Intro to AI
Homework Assignment 1

Nikita Zagainov DSAI-01

October 31, 2024

1 Implementation Details

1.1 A Star Algorithm

The code submitted to the assignment is an implementation of Agent that uses A* algorithm to find best path to the goal. The implementation uses Dijkstra's algorithm to traverse the map, the heuristic function is implemented as Manhattan distance between the current position and the keymaker position. No modifications were introduced to the A* algorithm in the implementation.

1.2 Backtracking Algorithm

The code submitted to the assignment is an implementation of Agent that uses backtracking algorithm to find best path to the goal. The implementation uses BFS to traverse the map considering known threats, and whenever new information is available on dangers on built path, the agent backtracks to the last known safe position and rebuilds the path to the goal. No significant modifications were made to the backtracking algorithm.

1.3 Agent Behavior

In both implementations, agent is aiming to find the shortest path to the keymaker, ignoring backdoor key (for simplicity), and avoiding threats. A* agent uses A* for finding best path to the keymaker, Backtracking agent uses BFS algorithm with backtracing on running into a threat for finding best path to the keymaker.

1.4 Code

Link to the code repository

2 Evaluation results

The agents were evaluated on 1000 randomly generated maps below is the summary of the results:

Metric	A* Algorithm	Backtracking Algorithm
Mean Execution Time (s)	0.009	0.005
Std Execution Time (s)	0.028	0.013
Var Execution Time (s^2)	0.00079	0.00018
Mean Moves	9.82	10.221
Std Moves	6.085	6.095
Var Moves	37.03	37.14
Win Fraction	0.881	0.909

Table 1: Evaluation results of A* and Backtracking algorithms

As can be seen from the table, the algorithms do not have a big statistical difference in terms of mean moves. However, the execution time of the A* algorithm implementation provided is slightly higher than the execution time of the backtracking algorithm, and the win fraction difference is also not very high.

3 Unsolvable map example

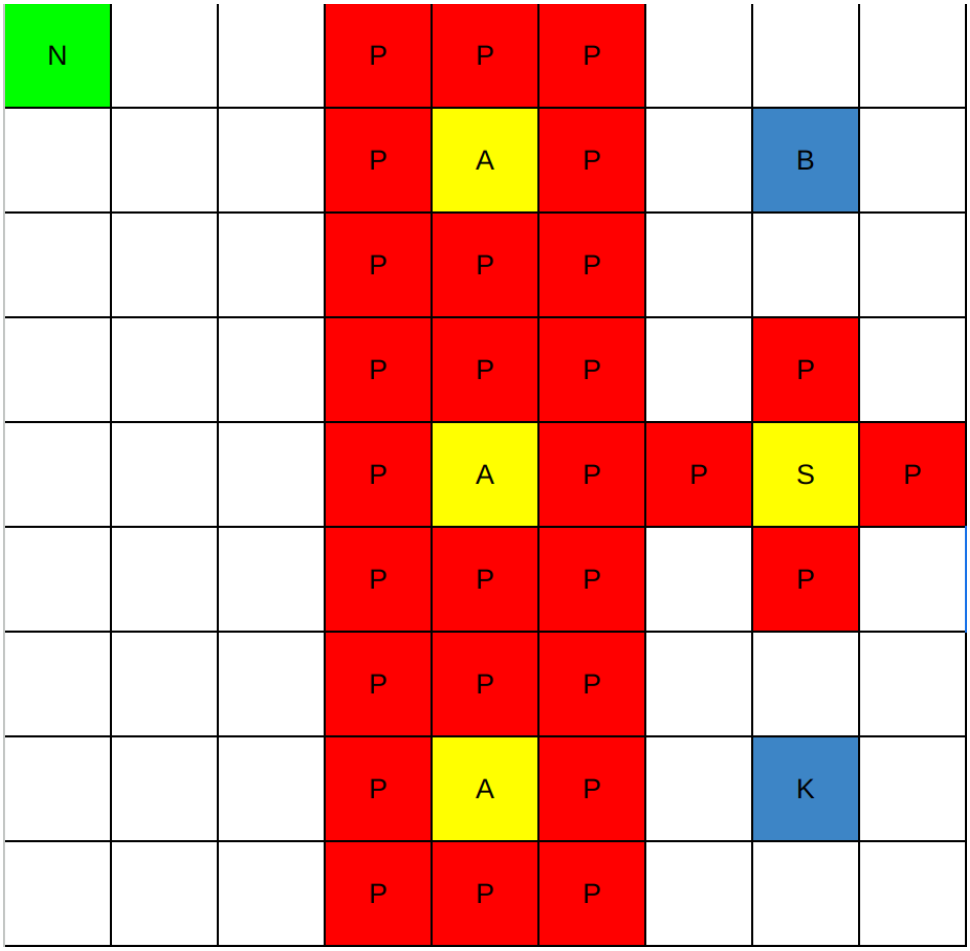

Figure 1: Example of an unsolvable map

3.1 Unsolvable Maps Analysis

During the evaluation, some maps were found to be unsolvable by both A* and Backtracking agents. These maps typically contained isolated sections or completely blocked paths that made it impossible for the agents to reach the keymaker. The primary reasons for unsolvability included:

- Completely blocked paths with no possible route to the goal.
- Maps where the keymaker was surrounded by threats with no safe path available.

The example shown in Figure 1 illustrates a scenario where the path to the keymaker and backdoor key is completely blocked, making it impossible for the agent to find a solution.

4 Conclusion

This assignment solution provides two implementations of the agent that can solve the problem of path finding. Both algorithms were implemented without any modifications. The evaluation results show that the backtracking algorithm has slightly better performance in terms of execution time, but both algorithms show approximately the same performance in terms of mean moves and win fraction.