

Programming Exercise: Optimal circle placement

Léo Simpson, Prof. Dr. Moritz Diehl

I General information

The goal of this exercise is to learn how to model, formulate, and solve optimization problems, and to get some intuition about how the solutions behave. For this purpose, we first focus on a specific example problem called the *optimal circle placement problem*, which is described in the next section as the *original problem*. You will first analyze the problem (Task 1) and implement a solution (Task 2).

The **final goal** of this exercise is that you come up with **your own variation** of the problem that you find interesting, and implement and solve it. To trigger your creativity, you will first be asked to explore some proposed modifications/extensions to the original problem (Task 3), and only then will you be asked to come up with your own modifications/extensions and implement them (Task 4).

Together with the code, you will also be asked to give a couple of interpretations and some mathematical formulations of the problem you are solving. For that, you can write your answers on paper, but the best would be to use a Jupyter notebook, where you can mix the Python code and some markdown cells with your explanations.

Passing the exercise

In order to pass the exercise, you need to finalize your code and explanations by **June 2nd, 2026**. Then, you will have to give a **live demonstration** of your code in one of the sessions on either June 2nd or June 9th. You should be able to (a) explain your code, (b) execute your code live, and (c) do spontaneous modifications based on requests by the teachers. You can also use the exercise session on June 2nd to ask questions, get preliminary feedback, etc., and then show your final solutions on June 9th.

It is very important that **you understand everything that you have coded and written**. Therefore, to pass the exercise, it is important to ensure that when you show us your code, you can explain every part of it (except the visualization part, which is provided anyway).

You can think about the exercise with some of your classmates, but the final code and explanations should be your own, and it is expected that each student gets its own creative modifications/extensions to the original problem.

II Optimal circle placement

Imagine you are given a square of side length $a = 10$ cm, and you want to cover it as much as possible with n circles, by choosing their positions and radius. One can formulate this as an optimization problem, where all centers' positions and radii are decision variables, and the objective is to maximize the covered area while ensuring that all circles are inside the square and do not overlap. Mathematically, this optimization problem can be formulated as follows:

$$\begin{aligned}
 & \underset{x, y, r \in \mathbb{R}^n}{\text{minimize}} && - \sum_{i=1}^n \pi r_i^2 \\
 & \text{subject to} && (x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2, \quad \text{for } i = 1, \dots, n, \text{ and } j = 1, \dots, i - 1, \\
 & && r_i \geq 0, \quad \text{for } i = 1, \dots, n, \\
 & && x_i \leq a - r_i, \quad \text{for } i = 1, \dots, n, \\
 & && y_i \leq a - r_i, \quad \text{for } i = 1, \dots, n, \\
 & && x_i \geq r_i, \quad \text{for } i = 1, \dots, n, \\
 & && y_i \geq r_i, \quad \text{for } i = 1, \dots, n,
 \end{aligned} \tag{1}$$

where (x_i, y_i) are the coordinates of the center of the i -th circle, and r_i denotes its radius. The first constraint ensures that the circles do not overlap, while the other constraints ensure that the circles are fully inside the square.

A solution of a similar problem is depicted in Figure 1 below.

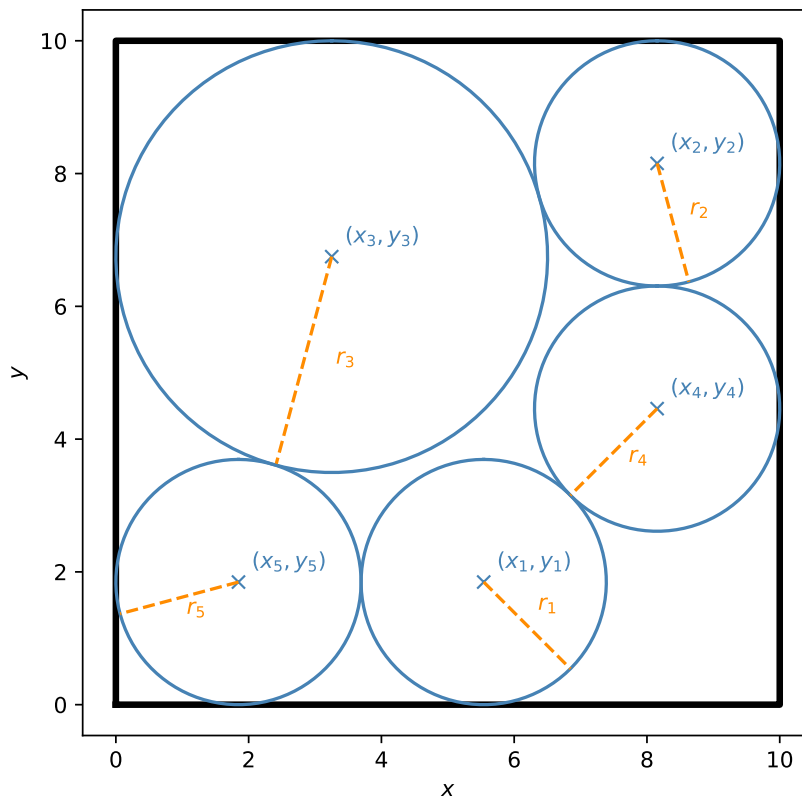


Figure 1: Example of a solution to the optimal circle placement problem with $n = 5$ circles. Note that this solution was computed for some variation of the original problem

III Your task

You will have to solve the following tasks, divided into four sections.

Task 1: Basic analysis of the problem

Solve the following questions (in paper or markdown format inside a Jupyter notebook).

- How many decision variables do we have in this problem?
- How many constraints do we have in this problem?
- What type of problem is this?
- Is it convex?

Task 2: Implementation and solution of the problem

Implement the problem as described above, solve it using CasADi and IPOPT, as in the previous exercises, and visualize the solution.

For the visualization part, you can use the Python file `plotting_circles.py` provided together with this exercise.

Task 3: Implement some modification/extension to the problem

Implement two of the four following possible modifications/extensions to the problem. For each modification, implement the task and explain how you modified the optimization problem to solve it (in paper or Markdown format within a Jupyter notebook). Then, interpret the results you obtained, and compare them to the original problem.

- Instead of maximizing the covered area, maximize the minimum radius of the circles. You might also make a trade-off between the two objectives.
- Change the shape of the objects: instead of the circles in a square, place some squares in a circle or in a rectangle.
- Study the behavior of the solution when the number of circles/squares increases, and plot the optimal value with respect to the number of circles/squares. What do you observe, and how do you interpret this behavior?
- Find at least two different local minima of the problem, and compare their optimal values. You might do so by changing the initial guess.

Task 4: Create your own modification to the problem

Invent your own modifications of the problem that you find interesting, by changing the objective **and** the constraints of the problem.

Your modifications should be non-trivial, and there should be at least 2 of them.

Describe these modifications, implement them, and show the results.