Online Stochastic Planning for Multimodal Sensing and Navigation under Uncertainty

Shushman Choudhury,* Nate Gruver,* and Mykel J. Kochenderfer*†
*Department of Computer Science
†Department of Aeronautics and Astronautics
{shushman,ngruver,mykel}@stanford.edu
Stanford University

***Abstract*—Small unmanned aircraft are typically equipped with multiple sensors for flight through unknown environments. Due to onboard energy constraints, they must balance sensing and movement. We frame this problem as a Partially Observable Markov Decision Process (POMDP) and show that online stochastic planning performs well compared to a deterministic myopic baseline. In contrast to most online POMDP solvers, which sample states, we motivate planning with a belief space MDP to simulate and heuristically encourage sensing appropriately, and show that this outperforms the particle-sampling POMCP solver. Our initial experiments with partially observed gridworlds support our hypotheses and raise interesting questions about contending with huge observation spaces, rollouts in belief space and reward sparsity in online planning.**

I. Introduction

We consider the problem of navigation in an unknown stochastic environment by an agent with multiple sensors and limited energy. This problem is relevant for small unmanned aircraft, where the onboard battery life is a crucial factor. A judicious strategy for sensor usage and movement is critical to the success of the mission. The stochasticity and partial observability of the environment make the problem challenging, as does the need to reason about both sensing and movement.

There are many related works that we only briefly comment on here. The dynamic sensor selection problem has been addressed through various theoretically justified methods - maximum flow graphs [1], minimizing error covariance [2], Rényi divergence [3], and convex optimization [4]. However, these methods are open-loop and myopic, i.e. they do not reason about the future effect of sensing on the environment. More recently, closed-loop policy frameworks for sensor selection have been developed by Spaan and Lima [5] and Satsangi et al. [6], but while their formulation is similar to ours, they consider offline solution methods which have trouble scaling to large state and observation spaces.

Our key idea is to formulate an adaptive approach using online POMDP planning [7], described in Section II, that reasons jointly about multimodal sensing and navigation actions, their outcomes on the environment, and their costs. We examine important issues like choosing between different sensors, the structure of the belief space, and overcoming reward sparsity in rollouts, as described in Section III.

We discuss our observations in Section IV. First, we design a simple but interpretable problem with complementary sensing modalities and show that online planning can do well in varying settings, compared to a greedy baseline. Additionally, we obtain better performance by solving a belief space MDP online rather than particle sampling. Though the belief-MDP approach requires a simplifying assumption about the observation space for tractability, it benefits from simulating the effects of sensing actions on the belief space. Finally, we show significant improvement in performance due to incorporating rollout heuristics for sensing and movement towards the goal. We conclude with some comments in Section V. Figure 1 shows an overview of our setup and an exemplary observation.

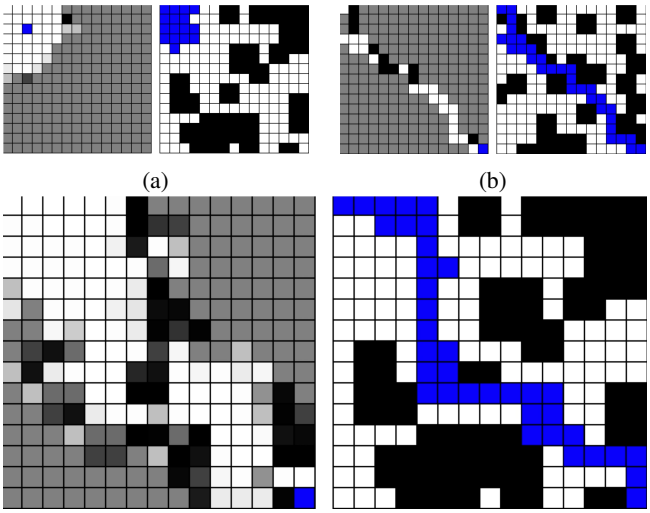


Fig. 1: Our POMDP formulation induces intuitive sensing and movement with an online solver. Three trajectories are shown on an example with different penalty parameters for entering a no-fly (black) cell. The left panels represent the belief of the solver about the environment and the right panels show the true environment and the agent movements. (a) Large penalty causes conservative behaviour with little exploration. (b) Small penalty causes a brash lack of sensing. (c) Reasonable penalty yields both exploration and progress.

II. Problem Formulation

Our problem environment is a binary gridworld, where cells are either fly zones or no-fly zones. The agent starts from

the top left and can move in the 4 principal directions. The state of a cell (fly/no-fly) is known for certain only after the agent occupies it. The agent has multiple sensors, with varying properties, for observing surrounding cells. The objective is to reach the lower right corner with minimum overall cost, where penalties are due to movement distance, sensing energy, and no-fly zones entered, if any.

A. POMDP Model

Here we describe the POMDP [8] components for our setting:

- State space $S \equiv \{G, E, \mathbf{x}\}$. $G_{n\times n}$ is the grid, i.e. $G[\mathbf{x}] = \mathbb{1}[\mathbf{x} \in NFZ]$, E is the current energy consumed and $\mathbf{x}$ is the current location.
- Belief space B. The belief is over the states of the grid cells. We assume cell states are independent of each other, i.e. $b(G[\mathbf{x}]) \equiv P(\mathbf{x} \in NFZ)$. Further comments on this are in Section III-B. We assume that the energy consumption and the current location are fully observed.
- Observation space Z. Each observation $\hat{G}_{n\times n}$ is a grid of tuples of the sensor observation and its confidence at each cell, $\hat{G}[\mathbf{x}] = (\mathbb{1}[\mathbf{x} \in NFZ], \mathrm{Conf}(\mathbf{x}))$.
- Action space $A = \{N, S, E, W, \mathrm{Sensor}_1 \ldots \mathrm{Sensor}_m\}$ for moving or taking a sensor reading.
- Transition function T. For simplicity we assume deterministic dynamics. For $a \in \{N, S, E, W\}$ the new location $\mathbf{x}'$ is the appropriate new cell and for $a = \mathrm{Sensor}_i$ the energy is updated by ΔE_i.
- Observation function O pertains only to sensing actions, $O(\mathbf{x}' \mid \mathbf{x}, a = \mathrm{Sensor}_i) = f_i(\mathbf{x}, \mathbf{x}')$ where f_i is the ith sensing function; $f_i(\mathbf{x}, \mathbf{x}') = P(G[\mathbf{x}] = G[\mathbf{x}'])$ depends on the sensor model and the distance between $\mathbf{x}$ and $\mathbf{x}'$.
- Reward function R has $-\lambda_{\text{move}}$ if $a \in \{N, S, E, W\}$ and ΔE_i if $a = \mathrm{Sensor}_i$. Additionally, a penalty of λ_{NFZ} is incurred if the agent enters an NFZ cell, and a reward of λ_{succ} is obtained when the agent reaches the goal. The λ's are tunable parameters but obey

$$\lambda_{\text{succ}} > \lambda_{NFZ} > \lambda_{\text{move}}$$

The start location is $\mathbf{x}_s = (0, 0)$ and the goal is $\mathbf{x}_g = (n, n)$ for all grids. Our problem is undiscounted.

B. Sensors

We define three simple sensor models for our experiments, each based on a realistic robotic sensor. They have somewhat complementary characteristics in terms of fidelity, spread and energy consumption.

- **Time-of-Flight (TOF).** - This sensor observes cells along a line of width 1 in a certain direction. It has moderate fidelity and consumes minimum energy. This sensor has 4 associated actions, one for sensing in each direction.
- **Spinning LIDAR.** This sensor observes cells all around the agent, up to a certain radius. It has fidelity similar to the ToF and consumes more energy than it.
- **HD Camera.** This sensor observes a block of cells of width 3 in a certain direction. It has the highest fidelity and consumes the most energy. It also has 4 actions for the 4 directions.

In each case, the confidence of the observation falls with the Manhattan distance of the observed cell from the agent's location, using exponential decay. The parameter details are omitted for space.

III. Approach

We considered online solution methods due to the very high dimensional state and observation spaces. In this section we describe two related online planning algorithms that we used, and a greedy baseline approach based on established ideas.

A. POMCP

The POMCP algorithm [9] determines the best action to take from the current belief state, using Monte-Carlo Tree Search (MCTS) to evaluate the utility of actions using forward-simulations, and particle filters to track belief propagation. To use POMCP, we require generative models for states and observations, according to our earlier rules.

We also define a function $b' \leftarrow \mathrm{UpdateBelief}(b, a, o)$ that the outer loop of POMCP invokes to update the belief state after an action is taken and a true observation received from the world. After some sensing action, we update the belief of the cells for which the sensor's confidence is > 0.5. For each such cell (i, j), we do a Bayesian belief update:

$$P\left(G'[i,j] = 1 \mid \hat{G}(i,j), b(G[i,j])\right) =$$

$$\begin{cases} \frac{\hat{G}[i,j][2]b(G[i,j])}{\hat{G}[i,j][2]b(G[i,j])+(1-\hat{G}(i,j)[2])(1-b(G[i,j]))} & \text{if } \hat{G}[i,j][1] = 1 \\ \frac{(1-\hat{G}[i,j][2])(b(G[i,j]))}{(1-\hat{G}[i,j][2])(b(G[i,j]))+\hat{G}(i,j)[2](1-b(G[i,j]))} & \text{if } \hat{G}[i,j][1] = 0 \end{cases}$$

where $\hat{G}[i,j][1]$ is the binary observation at (i, j) and $\hat{G}[i,j][2]$ is the confidence at (i, j) as per the sensor model.

There are two important limitations. First, positive rewards are very sparse, occurring only at the goal state. With a large action space, rollouts are unlikely to reach the goal for most grid cells. We handle sparsity somewhat by incorporating a heuristic in the rollout reward that encourages movement towards the goal (described in Section IV-B). Second, POMCP uses an unweighted particle filter representation for the rollouts, where each particle state maintains a possible world map sampled from the current belief state. The effects of sensing actions on the belief map are not explicitly captured by the rollouts as the sampled particles do not carry around the belief state. This observation motivated our second approach.

B. Belief Space MDP with MCTS

Any POMDP has an equivalent MDP in belief space. State-of-the-art online solvers like POMCP typically do not solve the Belief-MDP directly as the belief space is exponential in the size of the state space. Instead, they sample states or scenarios. This particle sampling does not allow changes in belief state to be incorporated into the rollout. Working directly with the Belief-MDP, however, avoids this drawback, and lets us simulate the effect of sensing on the belief during rollouts.

Reasoning about belief space has been presented in related contexts. The typical state-based reward function is unsuitable when the agent has an explicit incentive to reduce uncertainty. Araya et al. [10] introduced the idea of belief-based rewards for POMDPs, and showed how to incorporate them in existing offline solvers. More specifically, Spaan and Lima [5] introduced the POMDP with Information Rewards framework for active perception, Satsangi et al. [6] considered dynamic sensor selection, and Dressel and Kochenderfer [11] extended a state-of-the art offline solver to take information seeking actions for localization. Unlike these works, however, we consider an online planning framework which is more scalable to large spaces and supports more general reward functions.

Due to our independence assumption for grid cell states, the observation space is linear in the grid size, making the Belief-MDP tractable with online planning. This assumption is restrictive, but we wanted to examine if the benefit of doing rollouts in belief space outweighs the simplifying effect of the assumption. The Belief-MDP differs from the POMDP only in the transition and reward functions; the transformation is standard and we omit it here for space. We do discuss an additional heuristic we used for the Belief-MDP in Section IV-B. We use the online MCTS solver for this Belief-MDP.

C. Greedy Baseline

Our baseline approach has two standard components - greedily sensing based on expected information gain [12] and moving via receding horizon path planning [13]. This method does not depend on rollouts and is less affected by reward sparsity, thereby offering a decent benchmark and comparison, although it is deterministic and myopic. There are two steps:

1) IG-based Sensing:

- Evaluate the change in confidence for $a = \text{Sensor}_i$

$$\Delta\text{conf}(G, \mathbf{x}, a) = \sum_{\mathbf{x}' \in G} \sum_{o} |P(o \mid \mathbf{x}, a)(b'(\mathbf{x}') - b(\mathbf{x})|$$

- Execute the sensing action with largest Δconf.
- Repeat until $\max_a \Delta\text{conf}(G, \mathbf{x}, a) < \epsilon$, where ϵ is a parameter for minimally useful information gain.

2) Movement along unobstructed path:

- Locate the waypoint $\mathbf{w}$ closest to the goal coordinates (by Manhattan distance) such that $b(G[\mathbf{w}]) < 0.5$.
- Plan the minimally obstructed path from current location $\mathbf{x}$ to $\mathbf{w}$ and take the first movement along this path.

The behaviour of this algorithm is thus to sense until fairly confident of its surroundings and then move along the apparently unobstructed path towards the goal.

IV. Results and Discussions

We ran various experiments in our gridworld setup with the three methods described. In the interest of space, we discuss only representative results that give us insights about the overall framework. Some implementation comments are in order. First, though our problem is an infinite-horizon one, and terminates when the goal state is reached, for our tests we terminated a trial if it took more than 1000 iterations. The accumulated reward for that case is the cost incurred till that point. Second, there are several parameters, for the problem as well as the solvers, that could affect the performance. We simply chose a realistic set of problem parameters and a consistent set of solver parameters that they do well with. The specific values of the rewards accrued are immaterial; only the relative performance matters.

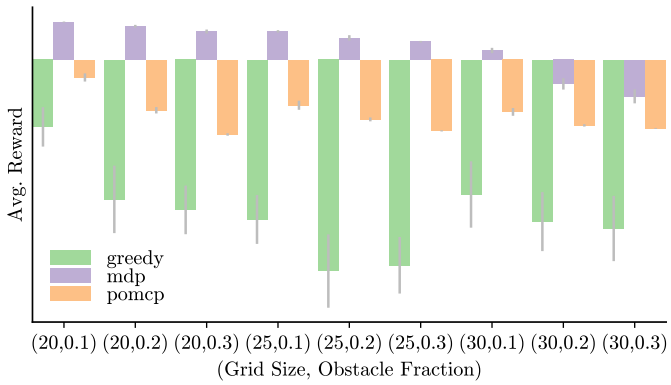


Fig. 2: The mean (bar) and standard error (line) of the rewards for the three approaches over 9 different problem settings. The Belief-MDP with MCTS does the best, but it is more affected by increasing problem size than POMCP is. The performance of the greedy baseline has very high deviation.

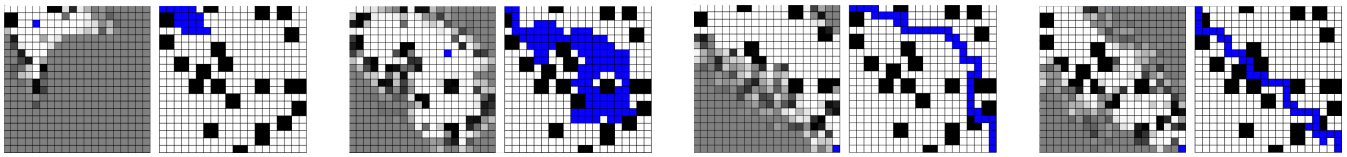
A. General Performance

Our first set of results compares the three approaches described above over a range of problem characteristics, defined by grid side length and the fraction of no-fly cells. For each setting, we averaged over 5 different gridworlds and for the stochastic solvers (other than greedy), over 10 trials for each gridworld. The statistics are shown in Figure 2. POMCP used a movement heuristic in the rollout reward as mentioned in Section III-A, while MCTS for Belief-MDP used movement and sensing heuristics (explored more in Section IV-B).

The results suggest that there is a benefit to rollouts in a belief space MDP when sensing is an action, even if it requires a simplifying independence assumption for tractability. However, as the grid becomes larger, so do the state and (true) observation spaces, and this assumption becomes more of a problem. Consequently, the performance of Belief-MDP degrades much more than that of POMCP, and the relative gap in their performance reduces significantly.

B. Effect of Rollout Heuristics

We examined the effect of incorporating heuristics for MCTS to help overcome the extreme sparsity of rewards in this problem which we motivated earlier. The idea of directing the search according to heuristic knowledge was introduced as 'progressive bias' by Chaslot et al. [14], and it modifies the selection strategy, but they flag an issue of scalability to large state spaces. In contrast, we applied time-inexpensive



(a) No heuristic (b) Sensing heuristic (c) Movement heuristic (d) Both heuristics

Fig. 3: A comparison of different heuristics used in search with the Belief-MDP solver. (a) Having no heuristic leads to some progress towards goal and little exploration of the terrain. (b) Sensing heuristic alone leads to heavy exploration but low progress towards the goal. (c) Movement heuristic alone leads to the avoidance of crowded obstacles. (d) Using both heuristics yields trajectories that do not shy away from obstacle-dense areas.

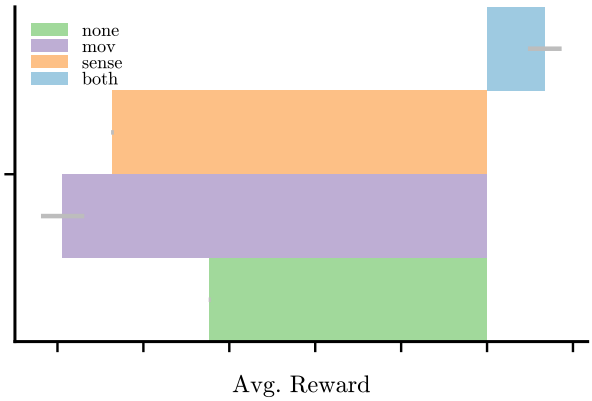


Fig. 4: A quantitative comparison of different heuristics used in search over the Belief-MDP over an intermediate (25,0.2) grid. The performance increases significantly with both heuristics. Note that the performance with only one heuristic is worse than that for neither, which indicates that a heuristic for only one kind of action may be detrimental. The specific numerical values are omitted as the relative behaviour is what matters.

modifications to the rollout reward function, which led to a considerable improvement in performance.

First, we added the negative Manhattan distance between the location in the rollout state and the goal coordinates (this was done in POMCP too). Second, we subtracted $(\Delta\text{confidence}(G, x) - k)$ when taking sensor action x. The k parameter was calibrated to ensure only worthwhile and not just positive changes in confidence were incentivized. The qualitative impact of these heuristics is shown in Figure 3 and a quantitative comparison over a specific intermediate setting is shown in Figure 4.

These results suggest that simple heuristics can impact online solvers in large problems with reward sparsity, provided they do not encourage only one type of action. The sparsity is particularly an issue in the online setting, where the idea of eligibility traces, introduced to overcome sparsity in offline problems, cannot directly be applied.

V. Conclusion

We investigated the effectiveness of an online stochastic planning framework for the multimodal sensing and navigation problem, which we modeled as a POMDP. In particular we considered if rollouts in belief space are useful for sensing actions. Our results are quite promising and bring up directions for future work, such as a more formal analysis of rollout heuristics, using reinforcement learning for good parameters, and tractable extensions to real-world problem settings.

References

[1] M. A. Perillo and W. B. Heinzelman, "Optimal sensor management under energy and reliability constraints," in *IEEE Wireless Communications and Networking Conference*, vol. 3, 2003, pp. 1621–1626.

[2] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.

[3] C. Kreucher, K. Kastella, and A. O. Hero, "Sensor management using an active sensing approach," *Signal Processing*, vol. 85, no. 3, pp. 607–624, 2005.

[4] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.

[5] M. T. Spaan and P. U. Lima, "A Decision-Theoretic Approach to Dynamic Sensor Selection in Camera Networks," in *International Conference on Automated Planning and Scheduling (ICAPS)*, 2009.

[6] Y. Satsangi, S. Whiteson, F. A. Oliehoek *et al.*, "Exploiting Submodular Value Functions for Faster Dynamic Sensor Selection," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2015, pp. 3356–3363.

[7] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa, "Online planning algorithms for POMDPs," *Journal of Artificial Intelligence Research*, vol. 32, pp. 663–704, 2008.

[8] M. J. Kochenderfer, *Decision making under uncertainty: Theory and application*. MIT Press, 2015.

[9] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 2164–2172.

[10] M. Araya, O. Buffet, V. Thomas, and F. Charpillet, "A POMDP extension with belief-dependent rewards," in *Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 64–72.

[11] L. Dressel and M. J. Kochenderfer, "Efficient decision theoretic target localization," in *International Conference on Automated Planning and Scheduling (ICAPS)*, 2017.

[12] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[13] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *American Control Conference (ACC)*, vol. 6, 2004, pp. 5576–5581.

[14] G. Chaslot, M. H. Winands, H. J. V. D. Herik, J. W. Uiterwijk, and B. Bouzy, "Progressive strategies for Monte-Carlo tree search," *New Mathematics and Natural Computation*, vol. 4, no. 03, pp. 343–357, 2008.