Stealthy Path Planning

Vikram Ramanathan

GitHub Repo: stealthy_path_planning

1 Objective

This project involved:

1. Developing a graphical representation of a grid of possible robot locations and movements
2. Placing static obstacles or "monsters" randomly in the grid
3. Developing an algorithm that finds an optimal path while taking into consideration the robot movement costs (or edge weights), proximity to "monsters" and proximity to the destination

2 Approach

The stealthy navigation problem entails finding a path that minimizes the exposure of the object under consideration to "areas of consequence" on the grid. Areas of consequence could range from capture zones in games to the operational range of missiles and mines. In my solution to this problem, I decided to consider the player or robot as omnidirectional or capable of moving to any of the 8 cells that surround it. Therefore, while creating the graphical representation of the grid, an individual cell could have 3,5,or 8 neighbors. No obstacles were placed in the environment except for monsters.

There are several methods have been developed to solve the global path planning problem. The majority of solutions often fall under (or have a hint of) either Cost Minimizing Algorithms or Potential Field Methods [HA92]. Professor Oussama Khatib is known for pioneering the potential field method in the context of obstacle avoidance [Kha86]. The potential field method essentially creates a global artificial field of the map by assigning attractive and repulsive potentials to particular areas in the environment (obstacles are assigned repulsive potentials while the end goal and desired waypoints are assigned an attractive potential). The robot experiences a net force which can be evaluated from the potential gradient. This encourages the robot to follow paths through attractive potentials and thereby avoid obstacles. However, the potential field method is known to avoid paths through closely placed obstacles and may define destination nodes that are not reachable (local minima caused by proximity to obstacles) [KB91].

A potential field method in stealth navigation is definitely an interesting problem. In fact, obstacles could be assigned different repulsive potentials to represent their range of influence. However, a potential issue with this method could be excessive robot movement. This could hypothetically be addressed by fine tuning the potential field strengths of the obstacles and employing a cost minimizing algorithm.

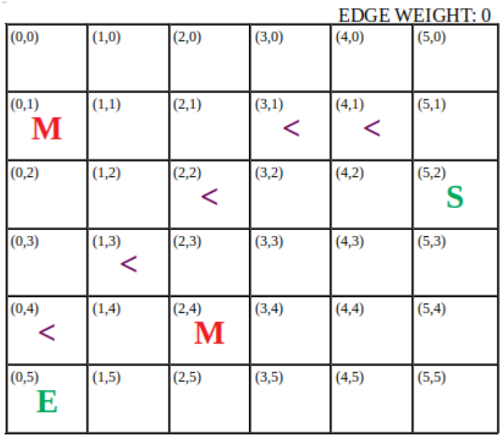
In this solution, I elected to use a cost minimizing algorithm, specifically, A* search with a custom-made heuristic. This heuristic evaluates the minimum Euclidean distance to an obstacle on the grid from the node under study and then subtracts it from the Euclidean distance to the destination node. The result is assigned to the priority of the node in the priority queue used in the A* search.

3 Code

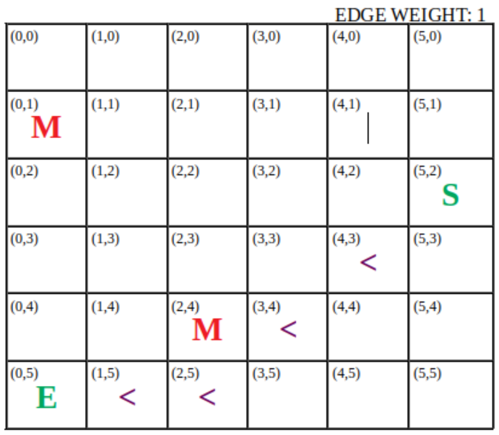
The code for this project can be found here: Stealthy Path Planning GitHub Repository

4 Results/Examples

4.1 6x6 world with 2 Monsters



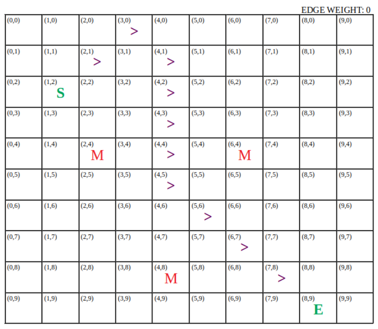
(a) Edge Weight = 0



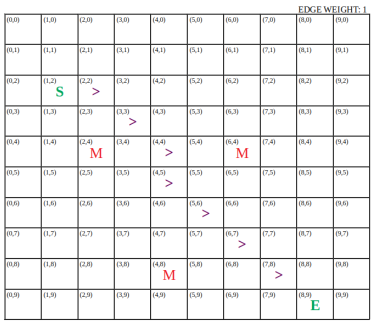
(b) Edge Weight = 1

Figure 1: Stealthy Routes on a 6x6 World with 2 Monsters

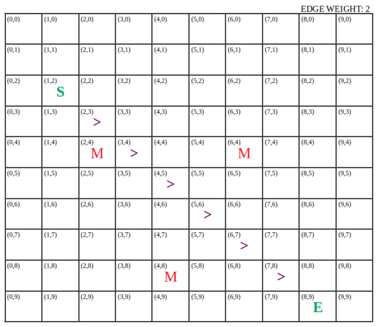
4.2 10x10 world with 3 Monsters



(a) Edge Weight = 0



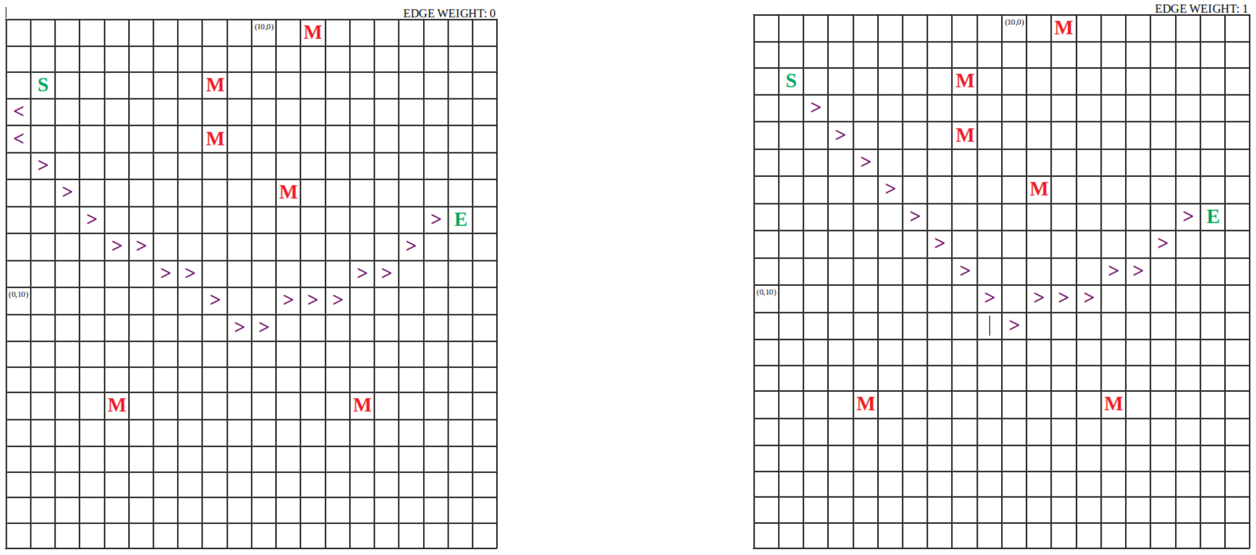
(b) Edge Weight = 1



(c) Edge Weight = 2

Figure 2: Stealthy Routes on a 10x10 World with 3 Monsters

4.3 20x20 world with 6 Monsters



(a) Edge Weight = 0

(b) Edge Weight = 1

Figure 3: Stealthy Routes on a 20x20 World with 6 Monsters

5 Conclusion

Upon completing this project, I thought of an interesting derivative of this problem. We could extend the initial problem statement to involve dynamic obstacles (or moving monsters). A modified D* search would seem like a viable solution. If the motion of the monsters is not random i.e. each monster has a desired state it intends to reach on the grid, we could design an intelligent predictor of each monster's intent and derive path accordingly. Given the need to predict the monsters' intent and find an optimal path only in the beginning and maybe (as a check) sometime half-way, this solution could greatly reduce the need to check all nodes in the frontier of a cost minimizing algorithm.

All in all, I had a blast developing this project. I especially loved being able to compare the path I would think is optimal (solely based on intuition) to the path the program returns. If you wish to discuss my solution with me or talk about similar problems (like the reactive navigation problem above), please reach out. I'd love to!

References

[Kha86] Oussama Khatib. "The Potential Field Approach And Operational Space Formulation In Robot Control". In: *Adaptive and Learning Systems: Theory and Applications*. Ed. by Kumpati S. Narendra. Boston, MA: Springer US, 1986, pp. 367–377. ISBN: 978-1-4757-1895-9. DOI: 10.1007/978-1-4757-1895-9_26. URL: https://doi.org/10.1007/978-1-4757-1895-9_26.

[KB91] Y. Koren and J. Borenstein. "Potential field methods and their inherent limitations for mobile robot navigation". In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. Apr. 1991, 1398–1404 vol.2. DOI: 10.1109/ROBOT.1991.131810.

[HA92] Y. K. Hwang and N. Ahuja. "A potential field approach to path planning". In: *IEEE Transactions on Robotics and Automation* 8.1 (Feb. 1992), pp. 23–32. DOI: 10.1109/70.127236.