Algorithms for Planar Maximum Covering Location by Ellipses Problems[☆]

Danilo Tedeschi[1], Marina Andretta[1]

Abstract

Planar Maximum Covering Location 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, Covering, Combinatorial Optimization

1. Introduction

The Planar Maximum Covering Location Problem (PMCLP) was first introduced in [1], and can be seen as a category of problems where the coverage of a demand set, a collection of subsets of $\mathbb{R}^2$, is to be maximized by determining the location of facilities in $\mathbb{R}^2$, with coverage being determined by a distance function. In [1], methods for Euclidean and Rectilinear distances versions of the problem were proposed. In [2, 3], exact algorithms for the Euclidean PMCLP with only one facility are proposed; and in [4] an approximation algorithm is proposed for the version with multiple unit disks as facilities. A property of the Euclidean PMCLP, which is utilized in the algorithms developed in [2, 3, 4], and in the method proposed in [1], is that there is an optimal solution which every facility is located in the demand points, or in the intersection of two circles centered at two demand points; we will prove a similar property for ellipses in our work.

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 [5], a mixed-integer non-linear programming method was proposed as a first approach to the problem. For some instances, the method took too long and did not find an optimal solution. 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 [6], 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 [5] could not. The exact method for the version with rotation could not obtain optimal solutions within a predefined time limit for several instances, the

[☆]This paper is the results of the research project funded by CAPES.

Email addresses: danilo.tedeschi@usp.br (Danilo Tedeschi), andretta@gmail.com (Marina Andretta)

heuristic method though returned solutions for every instance, and impressively enough, obtained optimal solutions for every verifiable instance.

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 [5], all the ellipses are restricted to be axis-parallel, while in the second version, introduced in [6], this constraint is dropped, and all the ellipses can be freely rotated. The first version will be referred to as Planar Maximum Covering Location by Ellipses Problem (MCE) and the second one as Planar Maximum Covering Location by Ellipses with Rotation Problem (MCER).

2. Definition

An instance of MCE and MCER is given by n distinct demand points $\mathcal{P} = \{p_1, \ldots, p_n\}$, $p_j \in \mathbb{R}^2$; n weights $\mathcal{W} = \{w_1, \ldots, w_n\}$, with $w_j \in \mathbb{R}_{>0}$ being the weight of the j-th point; and m shape parameters $\mathcal{R} = \{(a_1, b_1), \ldots, (a_m, b_m)\}$, with $(a_j, b_j) \in \mathbb{R}^2_{\geq 0}$ being the semi-major and semi-minor axis of the j-th ellipse, with $a_j > b_j > 0$.

For MCE, the shape parameters describe axis-parallel ellipses, and the problem is to determine a center for each ellipse to maximize the weight of covered demand points. For MCER, besides the center, because the ellipses are not necessarily axis-parallel, for each facility, an angle of rotation is also part of the solution, and the problem is to maximize the weight of covered demand points as well.

Let $\mathcal{E} = \{E_1, \ldots, E_m\}$ be a list of functions representing the coverage area of each facility, with $E_j \colon \mathbb{R}^2 \to \mathbb{R}^2$ for MCE, and $E_j \colon \mathbb{R}^2 \times [0, \pi) \to \mathbb{R}^2$. Let $||\cdot||_{a,b,\theta} \colon \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ denote the elliptical norm given by

$$||x||_{a,b,\theta} = \left|\left| \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix} \begin{pmatrix} a_j & 0 \\ 0 & b_j \end{pmatrix} x \right|\right|_2,$$

then, for MCE we define $E_j(q) = \{p \in \mathbb{R}^2 \colon ||p - q||_{a_j,b_j,0} \leq 1\}$; and for MCER we define $E_j(q, \theta) = \{p \in \mathbb{R}^2 \colon ||p - q||_{a_j,b_j,\theta} \leq 1\}$.

To make the notation more clear, we define a solution of MCE as $Q := (q_1, \ldots, q_m)$, and a solution of MCER as $Q := ((q_1, \theta_1); \ldots; (q_m, \theta_m))$. Let $w : A \subset \mathcal{P} \to \mathbb{R}$ be a function which takes a subset of the demand set and returns the sum of the weights of the points in A. Then, we define MCE as the optimization problem

$$\max_Q w\left(\cup_{j=1}^m \mathcal{P} \cap E_j(q_j)\right),$$

and similarly MCER as

$$\max_Q w\left(\cup_{j=1}^m \mathcal{P} \cap E_j(q_j, \theta_j)\right).$$

Additionally, whenever we have an instance $(\mathcal{P}, \mathcal{W}, \{(a, b)\})$ with only one ellipse, we omit the index referring to the facility, and we use ∂ as the boundary operator, for example, given an instance of MCE, $\partial E_1(q_1)$ denotes an ellipse with shape parameters (a_1, b_1) centered at q_1.

2.1. Facility Cost

In [5, 6], as a result of having costs assigned to each facility, two other parameters are present in the definition of the problem. These new parameters are: the list of costs $\mathcal{C} = \{c_1, \ldots, c_m\}$, with

$c_j \in \mathbb{R}_{\geq 0}$ being the j-th ellipse's cost; and an integer $k \in \mathbb{N}$, $k \leq m$, which introduces a constraint requiring that exactly k facilities have to be utilized. Because of that, a solution also needs an additional parameter $I := \{i_1, \ldots, i_k\} \subset \{1, \ldots, m\}$ to express the indexes of the utilized ellipses. We refer to this version of the problem as MCE-k and MCER-k.

From an algorithm for MCE (MCER), we can propose an algorithm for MCE-k (MCER-k), which just takes the best overall solution among the $\binom{m}{k}$ instances $(\mathcal{P}, \mathcal{W}, \mathcal{R}')$, such that $\mathcal{R}' \subset \mathcal{R}$ and $|\mathcal{R}'| = k$. Therefore, in our work, we focus on developing algorithms for MCE and MCER, and whenever we refer to an algorithm for MCE-k (MCER-k), we mean an algorithm as described here, which considers all the $\binom{m}{k}$ instances of MCE (MCER).

3. An algorithm for MCE

Similarly to the method developed in [1] for the Euclidean PMCLP, we will describe a Candidate List Set (CLS) of possible locations for each ellipse and then propose an algorithm that constructs solutions combining the possible locations in each ellipse's CLS. Based on the approach of [3, 4], we will construct the CLS for each ellipse by working with a problem equivalent to MCE for only one facility.

To define this equivalent problem, we need to first state an ellipse's property. Let $(\mathcal{P}, \mathcal{W}, \{(a, b)\})$ be an instance of MCE with one facility, and $p, q \in \mathbb{R}^2$, we have that

$$p \in E(q) \Leftrightarrow ||p - q||_{a,b,0} \leq 1 \Leftrightarrow q \in E(p). \tag{1}$$

The equivalent problem is given by n ellipses with shape parameters (a, b) centered at $\mathcal{P}$. Let $q \in \mathbb{R}^2$ be a solution of MCE for one facility, by applying Equation 1 to every point covered by $E(q)$, we obtain that

$$\mathcal{P} \cap E(q) = \{p_i \in \mathcal{P} \colon q \in E(p_i)\}, \tag{2}$$

which implies that the problem of determining $q \in \mathbb{R}^2$ to maximize $w(\{p_i \in \mathcal{P} \colon q \in E(p_i)\})$, is equivalent to MCE for one facility. This changes the problem from determining a location for an ellipse given n points to the problem of finding a point given n ellipses with fixed locations.

To construct the CLS for each ellipse, let us consider the intersection region $\cap_{p \in A} E(p)$, for some $A \subset \mathcal{P}$, $|A| > 1$. In [1], this region is said to have vertices that are in the set of pairwise intersections of the circles with centers in A. Using the results of [7], which develops an algorithm to determine the intersection region of n fixed-radii disks in an strictly convex normed space, it is possible to prove that this is also true for ellipses. Based on that, given an instance of MCE, we define the CLS for each ellipse considering also the case where an ellipse could cover only one point in an optimal solution.

Lemma 1. *Let E be the coverage region of an axis-parallel ellipse with shape parameters (a, b); and $v \in \mathbb{R}^2$, $v \neq 0$. Then $|\partial E(0) \cap \partial E(v)| \leq 2$, and $\partial E(0) \cap \partial E(v)$ can be determined analytically.*

Proof. By [7], we have that the number of intersections between two strictly convex circles of fixed radii is at most two. To determine the intersection points, consider the equality between the equations of $\partial E(0)$ and $\partial E(v)$: $x^2/a^2 + y^2/b^2 = (x - v_x)^2/a^2 + (y - v_y)^2/b^2$. This expression can be reduced to $y = \alpha x + \beta$, for some α, β. Using $\partial E(0)$'s equation, we obtain at most two values for x, which consequently, by $y = \alpha x + \beta$, determine the intersection points between the two ellipses. □

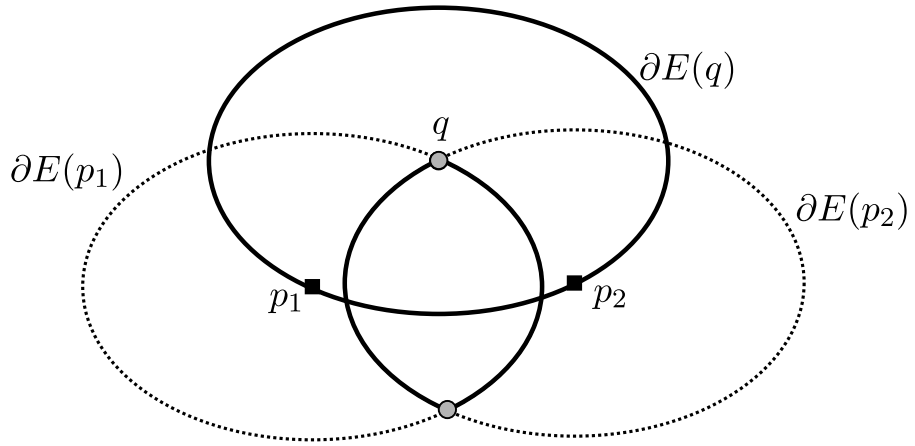


Figure 1: Transforming a solution of E3P into a solution of the circumcircle problem.

Definition 1. Given an instance of MCE, for all $k \in \{1, \ldots, m\}$, we define the CLS for the k-th ellipse as

$$S_k = \mathcal{P} \cup \left(\bigcup_{1 \leq i < j \leq n} \partial E_k(p_i) \cap \partial E_k(p_j) \right). \tag{3}$$

By Lemma 1, the CLS for each ellipse can be computed in $\mathcal{O}(n^2)$, and $|S_k| \leq n + 2\binom{n}{2}$. Next, we establish a lemma stating that the set of solutions obtained by combining the possible locations in each ellipse's CLS contains at least one optimal solution.

Theorem 1. *Given an instance of MCE, and $S_1, \ldots, S_m$ as defined by Definition 1, then the set $\Omega = \{(q_1, \ldots, q_m)\colon \text{ for all } q_k \in S_k\}$ contains an optimal solution of MCE and $|\Omega| \leq n^{2m}$.*

Proof. Let Q^* be an optimal solution of MCE for the given instance. Then, we are going to prove that there exists $Q' \in \Omega$, which is also optimal.

For each $k = 1, \ldots, m$, let $X_k = \{p_i \in \mathcal{P}\colon p_i \in E_k(q_k^*)\}$.

If $|X_k| \leq 1$, then there is at least one element in S_k that makes $X_k \subset \mathcal{P} \cap E_k(q_k)$.

If $|X_k| > 1$, then let $Y_k = \cap_{p \in X_k} E_k(p)$. By the results of [7], we have that the boundary of Y_k has vertices in the pairwise intersections of $\{\partial E_k(p)\colon p \in X_k\}$. Therefore, at least one vertex of Y_k is in S_k, and any of those vertices produce a solution that covers at least the same points covered by Q^*.

Lastly, we have that $|S_k| \leq 2\binom{n}{2} + n = n(n+1)/2 \leq n^2$. Hence, $|\Omega| \leq n^{2m}$. □

With all this in hand, we define Algorithm 1, which goes through every possible combination in the CLS of each ellipse. As evaluating each solution can be done in $\mathcal{O}(nm)$, we have that Algorithm 1 has $\mathcal{O}(mn^{2m+1})$ runtime complexity. In section 8, we give more details about the implementation of Algorithm 1 and analyze some numerical experiments for instances proposed in [5, 6], and for some new ones.

Algorithm 1 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.

procedure $MCE(\mathcal{P}, \mathcal{W}, \mathcal{R})$
    return $MCE_{bt}(\mathcal{P}, \mathcal{W}, \mathcal{R}, 1)$
end procedure

procedure $MCE_{bt}(Z, \mathcal{W}, \mathcal{R}, j)$
    if $j = |\mathcal{R}| + 1$ **then**
        return 0
    end if
    $(q_j^*, \ldots, q_m^*) \leftarrow (0, \ldots, 0)$
    Let S_j be the CLS for the j-th ellipse as defined by Definition 1.
    for $q_j \in S_j$ **do**
        $Cov \leftarrow \mathcal{P} \cap E_j(q_j)$
        $(q_{j+1}, \ldots, q_m) \leftarrow MCE_{bt}(Z \setminus Cov, \mathcal{W}, \mathcal{R}, j+1)\}$
        if $w(\cup_{k=j}^{m} Z \cap E_k(q_k)) > w(\cup_{k=j}^{m} Z \cap E_k(q_k^*))$ **then**
            $(q_j^*, \ldots, q_m^*) \leftarrow (q_j, \ldots, q_m)$
        end if
    end for
    return $(q_j^*, \ldots, q_m^*)$
end procedure

4. Determining every center and angle of rotation of an ellipse given its shape and three points that it must contain

In this section, 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 for MCER in the next section. It is important to point out that no prior studies were found on it, or even on related problems. We propose an algorithm for it 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. Problem definition

Given the shape parameters of an ellipse (a, b), $a > b > 0$, and three points $u, v, w \in \mathbb{R}^2$, let $E\colon \mathbb{R}^2 \times [0, \pi) \to \mathbb{R}^2$ be the coverage region of an ellipse with shape parameters (a, b), we refer to the problem of obtaining $(q, \theta) \in \mathbb{R}^2 \times [0, \pi)$, such that $\{u, v, w\} \subset \partial E(q, \theta)$ as Ellipse by Three Points Problem (E3P). Because of its application here in our work, we are only interested in a method that can obtain every solution of E3P.

4.2. Transforming E3P into a circle problem

Initially, E3P is a problem of determining the values of three unknown continuous variables (q_x, q_y), and θ. However, as it will be shown, we can reduce this number to only one, as it is possible to obtain q uniquely from θ. 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.

Applying a rotation of $-\theta$ to the coordinate system makes the ellipse in the original solution become axis-parallel. Then, that ellipse can be transformed into a circle of radius b by squeezing the x-axis by b/a. This two-step transformation can be written as a function $\varphi\colon \mathbb{R}^2 \to \mathbb{R}^2$ defined as

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

An example of this transformation can be seen in Figure 2. As φ^{-1} is well-defined, instead of solving E3P, we can work with the univariate problem of determining an angle of rotation $\theta \in [0, \pi)$ that makes the triangle with vertices $\varphi(u, \theta), \varphi(v, \theta), \varphi(w, \theta)$ be circumscribed in a circle of radius b. To make the notation less cluttered, we denote by $\Lambda(\theta)$ the triangle with vertices $\varphi(u, \theta), \varphi(v, \theta), \varphi(w, \theta)$.

As circles are uniquely defined by three non-collinear points, the circumcircle of $\Lambda(\theta)$ is unique, and its radius and center can be determined analytically [8]. Let $|\Lambda(\theta)|$ denote the area of $\Lambda(\theta)$, using the formula from [9, p. 189] for the radius of a circumcircle of a triangle, and imposing that radius to be equal b, we define a function $\xi\colon [0, \pi) \to \mathbb{R}$ as

$$\xi(\theta) = 16b^2|\Lambda(\theta)|^2 - \|\varphi(v, \theta)\|_2^2 \, \|\varphi(w, \theta)\|_2^2 \, \|\varphi(v, \theta) - \varphi(w, \theta)\|_2^2 , \tag{4}$$

whose roots are angles of rotation which determine solutions of E3P through the inverse transformation φ^{-1}. From a root $\hat{\theta}$ of ξ, let $\hat{q}$ be the center of the circumcircle of $\Lambda(\hat{\theta})$, the solution $(\varphi^{-1}(\hat{q}), \hat{\theta})$ of E3P is obtained.

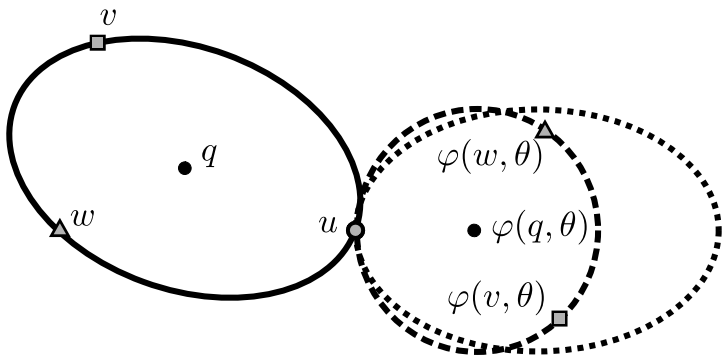


Figure 2: Transforming a solution of E3P into a solution of the circumcircle problem.

The algorithm for MCER described in the next section, of which E3P is a subproblem, goes through every solution of several instances of E3P. This is only possible if the number of solutions of E3P is finite, and it is only viable if that number is small. Next we introduce a lemma regarding that matter.

Lemma 2. *E3P has at most six solutions.*

Proof. The first thing to notice is that ξ is a real trigonometric polynomial of degree 6. Its term of highest degree is the multiplication of the norms $\|\varphi(v,\theta)\|_2^2 \|\varphi(w,\theta)\|_2^2 \|\varphi(v,\theta) - \varphi(w,\theta)\|_2^2$. In [10, p. 150], where a definition of real trigonometric polynomial is also given, it is stated that a n-degree real trigonometric polynomial can have up to $2n$ roots in $[0, 2\pi)$. Therefore, E3P has at most 12 solutions in $[0, 2\pi]$. Half of these solutions, though, are duplicated as ellipses are symmetric to their axis. □

4.3. Converting ξ into a polynomial

In [11, p. 195], a theorem is presented stating 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. 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 [12]). For example, for a degree-4 polynomial $\sum_{k=0}^{4} a_k x^4$, a possible companion matrix is given by

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

In practice, we can use the very well-known LAPACK software library to obtain the eigenvalues of a matrix[13]. This approach works for both real or complex polynomials, and, because of that, based on [14], we describe a way of converting ξ into a complex polynomial.

By using the identities $\cos\theta = (e^{i\theta} + e^{-i\theta})/2$, and $\sin\theta = (e^{i\theta} - e^{-i\theta})/(2i)$, which relate trigonometric functions with complex numbers in the unit circle $\mathbb{S} = \{z \in \mathbb{C} \colon |z| = 1\}$, we can rewrite ξ as a function of the variable $z = e^{i\theta} \in \mathbb{S}$. In [14], it is stated that this substitution when utilized for the task of determining the roots of a real trigonometric polynomial does not yield loss of accuracy.

As ξ is a real trigonometric polynomial of degree 6, z appears with exponents from -6 up to 6. Multiplying ξ by z^6 and extending the domain of z to $\mathbb{C}$, we obtain a complex polynomial $g(z) = \sum_{k=0}^{12} c_k z^k$, for some $c_0, \ldots, c_{12} \in \mathbb{C}$. In practice, we utilize symbolic computation to obtain the actual coefficients of g in terms of an instance of E3P.

Let $angle\colon \mathbb{C} \to [0, 2\pi)$ be a function that takes a complex number and returns its angle, then given a root $\hat{z}$ of g, if $|\hat{z}| = 1$ and $angle(\hat{z}) \in [0, \pi)$, then $\hat{\theta} = angle(\hat{z})$ is a root of ξ.

Observing that for any $z \in \mathbb{C}$, $angle(-z) = \pi + angle(z)$, and that for any ellipse the angles of rotation θ and $\theta + \pi$ are equivalent, we conclude that $g(-z) = g(z)$. This implies that all the odd-degree coefficients of g are zero. Therefore, we can use the substitution $y = z^2$ to obtain a degree-6 polynomial $f(y) = \sum_{k=1}^{6} c_{2k} y^k$ whose roots can be used to determine the roots of ξ: from a root $\hat{y}$ of f, $\hat{y} \in \mathbb{S}$, we have that $\hat{\theta} = angle(\hat{y})/2$ is a root of ξ.

Therefore, using the QR algorithm to obtain the eigenvalues of a companion matrix of the polynomial f, we can conclude that an algorithm to obtain every solution of E3P can be implemented, and that such algorithm takes a constant number of operations to do so.

4.4. Choosing a precision constant

In this section, we describe an experiment we made to choose a precision constant for comparing if a root of f returned as an eigenvalue of its companion matrix is in the unit circle. The implementation was coded in C++, and LAPACK's ZGEEV was utilized to obtain the eigenvalues of the companion matrix of f (more information about the implementation is given in section 7). For the experiment, we defined $K \in \mathbb{R}$, $K > 0$, and considered instances with the ellipse's shape parameters $(K, \frac{K}{2})$, for $K \in \{10^j : j = 0, \ldots, 10\}$.

The experiment considered instances of E3P where the three points are the vertices of an ellipse rotated by $\theta \in [0, \pi)$. Such instances only have one solution, and therefore, roots with multiplicity greater than one are expected. 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 root $\hat{z}$ which produced the closest solution to the known one. Then, for each K, as it can be seen in Figure 3a, we considered the maximum and the average distance to the unit circle $|1 - |\hat{z}||$; and, as it presented in Figure 3b, the maximum and average error $|f(\hat{z})|$.

From this experiment, we decided to adopt a precision constant of 10^{-6} to consider a root of f to be in the unit circle, and as an additional check, we adopted a precision constant of 10^{-9} to consider a root to be a solution of E3P.

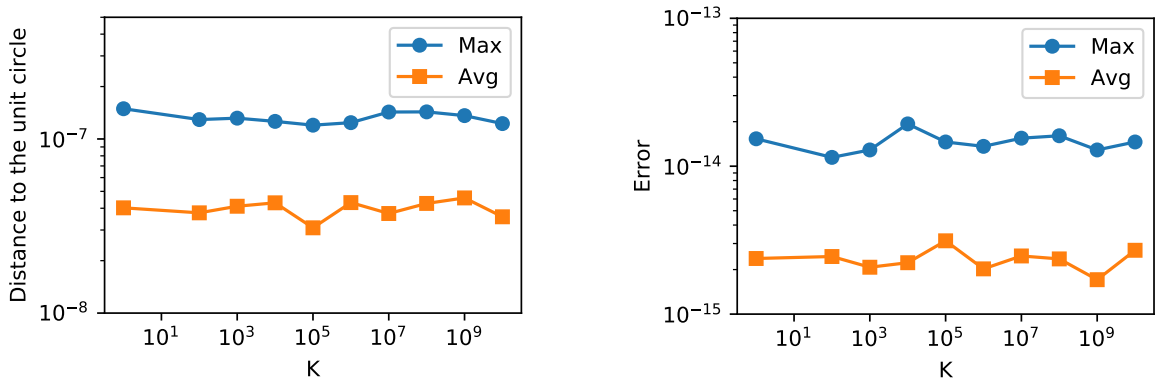
4.5. Instances with four and six solutions

Any instance of E3P, as stated by Lemma 2, can have up to six solutions. At first, though, this bound seemed to be loose as for randomly generated instances, we were not able to obtain instances with more than two solutions.

After some investigation, we were able to construct some four-solution instances (an example is displayed in Figure 4a). An interesting property of those solutions is that their three points form an isosceles triangle.

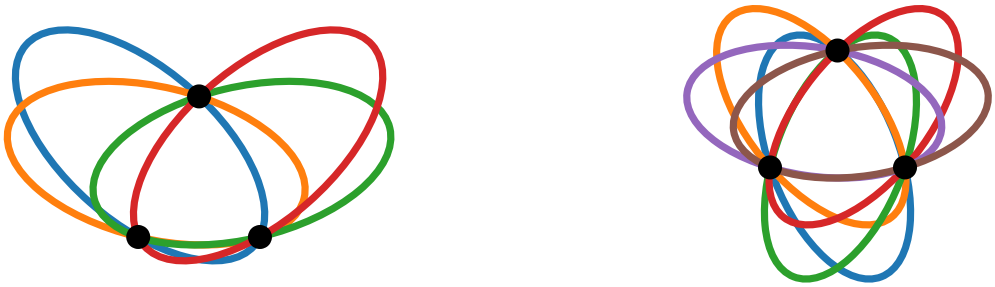
Six-solution instances were found by taking a particular case of the four-solution instances, we took the three points as the vertices of a equilateral triangle. An example of that is shown in Figure 4b.

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.



(a) (b)

Figure 3: (a) Shows the maximum and average distance to the unit circle $|1 - |\hat{z}||$. (b) Shows the maximum and average error $|f(\hat{z})|$.



(a) (b)

Figure 4: (a) The four solutions for an instance of E3P where the three points form an isosceles triangle. (b) The six solutions for an instance of E3P where the three points form an equilateral triangle.

5. An Algorithm for MCER

The version of PMCLP where the facilities are ellipses that can be freely rotated was first introduced in [6] where an exact and a heuristic method were developed for it. In comparison with MCE, this problem introduces a new variable that is responsible for determining the angle of rotation of every ellipse, making MCER a more challenging problem. We propose an algorithm for MCER which is able to obtain optimal solutions for every instance proposed in [6] including the ones its exact method could not, and its heuristic obtained non-optimal ones.

Definition 2. Two solutions Q and Q' of MCER, are said to be equivalent to each other if $\mathcal{P} \cap \cup_{j=1}^{m} E_j(q_j, \theta_j) = \mathcal{P} \cap \cup_{j=1}^{m} E_j(q'_j, \theta'_j)$. Also, if $\mathcal{P} \cap \cup_{j=1}^{m} E_j(q_j, \theta_j) \subset \mathcal{P} \cap \cup_{j=1}^{m} E_j(q'_j, \theta'_j)$, then we say that $Q' \succ Q$.

Next, we introduce a lemma which states that any optimal solution of MCER has an equivalent solution where every ellipse that covers at least two points has two points on its border.

Lemma 3. *Let Q^* be a solution of MCER for an instance $(\mathcal{P}, \mathcal{W}, \{(a,b)\})$. If $|\mathcal{P} \cap E(q^*, \theta^*)| \geq 2$, then there exists a solution Q for the same instance, such that $Q \succ Q^*$, and $|\mathcal{P} \cap \partial E(q, \theta)| \geq 2$.*

Proof. First, let $\theta = \theta^*$ and ignore the angle of rotation as it does not change, and assume that we are dealing with an axis-parallel ellipse.

Let $A = \mathcal{P} \cap E(q^*, \theta^*)$ and $X = \cap_{p \in A} E(p, \theta^*)$ be the region of intersection of ellipses centered at each point in A. By [7], the vertices of ∂X are in the set of pairwise intersections of $\{\partial E(p, \theta^*) \colon p \in A\}$. Setting q as any of these vertices makes $E(q, \theta)$ have two points on its border. □

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 3. 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, the set of (E, u, v)-feasible angles is referred to as

$$\Phi(u, v) := \{\theta \in [0, \pi) : \theta \text{ is a } (E, u, v)\text{-feasible angle}\}.$$

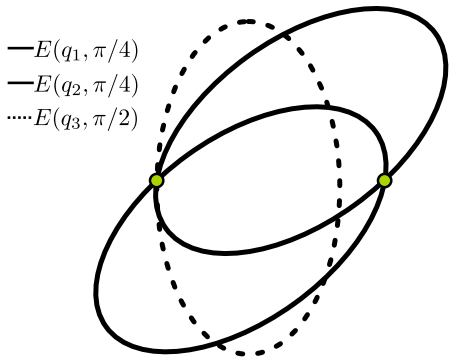


Figure 5: A (E, u, v)-feasible angle and a not (E, u, v)-feasible angle.

For any $x \in \mathbb{R}^2$, we denote by $\angle x \in [0, \pi)$ the minimal angle between x and the vector $(1, 0)$. Then, if $\Phi(u, v) \neq \emptyset$, then $\angle(u - v) \in \Phi(u, v)$ as it is the angle that makes the ellipse's major-axis be parallel to the line that passes through u and v.

Next we open a parenthesis to discuss the problem of deciding for what angles of rotation it is possible to find a center for an ellipse, so it contains two given points. We give the result for two points that have the same y-coordinate, but this can be generalized.

Lemma 4. *Given an instance $(\mathcal{P}, \mathcal{W}, \{(a,b)\})$ of MCER, if $u, v \in \mathcal{P}$ have the same y-coordinate and $||u - v||_2 \leq 2a$, then $\Phi(u, v) = [0, \alpha] \cup [\pi - \alpha, \pi)$, for some $\alpha \in [0, \pi/2]$.*

Proof. Consider an axis-parallel 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. By plugging the line's equation into $x^2/a^2 + y^2/b^2 = 1$, 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). If $D(m, c) = ||u - v||_2$, then there exists $q_1, q_2 \in \mathbb{R}^2$, such that $\{u, v\} \subset \partial E(q_1, \tan m)$ and $\{u, v\} \subset \partial E(q_2, \pi - \tan m)$. 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 $c = 0$. Following that, we define a function $L : \mathbb{R} \mapsto \mathbb{R}$ as

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

which describes the maximum distance between points of an ellipse-line intersection considering all lines with m angular coefficient. From that, if $L(m) \geq ||v - u||_2^2$, then there exists $q_1, q_2 \in \mathbb{R}^2$, such that $\{u, v\} \subset \partial E(q_1, \tan m)$, and $\{u, v\} \subset \partial E(q_2, \pi - \tan m)$.

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.

If $\inf L \geq ||u - v||_2^2$, then $\Phi(u, v) = [0, \pi)$. Otherwise, let $\beta \in \mathbb{R}$, $\beta \geq 0$, such that $L(\beta) = ||u - v||_2^2$, then as $m > \beta$, we have $L(m) < ||u - v||_2^2$, which means that it is impossible to make the ellipse contain u, and v. As L is an even function, the same can be said for $m < \beta$. Therefore, we conclude that $\Phi(u, v) = [0, \tan(\beta)] \cup [\pi - \tan(\beta), \pi)$. □

Following that, we introduce a lemma that is responsible for connecting the developments of this chapter with the results of section 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. *Let Q^* be a solution of MCER for the instance $(\mathcal{P}, \mathcal{W}, \{(a, b)\})$, such that $|\mathcal{P} \cap E(q^*, \theta^*)| \geq 2$. If for all $\bar{Q} \succ Q^*$, $|\mathcal{P} \cap \partial E(\bar{q}, \bar{\theta})| < 3$, then there exists $\{u, v\} \subset \mathcal{P} \cap E(q^*, \theta^*)$, such that for all $\theta \in \Phi(u, v)$ there exists $q \in \mathbb{R}^2$, such that (q, θ) is equivalent to Q^*.*

Proof. According to Lemma 3, there exists $\{u, v\} \subset \mathcal{P} \cap E(q^*, \theta^*)$, such that $Q' \succ Q^*$ exists, and $\{u, v\} \subset \partial E(q', \theta^*)$. Therefore, $\theta^* \in \Phi(u, v)$.

Suppose that u and v have the same y-coordinate (if they do not, a rotation can be applied to make them do). Then, by Lemma 4, $\Phi(u, v) = [0, \alpha] \cup [\pi - \alpha, \pi)$, for some $\alpha \in [0, \pi/2]$. Then, if we rotate the coordinate system by $\pi - \alpha$, we obtain $\Phi(u, v) = [0, 2\alpha]$.

With this result in hand, we can use a continuity argument to complete our proof as follows. Let $\delta : \Phi(u, v) \mapsto \mathbb{R}^2$ be a continuous function which takes an angle $\theta \in \Phi(u, v)$ and returns a center, such that $\{u, v\} \subset \partial E(\delta(\theta), \theta)$, and, from solution Q', $\delta(\theta') = q'$. Notice that, in general, for any angle in $\Phi(u, v)$, there are two possible centers that make $\{u, v\} \subset \partial E(\delta(\theta), \theta)$ (see Figure 5 for an example), however, imposing $\delta(\theta') = q'$ makes δ be a well-defined continuous function. This is shown in Figure 6 where δ is plotted for the whole interval $\Phi(u, v)$.

Let $w \in \mathcal{P} \setminus \{u, v\}$, then we define $f_w : [0, \pi) \mapsto \mathbb{R}_{\geq 0}$ to be a function that takes an angle of rotation θ and returns the elliptical distance $|| \cdot ||_{a,b,\theta}$ to the center $\delta(\theta)$; that is $f_w(\theta) = ||w - \delta(\theta)||_{a,b,\theta}$. We have that if $w \in \mathcal{P} \cap E(q^*, \theta^*)$, then $f_w(\theta^*) \leq 1$; and if $w \notin \mathcal{P} \cap E(q^*, \theta^*)$, then $f_w(\theta^*) > 1$.

Therefore, if there exists $\theta \in \Phi(u,v)$, such that for all $q \in \mathbb{R}^2$, (q,θ) is not equivalent to Q^*, then there exists either $w \in \mathcal{P} \cap E(q^*,\theta^*)$, with $f_w(\theta) > 1$, or $w \notin \mathcal{P} \cap E(q^*,\theta^*)$, with $f_w(\theta) \leq 1$. Because f_w is continuous, there exists $\bar{\theta} \in \Phi(u,v)$, such that $f_w(\theta) = 1$, implying that $|\mathcal{P} \cap \partial E(\delta(\bar{\theta}),\bar{\theta})| \geq 3$. □

In Figure 6, a visualization of Lemma 5 is presented. An initial 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 $\Phi(u,v)$ while maintaining u,v on it. Six angles were chosen from $\Phi(u,v)$ to be shown in Figure 6, among those were 0 and $\max\{\Phi(u,v)\}$; their corresponding ellipses are displayed with solid-line borders. Consistently with Lemma 5, 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, which is a solution of E3P.

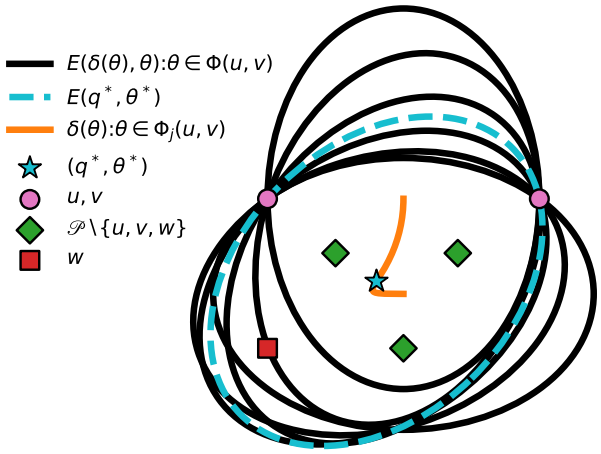


Figure 6: A visualization of Lemma 5.

Definition 4. Let $(\mathcal{P},\mathcal{W},\mathcal{R})$ be an instance of MCER. Then, for all $j \in \{1,\ldots,m\}$, we define the CLS of the j-th ellipse as $S_j = S_j^{(1)} \cup S_j^{(2)} \cup S_j^{(3)}$ with

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

$$S_j^{(2)} = \bigcup_{\{u,v\} \subset \mathcal{P}} \{(q, \angle(u-v)) \in \mathbb{R}^2 \times [0,\pi) : \{u,v\} \subset \partial E_j(q, \angle(u-v))\}$$

$$S_j^{(3)} = \bigcup_{\{u,v,w\} \subset \mathcal{P}} \{(q,\theta) \in \mathbb{R}^2 \times [0,\pi) : \{u,v,w\} \subset \partial E_j(q,\theta)\}.$$

This definition breaks the construction of the CLS 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.

To compute $S_j^{(2)}$, we can observe that, given two points u, v, determining every $q \in \mathbb{R}^2$, such that $\{u, v\} \subset \partial E_j(q, \angle(u-v))$ can be transformed into the problem of determining the set $\partial E_j(u, \angle(u-v)) \cap \partial E_j(v, \angle(u-v))$, which, by Lemma 1, is composed of at most two points, which can be determined analytically. Therefore, we have that $S_j^{(2)}$ can be computed in $\mathcal{O}(n^2)$ operations.

To compute $S_j^{(3)}$, we have to call the algorithm described in section 4 to determine every solution of E3P for every triplet of points in $\mathcal{P}$. Even though that algorithm is $\mathcal{O}(1)$, it has a high constant factor, thus skipping it, in practice, is a good suggestion. Given three points and an ellipse with shape parameters (a, b), the following two conditions are sufficient for E3P to have no solutions, and therefore, if any of them is true, we can skip calling the algorithm to determine every solution of E3P for that instance:

- The maximum distance between any of the points is greater than $2a$;
- 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).

Overall, constructing every ellipse's CLS can be implemented to have a $\mathcal{O}(n^3)$ runtime complexity. Following this, we introduce a theorem, which connects the results for MCER so far, to prove that the set of solutions constructed using the CLSs described by Definition 4 contains an optimal solution.

Theorem 2. *Let $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ be an instance of MCER, and Ω be a set of solutions defined as*

$$\Omega = \{Q \in (\mathbb{R}^2 \times [0, \pi))^m \colon (q_j, \theta_j) \in S_j \text{ for all } j \in \{1, \ldots, m\}\},$$

Then there exists an optimal solution $Q^* \in \Omega$, and $|\Omega| \leq n^{3m}$.

Proof. The first thing to notice is that Ω is defined as the combination of every possible solution from each CLS. To prove that it contains an optimal solution Q^*, we only need to prove that for all $j \in \{1, \ldots, m\}$, there exists $(q_j, \theta_j) \in S_j$, such that $\mathcal{P} \cap E_j(q_j^*, \theta_j^*) \subset \mathcal{P} \cap E_j(q_j, \theta_j)$. To do that, we use Lemma 5 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 Ω 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 there is no $Q' \succ Q^*$, such that $|\mathcal{P} \cap \partial E_j(q_j', \theta_j')| \geq 3$. This case is addressed by Lemma 5, which states that there are equivalent solutions to Q^* with two points $u, v \in \mathcal{P} \cap E_j(q_j^*, \theta_j^*)$ on the j-th ellipse for every (E_j, u, v)-feasible angle. As $\angle(u - v)$ is a (E_j, u, v)-feasible angle, we have that there exists $(q_j, \theta_j) \in S_j^{(2)}$, such that $\mathcal{P} \cap E_j(q_j, \theta_j) = \mathcal{P} \cap E_j(q_j^*, \theta_j^*)$. Also, by Lemma 1, 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 Ω. Also, by Lemma 2 we can conclude 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 $|S_j|$:

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

Therefore, we conclude that $|\Omega| \leq |S_1| \times \cdots \times |S_m| \leq n^{3m}$. □

Finally, we define Algorithm 2, which backtracks every possible combination of solutions considering the CLS of every ellipse. As evaluating each solution can be implemented to take $\mathcal{O}(nm)$ operations, we have that Algorithm 2 has a $\mathcal{O}(mn^{3m+1})$ runtime complexity. In the next section, we describe some implementation details and improvements that, in practice, can lower the size of S_j significantly, and also can make the backtracking process described in Algorithm 2 skip many non-optimal solutions.

Algorithm 2 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.

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

procedure $MCER_{bt}(Z, \mathcal{W}, \mathcal{R}, j)$
    $(q_j^*, \theta_j*); \ldots; (q_m^*, \theta_m^*) \leftarrow (0,0); \ldots; (0,0)$ ▷ Setting to 0 as a default value.
    Let S_j be the CLS for the j-th ellipse as defined in Definition 4
    for all $(q_j, \theta_j) \in S_j$ **do**
        if $j < m$ **then**
            $(q_{j+1}, \theta_{j+1}); \ldots; (q_m, \theta_m) \leftarrow MCER_{bt}(Z \setminus Cov, \mathcal{W}, \mathcal{R}, j+1)$
        end if
        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**
            $(q_j^*, \theta_j*); \ldots; (q_m^*, \theta_m^*) \leftarrow (q_j, \theta_j); \ldots; (q_m, \theta_m)$
        end if
    end for
    return $(q_j^*, \theta_j*); \ldots; (q_m^*, \theta_m^*)$
end procedure

6. Improvements

In this section, we describe some improvements that can be applied to the implementation of Algorithm 1 and Algorithm 2, which, in practice, has shown to increase the efficiency of the algorithms proposed by us.

6.1. Reducing the CLS size

As for the algorithms for both MCE and MCER, the number of solutions they go through is directly proportional to the size of each ellipse's CLS, reducing their size can significantly improve the performance of both algorithms.

For MCE (the MCER's case is analogous), let $q, q' \in S_j$ be two possible locations in the CLS for the j-th ellipse. If $\mathcal{P} \cap E_j(q') \subset \mathcal{P} \cap E_j(q)$, then q' is redundant and we can remove it from S_j, as it produces a solution which is either non-optimal or equivalent to an optimal one.

In [1], for the Euclidean PMCLP, this reduction to the CLSs is also employed, and after analyzing some experiments, they concluded that after the removal of redundant locations, the CLSs gets much smaller, and although the reduction step can be quite costly, in the end, it is worth it.

To remove redundant elements from a CLS, we use the same tree-like data structure described in [6], which keeps every maximal subset of covered points by an ellipse, and supports a query operation to verify if a subset is maximal or not. First, we sort the elements in S_j by the number of covered demand points, non-decreasingly. Then, we iterate over it, removing elements which make the ellipse cover non-maximal subsets of demand points, when compared to the elements of S_j that have already been processed.

6.2. Prunning the Backtracking Tree

In this section, we introduce a condition for skipping solutions in the backtracking process in the algorithms for MCE and MCER. The idea is to prune the backtracking tree by observing that any solution constructed going down the current branch will have a value less than or equal to the current best solution. As this condition can be applied for both problems, and their notation differs very little, we describe it only for MCE.

Given an instance $(\mathcal{P}, \mathcal{W}, \mathcal{R})$ of MCE, an upper-bound for the value of an optimal solution is the sum of the optimal solutions for each ellipse individually:

$$\max_{Q} w\left(\bigcup_{j=1}^{m} \mathcal{P} \cap E_j(q_j)\right) \leq \sum_{j=1}^{m} \max_{q_j} w(\mathcal{P} \cap E_j(q_j)). \tag{5}$$

Suppose that the first k ellipses are fixed at locations $(q_1, \ldots, q_k)$, and that we have an lower-bound L for the value of an optimal solution. Let $Z_k = \mathcal{P} \setminus \cup_{j=1}^{k} E_j(q_j)$ be the points not covered by the first k ellipses, then we can use Equation 5 to verify if we can skip every solution where the first k ellipses are fixed at $(q_1, \ldots, q_k)$ or not. If

$$w\left(\bigcup_{j=1}^{k} \mathcal{P} \cap E_j(q_j)\right) + \sum_{j=k+1}^{m} \max_{q_j} w(Z_j \cap E_j(q_j)) \leq L, \tag{6}$$

then, any solution with the first k ellipses fixed at $(q_1, \ldots, q_k)$ will have value less than or equal to the value of an optimal solution, therefore, we can cut the backtracking tree there. In practice, we can use the value of the best solution found at the moment as the lower-bound L.

It is worth mentioning that this improvement do not have an effect in a possible worst case scenario. We decided to adopt it in our implementation because it showed good results in practice. For example, without it, MCER-k's algorithm takes nine seconds to obtain an optimal solution for instance AB060 developed by [6], going through 336,494,451 solutions, while the implementation using Equation 6 to prune the backtracking tree for the same instance takes less than one second to return an optimal solution, and evaluates only 1809 solutions in total.

7. Implementation Details

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/`.

7.1. Determining the eigenvalues of a matrix

In the algorithm to obtain every solution of E3P described in section 4, we assumed that a procedure which returns every eigenvalue of a given complex matrix was available. In practice, we used the famous linear algebra package LAPACK (see [13] 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.

To compute every eigenvalue of a complex matrix, LAPACK offers a routine called ZGEEV, which is 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.

7.2. Symbolic computation

Back in section 4, we were faced with the problem of writing function ξ defined in Equation 4 as a complex polynomial in the new variable $z = e^{i\theta}$. We suggested that symbolic computation should be used for this task, as the expressions for that polynomial's coefficients become very long, and doing that by hand is, to say the least, a very tedious work. Symbolic computation is a vast topic, which deals with the problem of solving or manipulating mathematical expressions computationally. In practice, we utilized an external library for Python called SymPy (see [15] 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 wrote $\xi(\theta)(e^{i\theta})^6$ as a polynomial by replacing the sine and cosine functions by the identities $\cos\theta = (z + z^{-1})/2$ and $\sin\theta = (z + z^{-1})/(2i)$.

The actual coefficients of that polynomial would take more than ten pages if they were to be presented here, because of that, we made them available elsewhere in `https://sites.icmc.usp.br/andretta/tedeschi-2020/`.

8. 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. 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.

8.1. Numerical Results for known instances

In this section, we present the results of our algorithms for MCE-k and MCER-k for some instances proposed by [5, 6]. We present here the numerical results for the instances CM6-CM9, and AB097-AB120. For each instance, we display the selected ellipses and the income, which is the weight of every covered point minus the cost of the selected ellipses, 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 were skipped. We made available at `https://sites.icmc.usp.br/andretta/tedeschi-2020/` every instance used here, along with the graphical representation of every obtained solution.

In Table 1, the results for instances CM7-CM9 are shown for MCE-k. 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 CM7-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 [6]. For example, to obtain an optimal solution for the instance CM9, the method proposed by [6] took more than thirty minutes. In Table 3, we present the results for instances AB097-AB120. We can also observe here, that in practice, the bound for the CLS size of n^2 given by Theorem 1 seems to be very loose. The closest we got to this number was in instances CM7-CM9 where $|S_3| = 174$, which is still very far from $n^2 = 100^2 = 10{,}000$.

For MCER-k, the numerical results obtained by our implementation are shown in Table 2 for instances CM7-CM9, and in Table 4 for instances AB097-AB120. An optimal solution was obtained for every instance, and overall, at most six seconds of CPU time was taken.

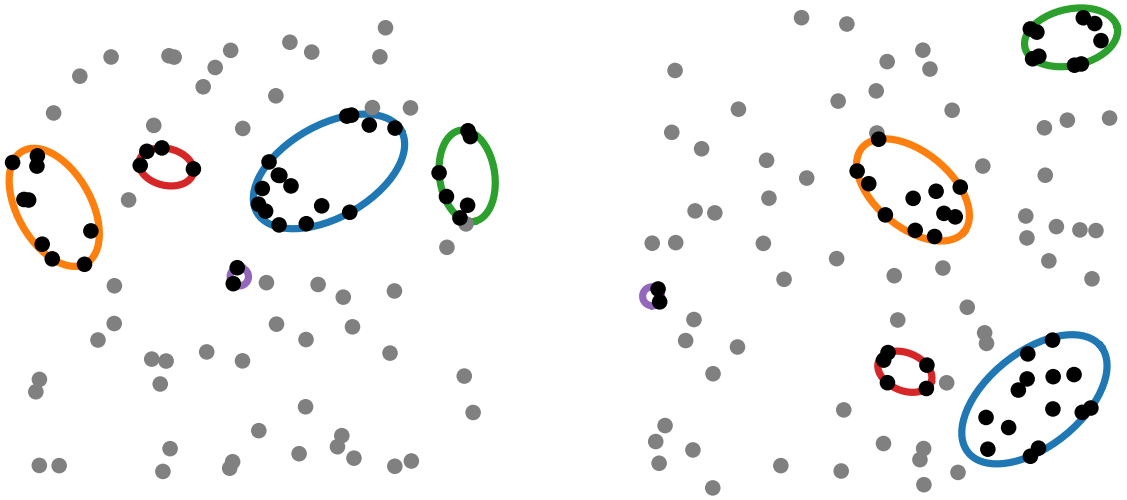
Looking at the numerical results of the heuristic method proposed in [6], the only non-optimal solutions it encountered were for instances AB105-AB108. For these instances, our algorithm obtained an optimal solution covering one more point. In Figure 7a, the optimal solution for AB108 is displayed. In general, our algorithm took much lower CPU time compared to the methods developed in [6]. For example, for instance CM9, their heuristic method took more than six hours to return a solution, and their deterministic one exceeded the predefined time limit of twelve hours, while our implementation of the MCER-k's algorithm took less than five seconds of CPU time.

8.2. New instances

After examining the results obtained for the formerly known instances, we decided to construct new ones to analyze the algorithms proposed by our work more thoroughly.

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, for some instances, we used a different probability distribution, other than the uniform distribution, to generate the location of the points. We set a time limit of two hours of CPU time for 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 47 new instances, which will be referred to as TA01, ..., TA47, we made all of them available at `https://sites.icmc.usp.br/andretta/tedeschi-2020/`.

The first set of instances, TA01-TA07, were constructed sampling each demand point 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



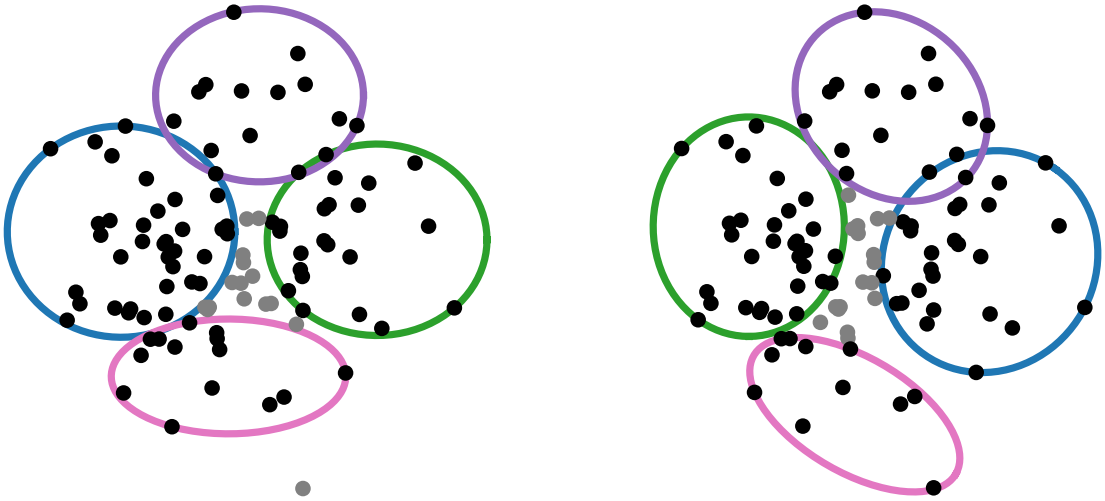
(a) (b)

Figure 7: An optimal solution of MCER-k for the instance AB108 (a), and for the instance AB120 (b).

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, with the most valuable ones 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 for MCE-k and MCER-k taking $k \in \{1, \ldots, m\}$. The results for MCE-k are presented in Table 5 and the results for MCER-k are displayed in Table 6. The optimal solutions for the instance TA04 for MCE-k and MCER-k are displayed in Figure 8. There it is possible to see that because of the normal distribution, most of the points are located close to each other, near the origin, making every ellipse's CLS end up being bigger compared to the previously introduced instances with the same number of demand points. This, and the 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 two hours for the instances TA05-TA07, while the algorithm for MCE-k did not finish in time only for the instance TA07.

For the second set of instances, TA08-TA22, we generated the demand set following the same process as for instances TA01-TA07. We kept the number of facilities at 3 and created five demand sets with $n \in \{200, 250, 300, 350, 400\}$. In total, we had 15 instances with $k \in \{1, \ldots, m\}$. The results for MCE-k are displayed in Table 7 and the results for MCER-k are presented in Table 8. Our implementation of the algorithm for MCER-k was not able to obtain a solution for the last instance TA22. Apart from instance TA13 for MCER-k, and instance TA22 for both algorithms, most of the CPU time was spent in constructing the CLSs. The graphical representation of solutions for the instance TA21 for MCE-k and MCER-k are shown in Figure 9.

The third set of instances, TA23-TA42, was constructed generating each demand point following a uniform distribution in $[0, 10]^2$, with each point having unitary weight; and the ellipses by the same process used for instances TA01-TA23. We created instances with $m = 5$, $n \in \{400, 500, 600, 700\}$,

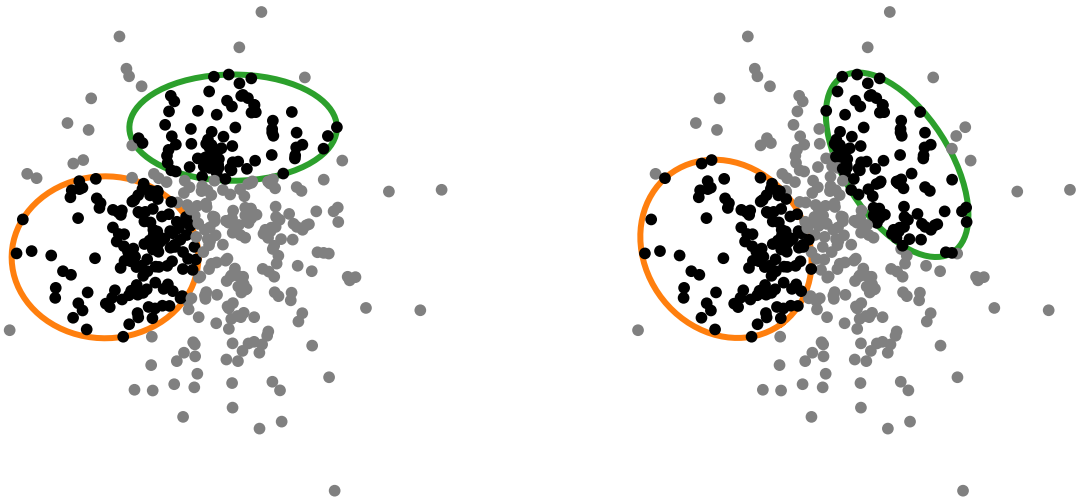


(a) (b)

Figure 8: Two optimal solutions for the instance TA04: (a) for MCE-k, and (b) for MCER-k.

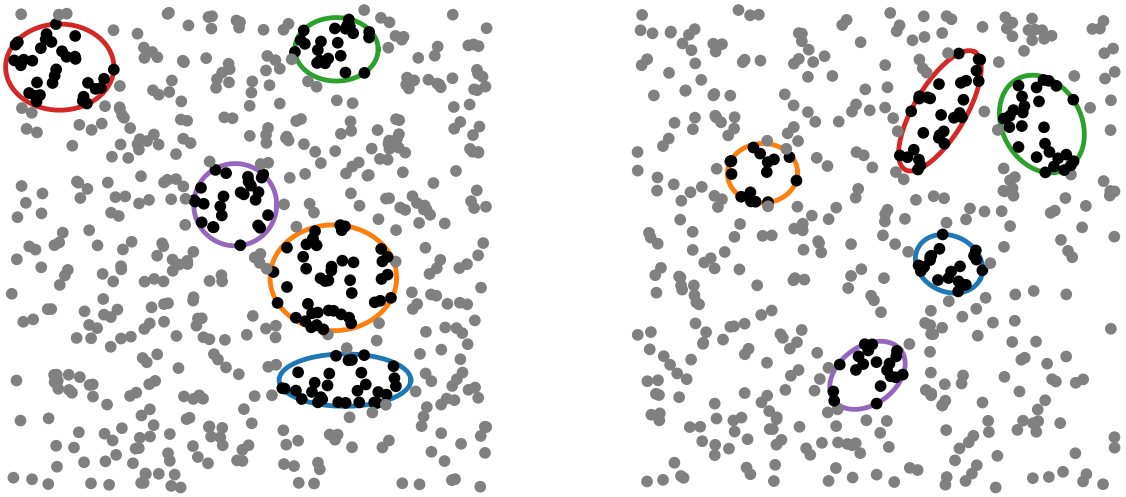
and $k \in \{1, \ldots, m\}$, with a total of 20 instances. The results for MCE-k can be seen in Table 9 and the results for MCER-k are presented in Table 10. Optimal solutions were obtained for every one of the instances in this set. It is possible to see that, compared with the first two sets of instances, 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 TA37 with $n = 500$ and $k = 5$ is shown in Figure 10.

The last set of instances, TA43-TA47, were constructed using two bivariate normal distributions with distinct means $\mathcal{N}(\mu^{(1)}, \mathbb{I})$ and $\mathcal{N}(\mu^{(2)}, \mathbb{I})$, $\mu^{(1)}, \mu^{(2)} \in \mathbb{R}^2$. Half of the points were generated following $\mathcal{N}(\mu^{(1)}, \mathbb{I})$, and the other half $\mathcal{N}(\mu^{(2)}, \mathbb{I})$; 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 $k \in \{1, \ldots, m\}$. We defined the values of $\mu^{(1)}$ and $\mu^{(2)}$ specifically to create such a counter-example. The results are shown in Table 11 for MCE-k and in Table 11 for MCER-k. In Figure 11, we show the solutions for the instances TA44-TA45 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.



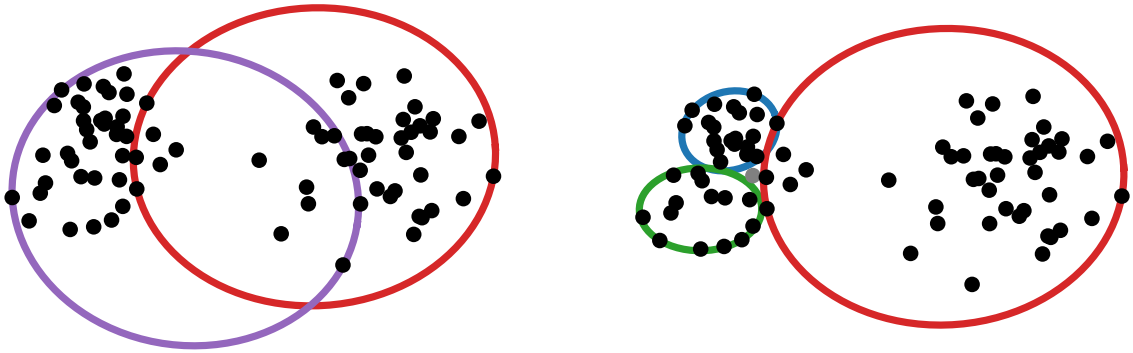
(a) (b)

Figure 9: Two optimal solutions for the instance TA21 with 400 points: (a) for MCE-k, and (b) for MCER-k.



(a) (b)

Figure 10: Two optimal solutions for the instance TA37: (a) for MCE-k, and (b) for MCER-k.



(a) (b)

Figure 11: Two optimal solutions for the instance TA44: (a) for MCE-k, and (b) for MCER-k.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	Backtracking Tree # nodes	Backtracking Tree # sol. leaves	CPU Time (s) CLS-MCE	CPU Time (s) Total
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 1: Numerical Results of MCE-k for instances CM7-CM9.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	# E3P subproblems	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) Total
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 2: Numerical Results of MCER-k for instances CM7-CM9.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	Backtracking Tree # nodes	Backtracking Tree # sol. leaves	CPU Time (s) CLS-MCE	CPU Time (s) Total
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 3: Numerical Results of MCE-k for instances AB097-AB120.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	# E3P subproblems	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) Total
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 4: Numerical Results of MCER-k for instances AB097-AB120.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) 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 5: Numerical Results of MCE-k for instances TA001-TA007.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	# E3P subproblems	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) 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 6: Numerical Results of MCER-k for instances TA001-TA007.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) 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
TA12	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
TA15	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 7: Numerical Results of MCE-k for instances TA008-TA022.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	# E3P subproblems	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) 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 8: Numerical Results of MCER-k for instances TA008-TA022.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) 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 9: Numerical Results of MCE-k for instances TA023-TA042.

Instance				Optimal Solution		Performance metrics				
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	# E3P subproblems	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) 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 10: Numerical Results of MCER-k for instances TA023-TA042.

Instance				Optimal Solution		Performance metrics			
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) 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 11: Numerical Results of MCE-k for instances TA043-TA047.

Instance				Optimal Solution		Performance metrics					
Name	n	m	k	Selected Ellipses	Income	CLS size $\lvert S_k \rvert$	# E3P subproblems	Backtracking Tree # nodes	Backtracking Tree # sol leaves	CPU Time (s) CLS-MCER	CPU Time (s) 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 12: Numerical Results of MCE-k for instances TA043-TA047.

References

[1] R. L. Church, The planar maximal covering location problem. (symposium on location problems: in memory of leon cooper), Journal of Regional Science 24 (2) (1984) 185–201. doi:10.1111/j.1467-9787.1984.tb01031.x.
URL https://doi.org/10.1111/j.1467-9787.1984.tb01031.x

[2] Z. Drezner, Note—on a modified one-center model, Management Science 27 (1981) 848–851. doi:10.1287/mnsc.27.7.848.

[3] B. M. Chazelle, D. Lee, On a circle placement problem, Computing 36 (1986) 1–16. doi:10.1007/BF02238188.

[4] M. de Berg, S. Cabello, S. Har-Peled, Covering many or few points with unit disks, Theory of Computing Systems 45 (3) (2008) 446–469. doi:10.1007/s00224-008-9135-9.
URL https://doi.org/10.1007/s00224-008-9135-9

[5] M. S. Canbolat, M. von Massow, Planar maximal covering with ellipses, Computers and Industrial Engineering 57 (2009) 201–208.

[6] M. Andretta, E. Birgin, Deterministic and stochastic global optimization techniques for planar covering with ellipses problems, European Journal of Operational Research 224 (1) (2013) 23–40. doi:10.1016/j.ejor.2012.07.020.
URL https://doi.org/10.1016/j.ejor.2012.07.020

[7] P. Martín, H. Martini, Algorithms for ball hulls and ball intersections in normed planes, Journal of Computational Geometry Vol 6 (2015) No 1 (2015)–. doi:10.20382/JOCG.V6I1A4.
URL https://journals.carleton.ca/jocg/index.php/jocg/article/view/187

[8] E. W. Weisstein, Circumcircle From MathWorld—A Wolfram Web Resource, last visited on 9/4/2020.
URL http://mathworld.wolfram.com/Circumcircle.html

[9] R. Johnson, Y. Young, Advance Euclidean Geometry (modern Geometry): An Elementary Treatise on the Geometry of the Triangle and the Circle, Dover books on advanced mathematics, Dover, 1960.
URL https://books.google.com.br/books?id=HdCjnQEACAAJ

[10] M. J. D. M. J. D. Powell, Approximation theory and methods, Cambridge [England] ; New York : Cambridge University Press, 1981, includes index.

[11] R. A. Horn, C. R. Johnson (Eds.), Matrix Analysis, Cambridge University Press, New York, NY, USA, 1986.

[12] D. S. Watkins, The qr algorithm revisited, SIAM Rev. 50 (1) (2008) 133–145. doi:10.1137/060659454.
URL http://dx.doi.org/10.1137/060659454

[13] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, 3rd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[14] P. Weidner, The durand-kerner method for trigonometric and exponential polynomials, Computing 40 (2) (1988) 175–179. doi:10.1007/BF02247945.
URL https://doi.org/10.1007/BF02247945

[15] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz, Sympy: symbolic computing in python, PeerJ Computer Science 3 (2017) e103. doi:10.7717/peerj-cs.103.
URL https://doi.org/10.7717/peerj-cs.103