Prathamesh Mehta

RBE 550 Motion Planning

Wildfire

Environment :

The purpose of this assignment report is to analyze and discuss a simulation environment designed for a firetruck to operate and extinguish fires that randomly emerge in a 250m x 250m field filled with obstacles. The obstacles consist of un-navigable thick brush, trees, and weeds shaped like tetrominoes with a base dimension of 15 meters. While the environment is not grid-based, an arsonist/wumpus sets a major conflagration at a random obstacle at 60-second intervals, and after 20 seconds, the fire spreads to all obstacles within a 30-meter radius. The simulation runs for a total of 3600 seconds, during which the firetruck starts at a random point in the map and has a path planner to navigate the environment and locate burning obstacles. The objective of this assignment is to assess the effectiveness of the firetruck's path planning algorithm in maximizing the number of obstacles extinguished within 10 meters of the firetruck.

Planner

The A* search-based planner utilizes kinematics equations to generate a list of possible neighbors and adds them to an open list based on their cost using a cost function, provided they are not already in the open or closed lists. The node with the lowest cost is then selected and added to the closed list. This process continues until the goal node is reached.

In contrast, the PRM sample-based planner selects n random points on the map and discards any that intersect with obstacles. It then connects the k nearest neighbors whose paths are collision-free to generate a graph, which is used by the A* algorithm to generate a global planner to reach the goal. From this global planner, a local planner is determined by finding the path between two points. In this simulation, the search-based planner described above is used as the local planner.

Run Iterations and Results:

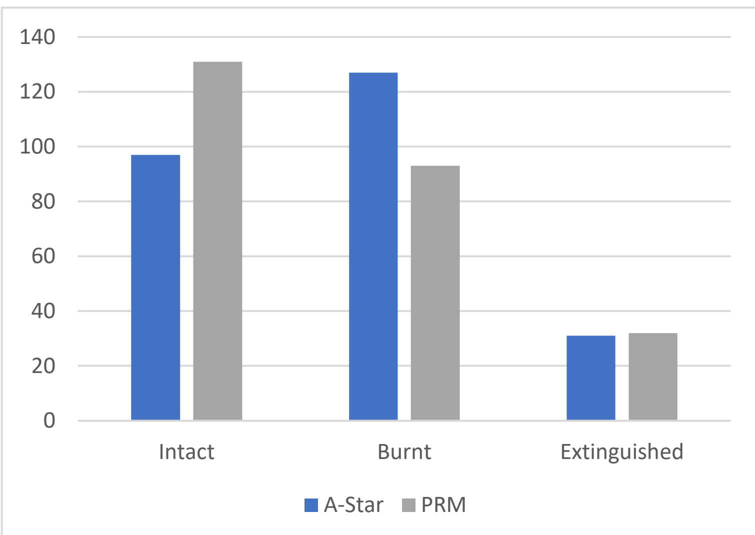
- 1ST Run

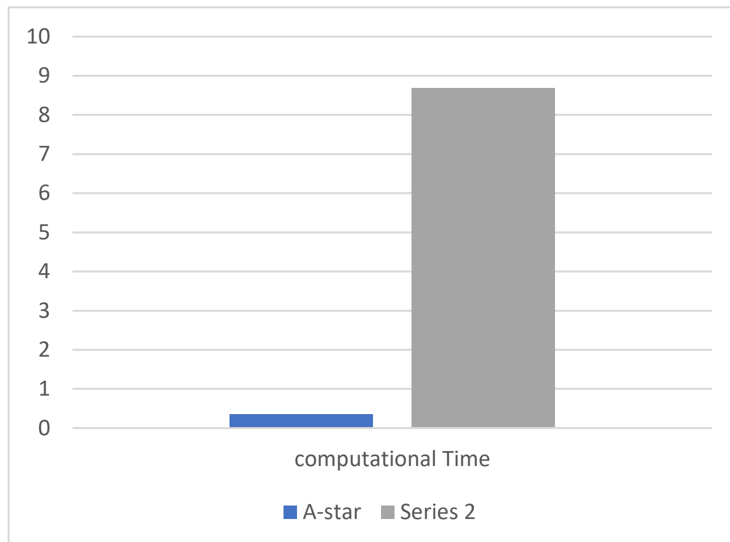
```
not working
Done
Intact 97
Burned 127
Extinguished 31
Total CPU time 0.34891843795776367
```

A-STAR

```
Done
Intact 131
Burned 93
Extinguished 32
Total CPU time 8.67925238609314
```

PRM



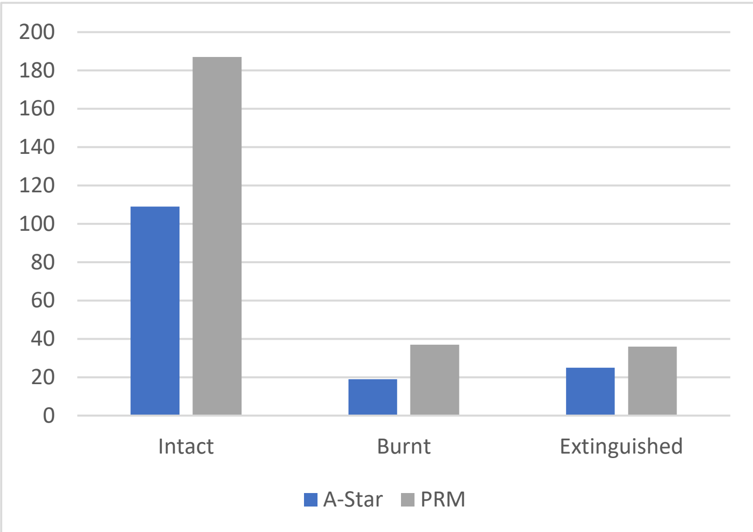


- 2nd Run

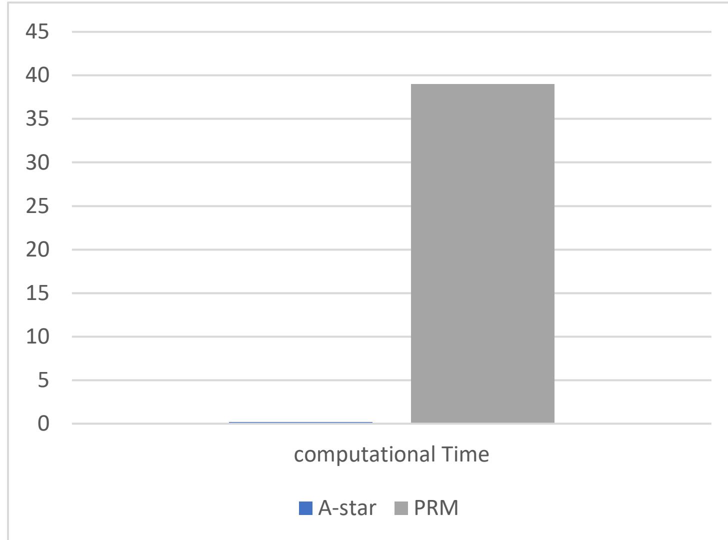
```
Intact 109
Burned 19
Extinguished 25
Total CPU time 0.20943498611450195
```

```
Done
Intact 187
Burned 37
Extinguished 36
Total CPU time 38.9136757850647
```

A-STAR



PRM

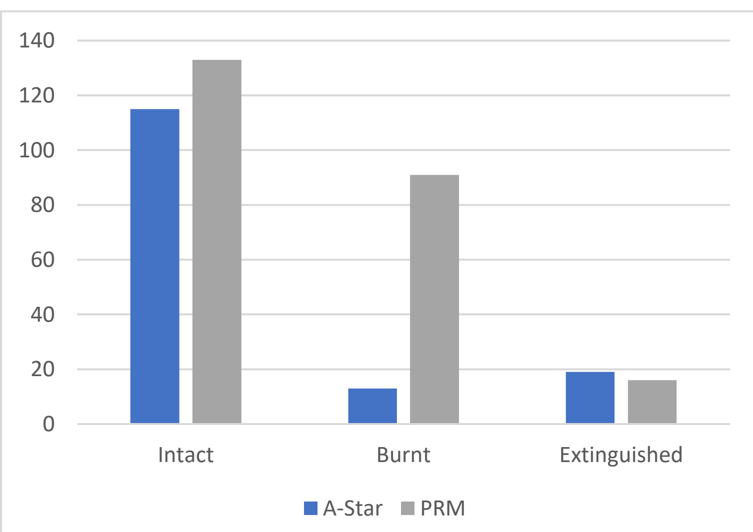


- 3rd Run

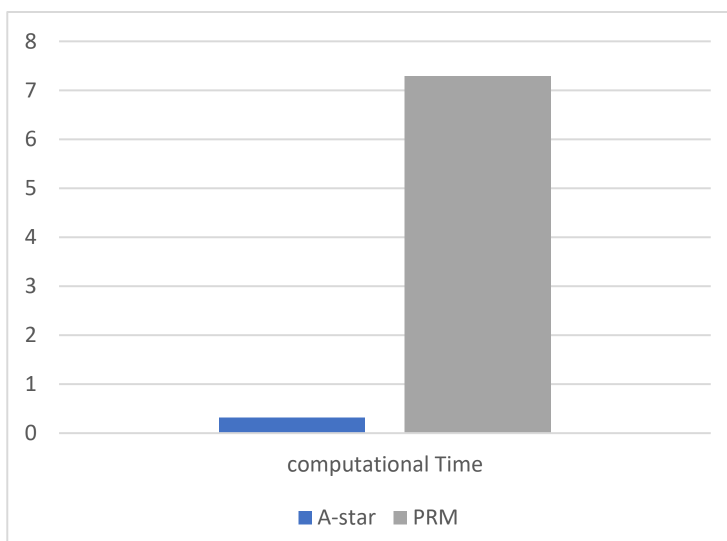
```
Intact 115
Burned 13
Extinguished 19
Total CPU time 0.313148736953735355
```

```
Intact 133
Burned 91
Extinguished 16
Total CPU time 7.280277729034424
```

A-STAR



PRM



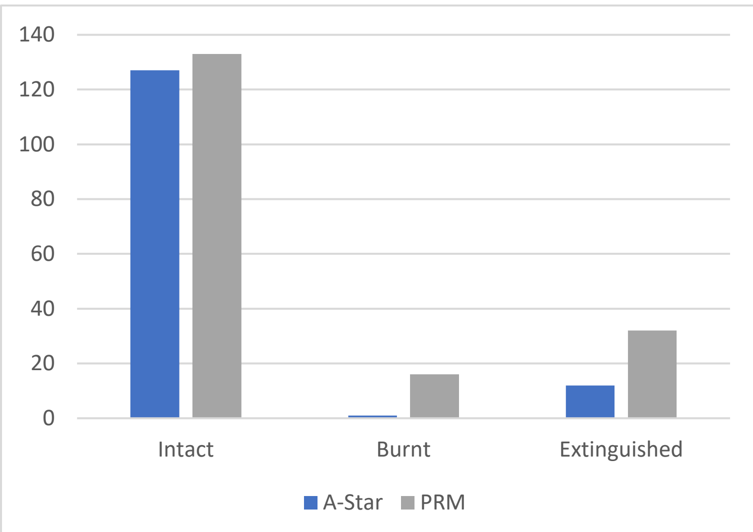
- 4th Run

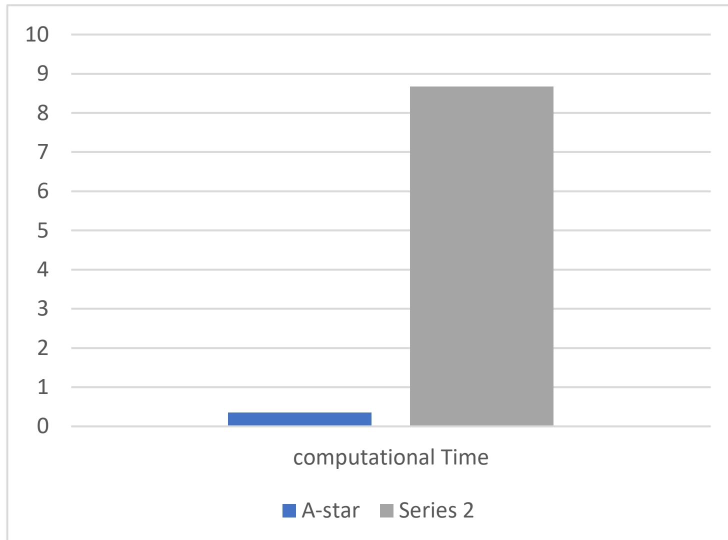
```
Intact 127
Burned 1
Extinguished 12
Total CPU time 0.13362455368041992
```

A-STAR

```
Intact 133
Burned 91
Extinguished 16
Total CPU time 7.280277729034424
```

PRM





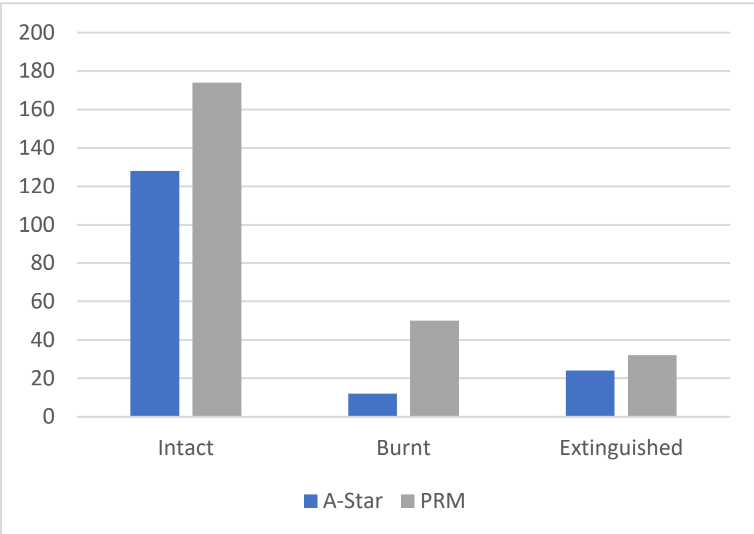
- 5th Run

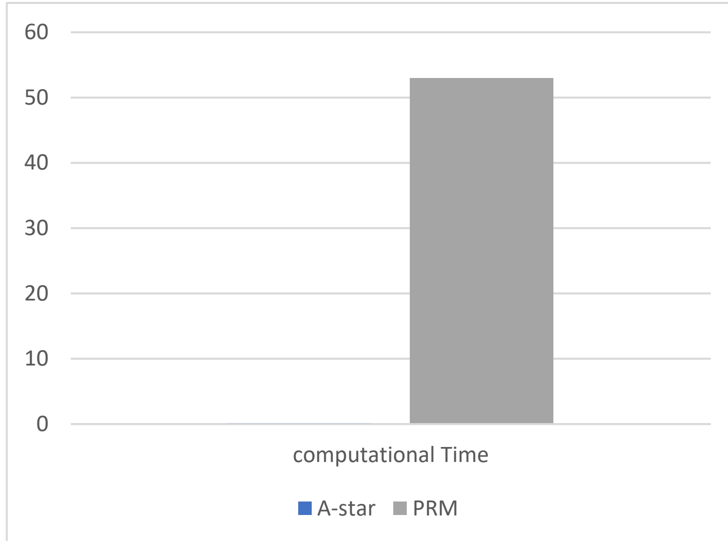
```
Intact 128
Burned 12
Extinguished 24
Total CPU time 0.17150211334228516
```

```
Done
Intact 174
Burned 50
Extinguished 38
Total CPU time 52.950637102127075
```

A-STAR

PRM





Discussion:

The experimentation yielded the following results:

- Hybrid A* was able to find optimal parts, but at the cost of exploring more space than PRM.
- Although PRM was able to generate a graph by selecting random points on the map, it required more computational resources than other methods, as shown in Figure 3.
- Pre-computing roadmaps can help the firetruck reach its destination faster, allowing it to cover more ground before the fire spreads, thus reducing its spread.
- The graph-based planner was observed to have a higher intact ratio than the sampling-based algorithm. This is because the graph-based planner selects an optimal path for the firetruck to follow.

Reference:

[1][GitHub - AtsushiSakai/PythonRobotics: Python sample codes for robotics algorithms.]

[2] [Planning Algorithms / Motion Planning (lavalle.pl)]