

Homework 8

Steve Gillet

October 28, 2024

Course: Algorithmic Motion Planning – ASEN 5254-001 – Fall 2024

Professor: Morteza Lahijanian

Teaching Assistant: Yusif Razzaq

Exercise 1. Extend your implementation of the GoalBiasRRT planner in *Homework 7* to plan for m disk robots translating and rotating in a workspace $W \subset \mathbb{R}^2$ using a coupled (centralized) approach. Every robot A^i for all $i \in \{1, \dots, m\}$ is a disk with radius R . The initial position of robot i (center of disk) is at x_{start}^i , and its goal is (center of disk) x_{goal}^i . Further assume that all m robots move at the same speed. That is, 1 step for robot A^i is equal to 1 step for robot A^j for all $i, j \in \{1, \dots, m\}$. Your planner should take as input a step size r , a goal bias probability p_{goal} , maximum number of iterations n , workspace obstacles, workspace boundaries, m robots and their corresponding initial position x_{start}^i and goal position x_{goal}^i , and radius ϵ (centered at x_{goal}^i) for the termination condition at goal as input and return a valid path for every robot from x_{start}^i to x_{goal}^i , the size of the RRT tree, and the computation time.

- Consider a 2D rectangular workspace $W = [0, 16] \times [0, 16]$, with obstacle space $WO \subset W$. Let $WO = \bigcup_{i=1}^6 WO_i$, where each WO_i is a rectangle in W with the following vertices:

$$WO_1 : \quad v_1^1 = (4, 6), \quad v_2^1 = (6, 6), \quad v_3^1 = (6, 7), \quad v_4^1 = (4, 7)$$

$$WO_2 : \quad v_1^2 = (4, 6), \quad v_2^2 = (5, 6), \quad v_3^2 = (5, 10), \quad v_4^2 = (4, 10)$$

$$WO_3 : \quad v_1^3 = (4, 9), \quad v_2^3 = (6, 9), \quad v_3^3 = (6, 10), \quad v_4^3 = (4, 10)$$

$$WO_4 : \quad v_1^4 = (10, 6), \quad v_2^4 = (12, 6), \quad v_3^4 = (12, 7), \quad v_4^4 = (10, 7)$$

$$WO_5 : \quad v_1^5 = (11, 6), \quad v_2^5 = (12, 6), \quad v_3^5 = (12, 10), \quad v_4^5 = (11, 10)$$

$$WO_6 : \quad v_1^6 = (10, 9), \quad v_2^6 = (12, 9), \quad v_3^6 = (12, 10), \quad v_4^6 = (10, 10)$$

- Consider a set of $m = 6$ disk robots, where $R = 0.5$ with the following start and goal configurations:

$$\begin{aligned}
x_{\text{start}}^1 &= (2, 2), & x_{\text{goal}}^1 &= (14, 14) \\
x_{\text{start}}^2 &= (2, 14), & x_{\text{goal}}^2 &= (14, 2) \\
x_{\text{start}}^3 &= (8, 14), & x_{\text{goal}}^3 &= (8, 2) \\
x_{\text{start}}^4 &= (2, 8), & x_{\text{goal}}^4 &= (14, 8) \\
x_{\text{start}}^5 &= (11, 2), & x_{\text{goal}}^5 &= (5, 14) \\
x_{\text{start}}^6 &= (11, 14), & x_{\text{goal}}^6 &= (5, 2)
\end{aligned}$$

(a) Answer the following questions:

i. What is the C-space of each individual disk robot?

The C-space (configuration space) of each individual disk robot is a two-dimensional space \mathbb{R}^2 , representing the possible x and y positions of the robot in the workspace.

ii. What is the C-space of the meta-agent (composed space)?

The C-space of the meta-agent, which combines the configuration spaces of all individual agents, is the Cartesian product of each robot's C-space. For m robots, this becomes \mathbb{R}^{2m} , where each pair of dimensions represents the x and y coordinates of each robot.

iii. Describe how the C-space grows with respect to the number of agents.

The C-space grows linearly with the number of agents in terms of dimensionality. For each additional robot, two more dimensions are added to the configuration space. So, for m robots, the dimension of the C-space is $2m$.

iv. Is the growth rate a concern for scaling to larger numbers of agents? Justify your answer.

Yes, the growth rate is a concern because the increased dimensionality makes the space exponentially harder to search for feasible paths. As the number of agents increases, the complexity of collision checking and path planning in higher-dimensional spaces grows significantly, which can impact computational efficiency and feasibility.

(b) Solve planning problem for $m = 2$ using the first two robots ($i = \{1, 2\}$). Let $n = 7,500$, $r = 0.5$, $p_{\text{goal}} = 0.05$, and $\epsilon = 0.25$. Plot the solution in the workspace showing both agents reaching their respective goals.

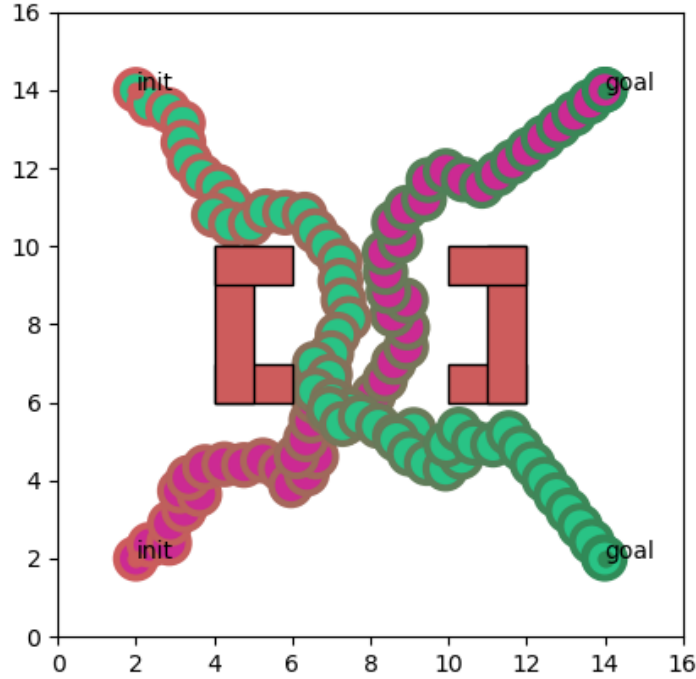


Figure 1: Solution plot for $m = 2$ with agents reaching their respective goals.

(c) Use 100 runs to benchmark your implementation in two categories:

- **Computation time:** Measure the time taken for each run.
- **Size of the tree:** Record the number of nodes in the RRT tree.

Show your results using boxplots. Additionally, save the data in a format suitable for later analysis.

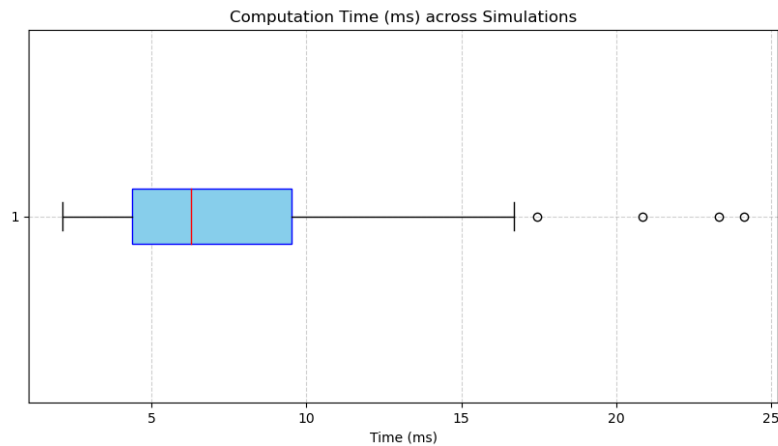


Figure 2: Box plot for $m = 2$ times over 100 iterations.

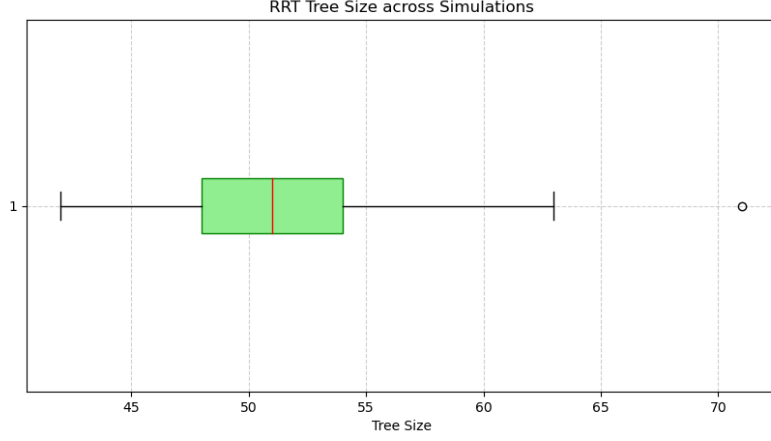


Figure 3: Box plot for $m = 2$ tree sizes over 100 iterations.

(d) Repeat parts (b) and (c) for all $m = 3, 4, 5, 6$ Incrementally add agents from the list in logical order (i.e., $m = 3$ uses agents 1, 2, and 3, etc.). Show your results for every benchmark using boxplots. Save the data separately for future use.

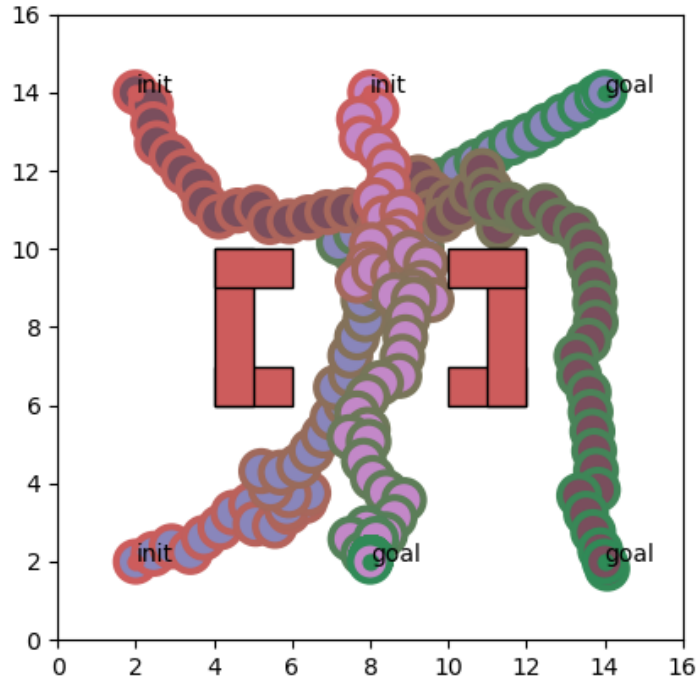


Figure 4: Solution plot for $m = 3$ with agents reaching their respective goals.

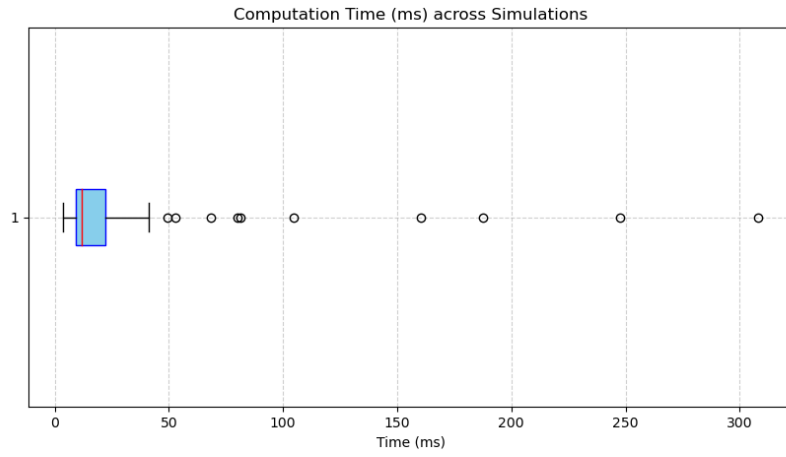


Figure 5: Box plot for $m = 3$ times over 100 iterations.

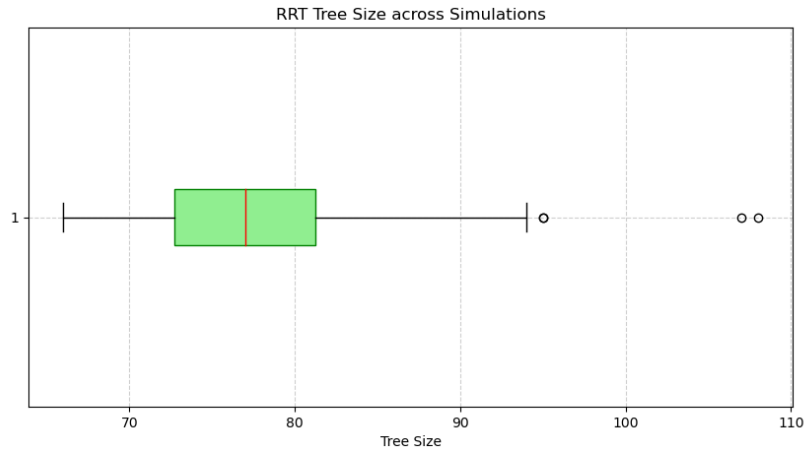


Figure 6: Box plot for $m = 3$ tree sizes over 100 iterations.

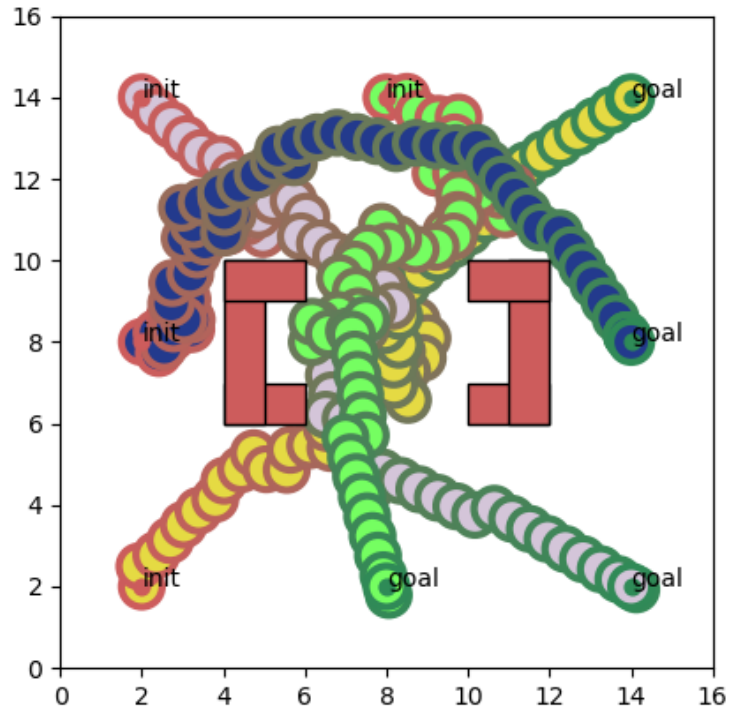


Figure 7: Solution plot for $m = 4$ with agents reaching their respective goals.

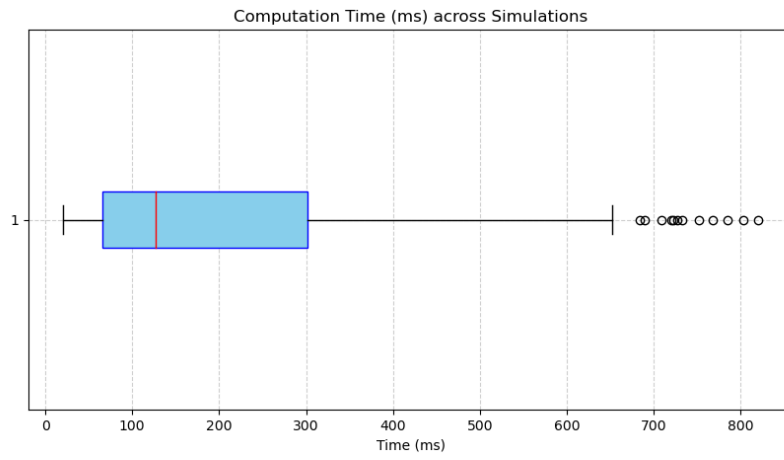


Figure 8: Box plot for $m = 4$ times over 100 iterations.

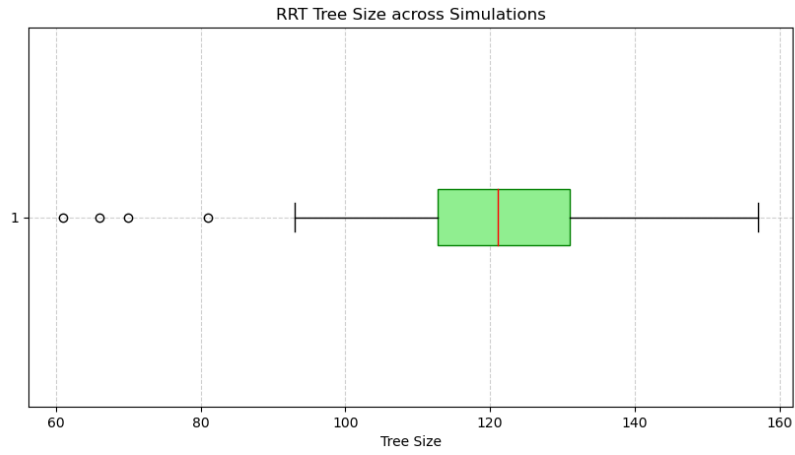


Figure 9: Box plot for $m = 4$ tree sizes over 100 iterations.

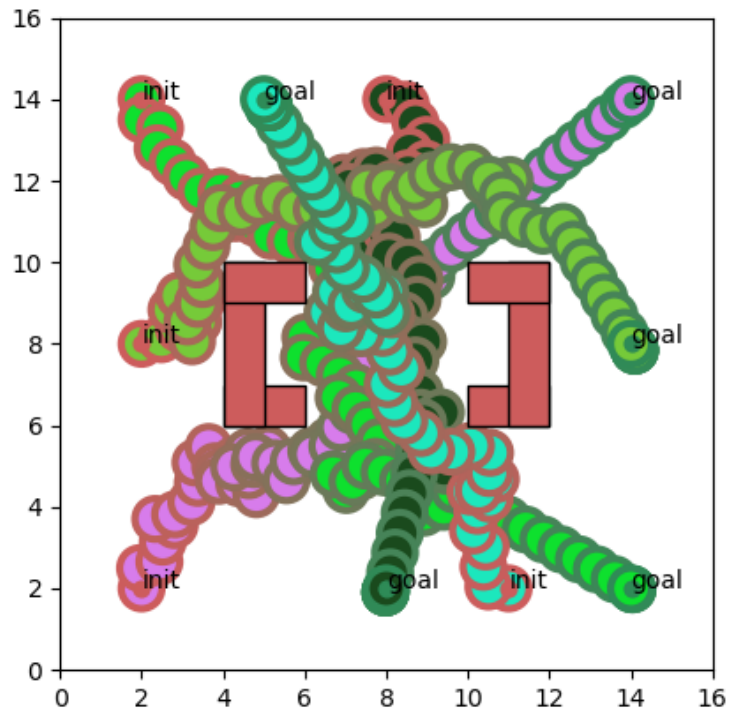


Figure 10: Solution plot for $m = 5$ with agents reaching their respective goals.

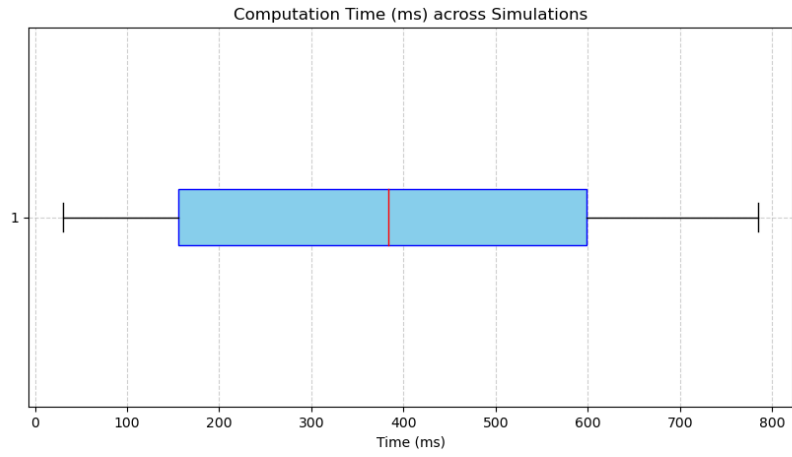


Figure 11: Box plot for $m = 5$ times over 100 iterations.

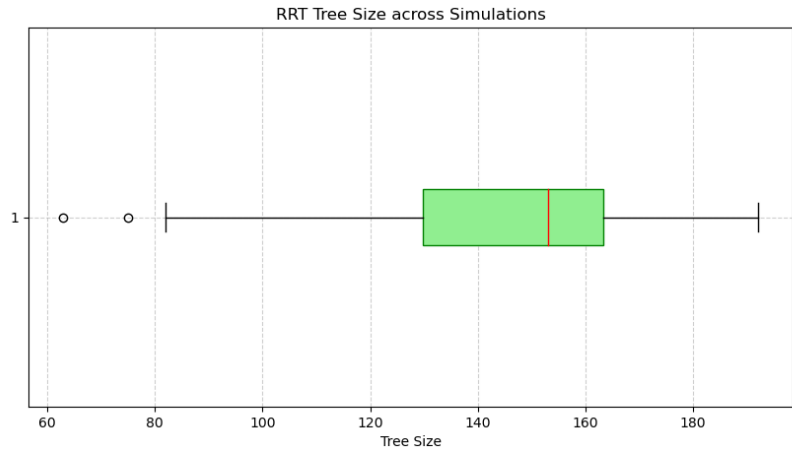


Figure 12: Box plot for $m = 5$ tree sizes over 100 iterations.

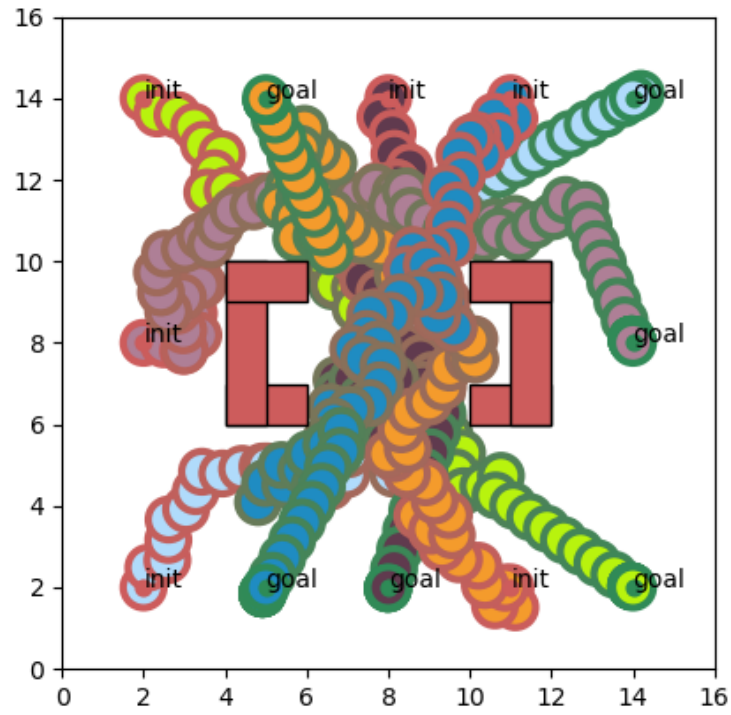


Figure 13: Solution plot for $m = 6$ with agents reaching their respective goals.

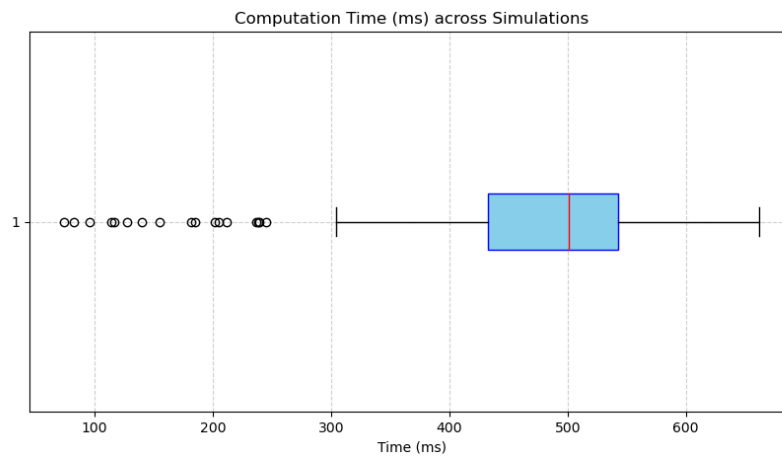


Figure 14: Box plot for $m = 6$ times over 100 iterations.

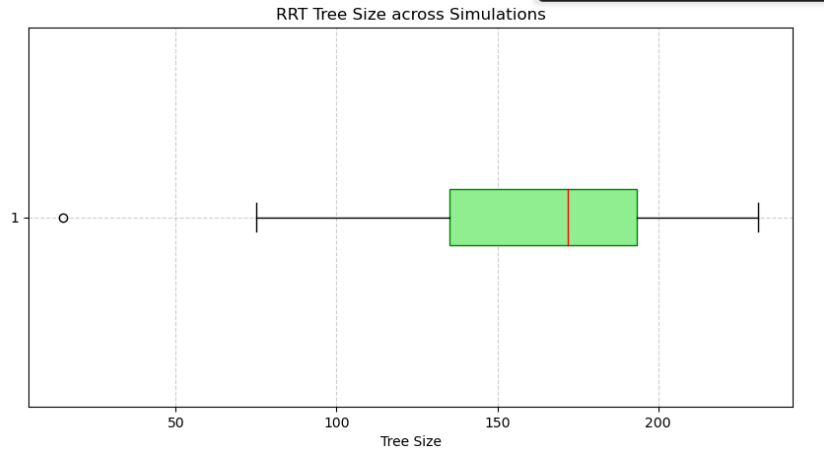


Figure 15: Box plot for $m = 6$ tree sizes over 100 iterations.

(e) Compute the average computation time and average size of tree over the 100 runs for each benchmark in (c) and (d). Use the 6 average computation times (1 for every value of m) and 6 average tree sizes to produce two plots:

- An average computation time (y-axis) vs. number of agents m (x-axis).
- An average size of tree (y-axis) vs. number of agents m (x-axis).

Comment on what these plots tell us about increasing the number of agents.

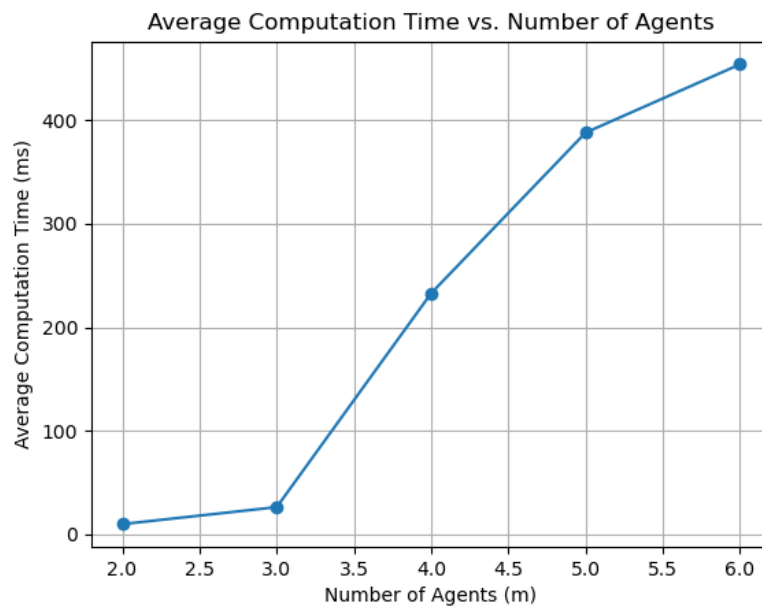


Figure 16: Average time for different m number of agents over 100 iterations.

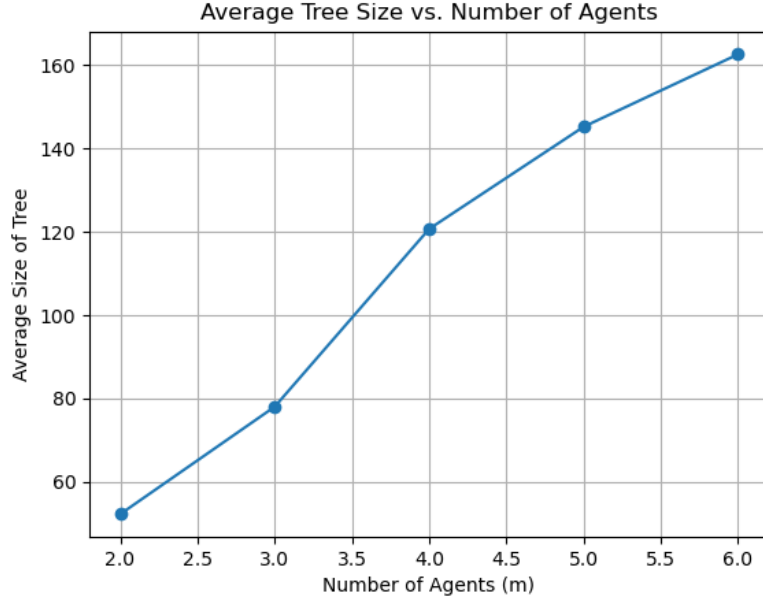


Figure 17: Average tree size for different m number of agents over 100 iterations.

- **Average Tree Size vs. Number of Agents:** The size of the tree increases as the number of agents m grows. This can be attributed to the fact that with more agents, there are more potential paths to explore to avoid collisions and reach each agent's goal configuration. Consequently, the search space grows exponentially, leading to a larger tree.
- **Average Computation Time vs. Number of Agents:** The computation time also increases as the number of agents grows. This is expected, as managing more agents requires additional computations for collision checking, path planning, and overall coordination. The increased complexity in multi-agent planning results in a steep rise in computation time.

In conclusion, both the tree size and computation time exhibit near-exponential growth with respect to the number of agents. This behavior highlights the scalability challenge in multi-agent RRT-based planning methods: as the number of agents increases, the required computational resources and time to find feasible paths rise substantially. This trend suggests that for larger numbers of agents, optimized or alternative methods may be necessary to maintain feasible computation times.