development of an algorithm in Chapter 5 as a subproblem. Because no studies were found on it, or even on related problems, we decide to devote a whole chapter to presenting a handful of approaches we attempted, going through the issues with the failing ones, as well as discussing the qualities of the ones that shown to be successful. In the end, we propose an algorithm for the problem that involves determining the eigenvalues of a 6×6 complex matrix. We also analyze its efficiency in terms of numerical accuracy and display some solutions that it was able to obtain.

4.1 Definition

We call the problem of finding a center and an angle of rotation for an ellipse given its shape parameters and three points that have to be on it Ellipse by Three Points (E3P). An instance of it is given by three points $u, v, w \in \mathbb{R}^2$, along with the ellipse's shape parameters $(a, b) \in \mathbb{R}^2_{>0}$, with $a > b$.

Let $E : \mathbb{R}^2 \times [0, \pi) \to \mathbb{R}^2$ be a function that takes the location of an ellipse with given shape parameters, and returns its coverage region as defined by Equation 2.9. Then a solution of E3P 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 5, a method would only be useful for our case if it can encounter every solution of E3P. This requirement makes the problem more challenging.

4.2   Transforming the problem

Initially, E3P is a problem with three unknown variables: the two coordinates of the ellipse's center point, q_x and q_y, and the angle of rotation θ. In this section, we transform E3P into the problem of finding the roots of a univariate function using a known circumcircle problem. This transformation, besides reducing the number of unknown variables to one, is later utilized in the demonstration that E3P has at most six distinct solutions.

To make the problem simpler, let us assume that point u is at the origin. If it is not, a simple translation by $-u$ applied to the three points can be made to put u at the origin. Assume as well that (q, θ) is a solution of E3P, which means that an ellipse rotated by θ, centered at q contains u, v, and w (see Figure 9 for an example). Taking this solution and applying a rotation of $-\theta$ to the coordinate system makes the ellipse become axis-parallel. After that, we can transform that axis-parallel ellipse into a circle of radius b by squeezing the x-axis by $\frac{b}{a}$. This two-step transformation can be written as a function $\varphi\colon \mathbb{R}^2 \to \mathbb{R}^2$ defined as

$$\varphi(p, \theta) = \left[\begin{array}{cc} \frac{b}{a} & 0 \\ 0 & 1 \end{array} \right] \left[\begin{array}{cc} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{array} \right] \left[\begin{array}{c} p_x \\ p_y \end{array} \right]. \tag{4.1}$$

An example of this two-step transformation being applied to a solution of E3P is shown in Figure 9. Notice that φ is a linear transformation. This means that given a final state, where after applying φ, the three points are on the radius-b circle, a solution of E3P can be obtained by using the well-defined inverse function φ^{-1}.

Additionally, to make the notation more clear, we denote by $\Lambda(\theta)$ the triangle formed by the points $\varphi(u, \theta); \varphi(v, \theta); \varphi(w, \theta)$, as long as they are not collinear. This being said, we can move forward and introduce a problem equivalent to E3P which is based on determining angles θ that make the triangle $\Lambda(\theta)$ be circumscribed in a circle of radius b.

4.2.1   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 contains the triangle's vertices. Given an instance of E3P, we define the circumradius problem as the problem of determining an angle $\theta \in [0, \pi)$, such that the circumradius of $\Lambda(\theta)$ is equal to b.

As it can be seen on Figure 9, given an instance of E3P and an angle of rotation $\theta \in [0, \pi)$ that makes $\Lambda(\theta)$ have a circumradius b, a solution for E3P can be obtained using the inverse transformation φ^{-1}. With that in mind, it is possible to conclude that both problems are equivalent because, from a solution of one, a 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 a radius b given a triangle is a convenient problem.

Figure 9 – Transforming a solution of E3P into a solution of the circumradius problem.
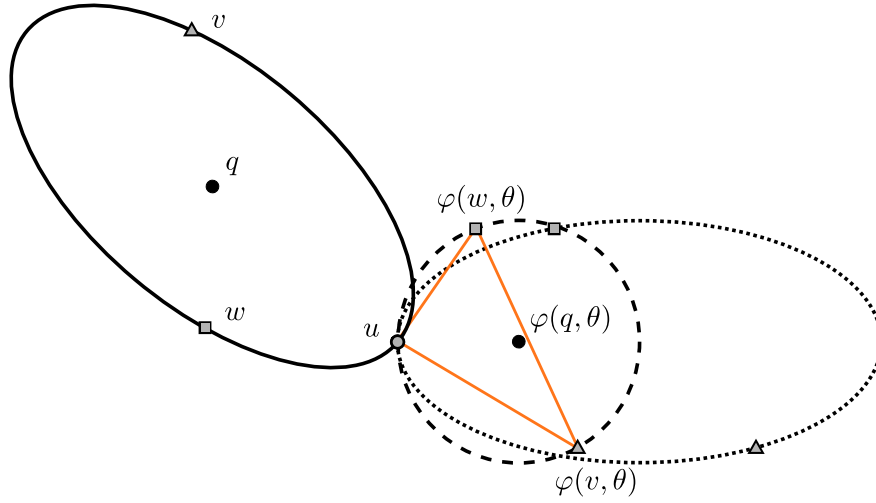
Source: Elaborated by the author.

It turns out that, for any triangle, there is always a unique circumscribed circle, which can be determined analytically. Given an instance of E3P and an angle of rotation $\theta \in [0, 2\pi]$, the circumradius R of $\Lambda(\theta)$ can be computed through the following expression

$$R = \frac{\|\varphi(v,\theta)\|_2 \, \|\varphi(w,\theta)\|_2 \, \|\varphi(v,\theta) - \varphi(w,\theta)\|_2}{4A(\theta)}, \tag{4.2}$$

with $A(\theta)$ being the area of $\Lambda(\theta)$ (for more details about Equation 4.2, or on how to determine the center of a circumscribed circle, see Johnson and Young (1960, p. 189)). It should be pointed out that this transformation does not preserve distance or area; if that was true, the radius defined by Equation 4.2 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, and consequently, solutions for E3P. Imposing the radius R to be equal b and squaring to eliminate the square roots present in the Euclidean distance, a function $\xi : [0, 2\pi) \mapsto \mathbb{R}_{>0}$ is defined as

$$\xi(\theta) = 16b^2 A(\theta)^2 - \|\varphi(v,\theta)\|_2^2 \, \|\varphi(w,\theta)\|_2^2 \, \|\varphi(v,\theta) - \varphi(w,\theta)\|_2^2. \tag{4.3}$$

Any root of ξ produces a triangle whose circumradius is b and subsequently provides a solution for E3P.

Before attempting to develop an algorithm to find every root of ξ, we address the question about the number of roots of ξ in the interval $[0, \pi)$.

4.2.2 The number of solutions of E3P

One of the steps of the method developed in Chapter 5 is to iterate over every solution of E3P. Of course, doing that is only possible if E3P has a finite number of solutions. Moreover,

even if the number of solutions is finite, discovering an upper-bound for that is essential for determining the algorithm's efficiency.

Lemma 4.1. Any instance of E3P has at most 6 solutions.

Proof. Back on Chapter 2, real trigonometric polynomials were introduced. It was stated that any 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 2.13. 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 2.13:

$$\|\varphi(v,\theta)\|_2^2 = (v_x \frac{b}{a}\cos\theta + v_y \frac{b}{a}\sin\theta)^2 + (v_y\cos\theta - v_x\sin\theta)^2 \tag{4.4}$$

$$A(\theta)^2 = \frac{1}{4}\det \begin{pmatrix} v_x\frac{b}{a}\cos\theta + v_y\frac{b}{a}\sin\theta & v_y\cos\theta - v_x\sin\theta \\ w_x\frac{b}{a}\cos\theta + w_y\frac{b}{a}\sin\theta & w_y\cos\theta - w_x\sin\theta \end{pmatrix}^2. \tag{4.5}$$

It is also possible to see that the term which has ξ's highest degree is the multiplication of the three squared lengths of $\Lambda(\theta)$'s sides. 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 ξ's degree. 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 $[\pi,2\pi)$. Because of that, half of the roots of ξ are in $[\pi,2\pi)$ and can be dismissed. $\square$

4.3   An attempt using the conic general equation

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

$$Ax^2 + Bxy + Cy^2 + Dy + Ex + F = 0, \tag{4.6}$$

with $A,B,C,D,E,F \in \mathbb{R}$ being fixed parameters. This equation actually represents any conic, for it to be an ellipse the condition $B^2 - 4AC < 0$ must be satisfied.

Given an instance of E3P, assuming u is at the origin, having that it satisfies Equation 4.6, 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 4.6 represents the same conic, we can set B to be equal to 1. Then, we end up with two variables, A and C, and still need to impose that the final equation represents an ellipse with the given shape parameters. Let $\Delta = 4AC - B^2 = 4AC - 1$, and assume $F = 0$, then

the expressions for both major-axis and minor-axis, respectively are

$$a^2 = \frac{2\dfrac{AE^2 - BDE + CD^2}{\Delta}}{A + C - \sqrt{1 + (A - C)^2}} \tag{4.7}$$

$$b^2 = \frac{2\dfrac{AE^2 - BDE + CD^2}{\Delta}}{A + C + \sqrt{1 + (A - C)^2}}. \tag{4.8}$$

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. This set can probably 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. At first, this function appeared to be monotonic, so 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.

4.4 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, like in our case, when the roots of a function need to be determined. In general, one has a function f that is part of a family of functions $\mathscr{A}$ and wants to select a simpler function f^* from a set of functions $\mathscr{A}^*$, such that f^* is close enough to f (POWELL, 1981, p. 3). For this problem, we consider the approximation of ξ on the interval $[0, \pi)$ by a function in the family of n-degree Chebyshev polynomials.

4.4.1 Chebyshev polynomial

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

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

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

$$T_n(x) = \cos(n \arccos(x)). \tag{4.9}$$

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

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x). \tag{4.10}$$

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

$$p_n(x) = \sum_{j=0}^{n} a_j T_j(x). \tag{4.11}$$

One of the greatest qualities of Chebyshev polynomials is their numerical stability. Gautschi (1979) showed that the matrix that maps polynomials onto its coefficients written in the power form has a condition number that grows exponentially with n. On the other hand, the matrix that converts polynomials to the Chebyshev basis as Equation 4.11 has a linear condition number bounded by $\sqrt{2}n$.

4.4.2 Chebyshev interpolation

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

$$f_n(x) = \sum_{j=0}^{n} f(x_j) \frac{\prod_{k \neq j}^{n+1}(x - x_k)}{\prod_{k \neq j}^{n+1}(x_j - x_k)}, \tag{4.12}$$

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

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

$$x_j = \cos\left(\frac{\pi(j - \frac{1}{2})}{n+1}\right), \tag{4.13}$$

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

$$\hat{x}_j = \frac{a+b}{2} + \frac{b-a}{2}x_j \tag{4.14}$$

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

Then, the Chebyshev interpolation of a function $f : [a,b] \mapsto \mathbb{R}$ can be determined using Lagrange's formula and the points $\hat{x}_1, \ldots, \hat{x}_n$. As it was mentioned in Chapter 2, finding the 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 values of n this works fine, however, converting the polynomial obtained by Equation 4.12 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 4.11, 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 are the roots of the Chebyshev interpolant f_n 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 satisfies 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 in a closed interval, its absolute difference to its respective 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 considered to be a good estimate for the error

$$r_n = \max_{0 \leq \theta < \pi} |f_n(\theta) - \xi(\theta)|, \tag{4.15}$$

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

Figure 10 – The maximum interpolation error.
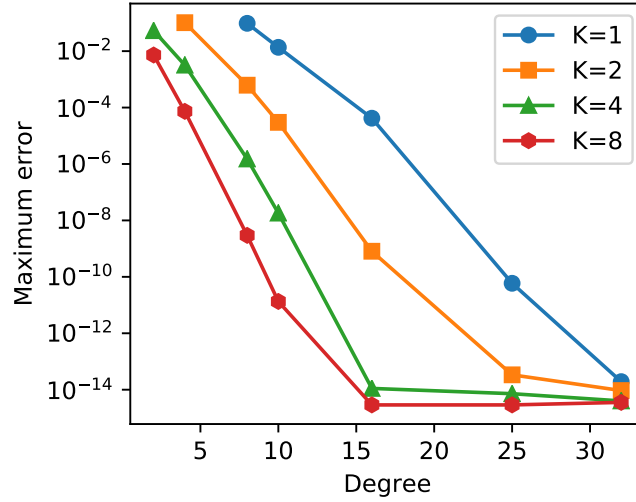
Source: Elaborated by the author.

4.4.3   *Testing different interpolant degrees*

In this section, we describe the results of an experiment we made to verify the accuracy of solutions found by the Chebyshev Interpolation method for different interpolation degrees. The main reason for doing this experiment was to obtain a practical lower-bound for the interpolation degree, which can be used later to decide whether to use this method or not. We also investigate if dividing the interpolation interval into K sub-intervals, which is a suggestion given in Boyd (2013), yields an improvement in the accuracy of solutions.

We used the Python programming language in the implementation of this approach for E3P. More specifically, we utilized the external library SciPy, which has routines already implemented for Chebyshev interpolation, and finding the roots of a Chebyshev polynomial. More information about SciPy can be found in (Virtanen *et al.*, 2020).

Let $\delta : \mathbb{R}^2 \to \mathbb{R}_{>0}$ be a function defined as the left-hand-side of Equation 2.8, then, for an instance of E3P with three points u, v, w, we define the error of a solution as

$$\max\{|\delta(u)|, |\delta(v)|, |\delta(w)|\}.$$

We created instances of E3P taking every triplet of points, and every ellipse from an MCE's instance named CM3 proposed in Canbolat and Massow (2009). We also tried dividing the interval $[0, \pi]$ into a different number of sub-intervals taking $K \in \{1, 2, 4, 8\}$.

In Figure 10, for each K, the maximum error observed among every instance of E3P for each interpolation degree is shown. It may be stated that adopting the strategy of dividing the interpolation interval into K sub-intervals provides a significant improvement in accuracy.

Also, as expected, in Figure 11, using the same instances, we were able to observe that

Figure 11 – The maximum absolute value of the last coefficient interpolation.
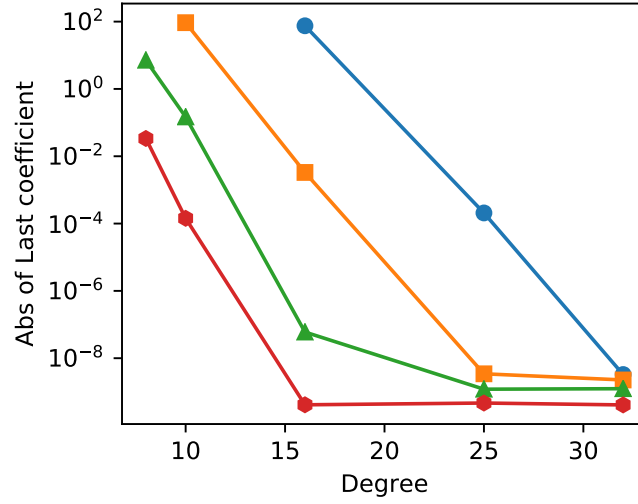
Source: Elaborated by the author.

the maximum absolute value of the last coefficient among every instance has the same behavior as its corresponding error in Figure 10.

Assuming that $K = 4$, we can say that for a small error to be achieved, we need to take an interpolation degree of at least 10, increasing it based on the last coefficient rule. A suggestion in Boyd (2013) says that if the last coefficient is not small, the interpolation degree must be doubled. This is only a suggestion of how to approach the problem of choosing a good interpolation degree. This procedure could still fail as a small last coefficient does not necessarily imply a small error everywhere in the interpolation interval.

4.5 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. The first one converts ξ into a real polynomial and the second one into a complex polynomial. For these two approaches we put symbolic computation into practice to obtain the coefficients of the polynomials in terms of the E3P's instance.

4.5.1 Real polynomial

From ξ, a real polynomial can be obtained by using the identity $x = \tan\left(\frac{\theta}{2}\right)$. We do not go in detail, but it is possible to show that a degree-12 polynomial can be obtained using that substitution.

At first, the root-finding algorithm described on Chapter 2 seemed to work fine and return

every solution of E3P. 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.

4.5.2 Complex polynomial

A complex polynomial can be obtained from ξ by using an idea published in Boyd (2006). There, the author uses the identities

$$\cos(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{2} \tag{4.16}$$

$$\sin(\theta) = \frac{e^{i\theta} - e^{-i\theta}}{2i}, \tag{4.17}$$

which relate complex numbers with trigonometric functions, 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 stated that computing the roots of a real trigonometric polynomial through this transformation does not yield loss of accuracy.

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

$$g(z) = \sum_{k=0}^{12} c_k z^{k-6}, \tag{4.18}$$

for some $c_0, \ldots, c_{12} \in \mathbb{C}$. As the equalities on Equation 4.16 and Equation 4.17 are valid for any $\theta \in \mathbb{R}$, function $g(e^{i\theta})$ and ξ are equivalent, since $g(e^{i\theta}) = \xi(\theta)$ for any $\theta \in [0, 2\pi]$. Notice that g is not a complex polynomial it has negative exponents and its domain is not $\mathbb{C}$.

We can get rid of negative exponents by multiplying g by z^6. 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 in the new polynomial, but they can be ignored as they are not roots of g. Finally, from g, the polynomial $h : \mathbb{C} \mapsto \mathbb{C}$ is defined as

$$h(z) = z^6 g(z) = \sum_{k=0}^{12} c_k z^k. \tag{4.19}$$

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.

4.5.2.1 Further improvements

It is possible to make another reduction and cut the size of the polynomial in half. As it has been mentioned in Chapter 2, an ellipse is symmetric with respect to its axis, which implies that rotating it by $\theta \in [0, \pi)$ is equivalent to rotating it by $\pi + \theta$. On the other hand, very conveniently, as given by Equation 2.10, angles of complex numbers of opposite signs are π apart from each other, which means that g has to produce the same output for both z and $-z$ as they represent equivalent angles of rotation for ellipses. From that, for all $z \in \mathbb{S}$ we have

$$h(-z) = (-z)^6 g(-z) = z^6 g(z) = h(z).$$

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

$$f(y) = \sum_{k=0}^{6} c_{2k} y^k. \tag{4.20}$$

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

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

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

$$c_k = \overline{c_{6-k}}, \tag{4.21}$$

for $k = 0, \ldots, 6$. For now, we neither have any ideas on how Equation 4.21 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 E3P.

4.6   An algorithm for E3P

Among the methods that have been described here, converting ξ into a complex polynomial, and then obtaining its roots by determining the eigenvalues of a companion matrix was the chosen one as the basis of Algorithm 3 for E3P. Despite the good results shown by the

Chebyshev interpolation method, it can still be classified as a heuristic as none of the approaches to determine the interpolation degree ensures a good approximation in the whole interval. On top of that, ultimately, the roots of the Chebyshev polynomial are computed through determining the eigenvalues of a companion matrix, which, unless a lower-than-seven interpolation degree is utilized, is going to be larger than the companion matrix whose eigenvalues are the roots of the complex polynomial h.

Details about getting the eigenvalues of a companion matrix, and determining the center of a circumscribed circle of a triangle are omitted from Algorithm 3 for the sake of clarity. In our implementation, we use symbolic computation to determine the coefficients of h in terms of the parameters of a E3P's instance. This way, we only need to compute once separately from the main algorithm. We get into more detail about that in Chapter 6.

Having all the coefficients of h available, basically, after building a companion matrix, Algorithm 3 applies the reverse transformations described by Equation 4.1 to every eigenvalue of that companion matrix to obtain a solution for E3P.

Algorithm 3 – 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** $e3p(u, v, w, a, b)$
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 $c_0, \ldots, c_{12}$ be the coefficients of polynomial h as Equation 4.19.
6:     Let A be a 6×6 zero matrix.
7:     **for** $i \in \{1, \ldots, 6\}$ **do**   ▷ Constructing the companion matrix.
8:         $A_{i,i+1} \leftarrow 1$
9:         $A_{6,i} \leftarrow -\dfrac{c_{2(i-1)}}{c_{12}}$
10:     **end for**
11:     $Q \leftarrow \{\}$
12:     **for all** $q \in eig(A)$ **do**   ▷ $eig(A)$ returns every eigenvalue of A.
13:         $\theta \leftarrow \min\{angle(-q)/2, angle(q)/2\}$
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. Algorithm 3 computes every solution for an instance of E3P in $\mathcal{O}(1)$ operations.

Proof. It has already been shown in Section 4.2 that computing every root of ξ through the

complex polynomial yields every solution of E3P. 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 $\mathcal{O}(1)$. □

4.6.1 Instances with six and four solutions

Any instance of E3P, as stated by Lemma 4.1 can have up to six solutions. At first, though, this bound seemed to be loose as for randomly generated instances like the ones generated by the model in the next section, only two solutions were returned by Algorithm 3. After some investigation, we were able to construct some four-solution instances (an example is displayed in Figure 12). An interesting property of those solutions is that every one of them has their three points form an isosceles triangle.

Figure 12 – An instance of E3P with four solutions.
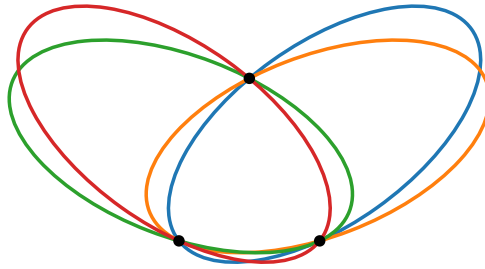
Source: Elaborated by the author.

Obtaining six-solution instances, on the other hand, was done by taking a particular case of the isosceles-triangle approach. As it can be seen in Figure 13, the three points on every one of the six ellipses' border form an equilateral triangle.

It should be pointed out that neither non-isosceles instances with four solutions nor non-equilateral instances with six solutions could be found. Further investigating these possible properties of E3P is left as future work.

4.6.2 Numerical Stability

In this section we show the results of some experiments made to study the numerical stability of Algorithm 3. For all the experiments, we define $K \in \mathbb{R}_{>0}$, and consider instances with ellipse's shape parameters $(K, \frac{K}{2})$, for $K \in \{10^j : j = 0, \ldots, 10\}$. Let $\delta \colon \mathbb{R}^2 \to \mathbb{R}$ be a function defined as the left-hand-size of Equation 2.8, then, for an instance with three points $u, v, w \in \mathbb{R}^2$, we define the error associated with a solution for that instance as $\max\{|\delta(u)|, |\delta(v)|, |\delta(w)|\}$.

The first experiment considers instances where the three points are the vertices of an ellipse rotated by $\theta \in [0, \pi)$. It is possible to see that such instances only have one solution,

Figure 13 – An instance of E3P with six solutions.
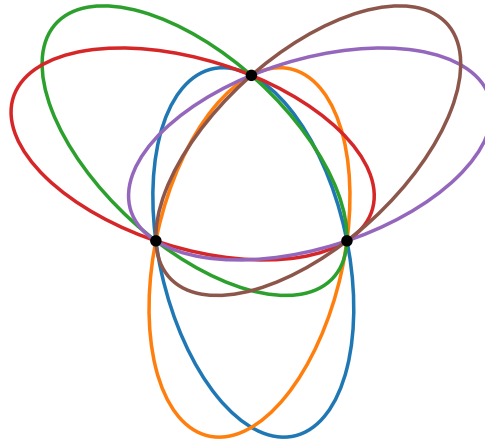
Source: Elaborated by the author.

and therefore, roots with multiplicity greater than one are expected, which can be seen as a special case. For each value of K, we ran the algorithm for 100 instances generated randomly by sampling θ according to a uniform distribution. For each instance, we took the closest solution to the priorly known one, and then, for each K, as it can be seen in Figure 14, we considered the maximum and the average error.

Figure 14 – The maximum and average error for instances with known solutions.
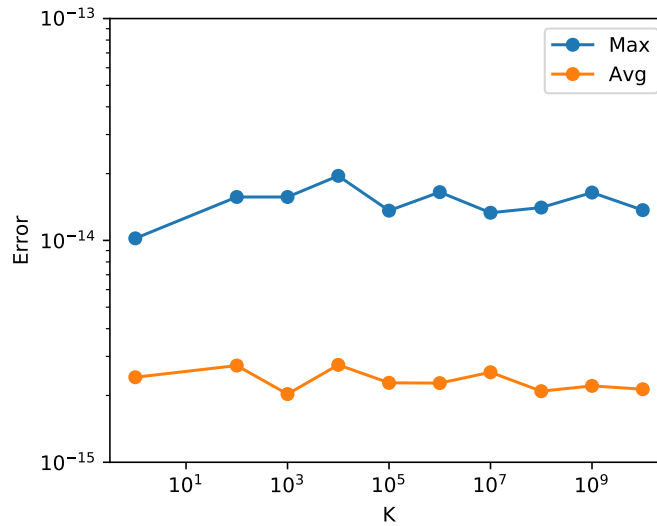
Source: Elaborated by the author.

The second experiment takes the same instances as the previous one, but this time, we analyze how close the roots corresponding to the priorly known solutions are to the unit circle. The distance of a root $\hat{x}$ of h to the unit circle is taken to be $|\hat{x} - 1|$. This experiment is utilized mostly to determine a good precision constant for floating point comparisons in the

Figure 15 – The distance of the roots to the unit circle.
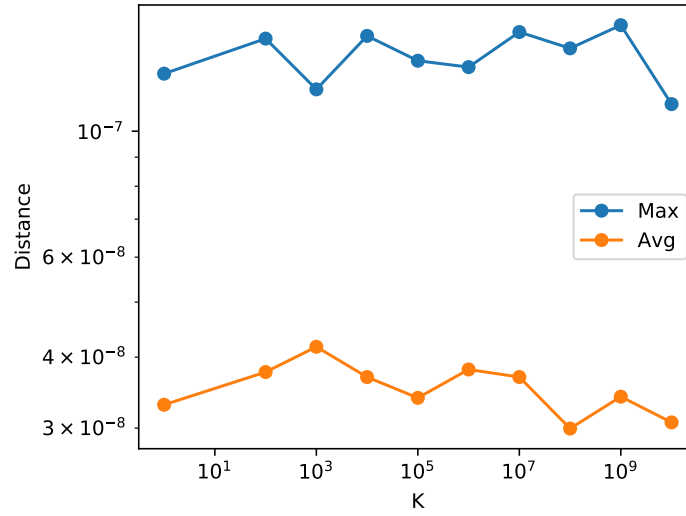
Source: Elaborated by the author.

implementation. The results are shown in Figure 15. As it can be seen that the distance always stays under 10^{-6} we concluded that a good precision constant would be 10^{-5}.

For the last experiment we considered 100 instances for each K with three points $(K\cos(t_j), \frac{K\sin(t_j)}{2})$, $j = 1\dots 3$, generated randomly by sampling t_j according to a uniform distribution in $[0, 2\pi]$. The average and the maximum error are plotted in Figure 16, and analyzing it, it is fair to say that Algorithm 3 is numerically stable for this example, and its error, in average, is expected to be small even if the instance's numerical values are big.

Figure 16 – The error measured on solutions found by Algorithm 3.
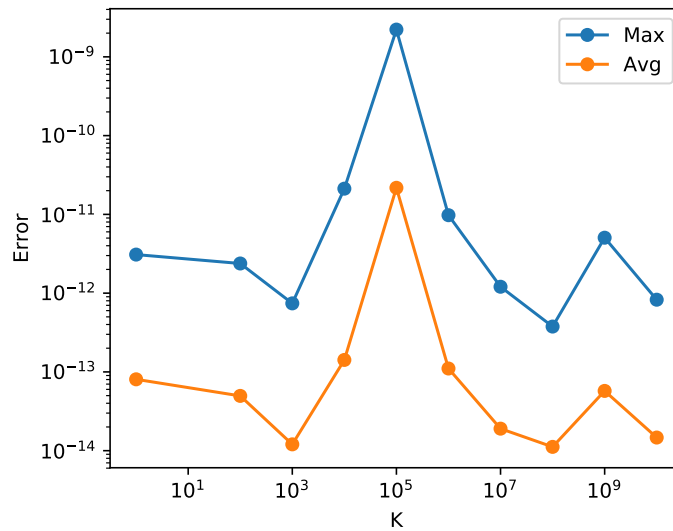
Source: Elaborated by the author.

MAXIMUM COVERING BY ELLIPSES WITH ROTATION PROBLEM

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 Maximum 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.

5.1 Definition

An instance of the non-axis-parallel is defined exactly like the axis-parallel one on **??**. It is given by a set of demand points $\mathcal{P} = \{p_1, \ldots, p_n\}$, $p_j \in \mathbb{R}^2$; a list of weights $\mathcal{W} := \{w_1, \ldots, w_n\}$, with $w_j \in \mathbb{R}_{\geq 0}$ being the weight of point p_j; and m ellipses given by their shape parameters $\mathcal{R} := \{(a_1, b_1), \ldots, (a_m, b_m)\}$, with $(a_j, b_j) \in \mathbb{R}^2_{>0}$ and $a_j > b_j$. Additionally, to make the text more clear, we define a set of m functions that represent the coverage regions of each ellipse as $\mathcal{E} = \{E_1, \ldots, E_m\}$, with $E_j \colon \mathbb{R}^2 \times \mathbb{R} \mapsto \mathbb{R}^2$ being a function that takes the center and angle of rotation where the j-th ellipse is located as input, and returns its coverage region as defined by Equation 2.9. Lastly, an instance of MCER is defined as the tuple $(\mathcal{P}, \mathcal{W}, \mathcal{R})$.

Given an instance of *MCER*, we define $Q := (q_1, \ldots, q_m) \in \mathbb{R}^{2m}$ as the centers of each ellipse, and $\Theta := (\theta_1, \ldots, \theta_m) \in [0, \pi)^m$ as the angles of rotation of each ellipse. Then, we define MCER as the problem of determining Q and Θ (placing and rotating each ellipse) to maximize the weight of the points covered by the m ellipses given by

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

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. That is, 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 finite set that contains at least one optimal solution for MCER.

5.2   An optimal and finite set of solutions

In this section, we construct a finite list of centers and angles of rotation, also referred to as a Candidate Locations Set (CLS), for each ellipse and show that at least one optimal solution is in the set of solutions created from those lists. The results presented in this section are strongly based on Chapter 4, more specifically on Lemma 4.1, which states that there exists at most six solutions for any instance of E3P.

We start by introducing a lemma, which says that given an optimal solution of MCER, it is always possible to find an equivalent one, such that every ellipse covering more than one point contains two of them.

Lemma 5.1. Let (Q^*, Θ^*) be an optimal solution of an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER. Then, for any $j \in \{1, \ldots, m\}$ with $|\mathcal{P} \cap E_j(q^*_j, \theta^*_j)| \geq 2$, an equivalent solution (Q', Θ^*) exists, such that $|\mathcal{P} \cap \partial E_j(q'_j, \theta^*_j)| \geq 2$.

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

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

As it was shown on **??**, 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 $|\mathcal{P} \cap \partial E_j(q'_j, \theta_j)| \geq 2$.

$\square$

What Lemma 5.1 states is that transforming any optimal solution of MCER into an equivalent optimal solution where every ellipse that covers more than one point contains two points is possible (an example can be seen in Figure 17). Lemma 5.1 also states that this equivalent optimal solution can always be achieved by just translating the ellipses; that is, no change in the angle of rotation is required.

Next, we define, for an optimal solution, a set of equivalent solutions, such that any ellipse covering more than one point contains at least two points.

Figure 17 – An optimal solution before and after applying [Lemma 5.1](#).
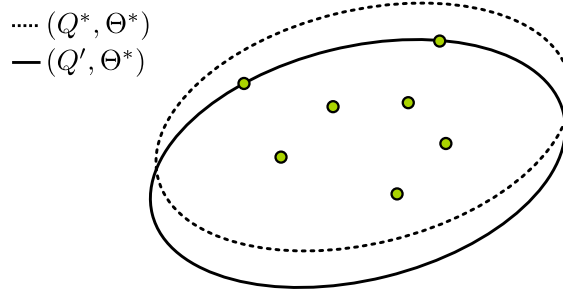
Source: Elaborated by the author.

Definition 5.1. Let (Q^*, Θ^*) be an optimal solution for an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER. We define $\Pi(Q^*, \Theta^*)$ as the set of every equivalent solution of (Q^*, Θ^*), such that for any $(Q, \Theta) \in \Pi(Q^*, \Theta^*)$, for $j \in \{1, \ldots, m\}$ with $|\mathcal{P} \cap E_j(q_j^*, \theta_j^*)| \geq 2$, we have $|\mathcal{P} \cap \partial E_j(q_j', \theta_j)| \geq 2$.

Next, we introduce a notation that helps us characterize angles which given an ellipse rotated by it and two points, it is possible to find a center for the ellipse, such that it contains both points.

Definition 5.2. Let E be the coverage region of an ellipse and $u, v \in \mathbb{R}^2$. An angle $\theta \in [0, \pi)$ is said to be (E, u, v)-feasible if there is $q \in \mathbb{R}^2$ such that $\{u, v\} \subset \partial E(q, \theta)$. In addition to that, given an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER, the set of (E_j, u, v)-feasible angles is referred to as

$$\Phi_j(u, v) := \{\theta \in [0, \pi) : \theta \text{ is a } (E_j, u, v)\text{-feasible angle}\}. \tag{5.2}$$

We also define $\tilde{\Phi}_j(u, v)$ as the angle which makes E_j's major-axis be parallel to the line that passes through u and v. Note that if $\Phi_j(u, v) \neq \emptyset$, then $\tilde{\Phi}_j(u, v) \in \Phi_j(u, v)$ as the longest segment that crosses an ellipse is its major-axis.

In [Figure 18](#) two examples for [Definition 5.2](#) are shown. The example with a solid border shows two given points on two different ellipses rotated by $\pi/4$, making $\pi/4$ a (E, u, v)-feasible angle. The other example, with a dashed border, presents a case where the two points cannot be on the ellipse rotated by $\pi/2$, no matter where it is placed; because of that, $\pi/2$ is said to be a non (E, u, v)-feasible angle.

Figure 18 – A (E, u, v)-feasible angle and a not (E, u, v)-feasible angle.
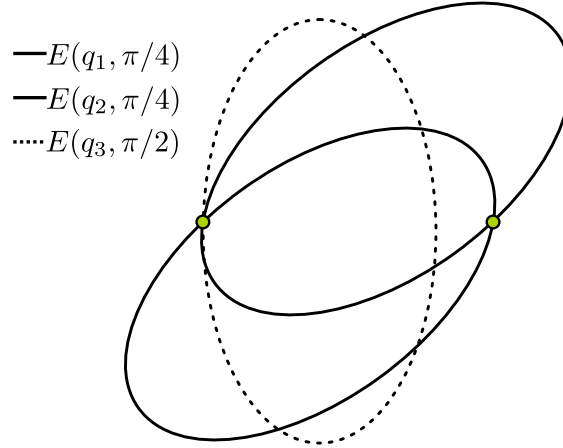
Source: Elaborated by the author.

Following that, we introduce a lemma that is responsible for connecting the developments of this chapter with the results of Chapter 4. This lemma makes it possible to describe a type of solution which, for sure, is part of the equivalence class of any optimal solution. It states that, for any ellipse that covers more than two points in a given optimal solution, an equivalent solution exists with at least one of the two properties:

- The ellipse contains at least three points.

- The ellipse contains two points for any feasible angle.

Lemma 5.2. Let (Q^*, Θ^*) be an optimal solution of an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER; $j \in \{1, \ldots, m\}$, such that $|\mathcal{P} \cap E_j(q_j^*, \theta_j^*)| \geq 2$; $(Q', \Theta') \in \Pi(Q^*, \Theta^*)$; and $\{u, v\} \subset \partial E_j(q_j', \theta_j')$. If, for all $(\hat{Q}, \hat{\theta})$ equivalent solution of (Q^*, Θ^*), $|\mathcal{P} \cap \partial E_j(\hat{q}_j, \hat{\theta}_j)| < 3$, 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 Lemma 5.1, there exists $\{u, v\} \subset \mathcal{P} \cap E_j(q_j^*, \theta_j^*)$, such that an equivalent optimal solution (Q', Θ') exists with u and v on the border of $E_j(q_j', \theta_j^*)$. Therefore, $\theta_j^* \in \Phi_j(u, v)$.

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

In Chapter 2, a function $L \colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ was defined in Equation 2.7. 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 consider the problem of rotating and translating the points in $\mathcal{P}$ instead.

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 on 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 hand, 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'), $\delta(\theta'_j) = q'_j$. Notice that, 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 18 for an example), however, imposing $\delta(\theta'_j) = q'_j$ makes δ be a well-defined function. This is shown in Figure 19 where δ is plotted for the whole interval $\Phi_j(u, v)$.

Let $w \in \mathcal{P} \setminus \{u, v\}$, then we define $f_w : \mathbb{R}^2 \times [0, \pi) \mapsto \mathbb{R}_{\geq 0}$ to be a function that takes a center $q \in \mathbb{R}^2$ and an angle of rotation $\theta \in [0, \pi)$, and returns the elliptical distance between w and $E_j(q, \theta)$ minus 1 as defined by the left-hand-side of Equation 2.8. Keep in mind that, for all $w \in \mathcal{P} \cap E_j(q_j^*, \theta_j^*) \setminus \{u, v\}$, we have that $f_w(q'_j, \theta'_j) < 0$ as they are covered by the j-th ellipse.

Then, to evaluate f_w for every center and feasible angle that maintains u and v on E_j's border, we introduce a new function $g_w : \Phi_j(u, v) \mapsto \mathbb{R}_{\geq 0}$, which is defined as $g_w(\theta) = f_w(\delta(\theta), \theta)$. As f_w and δ are both continuous functions, g_w 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 $g_w(\theta) > 0$. Then, by continuity, another angle $\bar{\theta} \in \Phi_j(u, v)$ must exist, such that $g_w(\bar{\theta}) = 1$, which means that $w \in \partial E_j(\delta(\hat{\theta}), \hat{\theta})$, contradicting the hypothesis. The same can be said about the case where there exists an angle $\theta \in \Phi_j(u, v)$, such that a point $w \in \mathcal{P} \setminus E_j(q_j^*, \theta_j^*)$ enters the coverage of $E_j(\delta(\theta), \theta)$. $\qquad \square$

What Lemma 5.2 states is that, for every ellipse in an instance of MCER, unless an equivalent optimal solution with three points on it exists, the angle of rotation can practically be ignored. Because of that, it will be shown that we can construct a CLS for each ellipse which is finite and also contains an optimal solution.

In Figure 19, a visualization of Lemma 5.2 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 it. Ten angles were chosen from $\Phi_j(u,v)$ to be shown in Figure 19, among those were 0 and $\max\{\Phi_j(u,v)\}$; their corresponding ellipses are displayed with solid-line borders. Consistently with Lemma 5.2, 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.

Figure 19 – A visualization of Lemma 5.2.
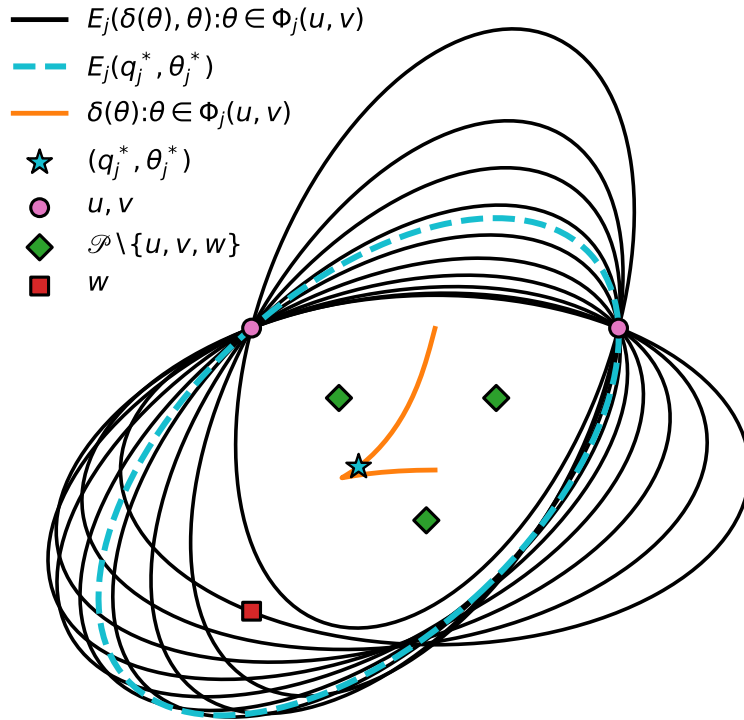
Source: Elaborated by the author.

Let (Q^*, Θ^*) be any optimal solution of an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER. We will define a set of solutions $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ 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 $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ for sure contains an optimal solution for the instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$. Before that, we introduce a definition for the CLS of every ellipse, which is then used to construct the set of solutions $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$.

Definition 5.3. Let $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ be an instance of MCER. Then, for all $j \in \{1, \ldots, m\}$, we define the Candidate Locations Set (CLS) of the j-th ellipse as $S_j = S_j^{(1)} \cup S_j^{(2)} \cup S_j^{(3)}$ with

$$S_j^{(1)} = \bigcup_{u \in \mathcal{P}} \{(u, 0)\} \tag{5.3}$$

$$S_j^{(2)} = \bigcup_{\{u,v\} \subset \mathcal{P}} \{(q, \tilde{\Phi}_j(u,v)) \in \mathbb{R}^2 \times \mathbb{R} : \{u,v\} \subset \partial E_j(q, \tilde{\Phi}_j(u,v))\} \tag{5.4}$$

$$S_j^{(3)} = \bigcup_{\{u,v,w\} \subset \mathcal{P}} \{(q, \theta) \in \mathbb{R}^2 \times \mathbb{R} : \{u,v,w\} \subset \partial E_j(q, \theta)\}. \tag{5.5}$$

This definition breaks the construction of the CLS S_j into three separated cases. The first one, $S_j^{(1)}$, represents solutions where the j-th ellipse covers only one point. The second one, $S_j^{(2)}$, takes into account solutions where the j-th ellipse covers at least two points, and no equivalent solution with three points on the ellipse exists. The last case, $S_j^{(3)}$, considers solutions where there exists an equivalent one with three points on the j-th ellipse. Following this, we move forward and introduce the main result of this section.

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 as

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

Then there exists an optimal solution $(Q^*, \Theta^*) \in \Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$, and $|\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})| \leq n^{3m}$.

Proof. The first thing to notice is that $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ is defined as the combination of every possible solution from each CLS. To prove that it contains an optimal solution (Q^*, Θ^*), we only need to prove that for all $j \in \{1, \ldots, m\}$, there exists $(q_j, \theta_j) \in S_j$, such that $\mathcal{P} \cap E_j(q_j^*, \theta_j^*) \subset \mathcal{P} \cap E_j(q_j, \theta_j)$. That is, we only need to show that the CLS of every ellipse contains a center and angle of rotation that makes the ellipse cover the same points (possibly some additional ones) that it covers in an optimal solution. To do that, we use Lemma 5.2 and break the possible optimal solutions into three cases.

In the first case, we consider solutions where the j-th ellipse covers less than one point, that is, $|\mathcal{P} \cap E_j(q_j^*, \theta_j^*)| \leq 1$. It is possible to see that $S_j^{(1)}$ takes this possibility into account as it includes in $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$ every solution that has an ellipse centered at a point from $\mathcal{P}$. From that, we can also conclude that $|S_j^{(1)}| \leq n$.

In the second case, we consider solutions where the j-th ellipse covers at least two points, and no equivalent solution exists, such that three points are on it. This case is addressed by Lemma 5.2, which says that an optimal solution (Q^*, Θ^*) of this type has equivalent solutions with two points $u, v \in \mathcal{P} \cap E_j(q_j^*, \theta_j^*)$ on the ellipse, for any angle of rotation $\theta_j \in \Phi_j(u, v)$. As $\tilde{\Phi}_j(u, v) \in \Phi_j(u, v)$, we have that there exists $(q_j, \theta_j) \in S_j^{(2)}$, such that $\mathcal{P} \cap E_j(q_j^*, \theta_j^*) = \mathcal{P} \cap E_j(q_j, \theta_j)$. Moreover, in Section **??**, we discuss the problem of determining the intersections

between two ellipses. This problem is equivalent to the problem of determining every center and angle of rotation that puts two points on an ellipse, which is the problem used in the definition of $S_j^{(2)}$ for every pair of points. In **??**, it is shown that this problem has at most two solutions. Therefore, we have that $|S_j^{(2)}| \leq 2\binom{n}{2}$.

For the last case, we are left with solutions where the j-th ellipse covers more than two points, and there exists an equivalent solution with three points on it. As $S_j^{(3)}$ contains every center and angle of rotation that puts three points on the j-th ellipse, an equivalent solution for this case is present in the set of solutions $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$. Also, in Lemma 5.2 it is stated that the number of centers and angles of rotation that make an ellipse contain three given points is at most six. Therefore, we have that $|S_j^{(3)}| \leq 6\binom{n}{3}$.

Combining the three cases, as $S_j = S_j^{(1)} \cup S_j^{(2)} \cup S_j^{(3)}$, we get the following bound for its size:

$$
\begin{aligned}
|S_j| &\leq 6\tbinom{n}{3} + 2\tbinom{n}{2} + n &= n(n-1)(n-2) + n(n-1) + n \\
|S_j| &\leq 6\tbinom{n}{3} + 2\tbinom{n}{2} + n &= n((n-1)^2 + 1) \leq n^3.
\end{aligned}
$$

Therefore, we conclude that $|\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})| \leq |S_1| \times \cdots \times |S_m| \leq n^{3m}$. $\square$

5.3   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, in Algorithm 4, we present a procedure called CLS-MCER which returns the CLS for an ellipse with shape parameters (a, b). Then, in Algorithm 5 we describe the procedure that returns an optimal solution for MCER.

Let E be the coverage region of an ellipse with shape parameters (a, b). We assume that in Algorithm 4, the procedure $e2p(u, v, a, b)$ returns every $(q, \tilde{\Phi}_j(u, v)) \in \mathbb{R}^2 \times [0, \pi)$, such that, $\{u, v\} \subset \partial E(q, \tilde{\Phi}_j(u, v))$. That is, this procedure returns every location for the ellipse with shape parameters (a, b), such that its angle of rotation is $\tilde{\Phi}_j(u, v)$, and the points u, v are on the ellipse. This can be done using the results of Section **??**.

After that, we define Algorithm 5 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 5. Instead, a complete search is done by backtracking the CLS of every ellipse returned by procedure CLS-MCER defined in Algorithm 4.

Two procedures are defined in Algorithm 5. 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 $j \in \{1, \ldots, m\}$, 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 with indexes $1, \ldots, j-1$.

Algorithm 4 – Algorithm that constructs a CLS for an ellipse.

Input: A set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$, 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 3.
11:     **end for**
12:     **return** S
13: **end procedure**

Algorithm 5 – Algorithm for MCER

Input: A set of points $\mathcal{P} = \{p_1, \ldots, p_n\}$, a list of weights $\mathcal{W} = \{w_1, \ldots, w_n\}$, and a list of shape parameters $\mathcal{R} = \{(a_1, b_1), \ldots, (a_m, b_m)\}$.
Output: An optimal solution for the given instance of MCER.

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

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

Corollary 5.1. Algorithm 5 takes $\mathcal{O}(mn^{3m+1})$ operations and returns an optimal solution for an instance of MCER.

Proof. For every $j \in \{1, \ldots, m\}$, unless $Z = \{\}$, when choosing the center and angle of rotation for the j-th ellipse, Algorithm 5 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 5 considers every solution in $\Omega(\mathcal{P}, \mathcal{W}, \mathcal{R})$. As evaluating each solution can be done in $\mathcal{O}(nm)$, we get the overall runtime complexity of $\mathcal{O}(mn^{3m+1})$. $\square$

In Figure 20, a solution returned by Algorithm 5 for the instance AB120 taken from Andretta and Birgin (2013) is displayed. The exact method developed by Andretta and Birgin (2013), for this instance, could not obtain an optimal solution within the established time limit, however, comparing with the solution obtained by our algorithm, their heuristic method does find an optimal solution. In the next chapter, we give more details about the solutions found by our algorithm, along with the proposal of some new instances for MCER.

Figure 20 – An optimal solution of MCER-*k* for the instance AB120.
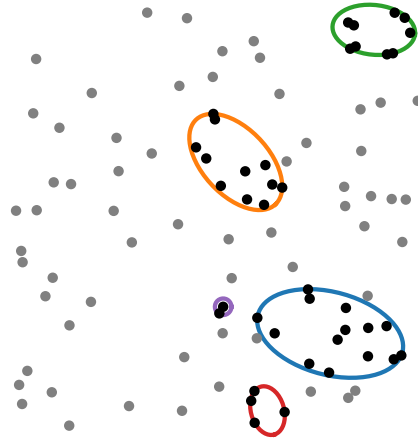
Source: Elaborated by the author.

5.3.1 *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 $\mathcal{C} := \{c_1, \ldots, c_m\}$, 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 $I := \{i_1, \ldots, i_k\} \subset \{1, \ldots, m\}$; $Q := (q_1, \ldots, q_k) \in \mathbb{R}^{2k}$, with q_j being the center of the i_j-th ellipse; and $\Theta := (\theta_1, \ldots, \theta_k) \in [0, \pi)^k$, 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 **??**, we introduce Algorithm 6 for MCER-k that uses Algorithm 5 for every $I := \{i_1, \ldots, i_k\} \subset \{1, \ldots, m\}$. Therefore, Algorithm 6 returns an optimal solution for MCER-k in $\mathcal{O}(k\binom{m}{k}n^{3k+1}) = \mathcal{O}(m2^m n^{3m+1})$ time.

Algorithm 6 – Algorithm for MCER-k

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

1: **procedure** MCER-$k(\mathcal{P}, \mathcal{W}, \mathcal{R}, \mathcal{C}, k)$
2:     $I^* = \{i_1^*, \ldots, i_k^*\} \leftarrow \{1, \ldots, k\}$
3:     $Q^* = (q_1^*, \ldots, q_k^*) \leftarrow (0, \ldots, 0)$
4:     $\Theta^* = (\theta_1^*, \ldots, \theta_k^*) \leftarrow (0, \ldots, 0)$
5:     **for all** $I = \{i_1, \ldots, i_k\} \subset \{1, \ldots, m\}$ **do**
6:         $\mathcal{R}' \leftarrow \{(a_j, b_j) \in \mathcal{R} : j \in I\}$
7:         $(q_1, \ldots, q_k); (\theta_1, \ldots, \theta_k) \leftarrow MCER(\mathcal{P}, \mathcal{W}, \mathcal{R}')$
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:             $Q^* \leftarrow (q_1, \ldots, q_k)$
10:            $\Theta^* \leftarrow (\theta_1, \ldots, \theta_k)$
11:            $I^* \leftarrow I$
12:         **end if**
13:     **end for**
14:     **return** I^*, Q^*, Θ^*
15: **end procedure**

NUMERICAL EXPERIMENTS

The goal of this chapter is to show the results of the algorithms for MCE-k and MCER-k proposed by us for instances proposed by other works as well as instances created by us. We also give implementation details and suggestions to improve performance, which we adopted in our implementations. All the experiments were run in a computer with the following specification:

- CPU Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz;

- 16Gib of RAM memory;

- Linux Operating System: Debian 4.19.5.

6.1 Implementation

In this section, we give more details about the implementation of the algorithms developed in our work.

All the algorithms were implemented using the C++ language, compiled with g++ (G++ 6.3.0) with the optimization flag -O3. The actual code is available in <https://sites.icmc.usp.br/andretta/tedeschi-2020/>.

6.1.1 Determining the eigenvalues of a matrix

In Algorithm 3 for E3P, we assumed that a procedure which returns every eigenvalue of a given square matrix was available. In practice, we used the very famous linear algebra package LAPACK (see Anderson *et al.* (1999) for more details). Even though LAPACK is a library for the FORTRAN programming language, its routines can be made available in a C/C++ environment by simply adding the -llapack linking flag to the compilation. The only remarks, though, are that FORTRAN represents matrices in a column-major fashion, and receives parameters only by

reference. Therefore, matrices must be transposed before being passed to a routine, and every parameter must receive a pointer to a variable containing its value.

LAPACK offers a routine called ZGEEV that computes every eigenvalue of a complex matrix by using an implementation of the QR algorithm. This routine optionally can also be asked to compute the right or left eigenvectors depending on two of its parameters. ZGEEV receives in total 14 parameters, with 4 of them being used for output. We show a brief description of them in Table 1, along with the specification of the value we set each parameter in our implementation.

Parameter	Description	Value
JOBVL	Indicates whether to compute the left eigenvalues	'N' (no eigenvectors should be computed)
JOBVR	Indicates whether to compute the right eigenvalues	'N' (no eigenvectors should be computed)
N	Order of matrix A	6
A	The square matrix whose eigenvalues are to be computed	The companion matrix
LDA	Leading dimension of A	6
W	The eigenvalues output array	A complex array of size 6
VL	The left eigenvectors output array	A complex array of size 1
LDVL	Leading dimension of VL	1
VR	The right eigenvectors output array	A complex array of size 1
LDVR	Leading dimension of VR	1
WORK	A workspace for the procedure to utilize	A complex array of size 12
LWORK	Dimension of WORK	12
RWORK	A real workspace of size 2N	A double array of size 12
INFO	An integer containing 0 if the algorithm was able to compute every eigenvalue	A pointer to an integer variable

Table 1 – The ZGEEV's parameter list.

6.1.2   Symbolic computation

Symbolic computation is a vast topic, which deals with the problem of solving or manipulating mathematical expressions computationally. Back in Chapter 4, we were faced with the problem of writing function ξ defined in Equation 4.3 as a complex polynomial in the power

format by replacing the sine and cosine functions with the identities given by Equation 4.16 and Equation 4.17.

As expected, computing the coefficients of that polynomial in terms of the E3P's instance by hand is very challenging; the expressions get too long, and it becomes humanly impossible not to make any mistake. For that reason, we resort to Symbolic computation for this task.

In practice, we utilized an external library for Python called SymPy (see Meurer *et al.* (2017) for more information). This tool can create expressions using arithmetic operators on predefined symbols, numbers, and other expressions. It can also convert expressions into polynomials in the power format, and output them directly into C code. Using these features, we can write $\xi(\theta)(e^{i\theta})^6$ as a polynomial by replacing the sine and cosine functions with expressions for the identities given by Equation 4.16 and Equation 4.17, and then import it into our C++ implementation of E3P's algorithm by printing the polynomial's list of coefficients as C code. The actual coefficients of that polynomial are presented in Appendix A in terms of a generic E3P's instance.

6.2 Some details and improvements

To achieve the results that are shown later in this chapter, an efficient implementation of **??** for MCE and Algorithm 5 for MCER had to be done. Just translating those algorithms into a programming language is not enough to obtain solutions for every instance previously published in Andretta and Birgin (2013). Therefore, we present here some improvements that can be applied to the implementation of those algorithms, which can result in significant growth of performance, especially in terms of CPU time.

6.2.1 CLS construction

In both algorithms, a subroutine to construct an ellipse's CLS is called inside the backtracking routine. This can potentially make the same combination of points be considered multiple times. To avoid this unnecessary computation, we compute the CLS for every ellipse beforehand in a preprocessing phase for the whole demand set. Then, in the backtracking, we only consider the options of locations that makes the ellipse cover at least one point that has not been covered before.

Another improvement that can be made in the construction of an ellipse's CLS is the elimination of redundant solutions. Let (Q, Θ) and (Q', Θ') be two solutions of MCER (the same can be said about MCE). If, for any $j \in \{1, \ldots, m\}$, we have $\mathcal{P} \cap E_j(q'_j, \theta'_j) \subset \mathcal{P} \cap E(q_j, \theta_j)$, then we can for sure dismiss solution (Q', Θ'). In our implementation, we use the same tree-like data structure as the one described in Andretta and Birgin (2013) to only keep solutions that are not redundant.

Calling E3P's algorithm for every triplet of points in an instance of MCER can be expensive. To avoid that, given three points and an ellipse with shape parameters (a,b), we can skip calling E3P's algorithm if the maximum distance between any of the points is greater than $2a$, or if the triangle's area with vertices on these three points have area greater than $\frac{3\sqrt{3}}{4}\pi ab$, which can be proved to be the greatest area of an inscribed triangle in an ellipse with shape parameters (a,b).

6.2.2 Backtracking

Without any improvement, backtracking through every possible combination of every ellipse's CLS can take a very long time, possibly going through a lot of non-optimal solutions. For this reason, we introduce a sufficient condition for the MCER's case (the MCE's case is analogous), based on MCER for one ellipse, which can be used to skip solutions that for sure are non-optimal.

Given an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCER, suppose that the first j ellipses are fixed at the locations $(q_1, \theta_1); \ldots; (q_j, \theta_j)$. Let Z_j be the points that are not covered by the first j ellipses, and OPT_j the value of the best solution with the location of the first j ellipses fixed at $(q_1, \theta_1); \ldots; (q_j, \theta_j)$.

Then, we can obtain an upper-bound for OPT_j by using, for $k \in \{j+1, \ldots, m\}$, the solutions (q'_k, θ'_k) of MCER for instances with demand set Z_j and only one ellipse with shape parameters (a_k, b_k). As these solutions only consider the best cover individually for each ellipse, we have the following inequality

$$OPT_j \leq w\left(\bigcup_{k=1}^{j} \mathcal{P} \cap E_k(q_k, \theta_k)\right) + w\left(\bigcup_{k=j+1}^{m} \mathcal{P} \cap E_k(q'_k, \theta'_k)\right). \tag{6.1}$$

This upper-bound for OPT_j can then be used in the backtracking process to skip solutions that are not better than any optimal solution. Let OPT_{lo} be a lower bound for the optimal solution, we have that if

$$w\left(\bigcup_{k=1}^{j} \mathcal{P} \cap E_k(q_k, \theta_k)\right) + w\left(\bigcup_{k=j+1}^{m} \mathcal{P} \cap E_k(q'_k, \theta'_k)\right) \leq OPT_{lo}, \tag{6.2}$$

then $OPT_j \leq OPT_{lo}$, which implies that OPT_j is less than or equal the value of any optimal solution. This defines a sufficient condition for us to dismiss every solution which have the location of the first j ellipses fixed at $(q_1, \theta_1); \ldots; (q_j, \theta_j)$. In practice, we can use the value of the best solution found so far as the lower-bound for the optimal solution.

It is worth pointing out that these improvement suggestions do not have an effect in a possible worst case scenario. We are adopting them in our implementation because they showed good results in practice. For example, without taking the suggestion given by Equation 6.2, MCER-k's algorithm takes nine seconds to obtain an optimal solution for instance AB060,

going through 336,494,451 solutions. In Table 6, we show the results of MCER-*k*'s algorithm implemented with all the improvement suggestions given here; for instance AB060, the algorithm takes less than one second to return an optimal solution, and evaluates only 1809 solutions.

6.3 Results for known instances

In this section, we present the results of **??** for MCE-*k* and Algorithm 6 for MCER-*k* for instances CM1,CM2, CM4, CM5, CM7, CM8 proposed by Canbolat and Massow (2009), and for instances CM3,CM6,CM9 and AB001-AB120 proposed by Andretta and Birgin (2013).

For each instance, we display the selected ellipses and the income of the found optimal solution. We also display some performance metrics with the intention of giving an idea of how much computation had to be done for the algorithms to find an optimal solution. These metrics are: the CLS size of every ellipse, the number of nodes in the backtracking tree, the number of leaves corresponding to a solution in the backtracking tree, the CPU time spent on constructing the CLSs, and the total CPU time. For the algorithms for MCER, we also have a column for the number of E3P subproblems that were solved, not counting the triplet of points which are dismissed by the improvements suggestions given in Section 6.2. All the tables containing results referenced in this section are presented in Appendix B. We also made available at <https://sites.icmc.usp.br/andretta/tedeschi-2020/> every instance used here, along with the graphical representation of every obtained solution.

6.3.1 MCE-k

In Table 2, the results for instances CM1-CM9 are shown. The algorithm proposed here showed great results as it was able to obtain optimal solutions in less than one second for every one of the instances CM1-CM9. Even though the experiments were run in a different environment, we can still say that this is a great improvement compared with the results from Andretta and Birgin (2013). For example, to obtain an optimal solution for the instance CM9, the method proposed by Andretta and Birgin (2013) took more than thirty minutes. In Table 4 and Table 5, we present the results for instances AB001-AB120. The only instance that our algorithm took more than one second to return an optimal solution was AB120, which it took 1.08 second.

Back in **??**, it was shown that the size of a CLS in an instance of MCE is less than or equal n^2. This bound, at least for these instances, seems to be very loose. Notice that the biggest CLS observed, which is 174 in instance CM9, is very far away from n^2, which in this case is 10^4. The same can be said about the bound for the running time of the algorithm, using the size of the backtracking tree times n (the number of steps needed to evaluate an option in a CLS) as an estimate for the number of computations, it can be seen that the asymptotic bound of $\mathcal{O}(m2^m n^{2m+1})$ is very far from the actual number of computations.

6.3.2   MCER-*k*

The numerical results obtained by our implementation of Algorithm 6 for MCER-*k* are shown in Table 3 for instances CM1-CM9, and in Table 6 and Table 7 for instances AB001-AB120. An optimal solution was obtained for every instance, and overall, at most six seconds of CPU time was taken.

Looking at the numerical results of the heuristic method proposed in Andretta and Birgin (2013), the only non-optimal solutions it encountered were for instances AB105-AB108. For these instances, our algorithm obtained an optimal solution covering one more point. In Figure 21, the optimal solution for AB108 is displayed. In general, our algorithm took much lower CPU time compared to the methods developed in Andretta and Birgin (2013). For example, for instance CM9, their heuristic method took more than six hours to return a solution, our implementation of MCER-*k*'s algorithm, on the other hand, obtained an optimal solution in less than five seconds.

Figure 21 – An optimal solution of MCER-*k* for the instance AB108.
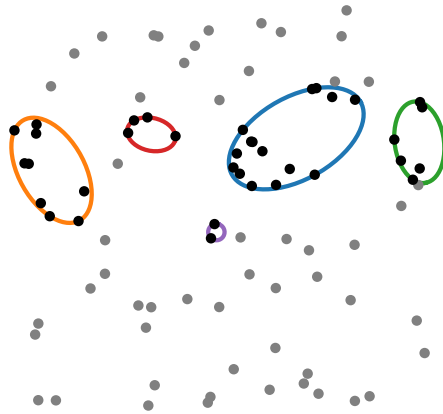
Source: Elaborated by the author.

As it was said for the results of MCE-*k*, in practice, the bounds for the CLS size and the number of operations taken by the algorithm is very loose. Notice that, the greatest CLS size was 730 obtained for the third ellipse in instances CM7-CM9. Notice that 730 is very distance from its upper-bound n^3, which in this case is 10^6. Also, looking at the size of the backtracking tree times *n* as an estimate for the number of computations taken by the algorithm, for example, for instance AB120, this number is 1579×100, which is very distant from its asymptotic upper-bound of $\mathcal{O}(m2^m n^{3m+1})$, which in this case would be $5 \times 2^5 \times 100^{16}$.

6.4   New instances

After examining the results obtained by our algorithms for the formerly known instances, constructing new ones, more challenging for our work, can be taken as essential for further

analyzing the algorithms proposed by us. Besides increasing the size of the demand set and the number of ellipses, we also designed instances with non-unitary weights, which is something none of the previous instances had. Moreover, to create very distinct instances from the previously introduced ones, we used a different probability distribution, other than the uniform one, to generate the location of points. We set a time limit of two hours of running time for solving every instance, meaning that if an algorithm did not stop in two hours, we report that it was not able to determine an optimal solution. In total, we designed 46 new instances, which will be referred to as TA01, . . . , TA46, and are available at <https://sites.icmc.usp.br/andretta/tedeschi-2020/>. The numerical results for every one of them are displayed in Appendix B.

The first set of instances were construct sampling each demand points from a bivariate normal distribution $\mathcal{N}([0,0]^T, \mathbb{I})$, with $\mathbb{I} \in \mathbb{R}^{2\times 2}$ being the identity matrix; and setting each point's weight as its squared distance to the origin. This is expected to produce a demand set with most points located near the origin, but with the most valuable points located far away from it. We generated a set of $n = 100$ points, with $m = 7$ ellipses, making the j-th ellipse have shape parameters randomly taken from a uniform distribution in $[0.5, 1.5]$, and cost $c_j = 10 \times a_j \times b_j$. From that, we created seven instances taking $k \in \{1, \ldots, m\}$. The results for MCE-k can be seen in Table 8 and the results for MCER-k are presented in Table 9. Because the normal distribution generates most of the points close to each other (see Figure 22 for an example), every ellipse's CLS's size ended up being significantly bigger if compared to previously introduced instances with 100 points. This, and the subtle increase in the number of ellipses, made the algorithms for MCER-k and MCE-k time out for some instances. The algorithm for MCER-k did not return an optimal solution within the predefined time limit for the last three instances ($k = 5, 6, 7$), while the algorithm for MCE-k did not finish in time for the last instance ($k = 7$). The optimal solution for MCER-k for the instance with $k = 4$ is displayed in Figure 22.

Figure 22 – An optimal solution of MCER-k for the instance TA04.
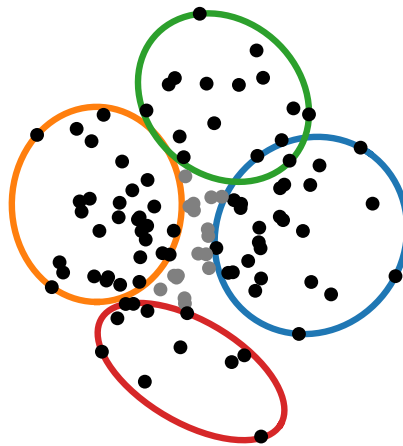
Source: Elaborated by the author.

For the second set of instances, we used the same process used for the first set to generate the demand points and the ellipses. We kept the number of ellipses at 3 and created five demand sets with $n \in \{200, 250, 300, 350, 400\}$. In total, we had 15 instances having $k \in \{1, \ldots, m\}$. The results for MCE-k can be seen in Table 10 and the results for MCER-k are presented in Table 11. The solution of MCER-k for the instance with $n = 400$, and $k = 2$ is shown in Figure 23.

Figure 23 – An optimal solution of MCER-k for the instance TA21.
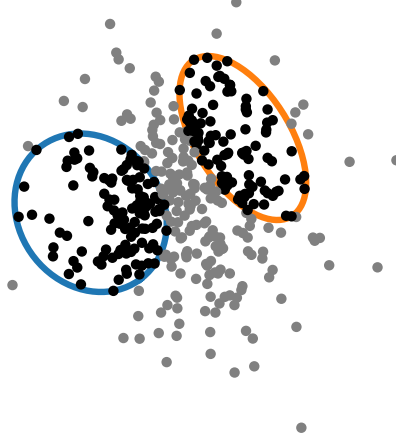
Source: Elaborated by the author.

The third set of instances were constructed generating the demand set following a uniform distribution in $[0, 10]^2$, with each point having unitary weight; and the ellipses by the same process used for the first two set of instances. We created instances with $m = 5$, $n \in \{400, 450, 500\}$, and $k \in \{1, \ldots, m\}$, with a total of 15 instances. The results for MCE-k can be seen in Table 12 and the results for MCER-k are presented in Table 13. Optimal solutions were obtained for every one of the instances in this set. It is possible to see that, compared with the first two sets of instances, the CLS sizes are smaller, mostly because of the size of the ellipses and the uniform distribution used to generate the points. The optimal solution returned by MCER-k's algorithm for the instance with $n = 500$ and $k = 5$ is shown in Figure 24.

The last set of instances were constructed using two bivariate normal distributions with unitary variance $\mathscr{N}(\mu^{(1)}, \mathbb{I})$ and $\mathscr{N}(\mu^{(2)}, \mathbb{I})$, $\mu^{(1)}, \mu^{(2)} \in \mathbb{R}^2$. Half of the points were generated following $\mathscr{N}(\mu^{(1)}, \mathbb{I})$, and the other half $\mathscr{N}(\mu^{(2)}, \mathbb{I})$; the weight of every point was set as its squared distance to the mean of the distribution from which it was generated. The ellipses were also divided into two halves, taking their shape parameters from uniform distributions in the intervals $[0.5, 1.5]$, and $[3, 4]$; setting the j-th ellipse's weight as $c_j = a_j \times b_j$. The purpose of this last set of instances was to create an example where the chosen ellipses in the solution of an instance of MCER-k is not a subset of the chosen ellipses in an optimal solution of that same instance for MCER-$(k + 1)$. We created seven instances with $n = 80$, $m = 6$ and $k \in \{1, \ldots, m\}$,

Figure 24 – An optimal solution of MCER-*k* for the instance TA37.
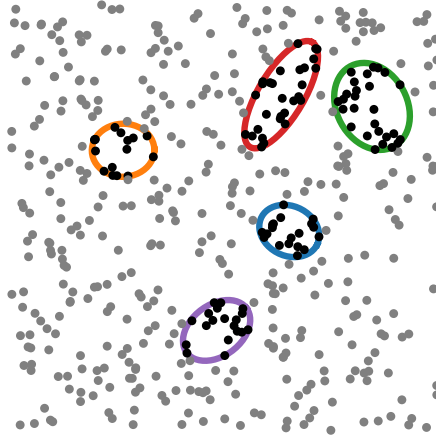
Source: Elaborated by the author.

and defined the values of $\mu^{(1)}$ and $\mu^{(2)}$ specifically to create such counter-example. The results are shown in Table 14 for MCE-*k* and in Table 14 for MCER-*k*. In Figure 25, we show the solutions for the instances with $k = 2$, where two of the bigger-sized ellipses are used, and $k = 3$, where one of the bigger-sized ellipses is replaced by two small ones.

Figure 25 – Optimal solutions for instances TA44 and TA45.

(a) TA44
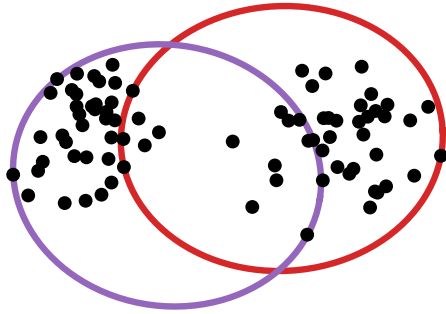
(b) TA45
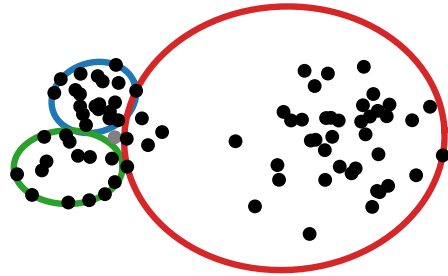
Source: Elaborated by the author.

CHAPTER

7

CONCLUSION

In this work, we studied two problems of maximum planar covering using ellipses, with most of the substantial results being for the less previously-studied problem where the ellipses can be freely rotated, which we referred to as MCER. We devoted Chapter 4 and Chapter 5 of our work for the development of an exact algorithm for this problem, which depended on the development of an algorithm for a never-studied-before geometric subproblem, which we referred to as E3P.

An exact algorithm was also developed for the other problem studied in this work, in which the ellipse could not be rotated. We created an algorithm for it in **??** based on an algorithm for maximum covering using disks, which we introduced in **??**.

In Chapter 6, numerical experiments were run for both algorithms. We first used instances found in the literature, and then proposed new instances to further analyze the performance of our algorithms. Even though both algorithms proposed by us are exponential in nature, in Chapter 6, we gave several improvements suggestions, which allowed our implementations to obtain optimal solutions for every previously published instance, including instances that no optimal solutions were obtained before, plus some fairly large new ones.

We believe there is plenty of room for furtherly working on these problems. Back in Chapter 4, we raised the attention for some possible properties which could be used in the development of an improved version of our algorithm for E3P. In Chapter 6, we observed that the bounds for the algorithms proposed by us might be a little loose, obtaining tighter bounds might be possible. Some other suggestions for future work are: modeling the problems as Linear Integer Programming problems, which could be an alternative to backtracking the optimal solution, done in the algorithms for both problems; adapting the approximation algorithm for the maximum covering by disks problem developed in Berg, Cabello and Har-Peled (2006) for the ellipses case; and developing and analyzing heuristics which avoid the backtracking phase of both algorithms.

BIBLIOGRAPHY

ANDERSON, E.; BAI, Z.; BISCHOF, C.; BLACKFORD, S.; DEMMEL, J.; DONGARRA, J.; CROZ, J. D.; GREENBAUM, A.; HAMMARLING, S.; MCKENNEY, A.; SORENSEN, D. **LA-PACK Users' Guide**. Third. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999. ISBN 0-89871-447-8 (paperback). Citations on pages 27 and 67.

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 16, 17, 30, 64, 69, 71, and 72.

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>. Citation on page 16.

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 20.

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 16.

BAREL, M. V.; VANDEBRIL, R.; DOOREN, P. V.; FREDERIX, K. Implicit double shift qr-algorithm for companion matrices. **Numerische Mathematik**, v. 116, n. 2, p. 177–212, Aug 2010. ISSN 0945-3245. Available: <https://doi.org/10.1007/s00211-010-0302-y>. Citation on page 27.

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 45.

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 16, 31, 33, and 77.

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 45, 46, and 47.

BOYD, J. P. **Chebyshev and Fourier Spectral Methods**. Second. Mineola, NY: Dover Publications, 2001. (Dover Books on Mathematics). ISBN 0486411834 9780486411835. Citation on page 45.

_____. 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 27 and 48.

BRANNAN, D.; ESPLEN, M.; GRAY, J. **Geometry**. Cambridge University Press, 1999. ISBN 9781107393639. Available: <https://books.google.co.id/books?id=HbytAQAAQBAJ>. Citation on page 20.

CANBOLAT, M. S.; MASSOW, M. von. Planar maximal covering with ellipses. **Computers and Industrial Engineering**, v. 57, p. 201–208, 2009. Citations on pages 16, 17, 30, 46, and 71.

CHAZELLE, B. M.; LEE, D. On a circle placement problem. **Computing**, v. 36, p. 1–16, 03 1986. Citations on pages 16 and 31.

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 15 and 16.

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 15.

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 16.

DREZNER, Z. Note—on a modified one-center model. **Management Science**, v. 27, p. 848–851, 07 1981. Citations on pages 16 and 33.

GAUTSCHI, W. The condition of polynomials in power form. **Mathematics of Computation - Math. Comput.**, v. 33, 01 1979. Citation on page 44.

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 45.

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 15.

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 26.

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 41.

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 15.

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 15.

MANI-LEVITSKA, P. Chapter 1.1 - characterizations of convex sets. In: GRUBER, P.; WILLS, J. (Ed.). **Handbook of Convex Geometry**. Amsterdam: North-Holland, 1993. p. 19 – 41. ISBN 978-0-444-89596-7. Available: <http://www.sciencedirect.com/science/article/pii/B9780444895967500067>. Citation on page 30.

MARTíN, P.; MARTINI, H. Algorithms for ball hulls and ball intersections in normed planes. **Journal of Computational Geometry**, Journal of Computational Geometry, Vol 6, p. No 1 (2015)–, 2015. Available: <https://journals.carleton.ca/jocg/index.php/jocg/article/view/187>. Citations on pages 30, 32, and 33.

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 43.

MEURER, A.; SMITH, C. P.; PAPROCKI, M.; ČERTÍK, O.; KIRPICHEV, S. B.; ROCKLIN, M.; KUMAR, A.; IVANOV, S.; MOORE, J. K.; SINGH, S.; RATHNAYAKE, T.; VIG, S.; GRANGER, B. E.; MULLER, R. P.; BONAZZI, F.; GUPTA, H.; VATS, S.; JOHANSSON, F.; PEDREGOSA, F.; CURRY, M. J.; TERREL, A. R.; ROUČKA, Š.; SABOO, A.; FERNANDO, I.; KULAL, S.; CIMRMAN, R.; SCOPATZ, A. Sympy: symbolic computing in python. **PeerJ Computer Science**, v. 3, p. e103, Jan. 2017. ISSN 2376-5992. Available: <https://doi.org/10.7717/peerj-cs.103>. Citation on page 69.

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 27, 43, and 44.

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 15.

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 15.

SKOPENKOV, A. A short elementary proof of the insolvability of the equation of degree 5. **arXiv preprint arXiv:1508.03317**, 2015. Citation on page 26.

Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Jarrod Millman, K.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C.; Polat, İ.; Feng, Y.; Moore, E. W.; Vand erPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; Contributors, S. . . SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. **Nature Methods**, 2020. Citation on page 46.

WADA, J. Strict convexity and smoothness of normed spaces. **Osaka Math. J.**, v. 10, p. 221–230, 1958. ISSN 0388-0699. Citation on page 30.

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 26.

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 48.

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 16.

COMPLEX POLYNOMIAL'S COEFFICIENTS

We present here the coefficients of the polynomial $f(y) = \sum_{k=0}^{6} c_{2k}y^k$ defined in Equation 4.20.

A.1 c_0

$$+\frac{x_1^4 x_2^2}{64} - 0.015625 x_1^4 y_2^2 + \frac{x_1^2 x_2^4}{64} + \frac{x_1^2 y_2^4}{64} - 0.015625 x_2^4 y_1^2 + \frac{x_2^2 y_1^4}{64} + \frac{y_1^2 y_2^2 \left(-y_1^2 + 2y_1 y_2 - y_2^2\right)}{64}$$

$$+0.09375 x_1^3 x_2 y_2^2 + 0.03125 x_1^3 \left(-x_2^3 + i y_2^3\right) + 0.09375 x_1^2 y_1^2 y_2^2 + 0.09375 x_1 x_2^3 y_1^2 + 0.03125 i x_2^3 y_1^3$$

$$+\frac{-6.0 a^6 \left(x_1^2 x_2^2 y_2^2 + y_1 \left(x_1^2 x_2^2 y_1 + y_2 \left(x_2^2 y_1^2 + y_2 \left(x_1^2 y_2 - x_2^2 y_1\right)\right)\right)\right) - b^6 x_1^4 x_2^2 + 2b^6 x_1^3 x_2^3 - b^6 x_1^2 x_2^4}{64 a^6}$$

$$+\frac{0.03125 i a^6 x_1 x_2^4 y_1 + 0.03125 i a^6 x_1 y_1 y_2^4 + 0.015625 b^6 \left(x_1^4 y_2^2 + x_2^4 y_1^2 + y_1^4 y_2^2 + y_1^2 y_2^4\right)}{a^6}$$

$$+i \left(0.0625 x_1^3 x_2^2 y_1 + 0.0625 x_1^2 x_2^3 y_2 + y_2 \left(0.0625 x_1 y_1^3 y_2 + 0.0625 x_2 y_1^2 y_2^2 + 0.03125 x_2 \left(x_1^4 + y_1^4\right)\right)\right)$$

$$+\frac{x_1 \left(0.046875 b^4 x_1^3 x_2^2 + 0.046875 b^4 x_1 y_2^4 + x_2 \left(0.125 a^4 y_1 y_2 \left(y_1^2 + y_2^2\right) + 0.046875 b^4 x_1 x_2^3\right)\right)}{a^4}$$

$$+\frac{0.046875 b^2 \left(a^2 x_1^4 y_2^2 + a^2 x_2^4 y_1^2 + a^2 y_1^4 y_2^2 + a^2 y_1^2 y_2^4 + b^2 x_2^2 y_1^4\right)}{a^4}$$

$$+\frac{0.28125 a^6 x_1^2 x_2^2 y_1 y_2 + 0.09375 a^4 b^2 x_1^3 x_2^3 + 0.09375 a^2 b^4 y_1^3 y_2^3 - 0.015625 b^6 \left(x_1^2 y_2^4 + x_2^2 y_1^4\right)}{a^6}$$

$$-\frac{0.0625 i a^6 x_1 \left(x_1^2 y_1 y_2^2 + x_1 x_2 y_2^3 + x_2^2 y_1^3\right) + 0.0625 i a^6 x_2 y_1^2 y_2 + 0.03125 b^6 y_1^3 y_2^3}{a^6}$$

$$-\frac{0.046875 b^4 x_2^4 y_1^2 + 0.046875 b^4 y_1^2 y_2^4 + x_1 y_2 \left(0.125 a^4 x_2 y_1 \left(x_1^2 + x_2^2\right) + 0.046875 b^4 x_1^3 y_2\right)}{a^4}$$

$$-\frac{0.09375 i a^4 x_1 y_1^2 y_2^3 + 0.046875 b^2 \left(a^2 x_1^4 x_2^2 + a^2 x_1^2 x_2^4 + a^2 x_1^2 y_2^4 + a^2 x_2^2 y_1^4 + b^2 y_1^4 y_2^2\right)}{a^4}$$

$$-\frac{0.28125a^4x_1x_2y_1^2y_2^2 + 0.09375a^2b^2y_1^3y_2^3 + 0.09375x_2\left(ia^4\left(x_1^3x_2y_2 + y_1\left(x_1^2x_2^2 + y_1^2y_2^2\right)\right) + b^4x_1^3x_2^2\right)}{a^4}$$

$$+\frac{0.09375b^4\left(ia^2\left(x_1^3y_2^3 + x_2^3y_1^3\right) + b^2x_1^2x_2^2y_1^2 + b^2x_1^2x_2^2y_2^2 + b^2x_1^2y_1y_2^3 + b^2x_2^2y_1^3y_2\right)}{a^6}$$

$$+\frac{0.28125\left(a^2b^2x_2^2y_1^3y_2 + x_1\left(a^2b^2x_1y_1y_2^3 + x_2\left(b^4x_1^2y_2^2 + y_1\left(ia^4y_2\left(x_1y_2 + x_2y_1\right) + b^4x_2^2y_1\right)\right)\right)\right)}{a^4}$$

$$+\frac{b^2\left(0.28125a^2\left(a^2x_1^2x_2^2y_2^2 + y_1^2\left(a^2x_1^2x_2^2 + b^2y_2^2\left(x_1^2 + x_2^2\right)\right)\right) - 0.03125ib^4x_1^3y_2^3 - 0.03125ib^4x_2^3y_1^3\right)}{a^6}$$

$$-\frac{0.09375b^2\left(ia^4\left(x_1^3y_2^3 + x_2^3y_1^3\right) + b^4x_1^3x_2y_2^2 + b^4x_1^2y_1^2y_2^2 + b^4x_1x_2^3y_1^2 + b^4x_2^2y_1^3y_2^2\right)}{a^6}$$

$$-\frac{x_1\left(a^2x_2\left(0.28125b^2x_1^2y_2^2 + y_1\left(0.1875ia^2y_2\left(x_1y_1 + x_2y_2\right) + 0.28125b^2x_2^2y_1\right)\right) + 0.28125b^4x_1y_1y_2^3\right)}{a^4}$$

$$+\frac{b^2\left(-0.28125a^2\left(a^2x_1^2y_1^2y_2^2 + a^2x_2^2y_1^2y_2^2 + b^2x_2^2\left(x_1^2y_2^2 + y_1^2\left(x_1^2 + y_1y_2\right)\right)\right) + 0.0625ib^4x_1x_2^2y_1^3\right)}{a^6}$$

$$+\frac{b^4\left(0.09375ia^2x_1x_2^4y_1 + b^2y_2\left(0.125x_1^3x_2y_1 + 0.125x_1x_2^3y_1 + 0.0625i\left(x_1^2y_2\left(x_1y_1 + x_2y_2\right) + x_2^3y_1^2\right)\right)\right)}{a^6}$$

$$+\frac{0.09375ib^4\left(b^2x_1^3x_2^2y_2 + b^2x_1^2x_2^3y_1 + y_2\left(a^2x_2y_1^4 + b^2x_2y_1^3y_2 + x_1\left(a^2x_1^3x_2 + y_1y_2^2\left(a^2y_2 + b^2y_1\right)\right)\right)\right)}{a^6}$$

$$+\frac{0.1875ib^2\left(a^2x_1^3y_1y_2^2 + a^2x_1^2x_2y_2^3 + b^2x_1^3x_2^2y_1 + b^2x_1^2x_2^3y_2 + y_1^2\left(b^2x_2y_2^3 + x_1y_1\left(a^2x_2^2 + b^2y_2^2\right)\right)\right)}{a^4}$$

$$+\frac{b^2x_2y_1y_2\left(0.375a^2x_1^3 + 0.375a^2x_1x_2^2 + 0.1875ia^2x_2^2y_1 + 0.375b^2x_1y_1^2 + 0.375b^2x_1y_2^2\right)}{a^4}$$

$$+\frac{0.28125b^2\left(ia^4\left(x_1^3x_2^2y_2 + y_1\left(x_1^2x_2^3 + y_1y_2^2\left(x_1y_2 + x_2y_1\right)\right)\right) + b^4x_1x_2y_1^2y_2^2\right)}{a^6}$$

$$-\frac{b^2x_1\left(0.03125ib^4x_1^3x_2y_2 + y_1\left(0.03125ib^4y_2^4 - x_2\left(0.84375a^2y_2\left(a^2y_1y_2 + b^2x_1x_2\right) - 0.03125ib^4x_2^3\right)\right)\right)}{a^6}$$

$$-\frac{ib^6\left(0.0625x_1^2x_2^3y_2 + y_1\left(0.0625x_1^3x_2^2 + y_1y_2\left(0.0625x_2y_2^2 + y_1\left(0.0625x_1y_2 + 0.03125x_2y_1\right)\right)\right)\right)}{a^6}$$

$$-\frac{b^2x_1\left(0.09375ia^4x_1^3x_2y_2 + y_1\left(0.09375ia^4y_2^4 + x_2\left(0.09375ia^4x_2^3 + 0.125b^4y_2\left(y_1^2 + y_2^2\right)\right)\right)\right)}{a^6}$$

$$-\frac{ib^2\left(0.1875a^2x_2y_1^2y_2^3 + 0.1875b^2x_1^2x_2y_2^3 + y_1^3\left(0.1875a^2x_1y_2^2 + x_2\left(0.09375a^2y_1y_2 + 0.1875b^2x_1x_2\right)\right)\right)}{a^4}$$

$$-\frac{b^2\left(0.375a^2x_1x_2y_1^3y_2 + 0.375a^2x_1x_2y_1y_2^3 + 0.1875i\left(a^2x_1^2x_2^3y_2 + y_1\left(b^2x_2^3y_1y_2 + x_1^3\left(a^2x_2^2 + b^2y_2^2\right)\right)\right)\right)}{a^4}$$

$$-\frac{b^4y_1\left(0.28125ix_1^2x_2^3 + y_2\left(x_1\left(0.375x_2\left(x_1^2 + x_2^2\right) + 0.28125iy_1y_2^2\right) + 0.28125ix_2y_1^2y_2\right)\right)}{a^4}$$

$$+\frac{b^2x_1x_2y_2\left(-0.84375a^4x_1x_2y_1 + 0.1875ib^4x_2y_1y_2 - b^2\left(0.84375a^2y_1^2y_2 + 0.28125x_1x_2\left(ia^2x_1 + b^2y_1\right)\right)\right)}{a^6}$$

$$+\frac{ib^2x_1x_2y_1y_2\left(0.5625a^4x_1y_1 + 0.5625a^4x_2y_2 + 0.84375a^2b^2x_1y_2 + 0.84375a^2b^2x_2y_1 + 0.1875b^4x_1y_1\right)}{a^6}$$

$$-\frac{ib^2x_1x_2y_1y_2\left(0.84375a^4x_2y_1 + b^2\left(0.5625a^2x_1y_1 + 0.5625a^2x_2y_2 + 0.28125b^2\left(x_1y_2 + x_2y_1\right)\right)\right)}{a^6}$$

$$-\frac{0.84375ib^2x_1^2x_2y_1y_2^2}{a^2}$$

A.2 c_2

$$+0.1875x_1^3x_2^3 - 0.09375x_1^2x_2^4 + \frac{x_2^4y_1^2}{32} + \frac{x_2^2y_1^4}{32} + \frac{y_2^2\left(x_1^4 + x_1^2y_2^2 + 3y_1^2\left(-y_1^2 + 2y_1y_2 - y_2^2\right)\right)}{32}$$

$$+x_1^2\left(x_2^2\left(-0.09375x_1^2 + 0.1875y_1^2 + 0.1875y_2^2\right) + 0.1875y_1^2y_2^2\right) - 0.1875x_1x_2^3y_1^2 + 0.1875x_2^2y_1^2y_2^2$$

$$+\frac{-6.0a^6y_2\left(x_1^2y_2\left(x_1x_2 + y_1y_2\right) + x_2^2y_1^3\right) - 3b^6x_1^4x_2^2 + b^6x_1^4y_2^2 + 6b^6x_1^3x_2^3 - 3b^6x_1^2x_2^4 + b^6x_1^2y_2^4}{32a^6}$$

$$+0.25ix_1y_1^3y_2^2 + 0.125ix_1y_1y_2^4 + 0.125ix_2y_1^4y_2 + 0.25ix_2y_1^2y_2^3 + \frac{0.03125b^6x_2^4y_1^2}{a^6} + \frac{0.03125b^6x_2^2y_1^4}{a^6}$$

$$+\frac{0.09375b^4y_1^2y_2^4 + x_1x_2\left(0.25a^4y_1y_2\left(x_1^2 + x_2^2 + y_1^2 + y_2^2\right) + 0.09375b^4x_1^3x_2 + 0.09375b^4x_1x_2^3\right)}{a^4}$$

$$+\frac{b^2\left(0.09375a^2\left(a^2x_1^4x_2^2 + a^2x_1^2x_2^4 + a^2y_1^4y_2^2 + a^2y_1^2y_2^4 + b^2y_1^4y_2^2\right) + 0.1875b^4y_1^3y_2^3\right)}{a^6}$$

$$+\frac{-0.03125b^4x_2^4y_1^2 - 0.03125b^4x_2^2y_1^4 + x_1^2\left(0.375ia^4x_2^2\left(x_1y_2 + x_2y_1\right) - 0.03125b^4x_1^2y_2^2 - 0.03125b^4y_2^4\right)}{a^4}$$

$$-\frac{0.125ia^2x_1^4x_2y_2 + 0.25ia^2x_1^3x_2^2y_1 + 0.125ia^2x_1x_2^4y_1 + 0.03125b^2\left(x_1^2y_2^2\left(x_1^2 + y_2^2\right) + x_2^4y_1^2 + x_2^2y_1^4\right)}{a^2}$$

$$-\frac{0.1875a^2b^4x_1^3x_2^3 + 0.1875a^2b^4y_1^3y_2^3 + y_2\left(0.25ia^6x_1^2x_2^3 + 0.09375b^6y_1^4y_2 + 0.09375b^6y_1^2y_2^3\right)}{a^6}$$

$$-0.5625x_1x_2y_1^2y_2^2 - 0.375ix_1y_1^2y_2^3 - 0.375ix_2y_1^3y_2^2 - \frac{0.1875b^2x_1^3x_2^3}{a^2} - \frac{0.1875b^2y_1^3y_2^3}{a^2}$$

$$+\frac{x_1x_2\left(0.1875a^2b^2x_1^2y_2^2 + 0.1875b^4x_1^2y_2^2 + x_2y_1\left(-0.5625a^4x_1y_2 + 0.1875a^2b^2x_2y_1 + 0.1875b^4x_2y_1\right)\right)}{a^4}$$

$$+\frac{0.1875b^2\left(b^4x_1^2x_2^2y_2^2 + b^4x_1^2y_1^2y_2^2 + y_1\left(a^2y_2\left(a^2x_2^2y_1^2 + b^2x_2^2y_1^2 + x_1^2y_2^2\left(a^2 + b^2\right)\right) + b^4x_1^2x_2^2y_1\right)\right)}{a^6}$$

$$-\frac{0.1875b^4\left(a^2x_1^2x_2^2y_1^2 + a^2x_1^2x_2^2y_2^2 + b^2\left(x_1^2y_1y_2^3 + x_2^2y_1^3y_2 + x_2\left(x_1^3y_2^2 + x_2y_1^2\left(x_1x_2 - y_2^2\right)\right)\right)\right)}{a^6}$$

$$-\frac{0.1875b^2\left(a^2x_1^2x_2^2y_2^2 + a^2x_1^2y_1^2y_2^2 + a^2x_2^2y_1^2y_2^2 + y_1^2\left(a^2x_1^2x_2^2 + b^2y_2^2\left(x_1^2 + x_2^2\right)\right)\right)}{a^4}$$

$$+\frac{0.125ib^2\left(a^4x_1^4x_2y_2 + a^2b^2x_1^4x_2y_2 + y_1\left(b^4x_2y_1^3y_2 + x_1\left(a^4x_2^4 + b^2\left(a^2x_2^4 + b^2y_2^4\right)\right)\right)\right)}{a^6}$$

$$+\frac{0.25ib^2\left(a^4x_1^2x_2^3y_2 + a^2b^2x_1^2x_2^3y_2 + y_1\left(a^4x_1^3x_2^2 + b^2\left(a^2x_1^3x_2^2 + b^2y_1y_2^2\left(x_1y_1 + x_2y_2\right)\right)\right)\right)}{a^6}$$

$$+\frac{b^2y_1y_2\left(0.375ia^2b^2x_2y_1^2y_2 + x_1\left(0.375ia^4y_1y_2^2 + b^2\left(0.375ia^2y_1y_2^2 + 0.25b^2x_2\left(x_1^2 + x_2^2 + y_1^2 + y_2^2\right)\right)\right)\right)}{a^6}$$

$$+\frac{b^2x_2\left(0.5625a^4x_1y_1^2y_2^2 + 0.5625a^2b^2x_1y_1^2y_2^2 + 0.375i\left(b^4x_1^3x_2y_2 + y_1\left(a^4y_1^2y_2^2 + b^4x_1^2x_2^2\right)\right)\right)}{a^6}$$

$$-\frac{b^2x_1\left(0.125ib^4x_1^3x_2y_2 + y_1\left(0.125ia^4y_2^4 + 0.125ia^2b^2y_2^4 - x_2^2\left(0.5625a^2x_1y_2\left(a^2 + b^2\right) - 0.125ib^4x_2^2\right)\right)\right)}{a^6}$$

$$-\frac{ib^2y_1^2y_2\left(0.25a^2x_2y_2^2+0.25b^2x_2y_2^2+y_1\left(0.25a^2x_1y_2+0.25b^2x_1y_2+0.125x_2y_1\left(a^2+b^2\right)\right)\right)}{a^4}$$

$$-\frac{0.25b^2x_1x_2\left(a^4y_1^3y_2+a^4y_1y_2^3+a^2b^2x_2^2y_1y_2+a^2b^2y_1^3y_2+b^2\left(a^2y_1y_2^3+ib^2x_1x_2\left(x_1y_1+x_2y_2\right)\right)\right)}{a^6}$$

$$-\frac{b^2y_1\left(0.375ia^2b^2x_1^2x_2^3+y_2\left(0.375ib^4x_2y_1^2y_2+x_1\left(0.25a^2x_2\left(a^2x_1^2+a^2x_2^2+b^2x_1^2\right)+0.375ib^4y_1y_2^2\right)\right)\right)}{a^6}$$

$$-\frac{b^2x_1x_2\left(0.375ia^2x_1x_2\left(a^2x_1y_2+a^2x_2y_1+b^2x_1y_2\right)+0.5625b^4x_1x_2y_1y_2+0.5625b^4y_1^2y_2^2\right)}{a^6}$$

A.3 c_4

$$+0.234375x_1^4x_2^2+0.234375x_1^2x_2^4-0.015625x_1^2y_2^4+\frac{x_2^4y_1^2}{64}+\frac{y_2^2\left(x_1^4+15y_1^2\left(-y_1^2+2y_1y_2-y_2^2\right)\right)}{64}$$

$$+0.09375x_1^2x_2^2y_1^2+0.09375x_1^2y_1y_2^3+0.09375x_2^2y_1^3y_2-x_2^2\left(0.46875x_1^3x_2+0.015625y_1^4\right)$$

$$-0.09375x_1^3x_2y_2^2-0.09375x_1^2y_1^2y_2^2-0.09375x_1^2y_2^2\left(ix_1y_2-x_2^2\right)-0.09375x_1x_2^3y_1^2-0.09375ix_2^3y_1^3$$

$$+\frac{8.0a^6x_1x_2^3y_1y_2+b^6x_1^2y_2^4+b^6x_2^2y_1^4+x_2^2\left(-6.0a^6y_1^2y_2^2-15b^6x_1^4+30b^6x_1^3x_2-15.0b^6x_1^2x_2^2\right)}{64a^6}$$

$$+\frac{0.046875a^2b^2x_1^4x_2^2+0.046875a^2b^2x_1^2x_2^4+y_1y_2\left(0.125a^4x_1^3x_2+0.046875b^4y_1^3y_2+0.046875b^4y_1y_2^3\right)}{a^4}$$

$$+\frac{0.1875ia^4x_1^3y_1y_2^2+0.1875ia^4x_1x_2\left(x_1y_2^3+x_2y_1^3\right)+0.09375a^2b^2y_1^3y_2^3+0.09375b^4x_1^3x_2^3}{a^4}$$

$$+i\left(0.15625x_1^4x_2y_2+0.3125x_1y_1^3y_2^2+0.15625x_2y_1^4y_2+y_1\left(0.15625x_1y_2^4+x_2^3\left(0.15625x_1x_2+0.1875y_1y_2\right)\right)\right)$$

$$+\frac{a^4x_2\left(0.28125x_1y_1^2y_2^2+0.3125i\left(x_1^2x_2^2y_2+y_1\left(x_1^3x_2+y_1y_2^3\right)\right)\right)+0.203125b^4x_1^2y_2^4+0.203125b^4x_2^2y_1^4}{a^4}$$

$$+\frac{b^2\left(0.203125a^4\left(x_1^4y_2^2+x_2^4y_1^2\right)-0.015625b^4x_1^4y_2^2+0.234375b^4y_1^2y_2^4+0.234375b^4y_1^4y_2^2\right)}{a^6}$$

$$-\frac{x_2\left(0.046875a^2b^4x_1^4x_2+0.046875a^2b^4x_1^2x_2^3+y_1\left(0.125a^6x_1y_2\left(y_1^2+y_2^2\right)+0.015625b^6x_2^3y_1\right)\right)}{a^6}$$

$$-\frac{0.28125a^4x_1^2x_2^2y_1y_2+b^2\left(0.09375a^2x_1^3x_2^3+y_1^2y_2^2\left(0.046875a^2\left(y_1^2+y_2^2\right)+0.09375b^2y_1y_2\right)\right)}{a^4}$$

$$-\frac{0.46875ia^4x_1y_1^2y_2^3+0.46875ia^4x_2y_1^3y_2^2+0.203125b^2\left(a^2x_1^2y_2^4+a^2x_2^2y_1^4+b^2\left(x_1^4y_2^2+x_2^4y_1^2\right)\right)}{a^4}$$

$$+\frac{-0.46875ia^6x_1^2x_2^2\left(x_1y_2+x_2y_1\right)+0.09375ib^6x_1^3y_2^3+0.09375ib^6x_2^3y_1^3-0.46875b^6y_1^3y_2^3}{a^6}$$

$$+\frac{b^4\left(0.21875ia^2x_1^3y_2^3+0.21875ia^2x_2^3y_1^3+0.09375b^2\left(x_1\left(x_1y_1^2y_2^2+x_2\left(x_1^2y_2^2+x_2^2y_1^2\right)\right)+x_2^2y_1^2y_2^2\right)\right)}{a^6}$$

$$+\frac{0.21875b^2\left(a^2x_2^2y_1^3y_2+b^2x_1^2y_1^2y_2^2+b^2x_2^2y_1^2y_2^2+x_1\left(a^2x_1y_1y_2^3+b^2x_2\left(x_1^2y_2^2+x_2^2y_1^2\right)\right)\right)}{a^4}$$

$$+\frac{x_1\left(a^4x_2\left(0.5625ia^2x_1y_1^2y_2+x_2\left(0.5625ia^2y_1y_2^2+0.21875b^2x_1\left(y_1^2+y_2^2\right)\right)\right)-0.09375b^6x_1y_1y_2^3\right)}{a^6}$$

$$-\frac{b^2\left(0.21875ia^4x_1^3y_2^3+0.21875ia^4x_2^3y_1^3+0.09375b^4x_2^2\left(x_1^2y_2^2+y_1^2\left(x_1^2+y_1y_2\right)\right)\right)}{a^6}$$

$$-\frac{0.21875b^2\left(b^2x_1^2x_2^2y_1^2+b^2x_1^2x_2^2y_2^2+b^2x_2^2y_1^3y_2+x_1\left(a^2x_2\left(x_1^2y_2^2+x_2^2y_1^2\right)+b^2x_1y_1y_2^3\right)\right)}{a^4}$$

$$+\frac{y_1\left(0.03125ib^2x_1x_2^4-y_2\left(0.84375ia^2x_1^2x_2y_2+y_1\left(0.84375ia^2x_1x_2^2+0.21875b^2y_2\left(x_1^2+x_2^2\right)\right)\right)\right)}{a^2}$$

$$+\frac{ib^2\left(0.0625x_1^3x_2^2y_1+y_2\left(0.0625x_1y_1^3y_2+0.03125x_1\left(x_1^3x_2+y_1y_2^3\right)+0.03125x_2y_1^4+0.0625x_2y_1^2y_2^2\right)\right)}{a^2}$$

$$+\frac{b^2x_1y_2\left(0.09375ia^2b^2y_1^2y_2^2+x_2\left(0.0625ia^4x_1x_2^2+0.125b^4y_1^3+0.125b^4y_1y_2^2\right)\right)}{a^6}$$

$$+\frac{b^2x_2\left(0.375a^2x_1y_1^3y_2+0.375a^2x_1y_1y_2^3+0.375b^2x_1x_2^2y_1y_2+0.09375ib^2\left(x_1^3x_2y_2+y_1\left(x_1^2x_2^2+y_1^2y_2^2\right)\right)\right)}{a^4}$$

$$+\frac{b^2x_1\left(0.4375ia^2x_1^2y_1y_2^2+x_2\left(0.4375ia^2x_1y_2^3+y_1\left(0.4375ia^2x_2y_1^2+0.375b^2x_1^2y_2\right)\right)\right)}{a^4}$$

$$+\frac{b^2x_2y_1y_2\left(0.34375a^4x_1^2x_2+0.34375a^2b^2x_1y_1y_2+x_2\left(0.4375ia^4x_2y_1+0.28125b^4x_1^2\right)\right)}{a^6}$$

$$+\frac{ib^4\left(-0.03125a^2x_1x_2^4y_1-0.03125a^2x_1y_1y_2^4+0.46875b^2\left(x_1^3x_2^2y_2+y_1\left(x_1^2x_2^3+y_1y_2^2\left(x_1y_2+x_2y_1\right)\right)\right)\right)}{a^6}$$

$$-\frac{ib^4\left(0.0625x_1^3x_2^2y_1+0.0625x_1^2x_2^3y_2+y_2\left(0.0625x_1y_1^3y_2+0.0625x_2y_1^2y_2^2+0.03125x_2\left(x_1^4+y_1^4\right)\right)\right)}{a^4}$$

$$-\frac{b^2y_1\left(0.09375ia^4x_1^2x_2^3+y_2\left(0.09375ia^4x_2y_1^2y_2+x_1\left(0.09375ia^4y_1y_2^2+0.125b^4x_2\left(x_1^2+x_2^2\right)\right)\right)\right)}{a^6}$$

$$-\frac{ib^2x_1\left(0.1875b^4x_1^2y_1y_2^2+x_2\left(0.1875b^4x_1y_2^3+x_2\left(0.09375a^4x_1^2y_2+0.1875b^4y_1^3\right)\right)\right)}{a^6}$$

$$-\frac{b^2x_2y_1y_2\left(0.375a^4x_1^3+0.375a^4x_1x_2^2+b^2\left(0.375a^2x_1y_1^2+0.375a^2x_1y_2^2+0.1875ib^2x_2^2y_1\right)\right)}{a^6}$$

$$-\frac{ib^6\left(0.3125x_1y_1^3y_2^2+0.15625x_1\left(x_1^3x_2y_2+y_1\left(x_2^4+y_2^4\right)\right)+0.15625x_2y_1^4y_2+0.3125x_2y_1^2y_2^3\right)}{a^6}$$

$$-\frac{ib^4x_1\left(0.4375a^2x_1^2y_1y_2^2+x_2\left(0.4375a^2x_1y_2^3+x_2\left(0.4375a^2y_1^3+0.3125b^2x_1\left(x_1y_1+x_2y_2\right)\right)\right)\right)}{a^6}$$

$$-\frac{b^2x_2y_1y_2\left(0.34375a^2b^2x_1^2x_2+y_1\left(0.34375a^4x_1y_2+b^2\left(0.4375ia^2x_2^2+0.28125b^2x_1y_2\right)\right)\right)}{a^6}$$

$$+\frac{ib^2x_1x_2y_1y_2\left(a^2\left(0.03125a^2\left(x_1y_2+x_2y_1\right)+0.6875b^2x_1y_1+0.6875b^2x_2y_2\right)+0.84375b^4x_2y_1\right)}{a^6}$$

$$+\frac{ib^4x_1x_2y_1y_2\left(-0.03125a^2x_1y_2-0.03125a^2x_2y_1-0.5625b^2x_1y_1+0.84375b^2x_1y_2-0.5625b^2x_2y_2\right)}{a^6}$$

$$-\frac{0.6875ib^2x_1x_2y_1y_2\left(x_1y_1+x_2y_2\right)}{a^2}$$

A.4 c_6

$$-0.0625x_1^4y_2^2 + 0.625x_1^3x_2^3 - 0.0625x_1^2y_2^4 - 0.0625x_2^4y_1^2 - 0.0625x_2^2y_1^4 - \frac{5y_1^2y_2^2\left(y_1^2 - 2y_1y_2 + y_2^2\right)}{16}$$

$$+ x_1\left(0.375x_1y_1y_2^3 + x_2\left(0.375x_1^2y_2^2 - x_2\left(0.3125x_1\left(x_1^2 + x_2^2\right) - 0.375x_2y_1^2\right)\right)\right) + 0.375x_2^2y_1^3y_2$$

$$+ \frac{-6.0a^6\left(x_1^2\left(x_2^2\left(y_1^2 + y_2^2\right) + y_1^2y_2^2\right) + x_2^2y_1^2y_2^2\right) + 64.0a^4b^4x_1^2y_2^2 - 5b^6x_1^4x_2^2 + 10b^6x_1^3x_2^3 - 5b^6x_1^2x_2^4}{16a^6}$$

$$+ \frac{b^2\left(a^2\left(0.375a^2x_1^3x_2^3 + 0.375a^2y_1^3y_2^3 + b^2\left(x_2^2\left(4.0a^2y_1^2 + 0.375x_1^3x_2\right) + 0.375y_1^3y_2^3\right)\right) + 0.625b^4y_1^3y_2^3\right)}{a^6}$$

$$- \frac{0.0625b^6x_2^4y_1^2 + 0.0625b^6x_2^2y_1^4 + x_1y_2\left(-1.125a^6x_2y_1\left(x_1x_2 + y_1y_2\right) + 0.0625b^6x_1^3y_2 + 0.0625b^6x_1y_2^3\right)}{a^6}$$

$$- \frac{0.1875b^4y_1^2y_2^4 + x_1x_2\left(0.5a^4y_1y_2\left(x_1^2 + x_2^2 + y_1^2 + y_2^2\right) + 0.1875b^4x_1^3x_2 + 0.1875b^4x_1x_2^3\right)}{a^4}$$

$$- \frac{b^2\left(0.1875a^2\left(a^2x_1^4x_2^2 + a^2x_1^2x_2^4 + a^2y_1^4y_2^2 + a^2y_1^2y_2^4 + b^2y_1^4y_2^2\right) + 0.3125b^4y_1^4y_2^2 + 0.3125b^4y_1^2y_2^4\right)}{a^6}$$

$$- \frac{0.4375b^2\left(a^2x_1^4y_2^2 + a^2x_1^2y_2^4 + a^2x_2^4y_1^2 + a^2x_2^2y_1^4 + b^2\left(x_1^2y_2^2\left(x_1^2 + y_2^2\right) + x_2^4y_1^2 + x_2^2y_1^4\right)\right)}{a^4}$$

$$+ \frac{b^2\left(0.625a^4x_1x_2^3y_1^2 + b^2\left(0.625a^2x_1x_2^3y_1^2 + 0.375b^2\left(x_1\left(x_1y_1y_2^3 + x_2\left(x_1^2y_2^2 + x_2^2y_1^2\right)\right) + x_2^2y_1^3y_2\right)\right)\right)}{a^6}$$

$$+ \frac{b^2\left(0.625a^2y_2\left(a^2x_2^2y_1^3 + b^2x_2^2y_1^3 + x_1^2y_2\left(a^2y_1y_2 + b^2y_1y_2 + x_1x_2\left(a^2 + b^2\right)\right)\right) - 0.375b^4x_1^2x_2^2y_1^2\right)}{a^6}$$

$$- \frac{b^4\left(0.625a^2x_1^2x_2^2y_1^2 + 0.625a^2x_1^2x_2^2y_2^2 + 0.625a^2x_1^2y_1^2y_2^2 + 0.375b^2y_2^2\left(x_1^2\left(x_2^2 + y_1^2\right) + x_2^2y_1^2\right)\right)}{a^6}$$

$$+ \frac{b^2\left(-0.625a^2x_1^2y_1^2y_2^2 - 0.625a^2x_2^2y_1^2y_2^2 + 0.5b^2x_1x_2y_1y_2^3 - 0.625x_2^2\left(a^2x_1^2y_2^2 + y_1^2\left(a^2x_1^2 + b^2y_2^2\right)\right)\right)}{a^4}$$

$$+ \frac{b^2x_1x_2y_1y_2\left(0.5a^2\left(a^2x_1^2 + a^2x_2^2 + a^2y_1^2 + a^2y_2^2 + b^2x_1^2 + b^2x_2^2 + b^2y_1^2\right) + 1.125b^4x_1x_2 + 1.125b^4y_1y_2\right)}{a^6}$$

$$- \frac{b^2x_1x_2y_1y_2\left(0.125a^2\left(a^2x_1x_2 + b^2x_1x_2 + y_1y_2\left(a^2 + b^2\right)\right) + 0.5b^4x_1^2 + 0.5b^4x_2^2 + 0.5b^4y_1^2 + 0.5b^4y_2^2\right)}{a^6}$$

$$- \frac{8.0b^4x_1x_2y_1y_2}{a^2}$$

A.5 c_8

$$+ 0.234375x_1^4x_2^2 + 0.234375x_1^2x_2^4 - 0.015625x_1^2y_2^4 + \frac{x_2^4y_1^2}{64} + \frac{y_2^2\left(x_1^4 + 15y_1^2\left(-y_1^2 + 2y_1y_2 - y_2^2\right)\right)}{64}$$

$$+ 0.09375ix_1^3y_2^3 + 0.09375x_1^2y_1y_2^3 + 0.09375ix_2^3y_1^3 + 0.09375x_2^2y_1^3y_2 - x_2^2\left(0.46875x_1^3x_2 + 0.015625y_1^4\right)$$

$$-\frac{6.0a^6\left(x_1\left(x_1y_1^2y_2^2+x_2\left(x_1^2y_2^2-x_2\left(x_1\left(y_1^2+y_2^2\right)-x_2y_1^2\right)\right)\right)+x_2^2y_1^2y_2^2\right)+15b^6x_1^4x_2^2+15b^6x_1^2x_2^4}{64a^6}$$

$$+\frac{4.0a^6x_1^3x_2y_1y_2+4.0a^6x_1x_2^3y_1y_2+1.5a^2b^4y_1^2y_2^4+b^6\left(x_1^2\left(15x_1x_2^3+0.5y_2^4\right)+0.5x_2^2y_1^4\right)}{32a^6}$$

$$+\frac{b^2\left(0.046875a^2x_1^4x_2^2+0.046875a^2x_1^2x_2^4+0.09375a^2y_1^3y_2^3+0.09375b^2x_1^3x_2^3+0.046875b^2y_1^4y_2^2\right)}{a^4}$$

$$+\frac{0.28125a^4x_1x_2y_1^2y_2^2+0.203125a^2b^2x_1^4y_2^2+0.203125b^4x_1^2y_2^4+0.203125b^4x_2^2y_1^4}{a^4}$$

$$+\frac{y_1^2\left(0.46875ia^6x_1y_2^3+0.46875ia^6x_2y_1y_2^2+b^2\left(0.203125a^4x_2^4+0.234375b^4y_1^2y_2^2+0.234375b^4y_2^4\right)\right)}{a^6}$$

$$+\frac{-0.125a^6x_1x_2y_1y_2^3-0.015625b^6x_2^4y_1^2+x_1^2\left(0.46875ia^6x_2^2\left(x_1y_2+x_2y_1\right)-0.015625b^6x_1^2y_2^2\right)}{a^6}$$

$$-\frac{0.046875a^2b^2y_1^4y_2^2+0.046875a^2b^2y_1^2y_2^4+x_1x_2\left(0.125a^4y_1^3y_2+0.046875b^4x_1^3x_2+0.046875b^4x_1x_2^3\right)}{a^4}$$

$$-\frac{0.1875ia^4x_1^3y_1y_2^2+0.1875ia^4x_1x_2\left(x_1y_2^3+x_2y_1^3\right)+0.09375a^2b^2x_1^3x_2^3+0.09375b^4y_1^3y_2^3}{a^4}$$

$$-i\left(0.15625x_1^4x_2y_2+0.15625x_2y_1^4y_2+y_1\left(0.15625x_1y_2^4+x_2^3\left(0.15625x_1x_2+0.1875y_1y_2\right)\right)\right)$$

$$-\frac{a^4\left(0.28125x_1^2x_2^2y_1y_2+0.3125i\left(x_1^2x_2^3y_2+y_1\left(x_1^3x_2^2+y_1y_2^2\left(x_1y_1+x_2y_2\right)\right)\right)\right)+0.203125b^4x_1^4y_2^2}{a^4}$$

$$+\frac{b^2\left(-0.203125a^2\left(a^2x_1^2y_2^4+a^2x_2^2y_1^4+b^2x_2^4y_1^2\right)+0.09375b^4x_1x_2^3y_1^2-0.46875b^4y_1^3y_2^3\right)}{a^6}$$

$$+\frac{b^2\left(0.21875ia^4x_2^3y_1^3+y_2^2\left(0.21875ia^4x_1^3y_2+0.09375b^4\left(x_1^2\left(x_1x_2+y_1^2\right)+x_2^2y_1^2\right)\right)\right)}{a^6}$$

$$+\frac{0.21875b^2\left(a^2x_2^2y_1^3y_2+b^2x_1^4y_1^2y_2^2+b^2x_2^2y_1^2y_2^2+x_1\left(a^2x_1y_1y_2^3+b^2x_2\left(x_1^2y_2^2+x_2^2y_1^2\right)\right)\right)}{a^4}$$

$$+\frac{x_1\left(a^4x_2\left(0.84375ia^2x_1y_1y_2^2+x_2\left(0.84375ia^2y_1^2y_2+0.21875b^2x_1\left(y_1^2+y_2^2\right)\right)\right)-0.09375ib^6x_1^2y_2^3\right)}{a^6}$$

$$-\frac{b^4\left(0.21875ia^2x_1^3y_2^3+0.09375b^2\left(x_1^2x_2^2y_2^2+y_1\left(x_1^2x_2^2y_1+x_1^4y_2^3+ix_2^3y_1^2+x_2^2y_1^2y_2\right)\right)\right)}{a^6}$$

$$-\frac{0.21875b^2\left(b^2x_1^2x_2^2y_1^2+b^2x_1^2x_2^2y_2^2+b^2x_1^4y_1y_2^3+b^2x_2^2y_1^3y_2+x_2\left(a^2x_1^3y_2^2+x_2^2y_1^2\left(a^2x_1+ib^2y_1\right)\right)\right)}{a^4}$$

$$+\frac{y_1\left(-a^2y_2\left(0.5625ia^2x_1^2x_2y_1+y_2\left(0.5625ia^2x_1x_2^2+0.21875b^2y_1\left(x_1^2+x_2^2\right)\right)\right)+0.03125ib^4x_1x_2^4\right)}{a^4}$$

$$+\frac{ib^4\left(0.0625x_1^3x_2^2y_1+y_2\left(0.0625x_1y_1^3y_2+0.03125x_1\left(x_1^3x_2+y_1y_2^3\right)+0.03125x_2y_1^4+0.0625x_2y_1^2y_2^2\right)\right)}{a^4}$$

$$+\frac{b^2y_2\left(0.09375ia^4x_2y_1^3y_2+x_1\left(0.09375ia^4y_1^2y_2^2+b^2x_2\left(0.0625ia^2x_1x_2^2+0.125b^2y_1^3+0.125b^2y_1y_2^2\right)\right)\right)}{a^6}$$

$$+\frac{ib^2x_1\left(0.1875b^4x_1^2y_1y_2^2+x_2\left(0.1875b^4x_1y_2^3+x_2\left(0.09375a^4x_1\left(x_1y_2+x_2y_1\right)+0.1875b^4y_1^3\right)\right)\right)}{a^6}$$

$$+\frac{b^2x_2y_1y_2\left(0.375a^4x_1y_1^2+0.375a^4x_1y_2^2+0.375a^2b^2x_1^3+0.375a^2b^2x_1x_2^2+0.1875ib^4x_2^2y_1\right)}{a^6}$$

$$+\frac{ib^6\left(0.3125x_1y_1^3y_2^2+0.15625x_1\left(x_1^3x_2y_2+y_1\left(x_2^4+y_2^4\right)\right)+0.15625x_2y_1^4y_2+0.3125x_2y_1^2y_2^3\right)}{a^6}$$

$$+\frac{ib^4x_1\left(0.4375a^2x_1^2y_1y_2^2+x_2\left(0.4375a^2x_1y_2^3+x_2\left(0.4375a^2y_1^3+0.3125b^2x_1\left(x_1y_1+x_2y_2\right)\right)\right)\right)}{a^6}$$

$$+\frac{b^2x_2y_1y_2\left(0.34375a^4x_1^2x_2+b^2\left(0.34375a^2x_1y_1y_2+x_2\left(0.4375ia^2x_2y_1+0.28125b^2x_1^2\right)\right)\right)}{a^6}$$

$$-\frac{ib^2\left(0.0625x_1y_1^3y_2^2+0.03125x_1\left(x_1^3x_2y_2+y_1\left(x_2^4+y_2^4\right)\right)+0.03125x_2y_1^4y_2+0.0625x_2y_1^2y_2^3\right)}{a^2}$$

$$-\frac{b^2x_1\left(0.09375ia^2b^2y_1^2y_2^3+x_2\left(0.125b^4x_1^2y_1y_2+x_2\left(0.0625ia^4x_1\left(x_1y_1+x_2y_2\right)+0.125b^4x_2y_1y_2\right)\right)\right)}{a^6}$$

$$-\frac{b^2x_2\left(0.375a^2x_1x_2^2y_1y_2+b^2\left(0.375x_1y_1^3y_2+0.375x_1y_1y_2^3+0.09375i\left(x_1^3x_2y_2+y_1\left(x_1^2x_2^2+y_1^2y_2^2\right)\right)\right)\right)}{a^4}$$

$$-\frac{b^2\left(x_1\left(0.4375ix_1^2y_1y_2^2+x_2\left(0.4375ix_1y_2^3+y_1\left(0.375x_1^2y_2+0.4375ix_2y_1^2\right)\right)\right)+0.4375ix_2^3y_1^2y_2\right)}{a^2}$$

$$-\frac{b^2x_1y_1y_2\left(0.46875ib^4y_1y_2^2+x_2\left(0.34375a^2b^2x_1x_2+y_1y_2\left(0.34375a^4+0.28125b^4\right)\right)\right)}{a^6}$$

$$+\frac{ib^4x_2\left(0.03125a^2x_1^2y_1y_2^2+0.03125a^2x_1x_2y_1^2y_2-0.46875b^2\left(x_1^3x_2y_2+y_1\left(x_1^2x_2^2+y_1^2y_2^2\right)\right)\right)}{a^6}$$

$$+\frac{ib^2x_1x_2y_1y_2\left(0.6875a^4x_1y_1-0.03125a^4x_1y_2-0.03125a^4x_2y_1+0.6875a^4x_2y_2+0.5625b^4\left(x_1y_1+x_2y_2\right)\right)}{a^6}$$

$$-\frac{ib^4x_1x_2y_1y_2\left(0.6875a^2\left(x_1y_1+x_2y_2\right)+0.84375b^2x_1y_2+0.84375b^2x_2y_1\right)}{a^6}$$

A.6 c_{10}

$$+0.1875x_1^3x_2^3-0.09375x_1^2x_2^4+\frac{x_2^4y_1^2}{32}+\frac{x_2^2y_1^4}{32}+\frac{y_2^2\left(x_1^4+x_1^2y_2^2+3y_1^2\left(-y_1^2+2y_1y_2-y_2^2\right)\right)}{32}$$

$$+x_1^2\left(x_2^2\left(-0.09375x_1^2+0.1875y_1^2+0.1875y_2^2\right)+0.1875y_1^2y_2^2\right)-0.1875x_1x_2^3y_1^2+0.1875x_2^2y_1^2y_2^2$$

$$+\frac{-6.0a^6y_2\left(x_1^2y_2\left(x_1x_2+y_1y_2\right)+x_2^2y_1^3\right)-3b^6x_1^4x_2^2+b^6x_1^4y_2^2+6b^6x_1^3x_2^3-3b^6x_1^2x_2^4+b^6x_1^2y_2^4}{32a^6}$$

$$+\frac{x_2\left(0.125ia^6x_1^4y_2+0.25ia^6x_1^3x_2y_1+0.25ia^6x_1^2x_2^2y_2+x_2y_1\left(0.125ia^6x_1x_2^2+0.03125b^6y_1\left(x_2^2+y_1^2\right)\right)\right)}{a^6}$$

$$+\frac{0.09375b^4y_1^2y_2^4+x_1x_2\left(0.25a^4y_1y_2\left(x_1^2+x_2^2+y_1^2+y_2^2\right)+0.09375b^4x_1^3x_2+0.09375b^4x_1x_2^3\right)}{a^4}$$

$$+\frac{b^2\left(0.09375a^2\left(a^2x_1^4x_2^2+a^2x_1^2x_2^4+a^2y_1^4y_2^2+a^2y_1^2y_2^4+b^2y_1^4y_2^2\right)+0.1875b^4y_1^3y_2^3\right)}{a^6}$$

$$+\frac{-0.03125b^4x_2^4y_1^2-0.03125b^4x_2^2y_1^4+y_2^2\left(0.375ia^4y_1^2\left(x_1y_2+x_2y_1\right)-0.03125b^4x_1^4-0.03125b^4x_1^2y_2^2\right)}{a^4}$$

$$-\frac{0.25ia^2x_1y_1^3y_2^2+0.125ia^2x_1y_1y_2^4+0.125ia^2x_2y_1^4y_2+0.03125b^2\left(x_1^2y_2^2\left(x_1^2+y_2^2\right)+x_2^4y_1^2+x_2^2y_1^4\right)}{a^2}$$

$$-\frac{0.1875a^2b^4x_1^3x_2^3 + 0.1875a^2b^4y_1^3y_2^3 + y_1^2y_2^2\left(0.09375b^6y_1^2 + y_2\left(0.25ia^6x_2 + 0.09375b^6y_2\right)\right)}{a^6}$$

$$-0.375ix_1^3x_2^2y_2 - 0.375ix_1^2x_2^3y_1 - 0.5625x_1x_2y_1^2y_2^2 - \frac{0.1875b^2x_1^3x_2^3}{a^2} - \frac{0.1875b^2y_1^3y_2^3}{a^2}$$

$$+\frac{x_1x_2\left(0.1875a^2b^2x_1^2y_2^2 + 0.1875b^4x_1^2y_2^2 + x_2y_1\left(-0.5625a^4x_1y_2 + 0.1875a^2b^2x_2y_1 + 0.1875b^4x_2y_1\right)\right)}{a^4}$$

$$+\frac{0.1875b^2\left(b^4x_1^2x_2^2y_2^2 + b^4x_1^2y_1^2y_2^2 + y_1\left(a^2y_2\left(a^2x_2^2y_1^2 + b^2x_2^2y_1^2 + x_1^2y_2^2\left(a^2 + b^2\right)\right) + b^4x_1^2x_2^2y_1\right)\right)}{a^6}$$

$$-\frac{0.1875b^4\left(a^2x_1^2x_2^2y_1^2 + a^2x_1^2x_2^2y_2^2 + b^2\left(x_1^2y_1y_2^3 + x_2^2y_1^3y_2 + x_2\left(x_1^3y_2^2 + x_2y_1^2\left(x_1x_2 - y_2^2\right)\right)\right)\right)}{a^6}$$

$$-\frac{0.1875b^2\left(a^2x_1^2x_2^2y_2^2 + a^2x_1^2y_1^2y_2^2 + a^2x_2^2y_1^2y_2^2 + y_1^2\left(a^2x_1^2x_2^2 + b^2y_2^2\left(x_1^2 + x_2^2\right)\right)\right)}{a^4}$$

$$+\frac{0.125ib^2\left(a^4x_2y_1^4y_2 + a^2b^2x_2y_1^4y_2 + x_1\left(b^4x_1^3x_2y_2 + y_1\left(a^4y_2^4 + b^2\left(a^2y_2^4 + b^2x_2^4\right)\right)\right)\right)}{a^6}$$

$$+\frac{0.25b^2\left(b^4x_1x_2y_1y_2^3 + i\left(b^4x_1^2x_2^3y_2 + y_1\left(a^2y_1y_2^2\left(a^2x_2y_2 + b^2x_2y_2 + x_1y_1\left(a^2 + b^2\right)\right) + b^4x_1^3x_2^2\right)\right)\right)}{a^6}$$

$$+\frac{b^4y_1\left(0.375ia^2x_1^2x_2^3 + b^2y_2\left(x_1\left(0.25x_2\left(x_1^2 + x_2^2 + y_1^2\right) + 0.375iy_1y_2^2\right) + 0.375ix_2y_1^2y_2\right)\right)}{a^6}$$

$$+\frac{b^2x_1x_2\left(0.5625a^2y_1^2y_2^2 + 0.5625b^2x_1x_2y_1y_2 + 0.5625b^2y_1^2y_2^2 + 0.375ix_1x_2\left(a^2x_1y_2 + a^2x_2y_1 + b^2x_1y_2\right)\right)}{a^4}$$

$$-\frac{b^2y_1\left(0.125ib^4x_2y_1^3y_2 + x_1\left(0.125ia^4x_2^4 + 0.125ia^2b^2x_2^4 - y_2\left(0.5625a^4x_1x_2^2 - 0.125ib^4y_2^3\right)\right)\right)}{a^6}$$

$$-\frac{ib^2\left(0.25a^2b^2x_1^3x_2^2y_1 + y_2\left(0.25b^4x_2y_1^2y_2^2 + x_1\left(0.125a^2x_1^3x_2\left(a^2 + b^2\right) + 0.25b^4y_1^3y_2\right)\right)\right)}{a^6}$$

$$-\frac{0.25b^2x_1x_2\left(a^2y_1^3y_2 + a^2y_1y_2^3 + b^2x_2^2y_1y_2 + b^2y_1^3y_2 + b^2y_1y_2^3 + ix_1x_2\left(a^2x_1y_1 + a^2x_2y_2 + b^2x_2y_2\right)\right)}{a^4}$$

$$-\frac{b^2y_1y_2\left(0.375ib^2x_2y_1^2y_2 + x_1\left(0.375ia^2y_1y_2^2 + 0.375ib^2y_1y_2^2 + 0.25x_2\left(a^2x_1^2 + a^2x_2^2 + b^2x_1^2\right)\right)\right)}{a^4}$$

$$-\frac{b^2x_2\left(0.5625b^4x_1^2x_2y_1y_2 + 0.5625b^4x_1y_1^2y_2^2 + 0.375i\left(b^4x_1^3x_2y_2 + y_1\left(a^4y_1^2y_2^2 + b^4x_1^2x_2^2\right)\right)\right)}{a^6}$$

A.7 c_{12}

$$+\frac{x_1^4x_2^2}{64} - 0.015625x_1^4y_2^2 + \frac{x_1^2x_2^4}{64} + \frac{x_1^2y_2^4}{64} - 0.015625x_2^4y_1^2 + \frac{x_2^2y_1^4}{64} + \frac{y_1^2y_2^2\left(-y_1^2 + 2y_1y_2 - y_2^2\right)}{64}$$

$$+x_1\left(0.09375x_1y_1^2y_2^2 + x_2\left(0.09375x_1^2y_2^2 + x_2\left(-0.03125x_1^2 + 0.09375y_1^2\right)\right)\right) + 0.09375x_2^2y_1^2y_2^2$$

$$-\frac{a^6\left(6.0x_1^2x_2^2y_1^2 + 6.0x_1^2x_2^2y_2^2 + 6.0x_1^2y_1y_2^3 + 6.0x_2^2y_1^3y_2 + 2.0i\left(x_1^3y_2^2 + x_2^3y_1^3\right)\right) + b^6x_1^4x_2^4}{64a^6}$$

$$+\frac{4.0ia^6x_1^2x_2y_2^3 + 4.0ia^6x_1x_2^2y_1^3 + b^6\left(x_1^3\left(x_1y_2^2 + x_2^2\left(-x_1 + 2x_2\right)\right) + x_2^4y_1^2 + y_1^4y_2^2 + y_1^2y_2^4\right)}{64a^6}$$

$$+\frac{a^4 y_1 y_2 \left(0.125 x_1 x_2 y_1^2 + 0.125 x_1 x_2 y_2^2 + 0.0625i \left(x_1^3 y_2 + x_2^3 y_1\right)\right) + 0.046875 b^4 x_1^2 x_2^4 + 0.046875 b^4 x_1^2 y_2^4}{a^4}$$

$$+\frac{0.09375 i a^4 x_1 y_1^2 y_2^3 + 0.046875 b^2 \left(a^2 x_1^4 y_2^2 + a^2 x_2^4 y_1^2 + a^2 y_1^4 y_2^2 + a^2 y_1^2 y_2^4 + b^2 x_2^2 \left(x_1^4 + y_1^4\right)\right)}{a^4}$$

$$+\frac{0.09375 \left(i a^4 x_2 \left(x_1^3 x_2 y_2 + y_1 \left(x_1^2 x_2^2 + y_1^2 y_2^2\right)\right) + a^2 b^2 x_1^3 x_2^3 + b^4 y_1^3 y_2\right)}{a^4}$$

$$+0.28125 x_1^2 x_2^2 y_1 y_2 - 0.03125 i x_1 x_2^4 y_1 - 0.03125 i x_1 y_1 y_2^4 - \frac{0.015625 b^6 x_1^2 y_2^4}{a^6} - \frac{0.015625 b^6 x_2^2 y_1^4}{a^6}$$

$$-0.03125 i x_1^4 x_2 y_2 - 0.0625 i x_1 y_1^3 y_2^2 - 0.03125 i x_2 y_1^4 y_2 - 0.0625 i x_2 y_1^2 y_2^3 - \frac{0.03125 b^6 y_1^3 y_2^3}{a^6}$$

$$-\frac{x_1 \left(a^4 x_2 \left(0.125 x_1^2 y_1 y_2 + x_2 \left(0.0625 i x_1 \left(x_1 y_1 + x_2 y_2\right) + 0.125 x_2 y_1 y_2\right)\right) + 0.046875 b^4 x_1^3 y_2^2\right)}{a^4}$$

$$-\frac{0.046875 b^2 \left(a^2 x_1^4 x_2^2 + a^2 x_1^2 x_2^4 + a^2 x_1^2 y_2^4 + a^2 x_2^2 y_1^4 + b^2 y_1^2 \left(x_2^4 + y_1^2 y_2^2 + y_2^4\right)\right)}{a^4}$$

$$+\frac{-0.28125 a^6 x_1 x_2 y_1^2 y_2^2 - 0.09375 a^4 b^2 y_1^3 y_2^3 - 0.09375 a^2 b^4 x_1^3 x_2^3 + 0.03125 i b^6 \left(x_1^3 y_2^3 + x_2^3 y_1^3\right)}{a^6}$$

$$+\frac{0.09375 b^2 \left(i a^4 \left(x_1^3 y_2^3 + x_2^3 y_1^3\right) + b^4 x_1^2 x_2^2 y_1^2 + b^4 x_1^2 x_2^2 y_2^2 + b^4 x_1^2 y_1 y_2^3 + b^4 x_2^2 y_1^3 y_2\right)}{a^6}$$

$$+\frac{x_1 \left(0.28125 a^2 b^2 x_1 y_1 y_2^3 + x_2 \left(0.28125 b^4 x_1^2 y_2^2 + y_1 \left(0.1875 i a^4 y_2 \left(x_1 y_1 + x_2 y_2\right) + 0.28125 b^4 x_2^2 y_1\right)\right)\right)}{a^4}$$

$$+\frac{b^2 \left(0.28125 a^2 x_1^2 x_2^2 y_2^2 - 0.09375 i b^2 x_1^3 y_2^3 + 0.28125 y_1^2 \left(a^2 x_1^2 x_2^2 + y_2 \left(a^2 x_2^2 y_1 + b^2 x_1^2 y_2 + b^2 x_2^2 y_2\right)\right)\right)}{a^4}$$

$$-\frac{0.28125 i a^6 x_1 x_2 y_1^2 y_2 + 0.09375 b^4 \left(b^2 x_1^2 y_1^2 y_2^2 + b^2 x_2^2 y_1^2 y_2^2 + x_2 \left(b^2 x_1^3 y_2^2 + x_2^2 y_1^2 \left(i a^2 y_1 + b^2 x_1\right)\right)\right)}{a^6}$$

$$-\frac{0.28125 b^4 x_2^2 y_1^3 y_2 + 0.28125 x_1 \left(a^2 x_2 \left(b^2 x_1^2 y_2^2 + y_1 \left(i a^2 x_1 y_2^2 + b^2 x_2^2 y_1\right)\right) + b^4 x_1 y_1 y_2^3\right)}{a^4}$$

$$+\frac{b^2 \left(-0.28125 a^2 \left(a^2 x_2^2 y_1^2 y_2^2 + x_1^2 \left(a^2 y_1^2 y_2^2 + b^2 x_2^2 \left(y_1^2 + y_2^2\right)\right)\right) + 0.03125 i b^4 x_1 x_2^4 y_1\right)}{a^6}$$

$$+\frac{i b^6 \left(0.0625 x_1^3 x_2^2 y_1 + y_2 \left(0.0625 x_1 y_1^3 y_2 + 0.03125 x_1 \left(x_1^3 x_2 + y_1 y_2^3\right) + 0.03125 x_2 y_1^4 + 0.0625 x_2 y_1^2 y_2^2\right)\right)}{a^6}$$

$$+\frac{b^2 x_1 \left(0.09375 i a^4 y_1 y_2^4 + x_2 \left(0.09375 i a^4 x_2^3 y_1 + b^4 y_2 \left(0.125 x_1^2 y_1 + x_2^2 \left(0.0625 i x_1 + 0.125 y_1\right)\right)\right)\right)}{a^6}$$

$$+\frac{i b^2 \left(0.1875 a^2 x_1 y_1^3 y_2^2 + 0.1875 b^2 x_1^2 x_2 y_2^3 + x_2 \left(0.09375 a^2 y_2 \left(x_1^4 + y_1^4\right) + 0.1875 b^2 x_1 x_2 y_1^3\right)\right)}{a^4}$$

$$+\frac{b^2 \left(0.375 b^2 x_1 x_2 y_1 y_2^3 + 0.1875 i \left(a^2 x_1^2 x_2^3 y_2 + y_1 \left(a^2 x_1^3 x_2^2 + b^2 x_2^3 y_1 y_2 + y_2^2 \left(a^2 x_2 y_1 y_2 + b^2 x_1^3\right)\right)\right)\right)}{a^4}$$

$$+\frac{b^2 y_1 y_2 \left(0.28125 i b^2 x_2 y_1^2 y_2 + x_1 \left(0.28125 i b^2 y_1 y_2^2 + 0.375 x_2 \left(a^2 x_1^2 + a^2 x_2^2 + b^2 y_1^2\right)\right)\right)}{a^4}$$

$$+\frac{b^2 x_1 x_2 \left(0.84375 a^4 y_1^2 y_2^2 + 0.84375 a^2 b^2 x_1 x_2 y_1 y_2 + 0.28125 b^2 \left(i a^2 x_1 x_2 \left(x_1 y_2 + x_2 y_1\right) + b^2 y_1^2 y_2^2\right)\right)}{a^6}$$

$$-\frac{b^6 \left(0.125 x_1 x_2 y_1^3 y_2 + 0.125 x_1 x_2 y_1 y_2^3 + 0.0625 i \left(x_1 \left(x_1^2 y_1 y_2^2 + x_2 \left(x_1 y_2^3 + x_2 y_1^3\right)\right) + x_2^3 y_1^2 y_2\right)\right)}{a^6}$$

$$-\frac{0.09375 i b^4 \left(a^2 x_2 y_1^4 y_2 + b^2 x_1^2 x_2^3 y_1 + b^2 x_2 y_1^3 y_2^2 + x_1 \left(a^2 x_1^3 x_2 y_2 + y_1 \left(a^2 \left(x_2^4 + y_2^4\right) + b^2 y_1 y_2^3\right)\right)\right)}{a^6}$$

$$-\frac{i b^2 \left(0.1875 a^2 b^2 x_2 y_1^2 y_2^3 + x_1 \left(0.1875 a^4 x_2^2 y_1^3 + b^2 y_2 \left(0.1875 a^2 y_1^3 y_2 + 0.09375 b^2 x_1^2 x_2^2\right)\right)\right)}{a^6}$$

$$-\frac{b^2 \left(0.375 a^2 x_1 x_2 y_1 y_2^3 + 0.1875 i \left(a^2 x_2^3 y_1^2 y_2 - x_1^2 \left(-a^2 x_1 y_1 y_2^2 - b^2 x_2^3 y_2 - x_2 \left(a^2 y_2^3 + b^2 x_1 x_2 y_1\right)\right)\right)\right)}{a^4}$$

$$-\frac{b^2 y_1 y_2 \left(0.28125 i a^2 x_2 y_1^2 y_2 + x_1 \left(0.28125 i a^2 y_1 y_2^2 + 0.375 x_2 \left(a^2 y_1^2 + b^2 x_1^2 + b^2 x_2^2\right)\right)\right)}{a^4}$$

$$-\frac{b^2 x_1 x_2 \left(0.84375 a^4 x_1 x_2 y_1 y_2 + 0.84375 a^2 b^2 y_1^2 y_2^2 + 0.28125 x_1 x_2 \left(i a^4 \left(x_1 y_2 + x_2 y_1\right) + b^4 y_1 y_2\right)\right)}{a^6}$$

$$+\frac{i b^2 x_1 x_2 y_1 y_2 \left(0.84375 a^4 x_2 y_1 + b^2 \left(0.5625 a^2 x_1 y_1 + 0.5625 a^2 x_2 y_2 + 0.28125 b^2 \left(x_1 y_2 + x_2 y_1\right)\right)\right)}{a^6}$$

$$+\frac{i b^2 x_1 x_2 y_1 y_2 \left(-0.5625 a^4 x_1 y_1 - 0.5625 a^4 x_2 y_2 - 0.1875 b^4 x_1 y_1 + y_2 \left(0.84375 a^4 x_1 - 0.1875 b^4 x_2\right)\right)}{a^6}$$

$$-\frac{0.84375 i b^4 x_1 x_2 y_1 y_2 \left(x_1 y_2 + x_2 y_1\right)}{a^4}$$

TABLES OF RESULTS

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $|S_k|$	Backtracking Tree # nodes	# sol. leaves	CPU Time (s) CLS-MCE	Total
CM1			1	1	2.0	19	124	59	0.00	0.00
CM2	25	3	2	1,2	3.8	21	159	57	0.00	0.00
CM3			3	1,2,3	3.0	19	58	18	0.00	0.00
CM4			1	3	4.2	43	56	50	0.01	0.01
CM5	50	3	2	1,3	8.2	47	237	100	0.01	0.01
CM6			3	1,2,3	10.0	50	329	50	0.01	0.01
CM7			1	3	12.2	101	180	174	0.07	0.07
CM8	100	3	2	2,3	20.0	135	689	348	0.06	0.06
CM9			3	1,2,3	27.0	174	1368	861	0.06	0.07

Table 2 – Solutions of MCE-k for instances CM1-CM9 obtained by **??**.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $|S_k|$	# E3P subproblems	Backtracking Tree # nodes	#sol leaves	CPU Time (s) CLS-MCER	Total
CM1			1	2	2.8	27		91	61	0.08	0.08
CM2	25	3	2	1,2	4.8	24	340	231	98	0.08	0.08
CM3			3	1,2,3	5.0	37		416	148	0.08	0.08
CM4			1	2	5.8	70		307	208	0.48	0.48
CM5	50	3	2	2,3	10.0	93	2028	212	115	0.48	0.48
CM6			3	1,2,3	13.0	115		651	115	0.49	0.49
CM7			1	3	13.2	204		736	730	5.93	5.93
CM8	100	3	2	2,3	22.0	370	18,693	1834	1460	5.99	5.99
CM9			3	1,2,3	28.0	730		13,838	3643	5.91	5.93

Table 3 – Solutions of MCER-k for instances CM1-CM9 obtained by Algorithm 6.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $|S_k|$	Backtracking Tree # nodes	# sol. leaves	CPU Time (s) CLS-MCE	Total
AB001			1	2	1.4	8	34	19	0.00	0.00
AB002	10	3	2	2,3	2.3	9	21	8	0.00	0.00
AB003			3	1,2,3	2.8	10	22	6	0.00	0.00
AB004			1	4	0.9	8	20	10	0.00	0.00
AB005	10	4	2	2,4	1.4	8	53	17	0.00	0.00
AB006			3	2,3,4	1.8	10	28	7	0.00	0.00
AB007			4	1,2,3,4	1.0	10	27	5	0.00	0.00
AB008			1	5	0.9	9	25	10	0.00	0.00
AB009			2	3,5	1.4	10	65	17	0.00	0.00
AB010	10	5	3	3,4,5	1.8	9	39	7	0.00	0.00
AB011			4	2,3,4,5	1.0	10	34	5	0.00	0.00
AB012			5	1,2,3,4,5	−1.5	10	52	7	0.00	0.00
AB013			1	2	1.4	15	62	38	0.00	0.00
AB014	20	3	2	2,3	2.3	18	40	18	0.00	0.00
AB015			3	1,2,3	2.8	20	48	16	0.00	0.00
AB016			1	2	1.5	13	108	52	0.00	0.00
AB017	20	4	2	2,3	2.9	14	137	51	0.00	0.00
AB018			3	2,3,4	3.8	18	50	15	0.00	0.00
AB019			4	1,2,3,4	4.0	20	90	25	0.00	0.00
AB020			1	4	2.4	13	67	36	0.00	0.00
AB021			2	3,4	3.9	11	96	31	0.00	0.00
AB022	20	5	3	3,4,5	4.8	15	88	29	0.00	0.00
AB023			4	2,3,4,5	4.0	16	109	12	0.00	0.00
AB024			5	1,2,3,4,5	2.5	20	217	19	0.00	0.00
AB025			1	1	2.5	17	130	67	0.00	0.00
AB026	30	3	2	1,2	4.9	23	169	70	0.00	0.00
AB027			3	1,2,3	6.8	27	61	22	0.00	0.00
AB028			1	2	2.5	21	148	72	0.00	0.00
AB029	30	4	2	2,3	4.9	22	187	70	0.00	0.00
AB030			3	1,2,3	6.1	22	363	39	0.00	0.00
AB031			4	1,2,3,4	7.0	28	311	19	0.00	0.00
AB032			1	3	2.5	24	139	67	0.01	0.01
AB033			2	3,4	4.9	19	182	68	0.01	0.01
AB034	30	5	3	2,3,4	7.1	17	139	37	0.01	0.01
AB035			4	2,3,4,5	9.0	23	73	17	0.01	0.01
AB036			5	1,2,3,4,5	9.5	27	195	39	0.01	0.01
AB037			1	1	2.5	28	187	95	0.01	0.01
AB038	40	3	2	1,2	4.9	30	243	97	0.00	0.01
AB039			3	1,2,3	6.8	37	122	65	0.00	0.00
AB040			1	1	5.2	25	276	114	0.01	0.01
AB041	40	4	2	1,4	7.1	25	359	97	0.01	0.01
AB042			3	1,2,4	8.6	27	449	87	0.01	0.01
AB043			4	1,2,3,4	10.0	37	285	53	0.01	0.01
AB044			1	3	3.5	26	183	89	0.01	0.01
AB045			2	1,3	7.0	24	406	136	0.01	0.01
AB046	40	5	3	1,2,3	9.2	26	470	100	0.01	0.01
AB047			4	1,2,3,5	11.1	27	696	99	0.01	0.01
AB048			5	1,2,3,4,5	12.5	36	398	60	0.01	0.01
AB049			1	1	5.5	36	226	113	0.01	0.01
AB050	50	3	2	1,2	7.9	35	331	151	0.01	0.01
AB051			3	1,2,3	9.8	42	181	111	0.01	0.01
AB052			1	1	5.2	41	377	150	0.01	0.01
AB053	50	4	2	1,2	8.7	31	709	214	0.01	0.01
AB054			3	1,2,3	11.1	32	786	219	0.01	0.01
AB055			4	1,2,3,4	13.0	46	453	138	0.01	0.01
AB056			1	1	3.5	42	337	120	0.01	0.01
AB057			2	1,4	6.9	42	798	211	0.01	0.01
AB058	50	5	3	1,3,4	9.4	36	1431	165	0.01	0.01
AB059			4	1,2,3,4	11.6	34	1286	89	0.01	0.01
AB060			5	1,2,3,4,5	13.5	44	617	28	0.02	0.02

Table 4 – Solutions of MCE-k for instances AB001-AB060 obtained by **??**.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $|S_k|$	Backtracking Tree # nodes	#sol. leaves	CPU Time (s) CLS-MCE	Total
AB061			1	1	3.5	38	256	133	0.01	0.01
AB062	60	3	2	1,2	5.9	41	339	141	0.01	0.01
AB063			3	1,2,3	7.8	54	126	48	0.01	0.01
AB064			1	1	5.2	51	475	185	0.02	0.02
AB065	60	4	2	1,2	8.7	44	693	207	0.01	0.01
AB066			3	1,2,3	12.1	39	747	194	0.01	0.01
AB067			4	1,2,3,4	14.0	51	277	77	0.01	0.01
AB068			1	3	4.5	62	279	137	0.02	0.02
AB069			2	1,3	9.0	59	1116	251	0.02	0.02
AB070	60	5	3	1,3,4	12.4	42	1860	283	0.02	0.02
AB071			4	1,2,3,4	14.6	43	2489	231	0.02	0.02
AB072			5	1,2,3,4,5	16.5	52	2453	222	0.02	0.02
AB073			1	1	4.5	54	323	161	0.01	0.01
AB074	70	3	2	1,2	7.9	48	414	155	0.01	0.01
AB075			3	1,2,3	9.8	59	151	51	0.01	0.01
AB076			1	1	5.2	55	505	201	0.02	0.02
AB077	70	4	2	1,2	9.7	43	657	191	0.02	0.02
AB078			3	1,2,3	13.1	43	598	136	0.02	0.02
AB079			4	1,2,3,4	16.0	60	177	46	0.02	0.02
AB080			1	1	5.5	68	849	266	0.03	0.03
AB081			2	1,3	10.0	53	1341	368	0.02	0.02
AB082	70	5	3	1,2,3	14.2	47	1452	298	0.02	0.02
AB083			4	1,2,3,4	17.6	43	2057	325	0.02	0.02
AB084			5	1,2,3,4,5	19.5	55	1663	73	0.02	0.03
AB085			1	1	4.5	66	363	178	0.01	0.01
AB086	80	3	2	1,2	7.9	47	458	165	0.01	0.01
AB087			3	1,2,3	10.8	65	167	56	0.01	0.01
AB088			1	1	7.2	83	680	260	0.02	0.02
AB089	80	4	2	1,2	12.7	52	955	281	0.02	0.02
AB090			3	1,2,3	16.1	57	975	287	0.02	0.02
AB091			4	1,2,3,4	18.0	68	235	54	0.02	0.02
AB092			1	2	6.2	90	714	275	0.03	0.03
AB093			2	2,3	10.7	77	1242	358	0.03	0.03
AB094	80	5	3	1,2,3	15.2	69	3169	545	0.03	0.03
AB095			4	1,2,3,4	18.6	55	3359	316	0.03	0.03
AB096			5	1,2,3,4,5	19.5	74	2051	109	0.03	0.03
AB097			1	1	5.5	77	439	216	0.02	0.02
AB098	90	3	2	1,2	9.9	63	561	203	0.02	0.02
AB099			3	1,2,3	11.8	76	205	67	0.02	0.02
AB100			1	1	6.2	87	757	292	0.02	0.02
AB101	90	4	2	1,2	10.7	68	1424	395	0.02	0.02
AB102			3	1,2,3	14.1	58	1030	260	0.02	0.02
AB103			4	1,2,3,4	17.0	79	267	65	0.02	0.02
AB104			1	2	8.2	130	770	287	0.04	0.04
AB105			2	2,3	12.7	96	1522	365	0.04	0.04
AB106	90	5	3	1,2,3	16.2	61	9612	352	0.04	0.04
AB107			4	1,2,3,4	19.6	58	26,173	206	0.04	0.05
AB108			5	1,2,3,4,5	21.5	72	16,033	211	0.04	0.05
AB109			1	1	5.5	90	511	249	0.02	0.02
AB110	100	3	2	1,2	10.9	76	653	230	0.02	0.02
AB111			3	1,2,3	13.8	83	238	74	0.02	0.02
AB112			1	1	7.2	119	928	339	0.03	0.03
AB113	100	4	2	1,2	12.7	80	1705	411	0.03	0.03
AB114			3	1,2,3	17.1	62	1217	258	0.03	0.03
AB115			4	1,2,3,4	20.0	78	313	63	0.03	0.03
AB116			1	1	8.5	142	1185	376	0.05	0.05
AB117			2	1,3	16.0	119	1445	369	0.05	0.05
AB118	100	5	3	1,2,3	22.2	76	1815	338	0.05	0.05
AB119			4	1,2,3,4	25.6	74	1796	249	0.05	0.05
AB120			5	1,2,3,4,5	27.5	84	723	118	0.05	0.05

Table 5 – Solutions of MCE-k for instances AB061-AB120 obtained by **??**.

Instance				Optimal Solution		Performance metrics						
Name	n	m	k	Selected Ellipses	Income	CLS size $|S_k|$	# E3P subproblems	Backtracking Tree # nodes	#sol leaves	CPU Time (s) CLS-MCER	Total
AB001			1	2	1.4	9		34	19	0.00	0.00
AB002	10	3	2	2,3	2.3	9	2	21	8	0.00	0.00
AB003			3	1,2,3	2.8	10		23	6	0.00	0.00
AB004			1	3	1.4	8		36	18	0.00	0.00
AB005	10	4	2	3,4	2.3	8	2	26	8	0.00	0.00
AB006			3	2,3,4	2.8	8		26	6	0.00	0.00
AB007			4	1,2,3,4	2.0	10		24	4	0.00	0.00
AB008			1	4	1.4	12		43	19	0.02	0.02
AB009			2	4,5	2.3	12		37	8	0.01	0.01
AB010	10	5	3	3,4,5	2.8	10	34	38	6	0.01	0.01
AB011			4	2,3,4,5	2.0	9		36	4	0.01	0.01
AB012			5	1,2,3,4,5	−0.5	10		81	2	0.01	0.01
AB013			1	1	1.5	15		105	52	0.00	0.00
AB014	20	3	2	1,2	2.9	17	5	131	50	0.00	0.00
AB015			3	1,2,3	3.8	20		46	15	0.00	0.00
AB016			1	2	1.5	23		109	51	0.02	0.02
AB017	20	4	2	2,3	2.9	17	57	134	47	0.02	0.02
AB018			3	1,2,3	4.1	14		107	25	0.01	0.01
AB019			4	1,2,3,4	5.0	20		60	11	0.02	0.02
AB020			1	4	2.4	20		65	35	0.02	0.02
AB021			2	3,4	3.9	18		92	31	0.02	0.02
AB022	20	5	3	3,4,5	4.8	12	77	112	29	0.03	0.03
AB023			4	2,3,4,5	5.0	15		57	11	0.02	0.02
AB024			5	1,2,3,4,5	3.5	20		396	17	0.02	0.02
AB025			1	1	3.5	24		151	74	0.02	0.02
AB026	30	3	2	1,2	5.9	23	60	187	68	0.02	0.02
AB027			3	1,2,3	7.8	27		66	21	0.02	0.02
AB028			1	2	2.5	36		168	79	0.06	0.06
AB029	30	4	2	2,3	4.9	28	197	210	72	0.05	0.05
AB030			3	1,2,3	7.1	23		237	63	0.05	0.05
AB031			4	1,2,3,4	8.0	28		370	37	0.06	0.06
AB032			1	3	2.5	45		152	70	0.13	0.13
AB033			2	1,3	5.0	36		347	87	0.13	0.13
AB034	30	5	3	1,3,4	7.4	24	545	301	49	0.13	0.13
AB035			4	1,3,4,5	9.3	19		278	31	0.13	0.13
AB036			5	1,2,3,4,5	9.5	27		569	39	0.14	0.14
AB037			1	1	3.5	32		195	97	0.02	0.02
AB038	40	3	2	1,2	6.9	28	66	245	92	0.02	0.02
AB039			3	1,2,3	8.8	37		88	30	0.02	0.02
AB040			1	1	5.2	47		386	144	0.08	0.08
AB041	40	4	2	1,2	7.7	31	333	701	178	0.09	0.09
AB042			3	1,2,4	9.6	29		676	88	0.08	0.08
AB043			4	1,2,3,4	11.0	37		349	51	0.08	0.08
AB044			1	3	3.5	73		190	91	0.25	0.25
AB045			2	1,3	7.0	55		514	137	0.26	0.26
AB046	40	5	3	2,3,4	10.1	29	1076	607	97	0.26	0.26
AB047			4	2,3,4,5	12.0	26		673	48	0.25	0.25
AB048			5	1,2,3,4,5	13.5	36		294	38	0.25	0.25
AB049			1	1	7.5	58		300	139	0.09	0.09
AB050	50	3	2	1,2	9.9	39	348	360	110	0.08	0.08
AB051			3	1,2,3	11.8	42		127	35	0.09	0.09
AB052			1	1	5.2	79		602	214	0.25	0.25
AB053	50	4	2	1,2	9.7	52	989	801	199	0.25	0.25
AB054			3	1,2,3	12.1	37		2450	178	0.24	0.24
AB055			4	1,2,3,4	14.0	46		6852	135	0.25	0.25
AB056			1	3	4.5	106		269	126	0.41	0.41
AB057			2	1,3	8.0	81		890	191	0.42	0.42
AB058	50	5	3	1,3,4	11.4	46	1777	1788	129	0.43	0.43
AB059			4	1,2,3,4	14.6	36		3361	146	0.42	0.42
AB060			5	1,2,3,4,5	16.5	44		1867	134	0.42	0.42

Table 6 – Solutions of MCER-*k* for instances AB001-AB060 obtained by Algorithm 6.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $\|S_k\|$	# E3P subproblems	Backtracking Tree # nodes	#sol leaves	CPU Time (s) CLS-MCER	Total
AB061			1	1	4.5	80		390	179	0.10	0.10
AB062	60	3	2	1,2	7.9	45	388	476	144	0.10	0.10
AB063			3	1,2,3	9.8	54		171	47	0.10	0.10
AB064			1	1	6.2	126		894	302	0.40	0.40
AB065	60	4	2	1,2	10.7	79	1696	1561	291	0.41	0.41
AB066			3	1,2,3	14.1	46		1443	198	0.40	0.40
AB067			4	1,2,3,4	16.0	51		755	152	0.41	0.41
AB068			1	1	6.5	154		1559	424	0.71	0.71
AB069			2	1,3	11.0	111		2444	290	0.70	0.70
AB070	60	5	3	1,2,3	14.2	64	2964	5536	293	0.70	0.70
AB071			4	1,2,3,4	16.6	43		15,793	179	0.70	0.71
AB072			5	1,2,3,4,5	18.5	52		11,730	104	0.70	0.70
AB073			1	1	5.5	105		492	220	0.18	0.18
AB074	70	3	2	1,2	8.9	56	753	584	159	0.19	0.19
AB075			3	1,2,3	10.8	59		206	50	0.19	0.19
AB076			1	1	6.2	112		841	298	0.34	0.34
AB077	70	4	2	1,2	11.7	71	1414	1116	274	0.34	0.34
AB078			3	1,2,3	16.1	55		1051	191	0.35	0.35
AB079			4	1,2,3,4	19.0	60		265	44	0.35	0.35
AB080			1	1	7.5	213		1969	524	1.07	1.07
AB081			2	1,2	12.7	119		3642	550	1.07	1.07
AB082	70	5	3	1,2,3	17.2	84	4543	5233	406	1.07	1.07
AB083			4	1,2,3,4	20.6	53		3293	276	1.07	1.07
AB084			5	1,2,3,4,5	23.5	55		1296	105	1.07	1.07
AB085			1	1	5.5	110		509	229	0.19	0.19
AB086	80	3	2	1,2	8.9	54	762	614	175	0.19	0.19
AB087			3	1,2,3	11.8	65		221	59	0.21	0.21
AB088			1	1	8.2	217		1416	464	0.73	0.73
AB089	80	4	2	1,2	13.7	112	2964	1750	355	0.74	0.74
AB090			3	1,2,3	17.1	67		2572	232	0.73	0.73
AB091			4	1,2,3,4	19.0	68		1521	166	0.73	0.73
AB092			1	1	6.5	321		2890	753	1.48	1.48
AB093			2	1,2	12.7	186		4770	737	1.48	1.48
AB094	80	5	3	1,2,3	18.2	108	6276	4389	602	1.50	1.50
AB095			4	1,2,3,4	22.6	64		4655	528	1.48	1.48
AB096			5	1,2,3,4,5	23.5	74		3232	411	1.49	1.49
AB097			1	1	5.5	160		728	319	0.29	0.29
AB098	90	3	2	1,2	9.9	83	1157	866	221	0.28	0.28
AB099			3	1,2,3	11.8	76		306	67	0.29	0.29
AB100			1	1	7.2	207		1465	494	0.72	0.73
AB101	90	4	2	1,2	12.7	132	3019	2593	481	0.73	0.73
AB102			3	1,2,3	16.1	76		1800	261	0.74	0.74
AB103			4	1,2,3,4	19.0	79		455	61	0.72	0.72
AB104			1	1	10.5	452		2820	703	2.46	2.46
AB105			2	1,2	16.7	249		5862	704	2.48	2.48
AB106	90	5	3	1,2,3	21.2	115	10,488	13,041	434	2.48	2.49
AB107			4	1,2,3,4	24.6	64		72,194	501	2.56	2.60
AB108			5	1,2,3,4,5	26.5	72		105,181	312	2.46	2.51
AB109			1	1	7.5	181		836	366	0.39	0.39
AB110	100	3	2	1,2	12.9	102	1614	1002	255	0.41	0.41
AB111			3	1,2,3	15.8	83		354	74	0.40	0.40
AB112			1	1	8.2	337		2091	660	1.33	1.33
AB113	100	4	2	1,2	14.7	165	5613	3604	527	1.35	1.35
AB114			3	1,2,3	19.1	80		2487	270	1.32	1.32
AB115			4	1,2,3,4	22.0	78		629	62	1.33	1.33
AB116			1	1	9.5	649		5571	1387	3.31	3.31
AB117			2	1,2	17.7	368		6671	1031	3.30	3.30
AB118	100	5	3	1,2,3	25.2	183	14,029	7344	609	3.32	3.32
AB119			4	1,2,3,4	29.6	103		6474	320	3.32	3.33
AB120			5	1,2,3,4,5	31.5	84		1579	119	3.30	3.30

Table 7 – Solutions of MCER-k for instances AB061-AB120 obtained by Algorithm 6.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\|S_k\|$	Backtracking Tree # nodes	#sol leaves	CPU Time (s) CLS-MCER	Total
TA01			1	1	48.9	218	3507	891	0.36	0.36
TA02			2	1,3	95.1	203	6596	1588	0.35	0.36
TA03			3	1,3,5	125.7	204	133,560	3576	0.36	0.49
TA04	100	7	4	1,3,5,7	148.8	204	960,460	5726	0.36	2.55
TA05			5	1,3,4,5,6	158.4	232	23,848,340	5945	0.36	87.26
TA06			6	2,3,4,5,6,7	162.0	248	523,396,293	5023	0.36	3454.29
TA07			7	-	-	237	-	-	-	-

Table 8 – Solutions of MCE-*k* for instances TA001-TA007 obtained by **??**.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $\|S_k\|$	# E3P subproblems	Backtracking Tree # nodes	#sol leaves	CPU Time (s) CLS-MCER	Total
TA01			1	3	52.4	470		10,026	4830	43.26	43.27
TA02			2	1,3	102.3	3015		30,072	11,475	43.21	43.23
TA03			3	1,3,5	135.2	755		1,259,300	24,958	43.28	46.97
TA04	100	7	4	1,3,5,7	157.1	721	146,116	57,430,353	74,709	43.30	462.09
TA05			5	-	-	1059		-	-	-	-
TA06			6	-	-	973		-	-	-	-
TA07			7	-	-	3132		-	-	-	-

Table 9 – Solutions of MCER-*k* for instances TA001-TA007 obtained by Algorithm 6.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\|S_k\|$	Backtracking Tree # nodes	#sol leaves	CPU Time (s) CLS-MCER	Total
TA08			1	2	82.1	836	2577	1760	1.19	1.19
TA09	200	3	2	1,2	157.2	811	9993	4238	1.19	1.21
TA10			3	1,2,3	192.6	949	38,939	7294	1.19	1.31
TA11			1	2	103.4	1349	3845	2610	2.11	2.11
TA012	250	3	2	2,3	196.5	1229	3995	2762	1.95	1.96
TA13			3	1,2,3	249.0	1381	23,598	12,416	1.96	2.09
TA14			1	1	112.1	2128	8493	4231	2.95	2.96
TA015	300	3	2	1,3	207.7	2152	10,602	4190	3.00	3.01
TA16			3	1,2,3	299.4	2103	12,726	4181	2.97	2.99
TA17			1	2	224.4	2561	6487	4550	9.54	9.55
TA18	350	3	2	1,2	379.7	1931	14,030	7603	10.47	10.54
TA19			3	1,2,3	460.1	2619	197,645	17,431	10.24	12.01
TA20			1	2	193.0	2716	9035	5993	15.82	15.84
TA21	400	3	2	2,3	339.6	3036	8939	5899	15.64	15.79
TA22			3	1,2,3	400.3	2957	633,779	14,754	15.58	49.86

Table 10 – Solutions of MCE-*k* for instances TA008-TA022 obtained by **??**.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $\|S_k\|$	# E3P subproblems	Backtracking Tree # nodes	# sol leaves	CPU Time (s) CLS-MCER	Total
TA08			1	1	85.9	8589		37,146	18,514	129.71	129.73
TA09	200	3	2	1,2	169.7	1448	681,627	53,908	25,243	129.22	129.27
TA10			3	1,2,3	202.6	8477		772,760	60,542	128.75	138.25
TA11			1	2	126.2	11,226		59,486	34,196	228.61	228.68
TA12	250	3	2	2,3	215.0	25,284	995,713	34,200	8912	232.41	233.87
TA13			3	1,2,3	262.8	8912		32,908,602	53,459	226.03	610.32
TA14			1	1	112.1	6693		42,702	29,310	383.00	383.05
TA15	300	3	2	1,3	214.2	22,954	1,755,415	81,519	45,175	410.92	411.05
TA16			3	1,2,3	311.2	22,617		257,865	22,558	401.90	402.44
TA17			1	2	225.9	63,315		54,419	43,151	775.78	775.85
TA18	350	3	2	1,2	398.1	11,262	2,961,709	191,753	83,386	771.38	772.47
TA19			3	1,2,3	483.3	31,889		2,421,540	274,754	800.72	874.46
TA20			1	2	199.6	17,691		178,589	141,413	922.98	923.47
TA21	400	3	2	2,3	364.7	37,170	2,432,988	245,472	208,298	903.69	912.19
TA22			3	-	-	112,932		-	-	-	-

Table 11 – Solutions of MCER-k for instances TA008-TA022 obtained by Algorithm 6.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\|S_k\|$	Backtracking Tree # nodes	# sol leaves	CPU Time (s) CLS-MCER	Total
TA23			1	5	14.5	830	1165	1150	0.96	0.96
TA24			2	3,5	27.4	627	2930	1150	0.95	0.95
TA25	400	5	3	3,4,5	36.8	880	26,520	3450	0.95	0.97
TA26			4	1,3,4,5	46.2	660	587,336	9200	0.95	1.48
TA27			5	1,2,3,4,5	54.2	1150	5,715,962	18,356	0.95	9.91
TA28			1	4	30.9	1396	4071	2028	2.48	2.48
TA29			2	4,5	57.8	1256	3983	1935	2.52	2.53
TA30	500	5	3	3,4,5	80.9	1678	19,478	9673	2.53	2.56
TA31			4	1,3,4,5	101.3	2028	101,334	9674	2.50	2.67
TA32			5	1,2,3,4,5	117.7	1939	2,040,107	17,428	2.56	6.11
TA33			1	2	42.5	1980	14,067	3513	4.79	4.80
TA34			2	2,4	73.5	3513	12,372	2663	4.70	4.71
TA35	600	5	3	1,2,4	101.8	1713	19,671	5325	4.80	4.82
TA36			4	1,2,4,5	126.0	2696	24,966	7960	4.71	4.73
TA37			5	1,2,3,4,5	147.4	2047	70,594	7949	4.74	4.81
TA38			1	5	63.0	4635	5557	5542	8.96	8.97
TA39			2	1,5	110.5	3243	24,102	5542	9.06	9.12
TA40	700	5	3	1,2,5	143.6	2212	19,804	5542	9.06	9.46
TA41			4	1,2,4,5	169.9	2536	341,942	44,336	9.04	14.92
TA42			5	1,2,3,4,5	195.3	5542	506,117	49,878	9.00	22.39

Table 12 – Solutions of MCE-k for instances TA008-TA022 obtained by **??**.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $|S_k|$	# E3P subproblems	Backtracking Tree # nodes	#sol leaves	CPU Time (s) CLS-MCER	Total
TA23			1	1	15.4	8939		44,710	8939	63.46	63.47
TA24			2	1,3	30.3	1116		31,689	4597	62.78	62.80
TA25	400	5	3	1,3,5	41.8	4597	207,056	549,510	2212	63.17	63.37
TA26			4	1,3,4,5	51.2	1317		10,524,741	8844	63.07	71.62
TA27			5	1,2,3,4,5	60.2	2212		100,446,086	19,904	63.16	219.73
TA28			1	4	32.9	9141		15,093	7539	198.29	198.30
TA29			2	4,5	60.8	12,541		9861	2302	196.96	196.99
TA30	500	5	3	3,4,5	84.9	15,986	655,969	146,030	4599	197.41	197.90
TA31			4	1,3,4,5	105.3	7539		14,107,397	16,124	197.83	238.85
TA32			5	1,2,3,4,5	123.7	2313		510,878,989	39,157	197.21	2347.94
TA33			1	2	44.5	34,585		71,347	17,833	379.90	379.93
TA34			2	2,4	77.5	17,833		61,168	12,741	378.37	378.44
TA35	600	5	3	1,2,4	105.8	5988	1,266,119	243,344	50,873	379.79	379.98
TA36			4	1,2,4,5	131.0	12,861		275,879	12,085	381.46	381.73
TA37			5	1,2,3,4,5	153.4	2090		280,278	16,108	380.39	380.87
TA38			1	5	64.0	7597		7195	7180	731.35	731.38
TA39			2	1,5	112.5	14,076		44,768	14,360	725.67	725.86
TA40	700	5	3	1,2,5	146.6	2386	2,500,817	271,740	14,360	729.36	732.77
TA41			4	1,2,4,5	174.9	26,697		938,333	57,437	725.81	750.48
TA42			5	1,2,3,4,5	199.3	7180		5,572,365	78,977	723.72	1242.04

Table 13 – Solutions of MCER-*k* for instances TA008-TA022 obtained by Algorithm 6.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $|S_k|$	Backtracking Tree # nodes	#sol leaves	CPU Time (s) CLS-MCER	Total
TA43			1	5	87.9	97	43	28	0.22	0.22
TA44			2	3,4	126.9	89	314	95	0.21	0.22
TA45	80	5	3	1,2,3	136.8	39	33,898	229	0.21	0.43
TA46			4	1,2,3,4	124.8	42	1,689,010	146	0.21	11.71
TA47			5	1,2,3,4,5	110.3	28	12,794,063	1	0.22	101.23

Table 14 – Solutions of MCE-*k* for instances TA008-TA022 obtained by **??**.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $|S_k|$	# E3P subproblems	Backtracking Tree # nodes	#sol leaves	CPU Time (s) CLS-MCER	Total
TA43			1	5	87.9	228		50	35	19.73	19.73
TA44			2	3,4	126.9	439		508	138	19.76	19.76
TA45	80	5	3	1,2,3	136.8	70	72,307	225,790	455	19.64	21.45
TA46			4	1,2,3,4	124.8	71		31,519,719	172	19.70	309.69
TA47			5	-	-	35		-	-	-	-

Table 15 – Solutions of MCER-*k* for instances TA008-TA022 obtained by Algorithm 6.