# A Comparison of Path Planning Algorithms in 2D: A\*, RRT\* and PRM (Probabilistic Road Map)

-Meera Ranjan

# Introduction

* This report compares the performance of three popular path planning algorithms: A\*, RRT\*, and PRM, in terms of efficiency, smoothness, clearance, and the ability to handle obstacles.
* The objective is to evaluate the effectiveness of each algorithm in navigating a robot through a grid environment with obstacles.

# Methodology

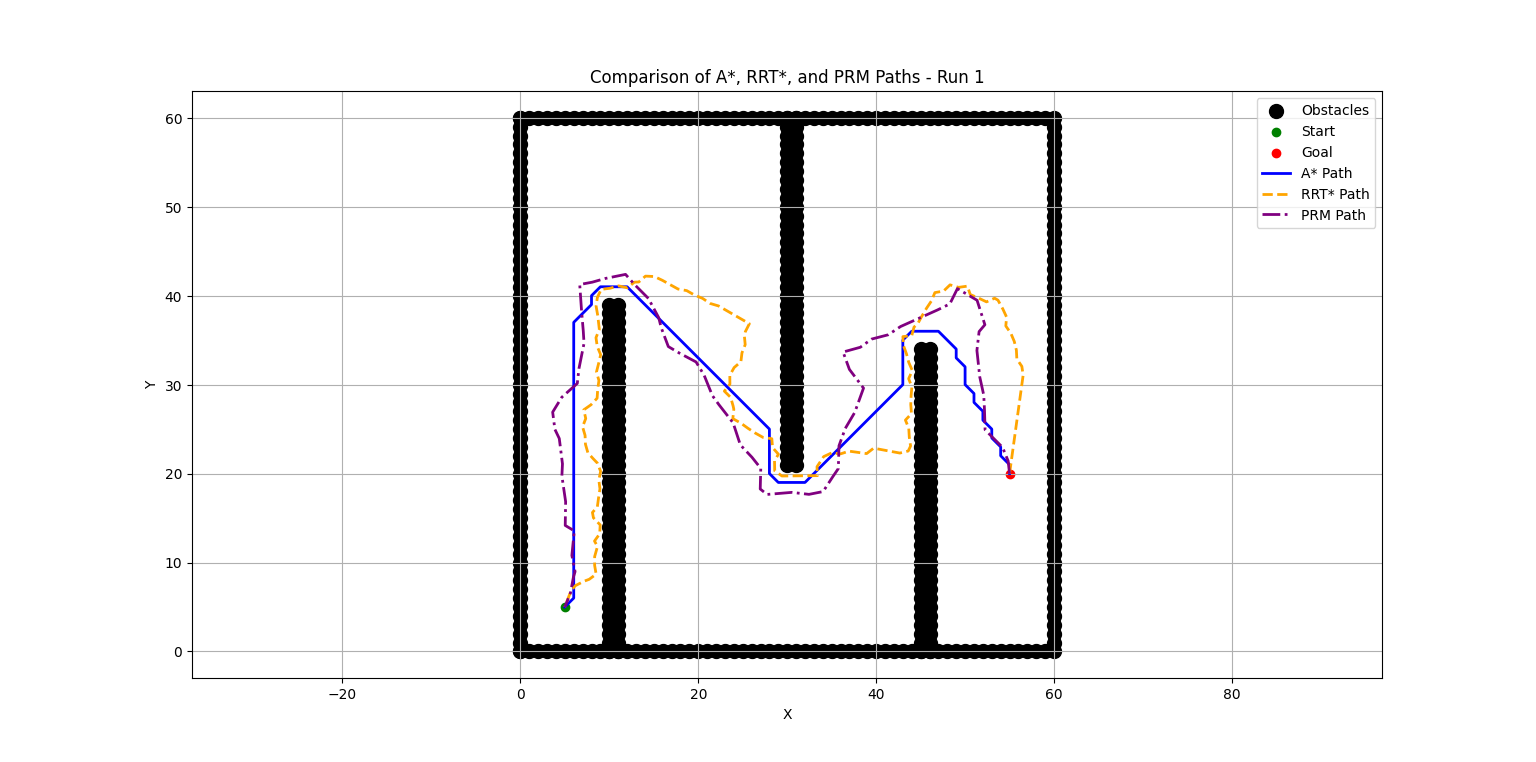
* **Environment Setup**:
  + **Grid Size**:
    - The environment is a 60x60 grid, with obstacles represented by small circles placed on the grid.
  + **Start and Goal Positions**:
    - Start position : (5, 5)
    - Goal position : (55, 20).
  + **Obstacle Representation**: Obstacles are represented as small circles across the grid, and their radius is set at 0.5 units.
* **Algorithms and custom parameters:**:
  + **A**\*:
    - Grid-based search, incorporates a heuristic to guide the search towards the goal efficiently.
    - resolution: Grid resolution for A\* algorithm (1.0).
    - rr: Robot radius (0.5).
    - obstacle\_radius: Radius for obstacle detection (2).
  + **RRT**\*:
    - Sampling-based approach, expanding a tree towards the goal.
    - expand\_dis: Expansion distance for each step (1.2).
    - robot\_radius: Radius of the robot (0.5).
    - obstacle\_radius: Radius for obstacle detection (2).
    - goal\_sample\_rate: Rate at which the goal is sampled (20).
    - max\_iter: Maximum number of iterations (2500).
  + **PRM**:
    - Sampling random points in the environment and connecting them with feasible paths.
    - obstacle\_radius: Radius of obstacles (2).
    - robot\_radius: Radius of the robot (0.5).
    - N\_SAMPLE: Number of samples to generate (1500).
    - N\_KNN: Number of nearest neighbors for connecting samples (10).
    - MAX\_EDGE\_LEN: Maximum edge length between nodes (3).
* **Metrics for Comparison**:
  + **Time Taken**: Time from the start to the goal.
  + **Path Length**: The total length of the path taken from start to goal.
  + **Nodes Expanded**: The number of nodes processed during the algorithm's execution.
  + **Smoothness**: Measured as the standard deviation of angles between consecutive path segments.
  + **Minimum Clearance**: The closest distance from the path to the obstacles.
* **Experiments**:  
  The experiment is run for 10 iterations, and the average and standard deviation for each metric are recorded for each algorithm.

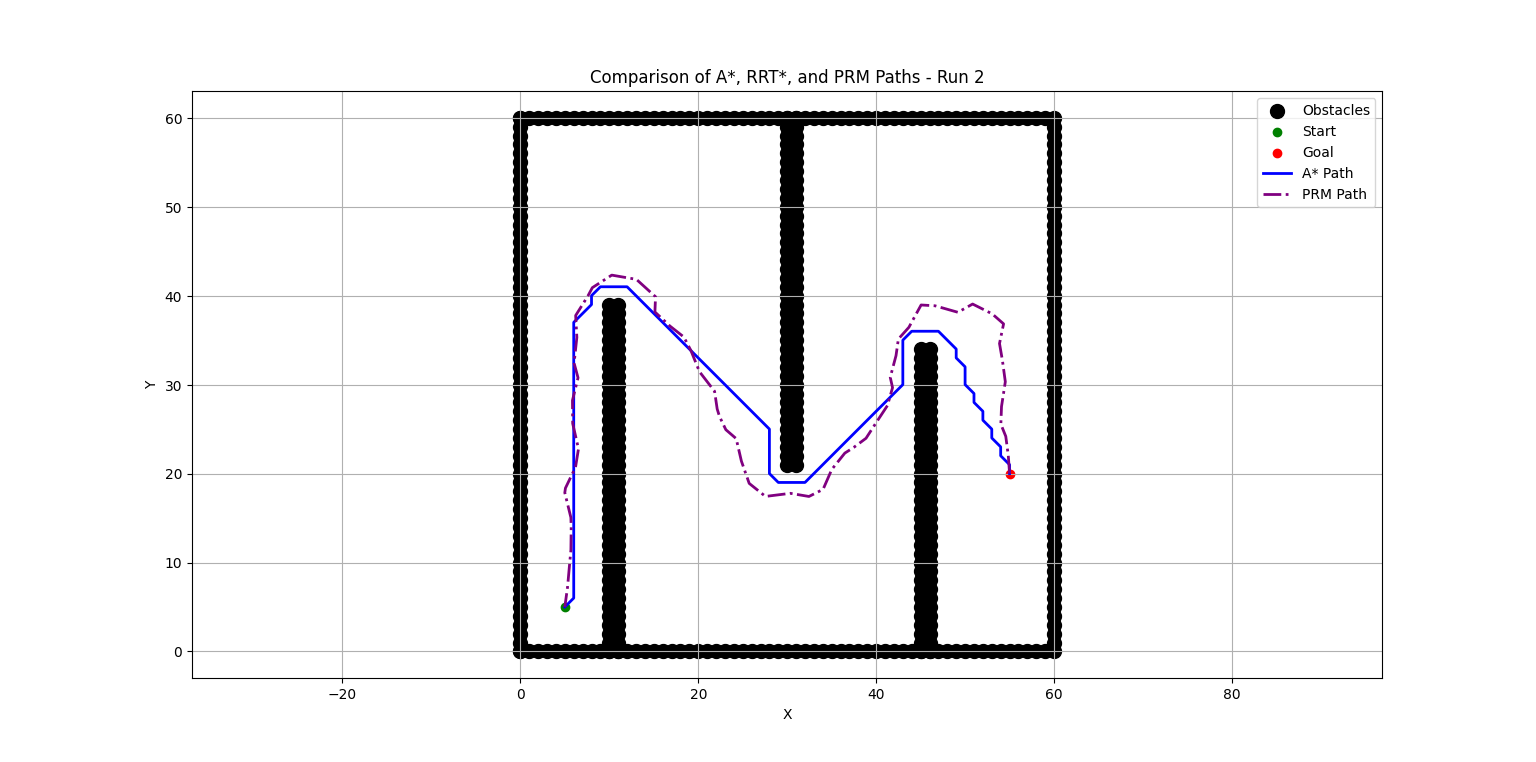
# Code

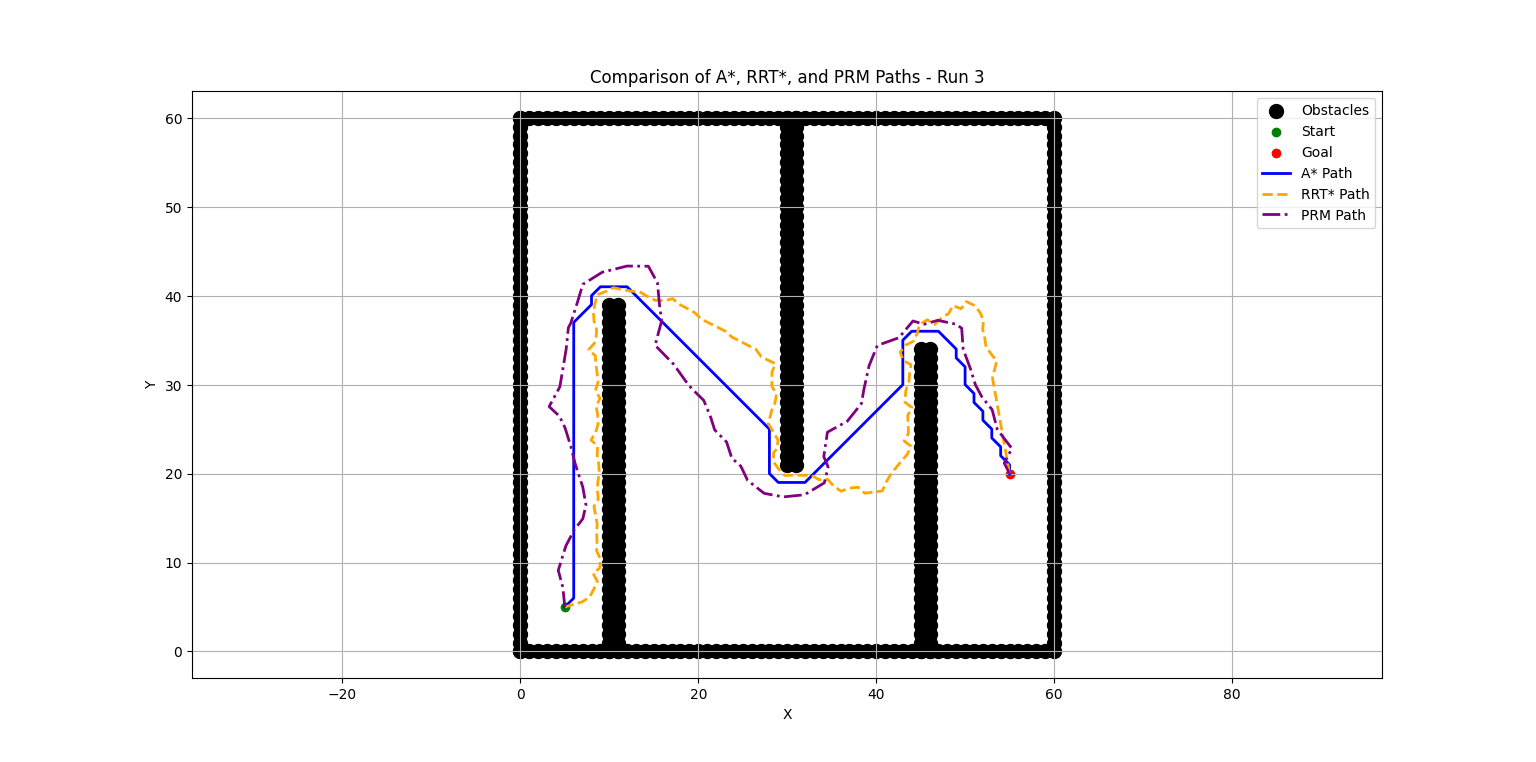
* Base code for algorithms: <https://github.com/AtsushiSakai/PythonRobotics>
* I wrote code that compared the three path planning algorithms—A\*, RRT\*, and PRM—on the same environment plot with obstacles.
* The code evaluates the performance of each algorithm based on path length, smoothness, clearance, and time taken, and visualises the resulting paths on the same plot.

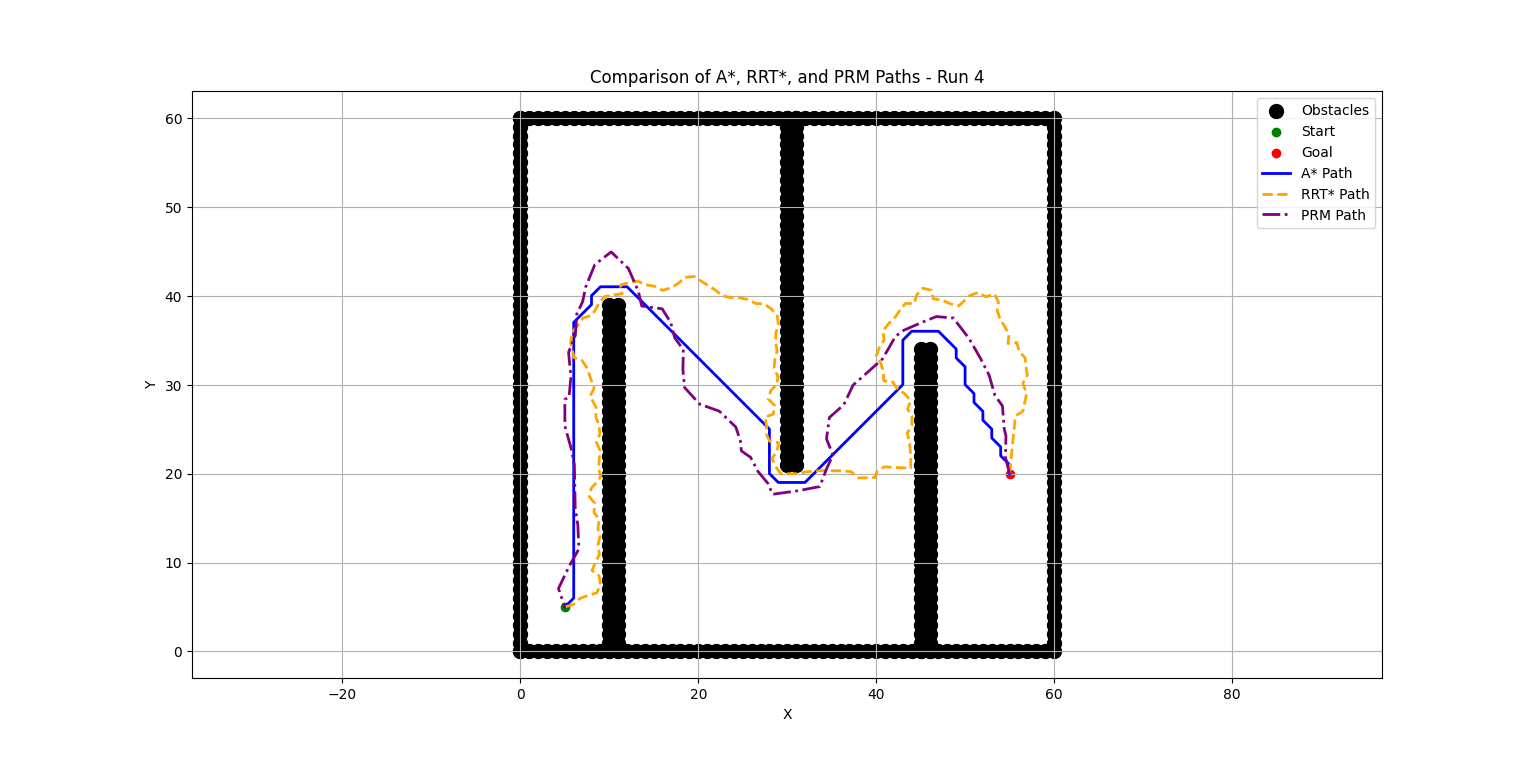
# Results

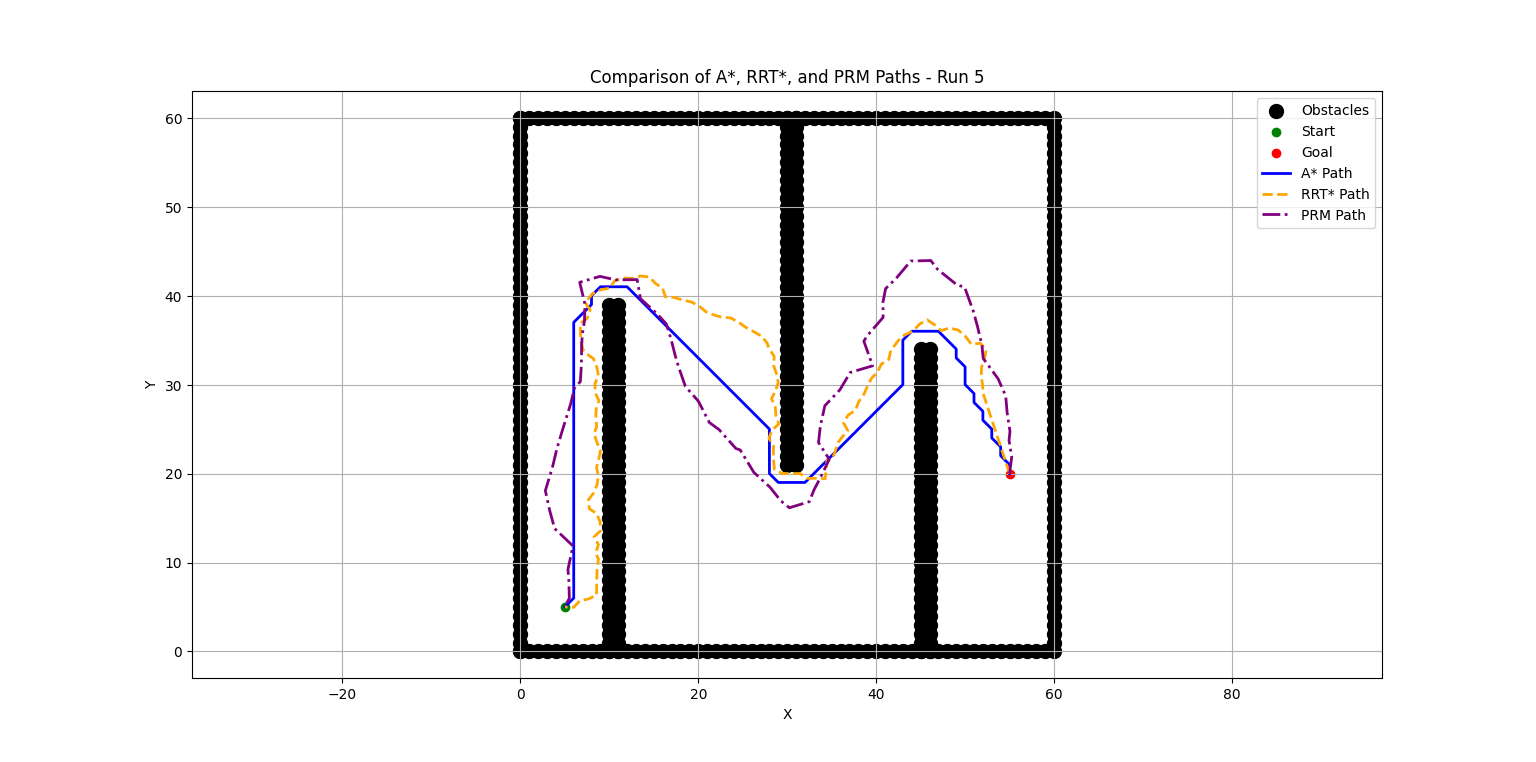
[Data table (Link)](https://docs.google.com/spreadsheets/d/1TMXHJNgraQ0qajV1Q4buTbpdln3gBcpZ04wtKfEdXXw/edit?usp=sharing)

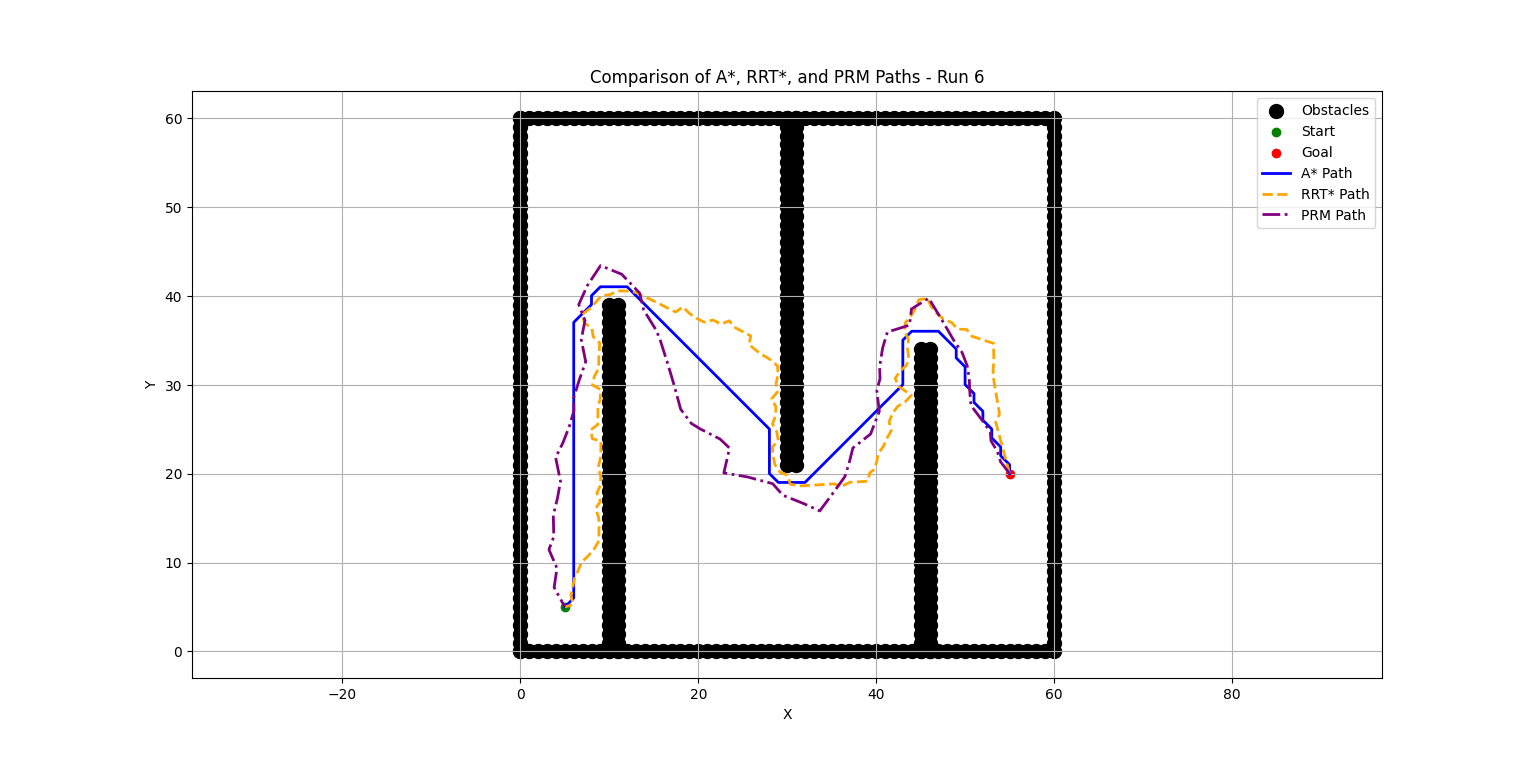


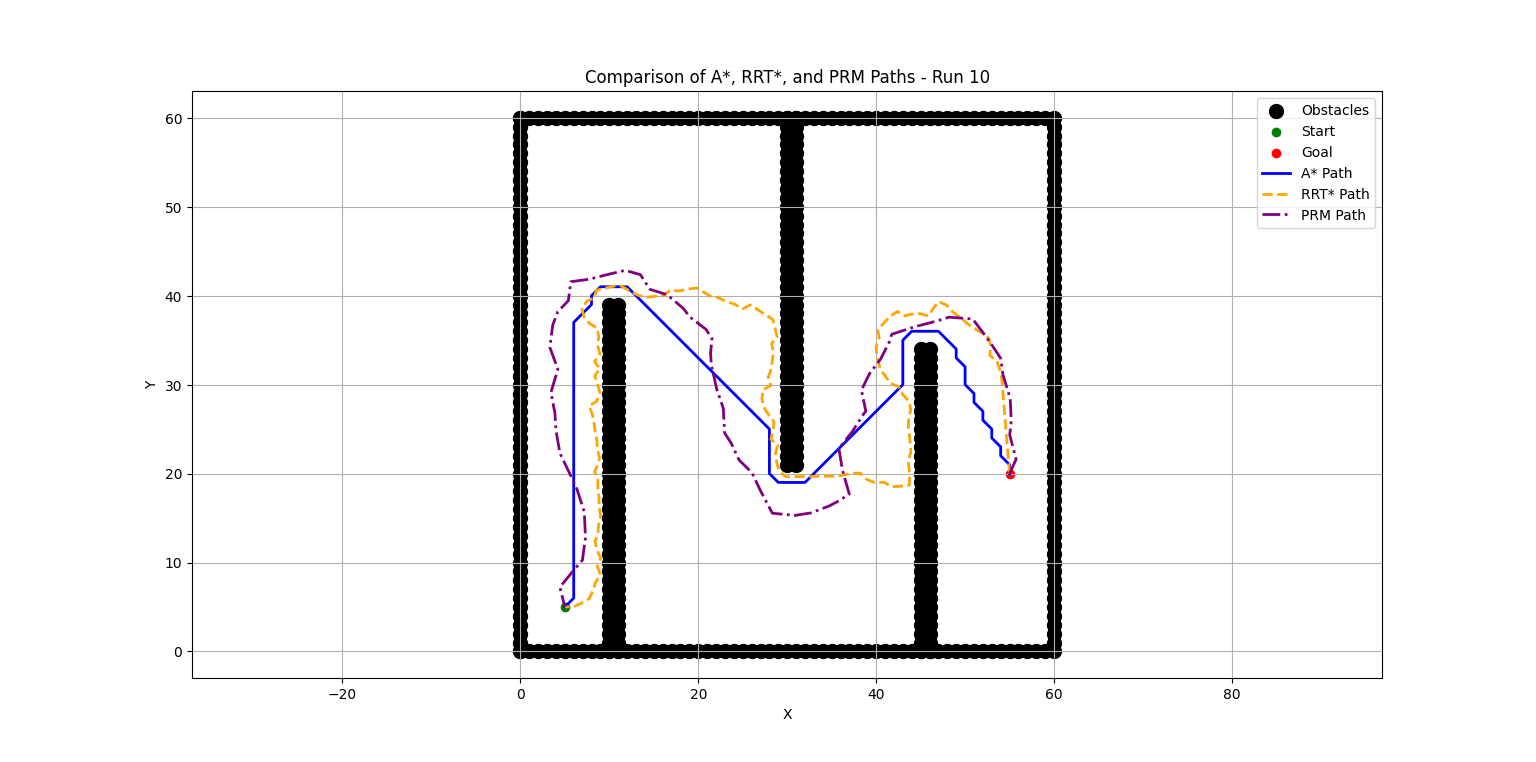
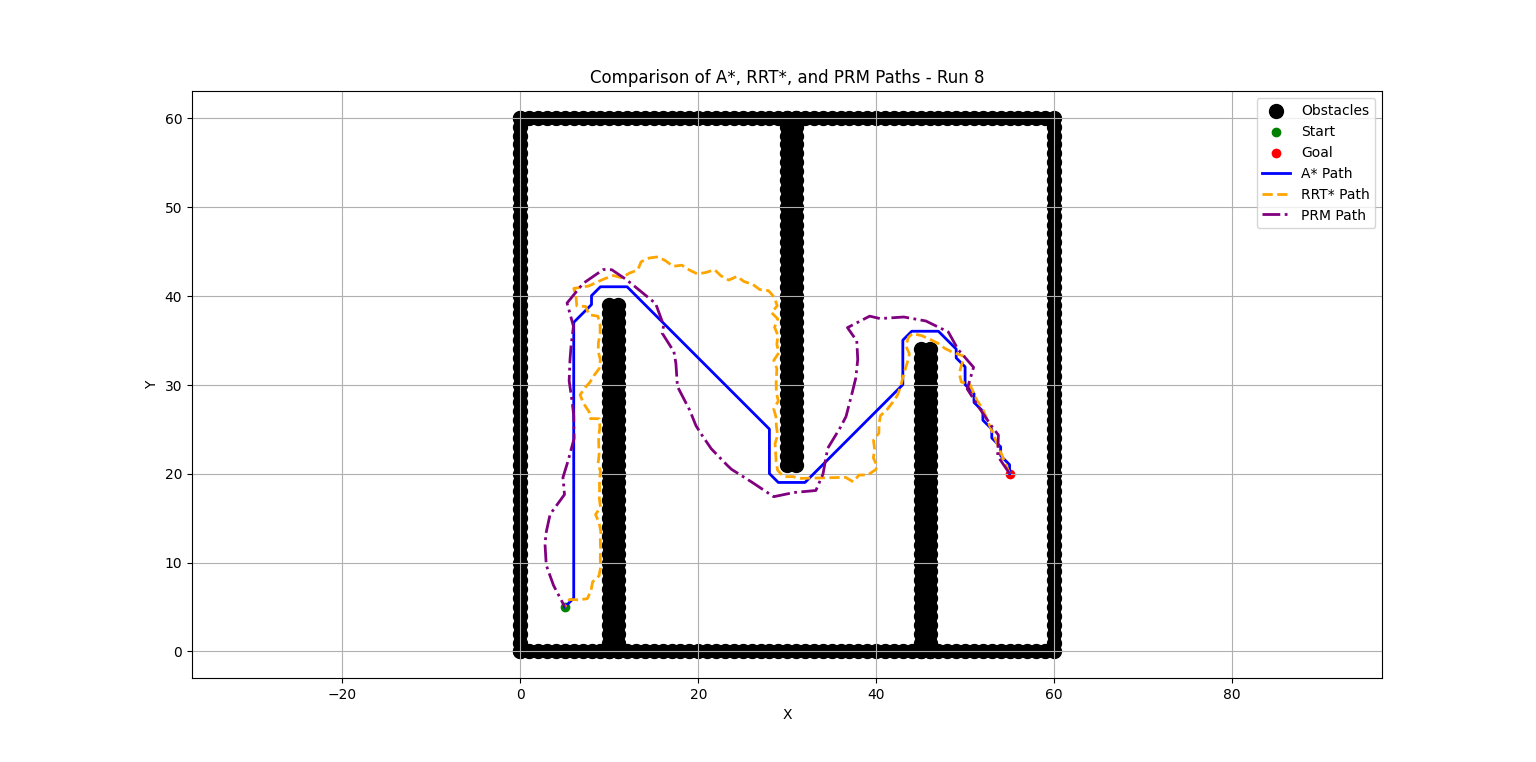
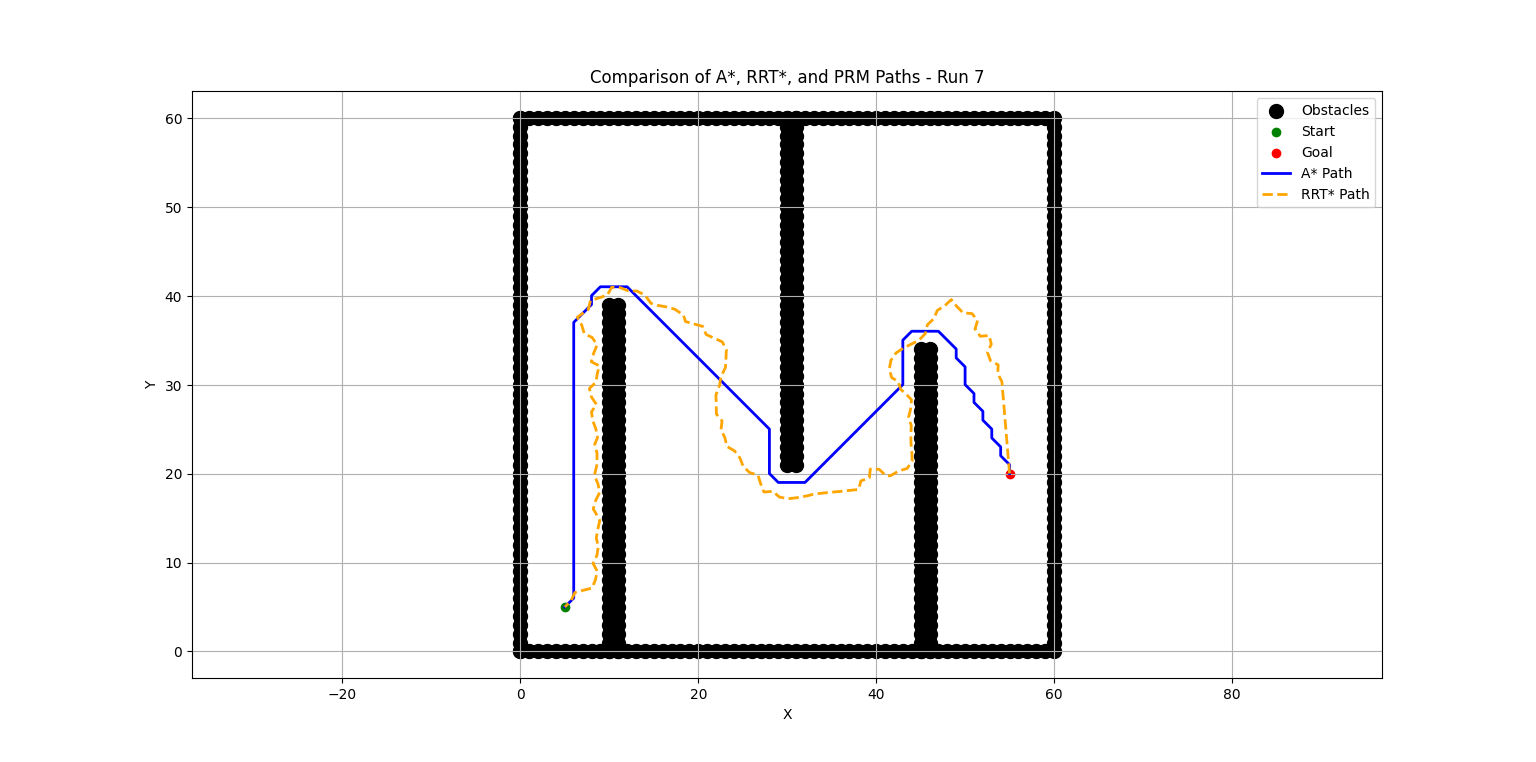


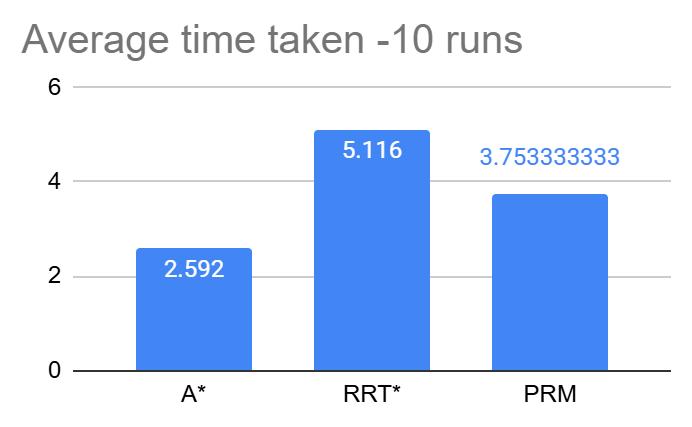


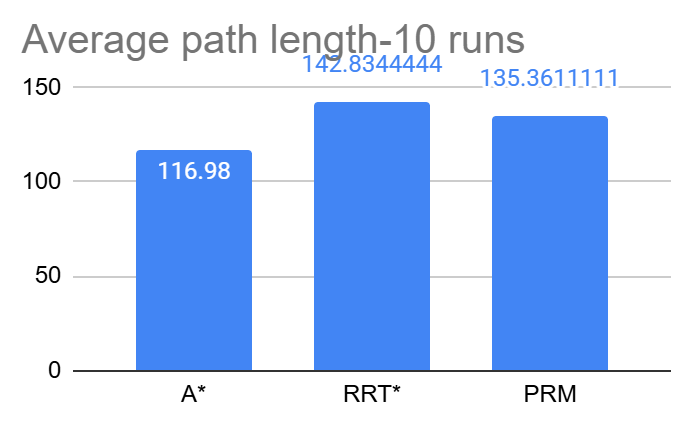


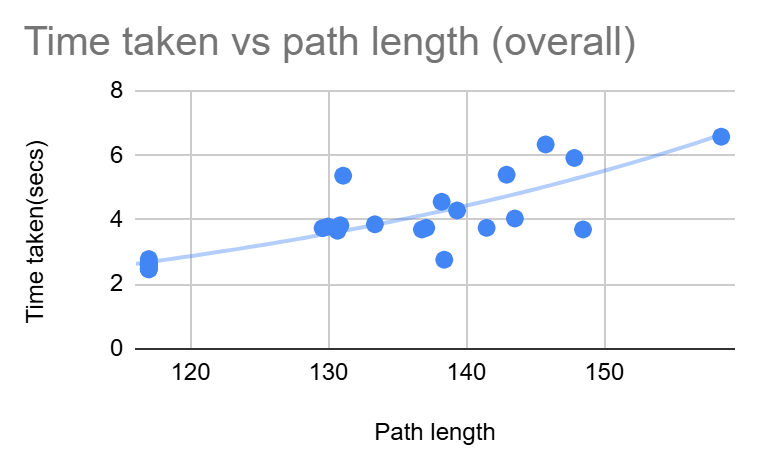


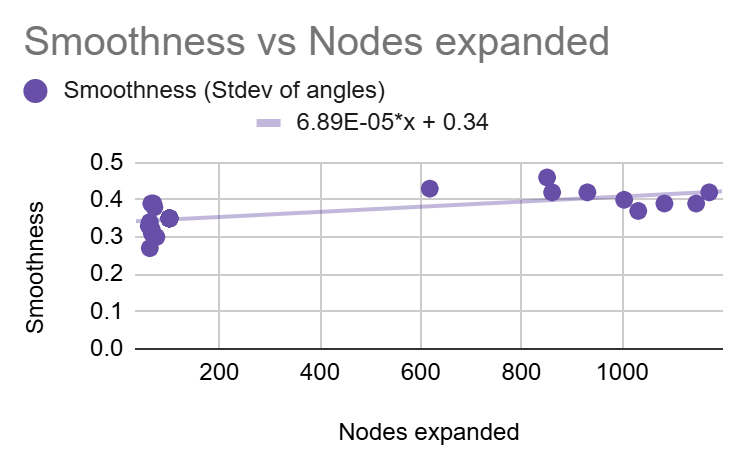












# Key observations

[Graphs and statistical analysis(Link)](https://docs.google.com/spreadsheets/d/1TMXHJNgraQ0qajV1Q4buTbpdln3gBcpZ04wtKfEdXXw/edit?gid=1118748668#gid=1118748668)

**Time Taken**:

* A\* consistently had the fastest execution time across all runs.
* PRM showed moderately slower performance compared to A\*.
* RRT\* had the longest computation time, particularly when no path was found (e.g., Run 2).

**Path Length**:

* A\* produced the shortest paths in all runs
* RRT\* and PRM had relatively longer paths, with PRM showing some improvements when more samples were used.

**Nodes Expanded**:

* A\* had fewer nodes expanded compared to RRT\* and PRM
* RRT\* expanded a significant number of nodes

**Smoothness**:

* RRT\* and PRM had similar smoothness scores, but they were generally higher than A\*’s smoother paths.

**Clearance**:

* PRM showed the best clearance from obstacles, as it uses a probabilistic approach to avoid collision during path construction.
* A\* had a moderate clearance, while RRT\* showed lower clearance in some runs.

**Other observations:**

* RRT\* had relatively high standard deviation all metrics.
* This Time Taken vs. Path Length graph shows an expected positive correlation between the two variables, since as the path length increases, the time taken to compute the path also increases.
* Smoothness vs Nodes expanded graph also shows a slight positive correlation. As the number of nodes expanded increases, the path generated tends to be smoother. This is because with more nodes, the algorithm has more opportunities to find a less jagged path.

# Reflection

* A\* performs very consistently with minimal variance, making it the most reliable and efficient algorithm for path planning in small and simple environments.
* RRT\* shows high variability in its performance, with long execution times and large path lengths. It may need better parameter tuning to improve efficiency.
* PRM offers a good balance between performance and consistency, perhaps making it a good choice for environments with dense obstacles.