Multi-Robot Surveillance

Aashi Manglik Navyata Sanghvi
{amanglik, nsanghvi}@andrew.cmu.edu

Abstract

This project aims to solve the problem of surveillance given multiple robots. Market-based approach has been previously proposed to plan for this task which is optimal within a threshold. The problem with market-based approach is that depending on the initialization, it might take large number of iterations to reach equilibrium. Here, we implemented a complete and optimal search-based planner using a three-level architecture. The proposed approach is evaluated against the greedy planner using planning time and path cost as metrics.

Video Link: *https://drive.google.com/file/d/1KiHVKWjCMLm0HymrAS_CxBcfjP546sEo/view?usp=sharing*

1. Problem Statement

We tackle the problem of surveillance given N robots with starting positions $\mathbf{x}_i^{st} \in \mathbb{R}^2$, goals $\mathbf{x}_i^{g} \in \mathbb{R}^2$, $\forall i \in \{1, ..., N\}$ and M waypoints $\mathbf{y}_j \in \mathbb{R}^2$, $\forall j \in \{1, ..., M\}$, which must be collectively visited by the robots on their paths to their goals. The planning problems here are threefold: (1) which subset of waypoints must be assigned to which robot, (2) once a subset of waypoints has been assigned, in what order must the robot visit them, and (3) how must a robot complete its tour while avoiding obstacles. An important application of this problem is in a fire extinguisher squad, where a limited set of resources must be optimally allocated to put out fires in a number of locations. For robot paths $\pi_i \subset \mathbb{R}^2$ beginning at $\mathbf{x}_i^{st}$, ending at $\mathbf{x}_i^{g}$, with path costs $c(\pi_i)$, $\forall i \in \{1, ..., N\}$, our problem is formally defined as:

$$\min_{\pi_i} \sum_{i=1}^{N} c(\pi_i) \quad \text{s.t.} \quad \bigcup_{j=1}^{M} \mathbf{y}_j \subseteq \bigcup_{i=1}^{N} \pi_i$$

2. Approach

To tackle the three parts of the problem described in the previous section, we construct a three-level graph [1].

Top-Level Graph

This level solves the problem of optimal waypoint assignment. The details are as follows:

- State Space: $\mathbf{z} \in \{0, 1, ..., N\}^M$, where each entry of the vector corresponds to the robot assigned that waypoint, and 0 indicates no assignment.
- Start: $\mathbf{0}^M$, Goal: optimal assignment of waypoints.
- Edges: Assign left-most unassigned waypoint to a robot i. Let the mid-level optimal cost for goal $\mathbf{q}$ be $c^*_{\mathbf{q}}$. Let robot i's source node assignment be $\mathbf{q}_i$, and its target node assignment be $\mathbf{q}'_i$. Then, edge cost: $c^*_{\mathbf{q}'_i} - c^*_{\mathbf{q}_i}$

Mid-Level Graph

For a given robot i with a certain subset of waypoints assigned to it, this level solves the problem of waypoint ordering - a TSP for the allotted waypoints and the robot's start and goal positions. The details are as follows:

- State Space: $\mathbf{q}_i = [\mathbf{z}_1, z_2]$ s.t. $\mathbf{z}_1 \in \{0, 1\}^M$, $z_2 \in \{0, 1, ..., M+1\}$. Here, each entry in vector $\mathbf{z}_1$ represents whether or not that waypoint has been visited by the robot under consideration, and z_2 represents which waypoint was visited last, where 0 represents the robot's start and $M+1$ represents the robot's goal.
- Start: $[\mathbf{0}^M, 0]$, Goal: $[\mathbb{I}(\text{visited waypoints}), M+1]$
- Edges: Robot i, originally at z_2, next visits another assigned waypoint z'_2. Edge cost: low-level cost of moving from z_2 to z'_2.

Low-Level Graph

For a robot i with a certain waypoint ordering, this level solves the low-level navigation problem between waypoints around obstacles. The details of the 8-connected $L \times B$ grid are as follows:

- State Space: $\mathbf{x}_i = [z_1, z_2]$ s.t. $z_1 \in \{1, ..., L\}$, $z_2 \in \{1, ..., B\}$
- Start: $\mathbf{x}_i^0$, Goal: $\mathbf{x}_i^{end}$
- Edges: 8-connected grid. Edge cost: Traversed cell cost.

3. Experiments and Results

Scalability

All experiments are conducted on the grid of size 500 $\times$ 500 with obstacles. Keeping the number of robots fixed as 3, the optimal planner could solve the multi-surveillance problem upto 5 waypoints whereas the greedy planner could scale upto 7 waypoints. An example of paths output by optimal planner and greedy planner are respectively shown in figure 3 and 4.

Planning time and Path Quality

The comparison of time taken by greedy planner and optimal planner is shown in figure 2. In the case of 2 robots and 7 waypoints, the optimal planner (315.146 seconds) took roughly 5 times the time taken by greedy planner (65.1326 seconds). On the other hand, the path quality of optimal planner (1480.12) is 1.5 times better than the greedy planner (2198.02) as shown in figure 1. Plots 2 and 1 show the relative performance of both planners in various tests ranging from 2 to 4 robots and 1 to 7 waypoints.

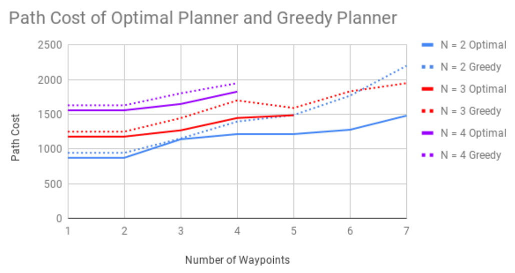

Figure 1. Path Quality of Optimal Planner vs Greedy Planner

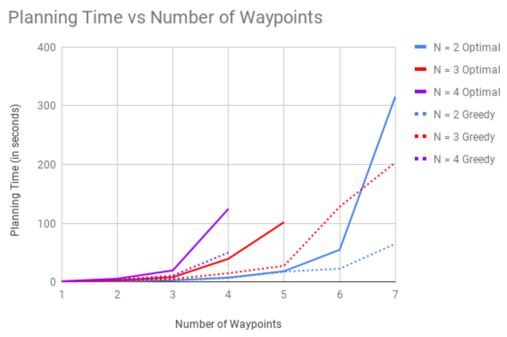

Figure 2. Planning Time of Optimal Planner vs Greedy Planner

4. Conclusions

Considering the pros and cons of two approaches presented in this report, it depends on the application domain whether we want to trade-off optimality for quicker planning or vice-versa. For the purpose of developing fire-extinguisher squad, the greedy approach might be better suited. As suggested by Professor Maxim Likhachev at the end of oral presentation, the planning time of optimal approach can be improved by reusing the paths of mid level and low level graphs. A trivial implementation of this suggestion will increase the memory requirement which is the main bottleneck in the scalability of planner. Thus, reusability of paths need to be thought of smartly without increasing the space complexity of algorithm significantly.

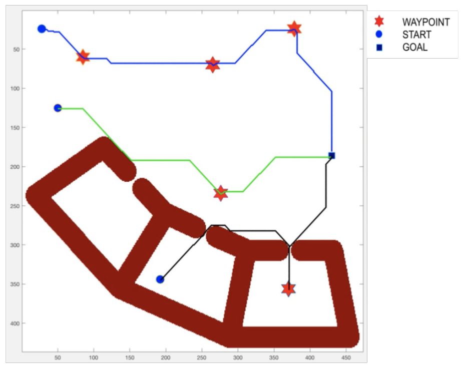

Figure 3. Test environment with 5 waypoints and 3 homogeneous robots: Paths computed by optimal planner

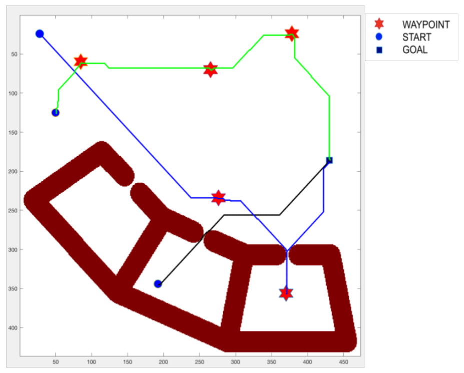

Figure 4. Test environment with 5 waypoints and 3 homogeneous robots: Paths computed by greedy planner

References

[1] D. Thakur, M. Likhachev, J. Keller, V. Kumar, V. Dobrokhodov, K. Jones, J. Wurz, and I. Kaminer. Planning for opportunistic surveillance with multiple robots. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5750–5757. IEEE, 2013. 1