Planar Maximal Covering with Ellipses

Danilo F. Tedeschi
Orientadora: Dra. Marina Andretta

Instituto de Ciências Matemáticas e Computação

May 4, 2019

Agradecimentos à CAPES.

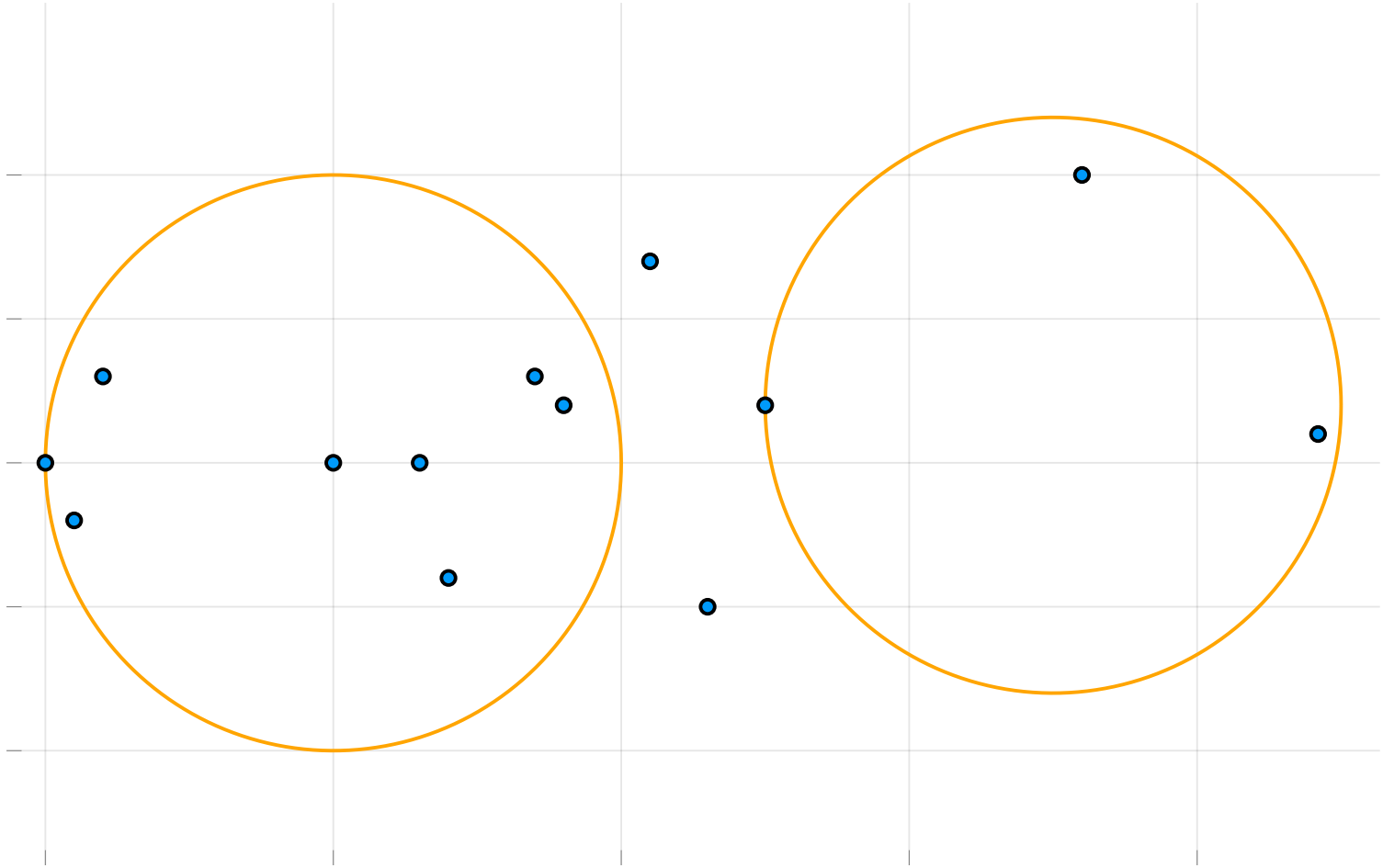
Contents

1. Introduction
2. Preliminaries
3. Maximal Covering by Disks
   - Maximum Weight Clique Problem
4. Maximal Covering by Ellipses
5. Future Work

Introduction

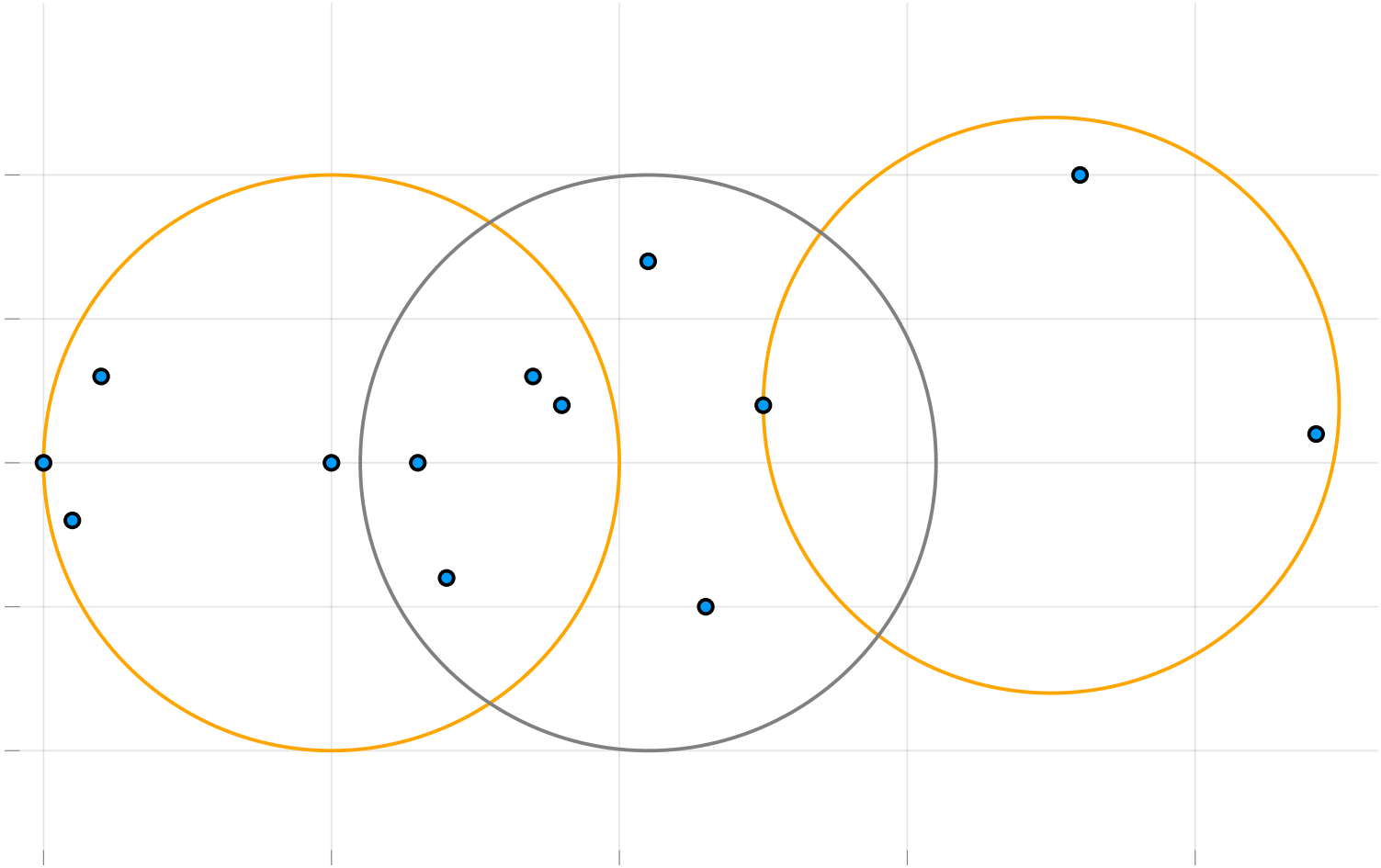
- Covering problems
  - Minimum Cover Problem
  - Maximal Covering Problem (**church:1974**)
- Maximal Covering Location Problem (MCLP)
- Planar Maximal Covering Location Problem (PMCLP)
  - One disk: $\mathscr{O}(n^2)$ and $\mathscr{O}(n^2 \log n)$ algorithms
  - m disks: $\mathscr{O}(n^{2m-1} \log n)$ algorithm
- Goals
  - Develop a $\mathscr{O}(n^2 \log n)$ algorithm for the one disk case
  - Adapt it for the m ellipses case creating a $\mathscr{O}(n^{2m})$ algorithm

Figure: Maximal cover by two disks.



Source: Elaborated by the author.

Figure: Minimum number of disks needed to cover the set of points.



Source: Elaborated by the author.

Maximal Covering by Disks

One disk

$MCD(\mathscr{P}, 1)$ is the problem of placing one disk on the plane to cover a subset of a demand set $\mathscr{P}$ maximizing the weights of the covered points.

$$\max_q w(\mathscr{P} \cap D(q)),$$

- $\mathscr{P} = \{p_1, \ldots, p_n\}$ is the demand set with weights $w(p_i) > 0$
- $w(A)$, $A \subset \mathscr{P}$, is the sum of weights of the points in A
- $D(q)$ is a unit disk with center at point q
- (**chazelle:1986**) proposed a $\mathscr{O}(n^2)$ algorithm
- (**drezner**) proposed a $\mathscr{O}(n^2 \log n)$ which our work is based on
- We will introduce an equivalent problem...

Maximum Weight Clique Problem

Let $\mathscr{D} = \{D_1, \ldots, D_n\}$ be a set of n unit disks with weights $w_i > 0$. The maximum weight clique is defined as

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

- The disks are fixed with centers at $\mathscr{P} = \{p_1, \ldots, p_n\}$ with $w_k = w(p_k)$
- A clique is a non-empty intersection area of a subset of disks. We search for only a point in an optimal clique.
- An optimal solution for the maximum weight clique is an optimal solution for $MCD(\mathscr{P}, 1)$.

Figure: An instance of the Maximum Weight Clique Problem obtained from an instance of $MCD(\mathscr{P}, 1)$.

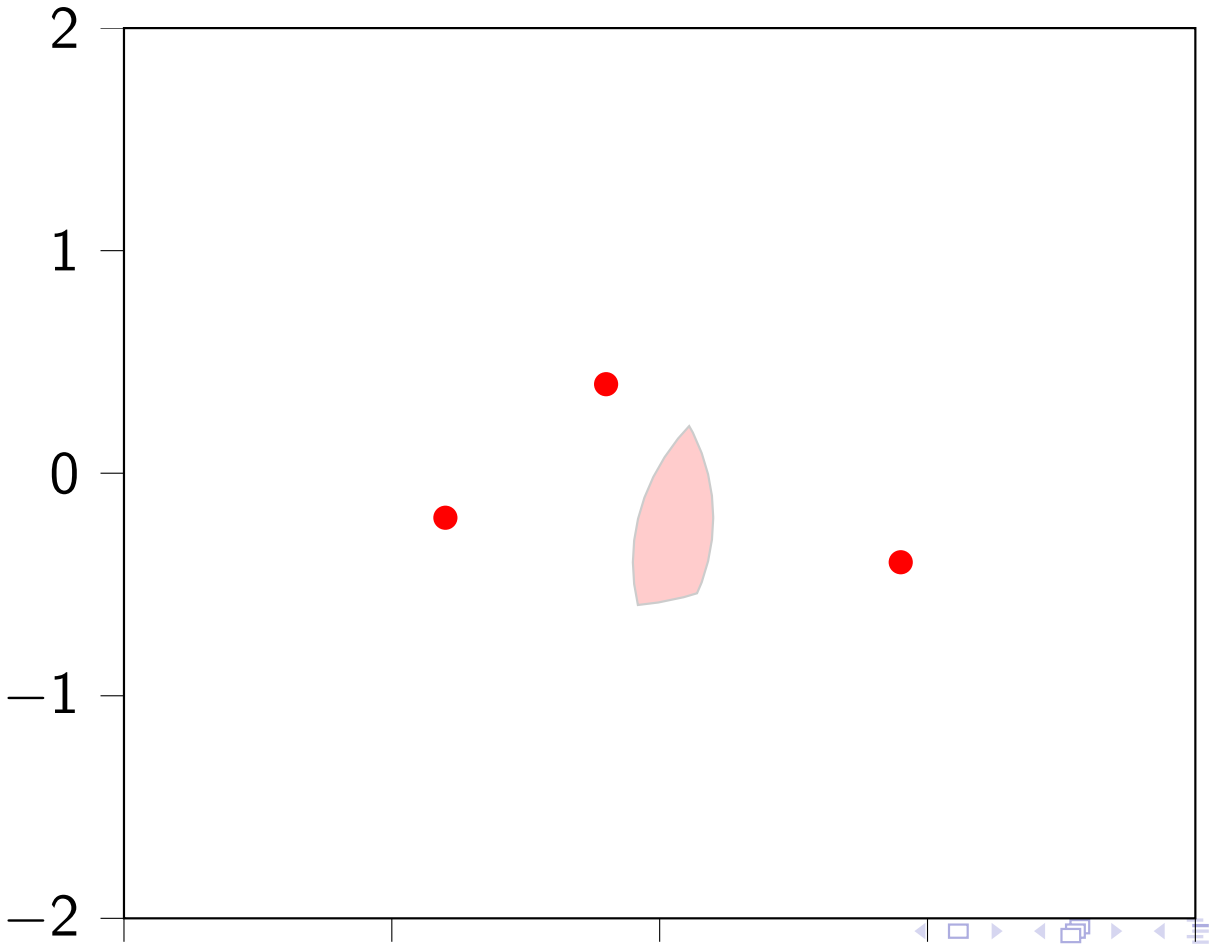
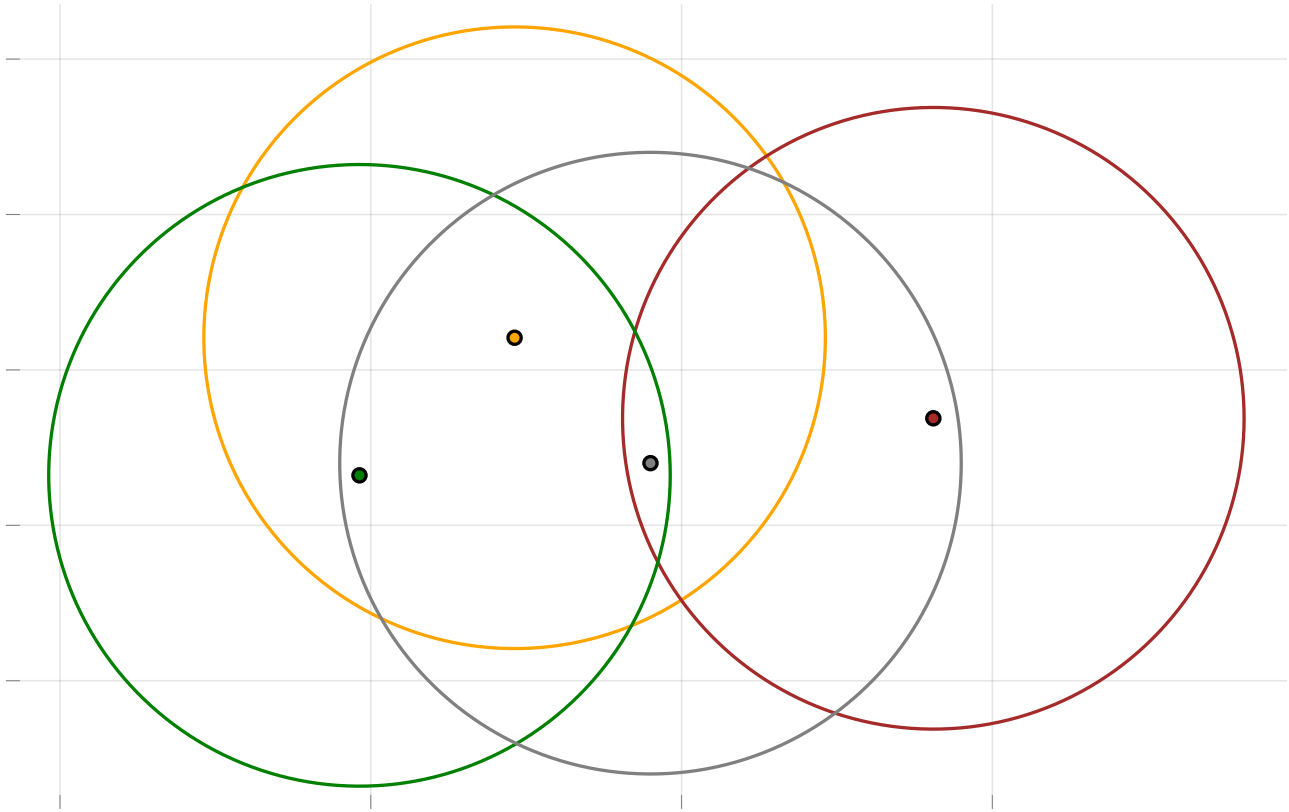


Figure: An instance of the Maximum Weight Clique Problem obtained from an instance of $MCD(\mathscr{P}, 1)$. In gray, the optimal solution.

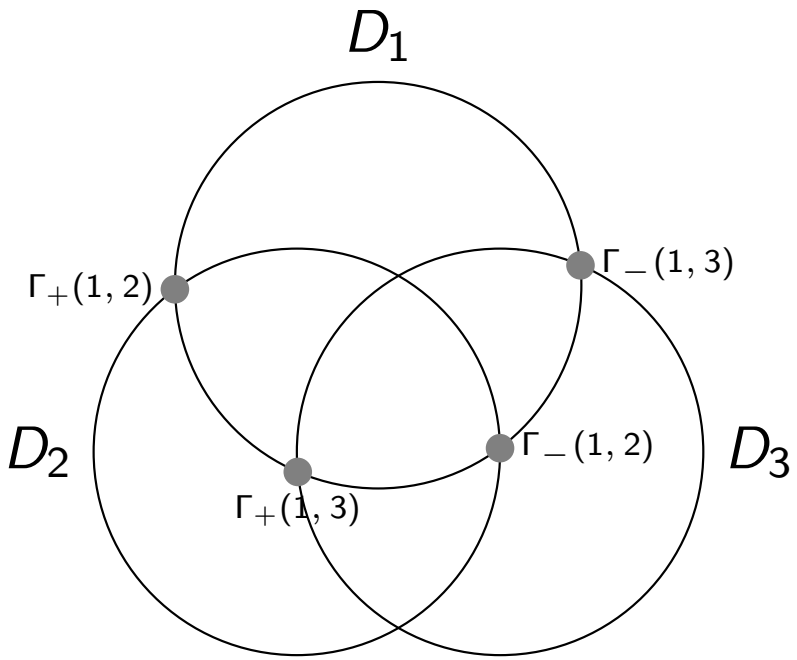


Source: Elaborated by the author.

Defining $\Gamma_+(i,j)$ and $\Gamma_-(i,j)$:
Let D_i and D_j be two unit disks that intersect at two points.

- Assume D_i is at the origin
- We can find the polar angle of the intersection points. Let's call them $\Gamma_+(i,j)$ and $\Gamma_-(i,j)$ with the following condition:
  - The arc defined by $\Gamma_+(i,j)$ and $\Gamma_-(i,j)$ bounds the region $D_i \cap D_j$
- Later it will be shown how to find which one is which algorithmically.

Figure: Three disks and their intersection points and angles.



Source: Elaborated by the author.

Maximum Weight Clique Problem

Algorithm

For a disk D_i, a counter-clockwise traversal visits every $\Gamma_+(i,j)$ and $\Gamma_-(i,j)$ in counter-clockwise order.

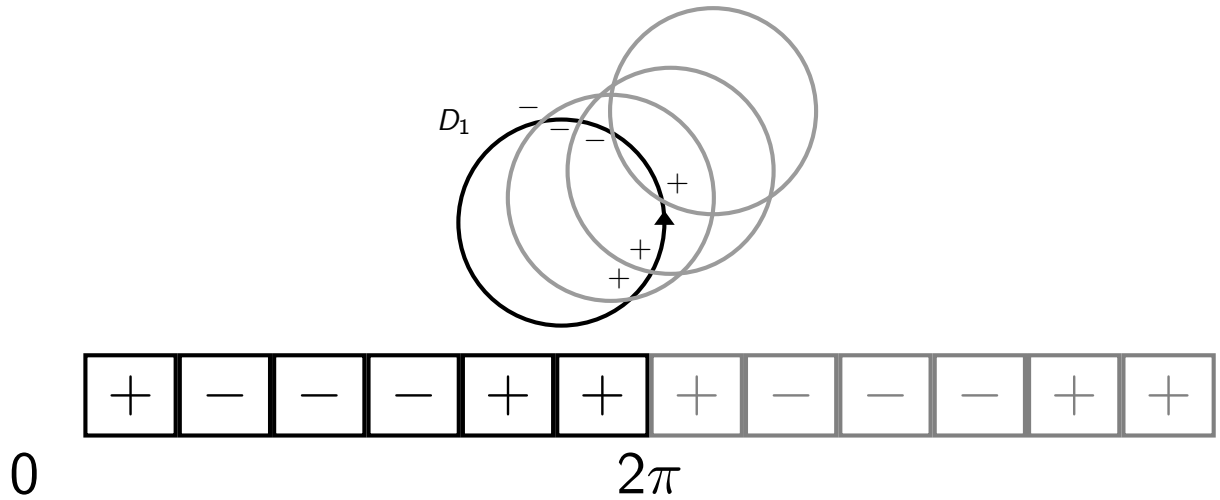
- An intersection region of disks is bounded by arcs.
- The arc $\Gamma_+(i,j), \Gamma_-(i,j)$ (counter-clockwise) determines a region where i and j intersect.
- In a counter-clockwise traversal, the arcs where $\Gamma_+(i,j) > \Gamma_-(i,j)$ can be a problem for the implementation. The traversal is on a circle, where to start?
- If the traversal reaches a opening angle, we mark the intersecting disk as active. If it reaches a closing angle we unmark it.
- We want to find a point in the traversal with the most active disks.

Maximum Weight Clique Problem

Algorithm

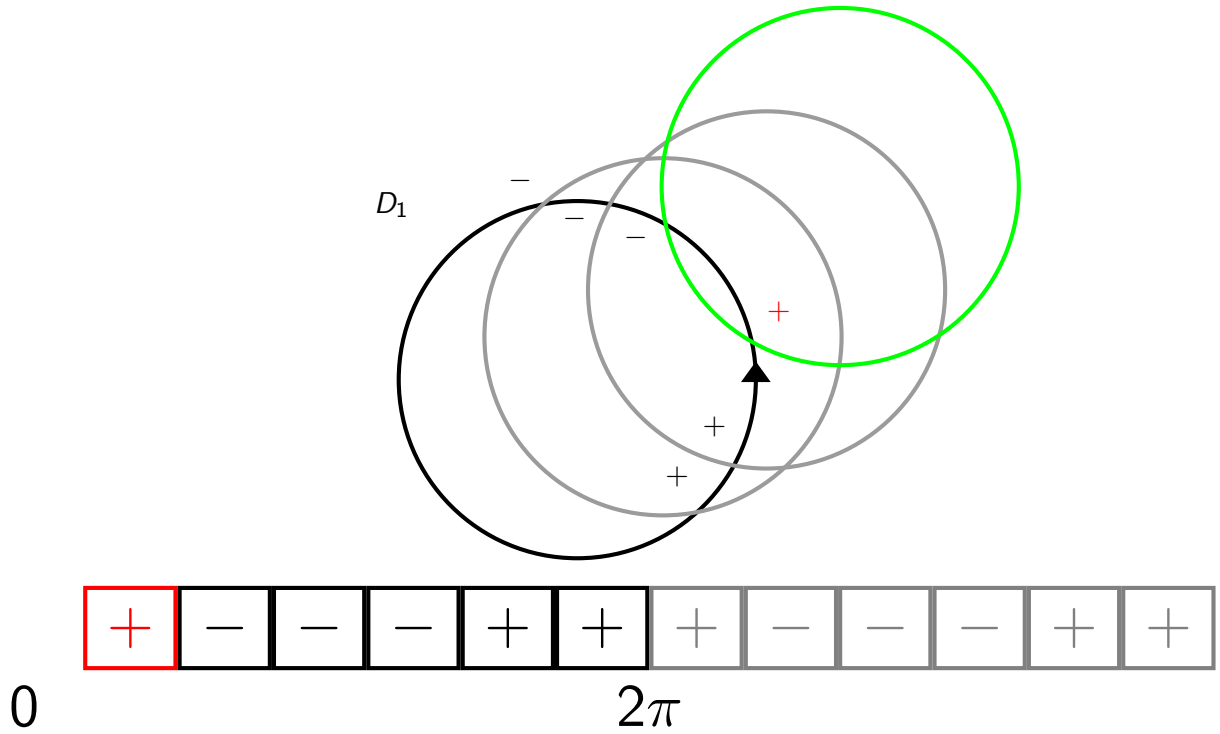
For every disk, traverse the sorted list of intersection angles twice, keeping a set of active disks, and the current best solution.
Doing the traversal twice solves the problem of deciding where to start.

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



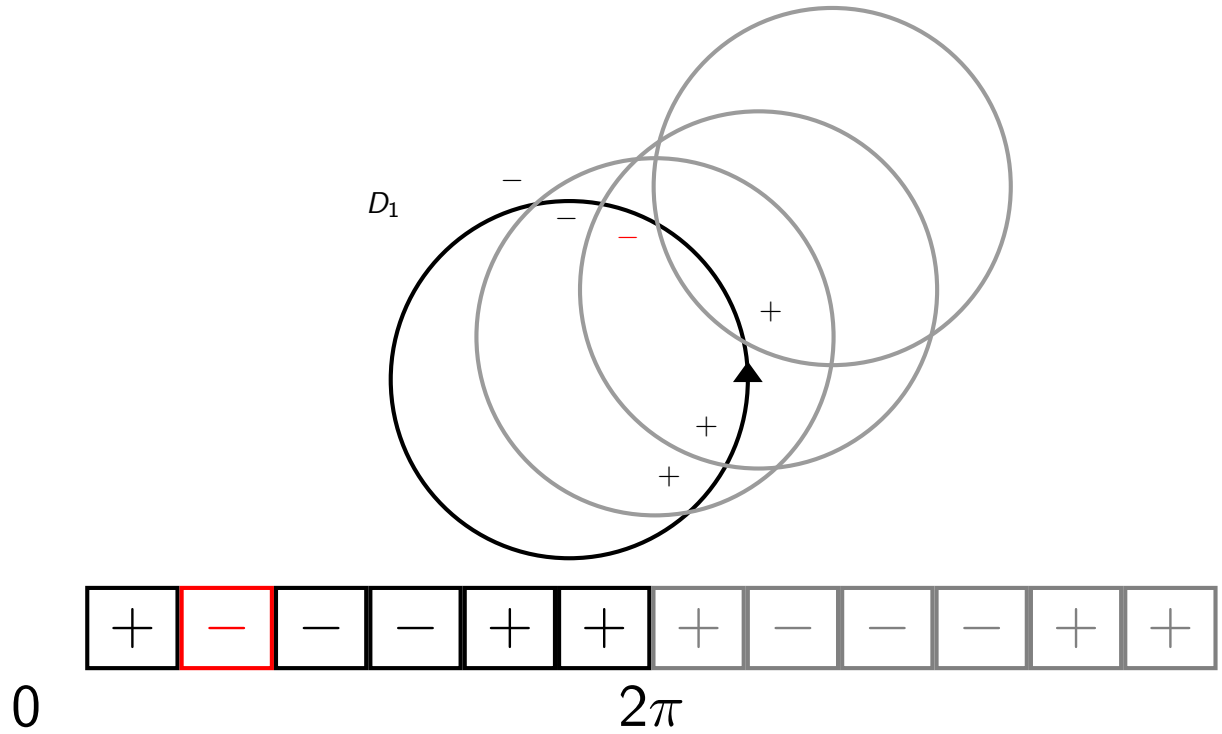
Source: Elaborated by the author.

Figure: A traversal for D_1 with green disks representing the active set and red signs representing the current angle being visited (some are omitted).



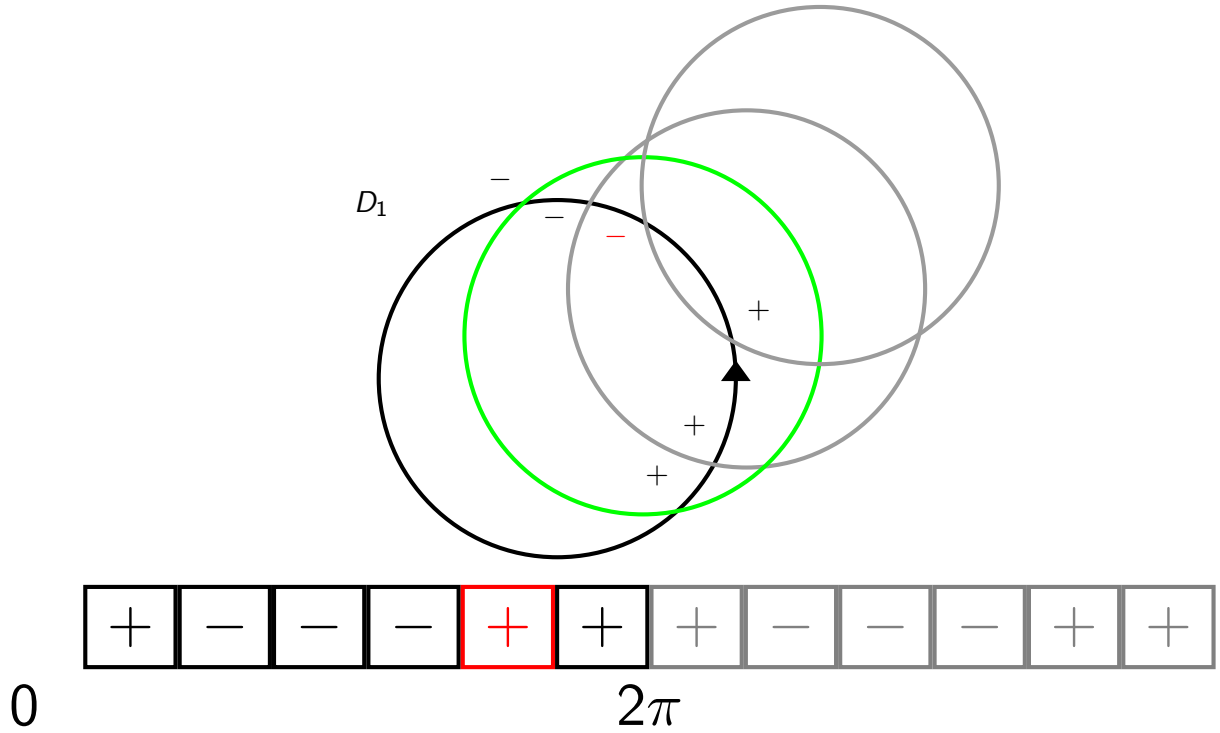
Source: Elaborated by the author.

Figure: A traversal for D_1 with green disks representing the active set and red signs representing the current angle being visited (some are omitted).



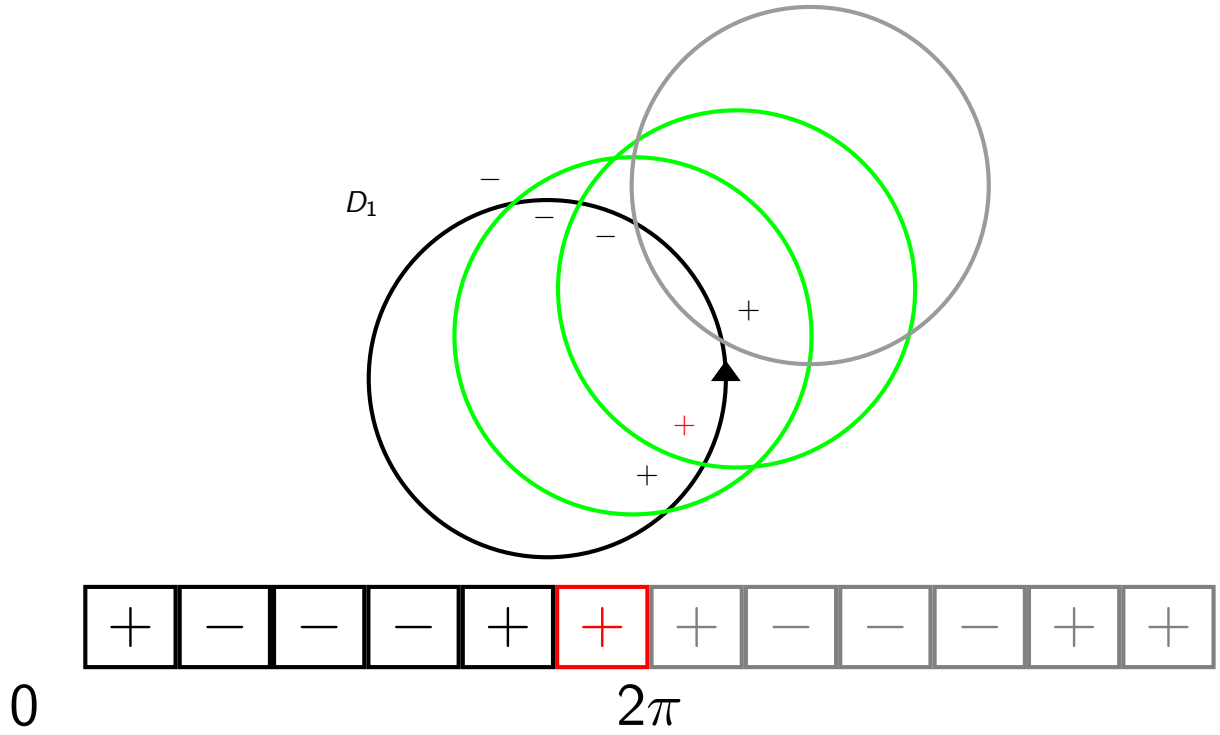
Source: Elaborated by the author.

Figure: A traversal for D_1 with green disks representing the active set and red signs representing the current angle being visited (some are omitted).



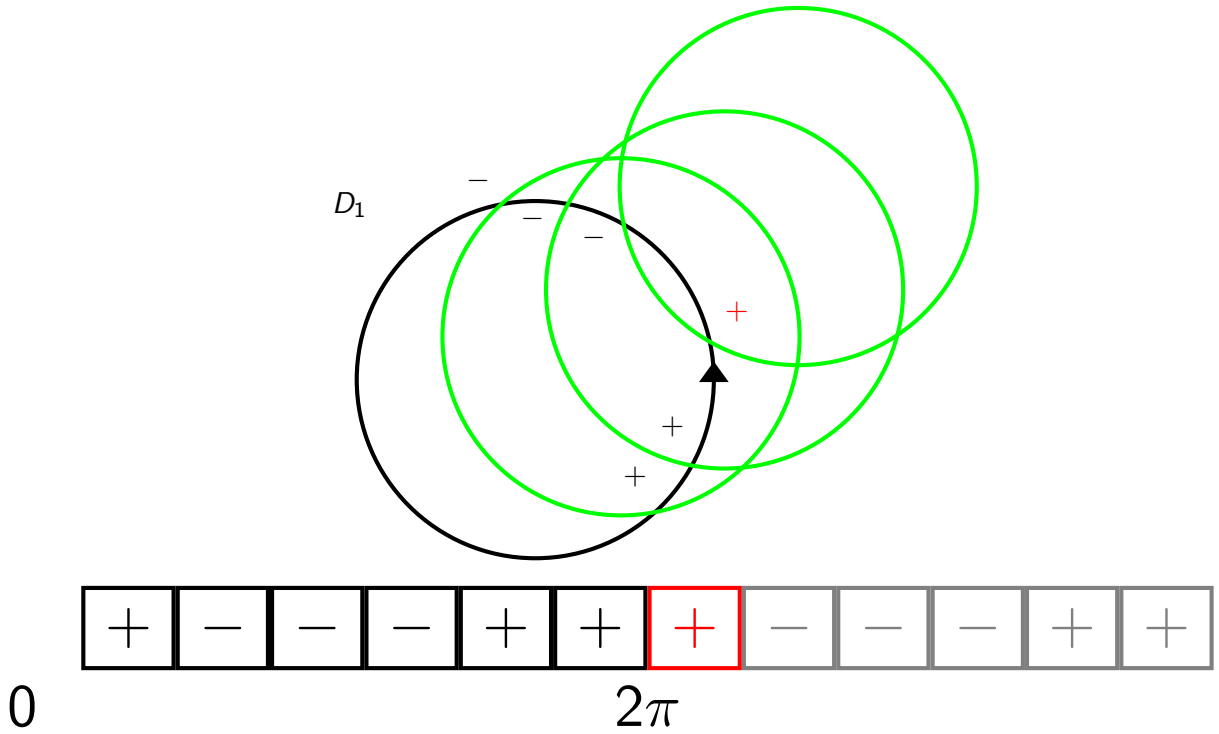
Source: Elaborated by the author.

Figure: A traversal for D_1 with green disks representing the active set and red signs representing the current angle being visited (some are omitted).



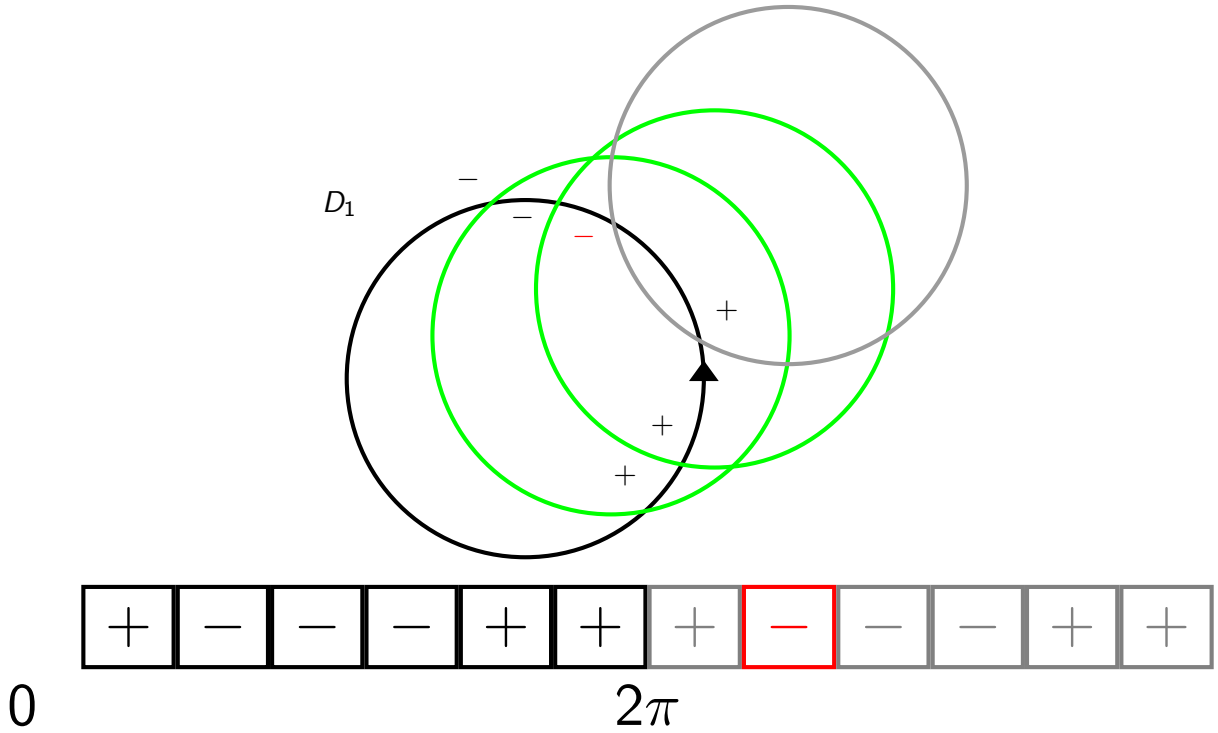
Source: Elaborated by the author.

Figure: A traversal for D_1 with green disks representing the active set and red signs representing the current angle being visited (some are omitted).



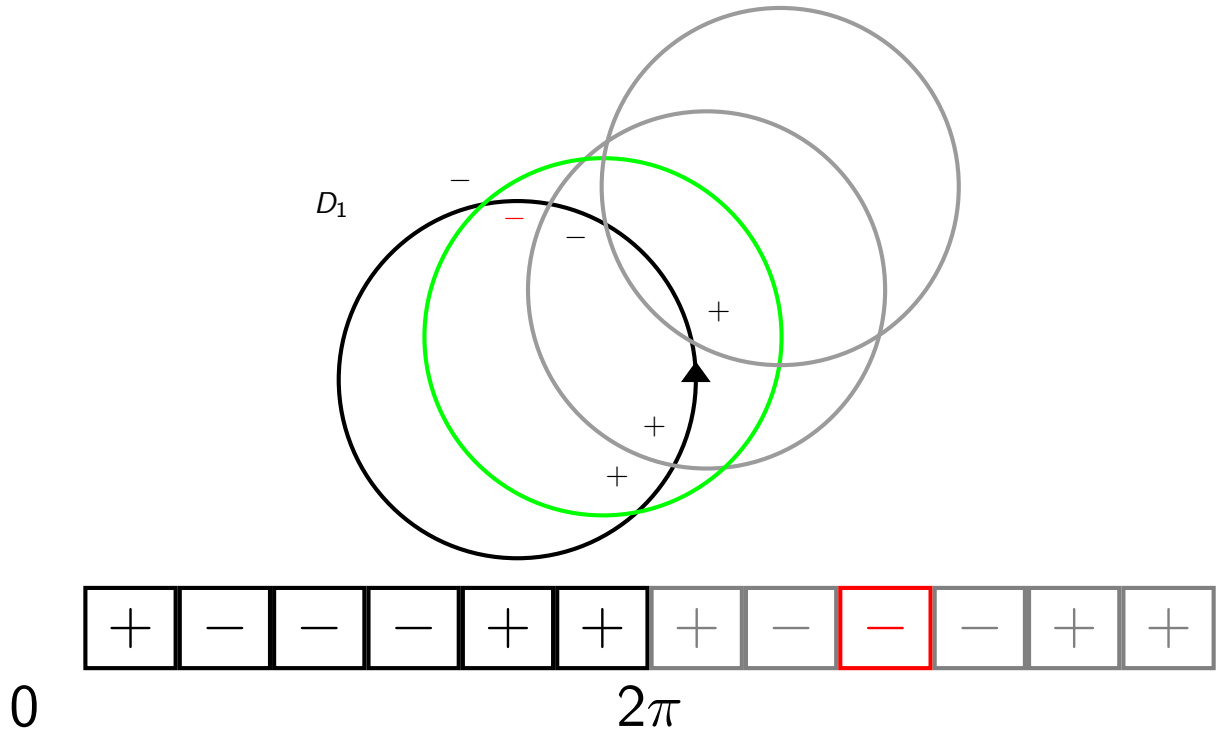
Source: Elaborated by the author.

Figure: A traversal for D_1 with green disks representing the active set and red signs representing the current angle being visited (some are omitted).



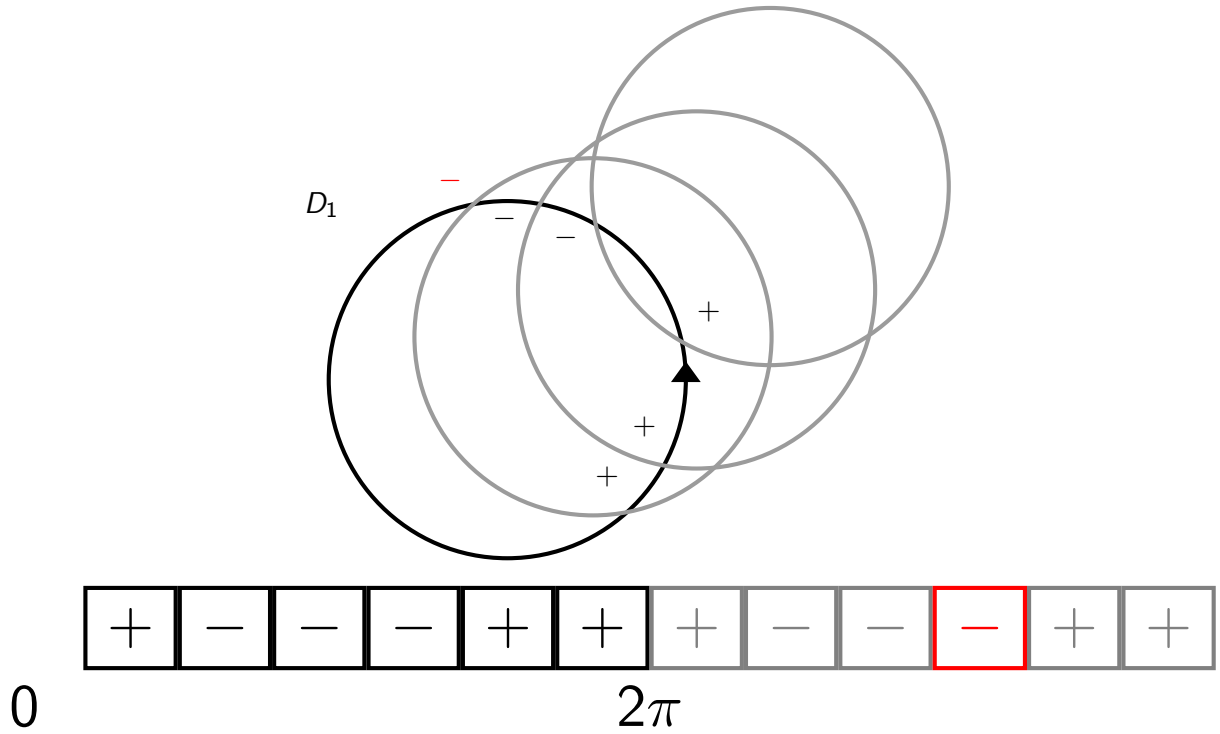
Source: Elaborated by the author.

Figure: A traversal for D_1 with green disks representing the active set and red signs representing the current angle being visited (some are omitted).



Source: Elaborated by the author.

Figure: A traversal for D_1 with green disks representing the active set and red signs representing the current angle being visited (some are omitted).



Source: Elaborated by the author.

Maximum Weight Clique Problem

Algorithm

The run-time complexity of the algorithm is $\mathcal{O}(n^2 \log n)$.

- There are $\mathcal{O}(n^2)$ intersections among n disks
- Sorting takes $\mathcal{O}(n^2 \log n)$
- The traversal takes $\mathcal{O}(n)$ for every disk.
- It can be implemented in $K \log n$ where K is the number of intersections.

Works found in the literature:

- In (**cabello:2006**) a $\mathcal{O}(n^{2m-1})$ algorithm was proposed. Also a $(1-\epsilon)-$approximation that runs in $\mathcal{O}(n \log n)$ was introduced.
- In (**zhou**) a heuristic method using an algorithm called mean-shift was developed. The mean-shift algorithm converges to a local density maxima of any probability distribution and it is used to find a smaller candidate list of centers for the disks.

Because of the similarities, we will discuss only the multiple ellipses algorithm later.

Ellipses

Ellipse

Given a center $c \in \mathbb{R}^2$ and $Q \in \mathbb{R}^{2x2}$, an ellipse is the set of points that satisfy

$$||u - c||_Q^2 = (u - c)^T Q(u - c) = 1,$$

with $\leq$ representing the set of covered points

Axis-parallel ellipse

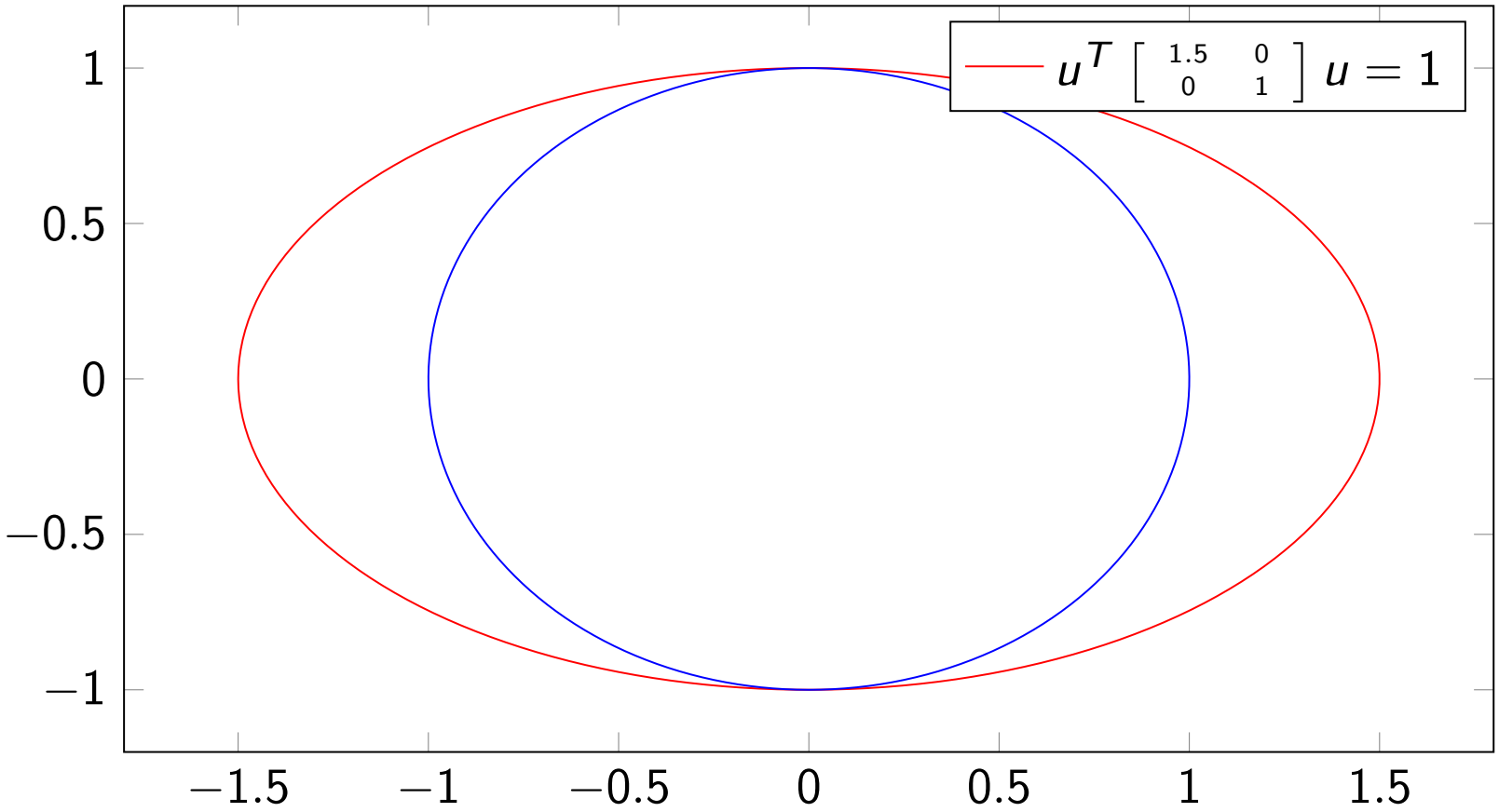
Any 2 by 2 diagonal d.p. matrix determines an axis-parallel ellipse, which can also be described by

$$\frac{(x - c_x)^2}{a^2} + \frac{(y - c_y)^2}{b^2} = 1,$$

where (a, b) are the shape parameters and $c = (c_x, c_y)$ is the center.

Ellipses

Figure: The ellipse seen as just a linear transformation of a circle.



Source: Elaborated by the author.

Maximal Covering by Ellipses

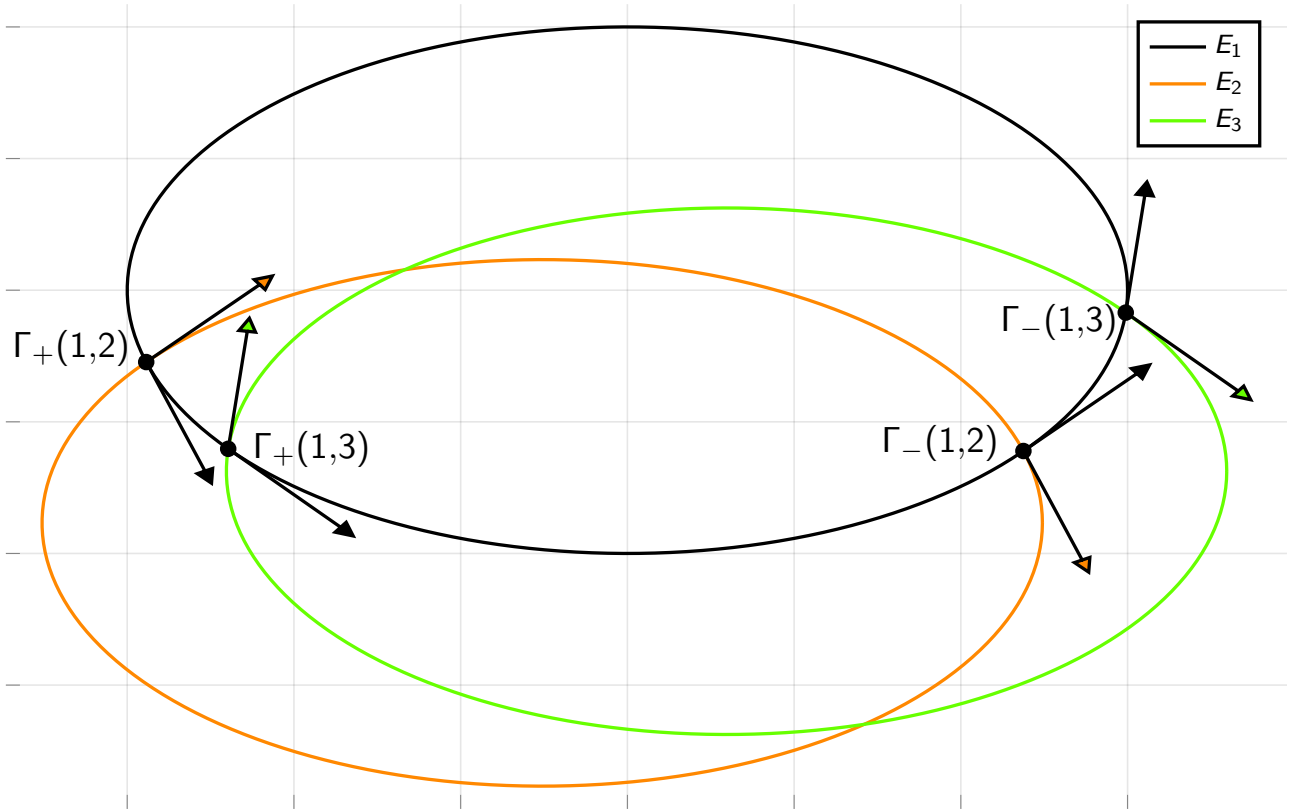
One ellipse

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

$$\max_q |\mathscr{P} \cap (q)|,$$

- $E(q)$ is an axis-parallel ellipse with center point q
- Assuming unit weights for now
- Same algorithm for one disk

Figure: Intersection points of E_1 with E_2 and E_3 along with opening and closing angles indicators.



Source: Elaborated by the author.

Maximal Covering by Ellipses

m ellipses

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

$$\max_{q_1,\ldots,q_m} \left| \bigcup_{i=1}^{m} \mathscr{P} \cap E_i(q_i) \right|,$$

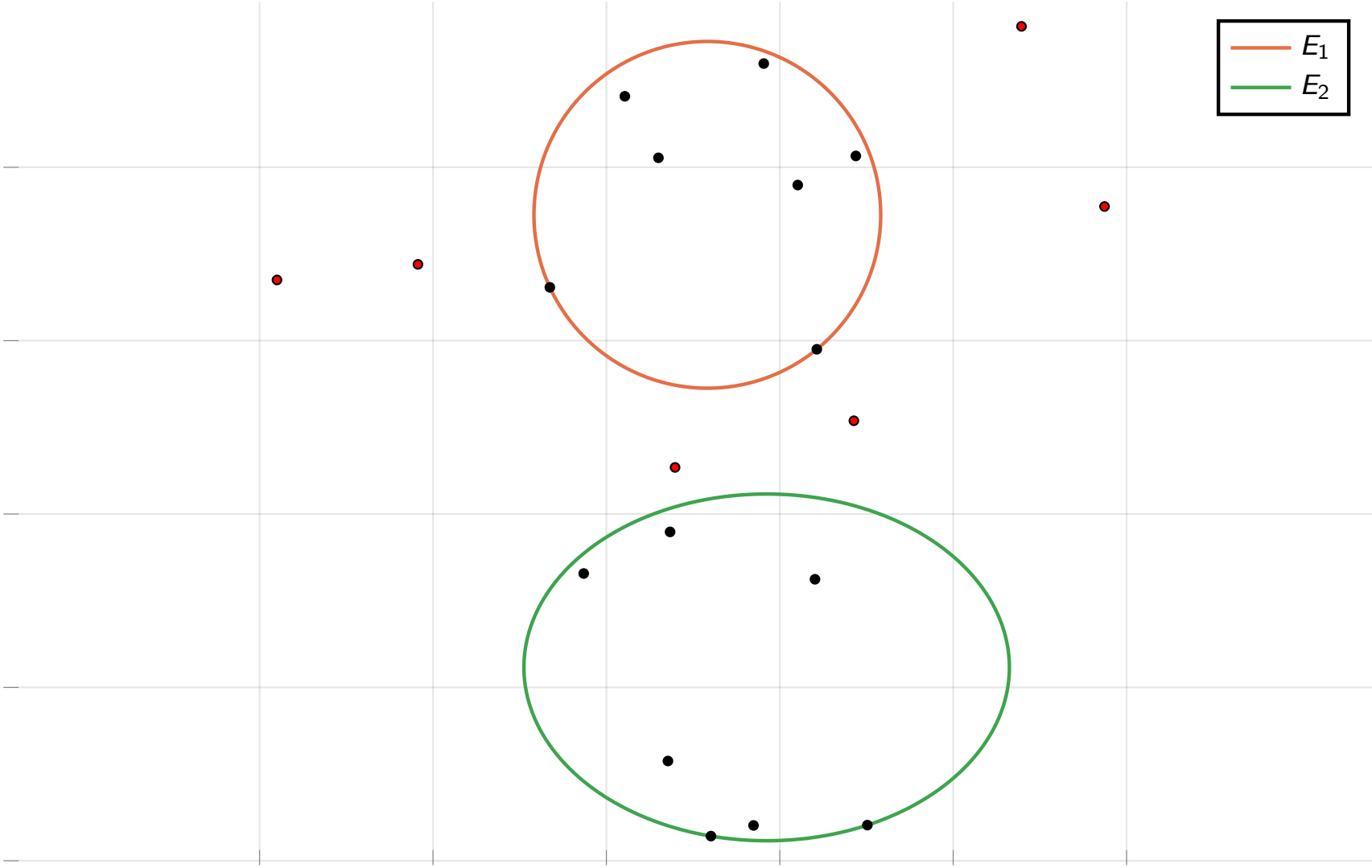
- $\mathscr{E}$ is a set of m ellipses
- (**canbolat**) is the very first study on the problem. Slow exact method, a heuristic one was proposed.
- (**andreta**) proposed a method that breaks the problem into smaller optimization ones. Also, they developed a method for the non-axis-parallel case.

Maximal Covering by Ellipses

Pre-processing that finds every possible coverage for ellipse E_i

1: $A \leftarrow \bigcup_{j \in I_i} \{\Gamma_+(i,j) \cup \Gamma_-(i,j)\}$
2: $Z \leftarrow \{\}$
3: **for** $cnt = 1..2$ **do**
4: **for** $a \in A$ **do**
5: Let p_a be the point that intersects E_i at angle a.
6: **if** a is a starting angle **then**
7: $Cov \leftarrow Cov \cup \{p_a\}$
8: **else**
9: $Cov \leftarrow Cov \setminus \{p_a\}$
10: **end if**
11: $Z \leftarrow Z \cup \{Cov\}$
12: **end for**
13: **end for**

Figure: Optimal solution with two ellipses for a random instance.



Source: Elaborated by the author.

Maximal Covering by Ellipses

- The algorithm for m ellipses tries every possible assignment of coverage for every one of the ellipses
- Run-time complexity of $\mathscr{O}(n^2)^m = \mathscr{O}(n^{2m})$
- Simpler than the m disks algorithm proposed by (**cabello:2006**). Achieves a similar complexity ($\mathscr{O}(n^{2m-1})$).
- Small improvements can be made in the pre-processing exhibited earlier in oder to reduce the size of the search space:
  - Non-maximal coverage sets.
  - Ellipses that are too distant do not need to be checked.
- The unit-weight assumption can be easily dropped

Future Work

Primary goals:

- Study the $(1-\epsilon)$-approximation method for the planar covering with disks in (**cabello:2006**) and develop an adapted version of the algorithm for ellipses with the same time complexity of $\mathcal{O}(n \log n)$.
- Develop an exact method for the version of the problem introduced in (**andreta**) where the ellipses can be freely rotated.

Future Work

Secondary goals:

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