RBE 550 - Wilfire Assignment (Assignment 4)

Colton Layhue

April 3, 2023

Introduction

For this assignment, the student was tasked to implement two motion planning algorithms within an adversarial world of a Wumpus versus a firetruck. The Wumpus has the ability to set obstacles within the environment ablaze, requiring the firetruck to plan and execute routes in order to extinguish the flames.

Wumpus

Path Planner: A*
Source Code: [1]

The paths determined for the Wumpus were found with an A* path planner [1]. This path planner demonstrates the core concepts of A* planning i.e. a transition cost between the past node and the current node is calculated in addition to a heuristic. This heuristic is referred as the "cost-to-go" to the goal, as it is calculated based on the Euclidean distance between the goal node and the current node expanded. New nodes are created based on motion primitives. It should be noted that for the purpose of the Wumpus and its ability to ignite the obstacles, collision detection was removed in lieu of allowing the target goal points to be set to the obstacle positions.

Firetruck

Path Planner: Probabalistic Roadmap (PRM) with Dijkstra's Search
Source Code: [1]

The paths determined for the firetruck were found with a PRM path planner [1]. This path planner begins planning by sampling the configuration space for the robot. A K-dimensional (KD) tree is utilized by this source code as an efficient means to cluster the sampled points. The sampled points are then filtered based on their distance to the robot, which reveals whether or not the robot can physically achieve the sampled point based on its assigned radius. 500 samples are taken and are further filtered based on whether the point would cause a collision as well as the number of edges that branch off of the sample. After

samples are taken, an initial roadmap is generated and a path is determined (for this assignment) using Dijkstra's. Dijkstra's follows a similar procedure to the A* described for the Wumpus, however, Dijkstra's does not calculate the heuristic for the "cost-to-go". Instead, Dijkstra's only considers the cost of the Euclidean distance traveled to the current node.

Methodology

For this assignment, the student must consider many different aspects for both the Wumpus and the firetruck:

- The total simulation time of 3600 seconds
- The Wumpus sets fire to obstacles instantaneously
- Burning objects spread the fire to obstacles within 30 meters after 10 seconds of being "burning"
- The firetruck cannot detect the Wumpus unless all obstacles are burned
- The firetruck must stop within 10 meters of a burning objects for 5 seconds to extinguish the object
- Extinguished objects can be relit

For sake of implementation, the student took the following approach:

- Each iteration of a loop is considered 1 second
- Generated paths and additional variables were calculated and stored to be simulated upon completion of 3600 seconds.
- Once either the Wumpus or the firetruck have identified a path, they must remain idle in a "waiting" state to simulate being unable to plan whilst moving
- A time of 25 seconds is set as the time for a "burning" object to transition to "burned"
- Both the Wumpus and firetruck choose goal obstacles at random

Wildfire: Combinatorial and Sampling Based Planner Implementations

Algorithm 1 Simulation

t ← 0
burning, burnable, intact, extinguished ← empty lists
Wumpus is Planning
Firetruck is Planning
Wumpus is not Waiting
Firetruck is not Waiting
Firetruck is not Extinguishing
b ← burning obstacle

while t < 3600 **do**
  if Wumpus is Planning **then** ▷ Wumpus Planning
    Plan wumpus path where goal is some obstacle in burnable
    if Path is found **then**
      Wumpus stops planning
      Wumpus starts Waiting
      Wumpus Wait Counter = 0
      burning ← b
      Starting location becomes goal location
    end if
  elseWumpus is waiting
    if 10 seconds have passed **then** since igniting previous goal
      Spread fire within 30 meters
    elseWumpus has waited the entirely of the path length ▷ This is to emulate moving without planning
      Wumpus is planning
      Wumpus is not waiting
    end if
  end if
  $\forall b \in B$
  if burn time of $b < 25$ **then**
    Add to burned
  end if

```
    if Firetruck is planning then                                  ▷ Firetruck Planning
        Pick a random obstacle, b, from the burning list
        Search for an area adjacent to b to set as goal
        Plan firetruck path
        if Path is found then
            Firetruck stops planning
            Firetruck begins waiting
            Firetruck Wait Counter = 0
            Starting location becomes goal location
        end if
    else if Firetruck is waiting then
        Waiting counter ← +1
        if Waiting counter is larger than the length of the path then
            Firetruck is not waiting
            Firetruck is extinguishing
            Extinguishing counter ← 0
        end if
    else if Firetruck is extinguishing then
        Extinguishing counter ← +1
        if Extinguishing counter is 5 then
            Add obstacles within 10 meter radius to extinguished
            Firetruck is not Extinguishing
            Firetruck is planning
        end if
    end if
end while
```

Plots

For each of the runs, the following three ratios were calculated:

Intact Ratio: $\frac{N_{Intact}}{N_{total}}$

Burned Ratio: $\frac{N_{Burned}}{N_{total}}$

Extinguished Ratio: $\frac{N_{Extinguished}}{N_{total}}$

Run 1

Starting Location, Wumpus: (10,10)
Starting Location, Firetruck: (225,10)
Intact Ratio: 0.848
Burned Ratio: 0.095
Extinguished Ratio: 0.00713

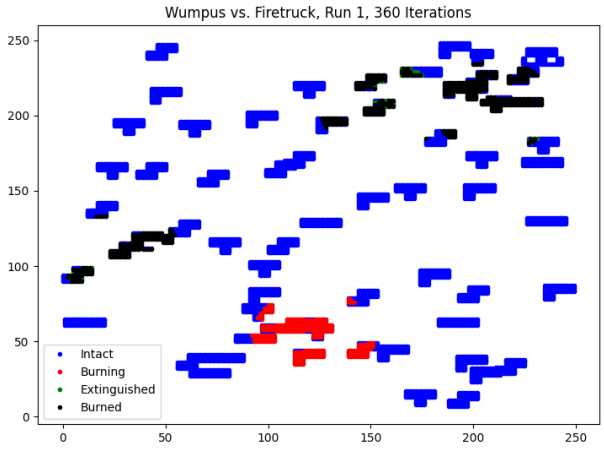

Figure 1: Run 1 Results, 360 Iterations

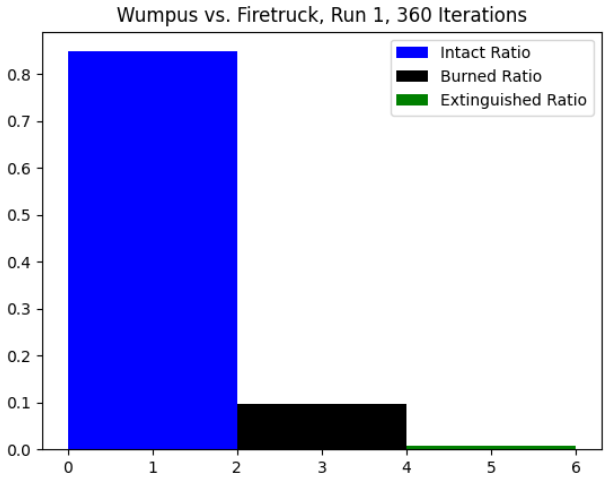

Figure 2: Run 1 Ratio Results, 360 Iterations

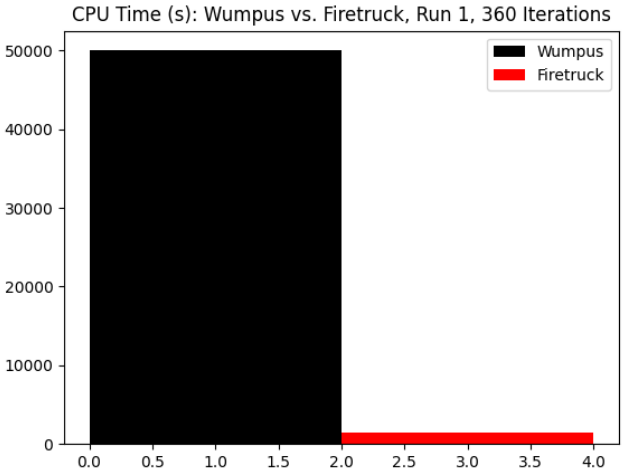

Figure 3: Run 1 CPU Time Results, 360 Iterations

Run 2

Starting Location, Wumpus: (10,10)
Starting Location, Firetruck: (225,10)
Intact Ratio: 0.873
Burned Ratio: 0.082
Extinguished Ratio: 0.025

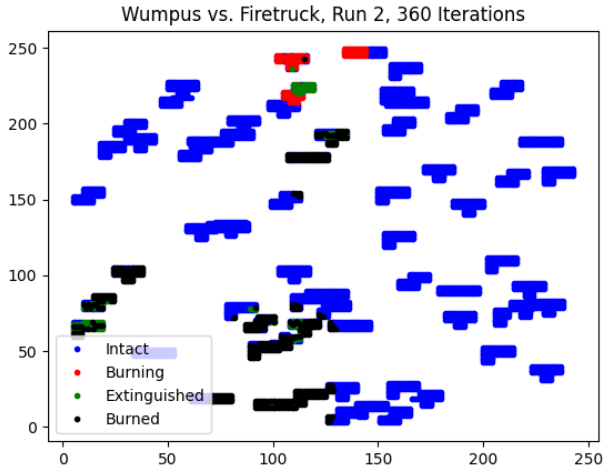

Figure 4: Run 2 Results, 360 Iterations

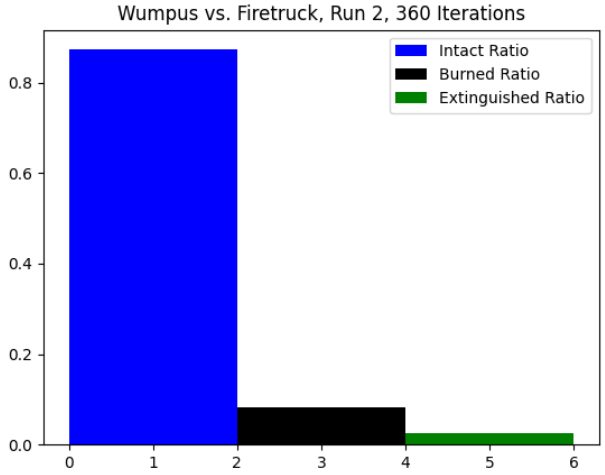

Figure 5: Run 2 Ratio Results Plot, 360 Iterations

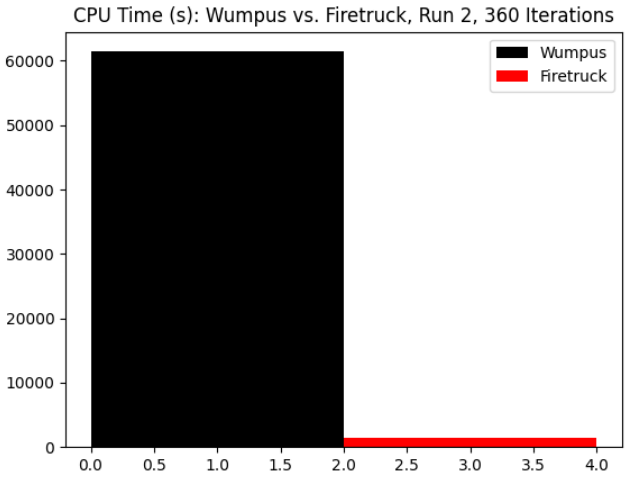

Figure 6: Run 2 CPU Time Results, 360 Iterations

Run 3

Starting Location, Wumpus: (10,10)
Starting Location, Firetruck: (225,10)
Intact Ratio: 0.956
Burned Ratio: 0.041
Extinguished Ratio: 0.002

Figure 7: Run 3 Results, 360 Iterations

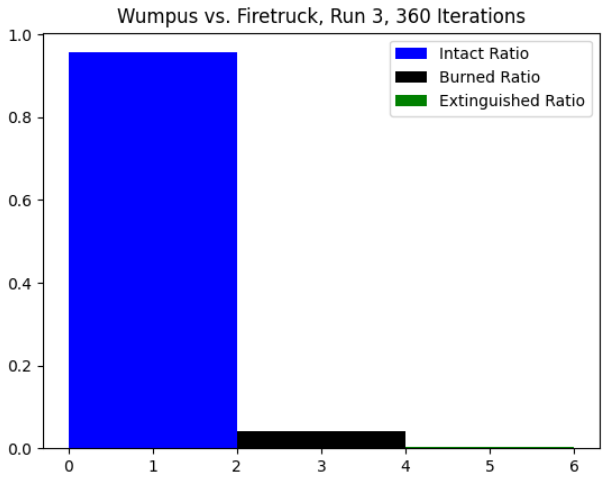

Figure 8: Run 3 Ratio Results Plot, 360 Iterations

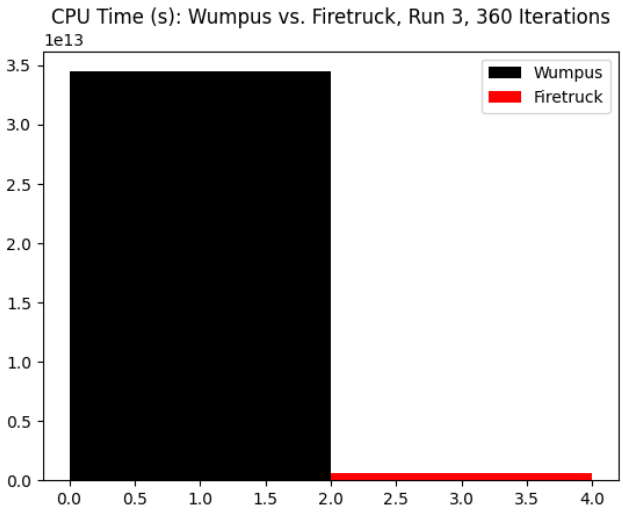

Figure 9: Run 3 CPU Time Results, 360 Iterations

Run 4

Starting Location, Wumpus: (10,10)
Starting Location, Firetruck: (225,10)
Intact Ratio: 0.942
Burned Ratio: 0.055
Extinguished Ratio: 0.002

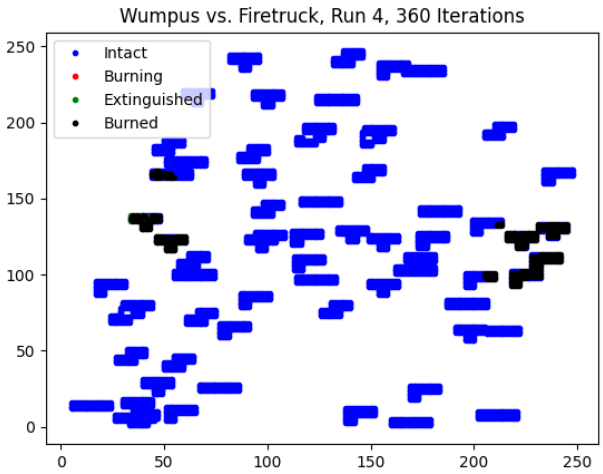

Figure 10: Run 4 Results, 360 Iterations

Figure 11: Run 4 Ratio Results Plot, 360 Iterations

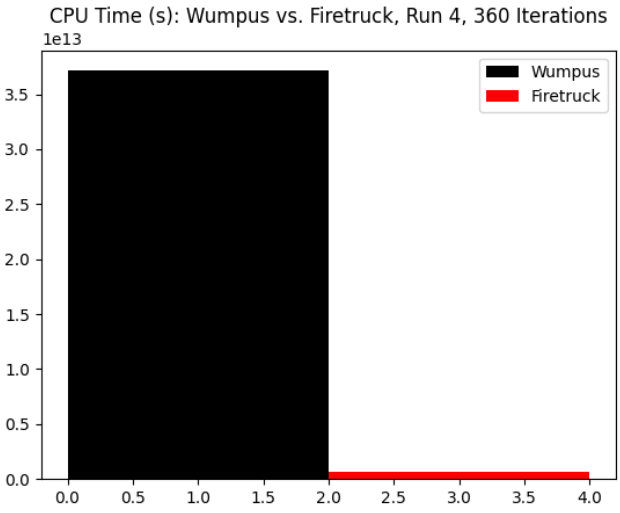

Figure 12: Run 4 CPU Time Results, 360 Iterations

Run 5

Starting Location, Wumpus: (10,10)
Starting Location, Firetruck: (225,10)
Intact Ratio: 0.912
Burned Ratio: 0.0879
Extinguished Ratio: 0.0

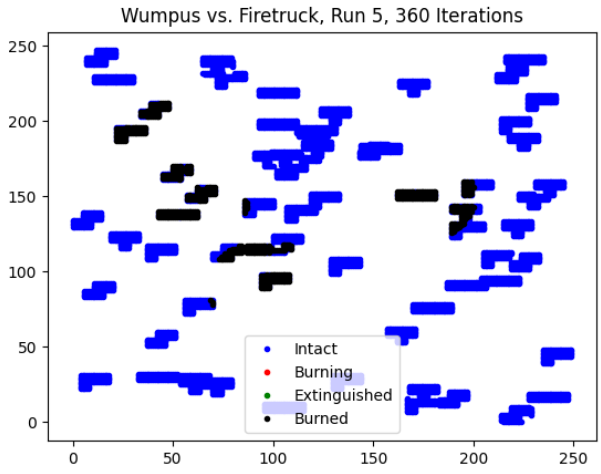

Figure 13: Run 5 Results, 360 Iterations

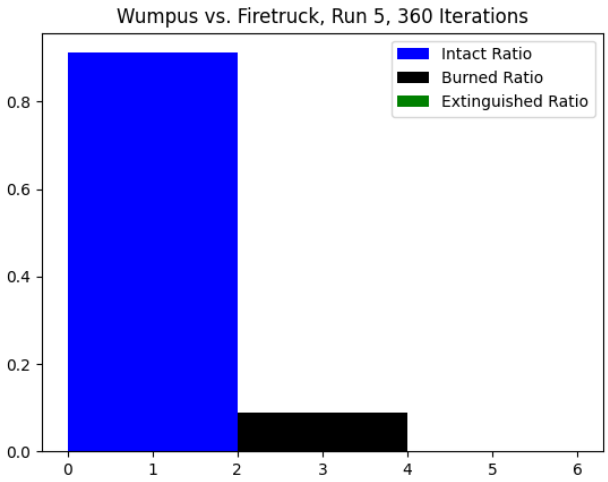

Figure 14: Run 5 Ratio Results Plot, 360 Iterations

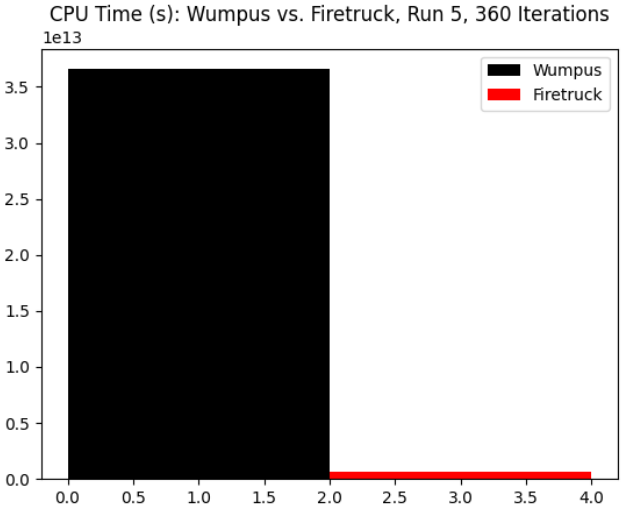

Figure 15: Run 5 CPU Time Results, 360 Iterations

Conclusions

As can be seen by the plots above, the Wumpus had higher computational times as opposed to the firetruck, even though most obstacles remained intact. It should be noted, however, that the number of iterations was shorted to 360 (10% of goal iterations) to help with overall processing time. The large difference

between the processing times of the firetruck vs. the Wumpus leads to the conclusion that generating a roadmap a priori can aid in overall path planning efficiency.

Source Code

Source Code will be attached on Canvas. Files named a_star.py, probabilistic_road_map.py, and pure_pursuit.py used for this assignment were provided by [1]. The repository is located at `https://github.com/AtsushiSakai/PythonRobotics`.

References

[1] Atsushi Sakai, Daniel Ingram, Joseph Dinius, Karan Chawla, Antonin Raffin, and Alexis Paques. Pythonrobotics: a python code collection of robotics algorithms, 2018.