Autonomous Avalanche Rescue

Yarden Kelmann
CU Boulder Smead Aerospace
yarden.kelmann@colorado.edu

Andrew Kabos
CU Boulder Smead Aerospace
andrew.kabos@colorado.edu

I. Introduction

Avalanches pose a significant risk to those venturing into the outdoors in the winter months throughout the mountainous areas of the world and especially in Colorado. Because of the high elevation and simultaneously sunny and cold climate throughout most of the year in the Rocky Mountains, Colorado's peaks have a tendency to form a dangerous form of unstable snow called persistent slab, whose top layer is hardened by repeated daily melting during the day and freezing throughout the night. The inherent instability of the snow, however, comes from the layer underneath, which remains frozen and granular, thereby lacking the structural integrity to maintain its form in all conditions, especially when found on slopes between 30° and 45°. Coincidentally, slopes between 30° and 45° are also some of the most popular for outdoor enthusiasts to visit while back country skiing, snowmobiling, and snowshoeing. This, along with the proliferation of outdoor adventure seeking as a common hobby has seen a large rise in avalanche injuries and deaths over the past couple of decades.

Fortunately, there exists a large number of preventative and corrective measures that outdoor enthusiasts can employ when traversing the winter mountains including observing avalanche forecasts closely and remaining conservative in their decisions to go out at all as well as knowing how to use any and all necessary safety equipment in the event that they do in fact get caught in an avalanche. One of the most key pieces of life saving equipment that one can carry is an avalanche transceiver, which when turned on emits magnetic flux lines that can lead a rescuer to the location of a buried avalanche victim, as shown in the visualization below.

Fig. 1. Caption of the image

This area is of great interest and concern especially for those exposed to conditions similar to those found in locations such as the Rocky Mountains, and in this paper we seek to simulate an avalanche search and rescue scenario as a POMDP with an autonomous searcher that navigates a simplified yet representative gridworld environment in search of an avalanche victim buried under the snow. Each cell of the gridworld environment represents a piece of avalanche terrain as well as a possible location of the searcher or victim. The searcher is to observe the readings from the avalanche transceiver mentioned in order to reach the victim in as short of a time as possible. Through the use of POMDP solvers in our simulation environment, we aim to test the efficacy of intelligent decision-making algorithms to the field of avalanche rescue operations.

II. Background and Related Work

Partially Observable Markov Decision Processes, or POMDPs, provide a robust format for modeling systems that can, to some degree, simulate a more real environment than those made with more traditional MDPs. This is largely because they contain environmental uncertainty within the problem, which, even when applied as a discrete state grid world, is more akin to the true reality of the world, where the true state of any environment is hardly ever known with full confidence. A commonly observed downside of using the POMDP approach to autonomous planning is the "Curse of Dimensionality" [1], where increasing the size of the state space linearly creates an exponentially larger amount of computational complexity.

With recent advancements in computation methods, POMDPs, however, have become an increasingly attractive option for modeling complex problems in this more grounded manner [1]. After the more recent major foray into the exploration of POMDPs, general POMDP formulations have seen use in various topics of research within localization and navigation including warehouse robots as well as mining operations [3].

This is especially applicable to the rising field of robot assisted search and rescue, where conditions are considerably variable and non-ideal. In our particular desired use case, error can accrue within the flux line measurement output for an avalanche transceiver from any number of interference sources including large metal objects in close proximity to the transceiver, e.g. ski poles, as well as other sources such as cell phones, cameras, radio equipment, power lines, electrical storms, and anything else that may generate electrical interference. This systematic proclivity to error drives those seeking to simulate such a situation to incorporate POMDPs into their research. This subsequently leads to explorations of topics such as beginnings of integration of POMDP based reasoning into search and rescue scenarios where one of the primary objectives is support of human personnel rather than just solo autonomous environmental search [2].

Some common solver algorithms that are implemented with POMDPs tend to follow either an online or an offline approach. Online solvers, like POMCP, use a computationally efficient variation of Monte-Carlo tree search in order to attempt to attack larger problems more effectively in real time, proving effective in problems that include a POMDP for searching for ships in the common game Battleship [4]. Offline solvers, like SARSOP, in a slightly less scalable and somewhat less realistic fashion, simulate the entire state space in order to develop a policy that takes an optimal step at every belief state; SARSOP in particular prunes its alpha vectors in order to quickly converge to optimal policies, which can be seen with favorable results tests in simultaneous localization and mapping environments [5]. Thus, we found these algorithms to be enticing choices for application in a search and rescue simulation.

III. Problem Formulation

We simplify the environment of a mountain side avalanche debris field to a discrete grid world, where each grid point represents a possible position within the field. The victim is located at a grid position and is carrying a transceiver that emanates magnetic flux lines. Within the field, there are impassable cliffs that the searcher must navigate around.

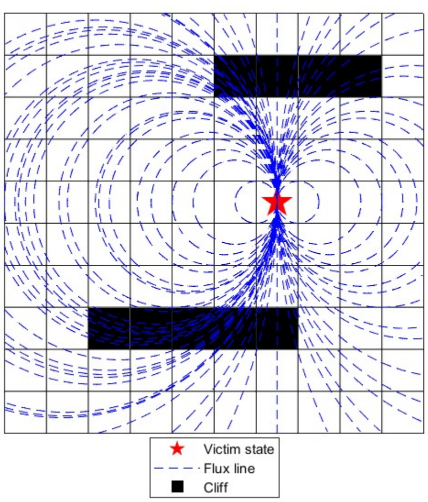

Fig. 2. Rescue Simulation Environment

The 10×10 visualized environment above is used for our rescue simulations. We then formulate the rescue mission as a POMDP in discrete space. Considering a $N \times N$ grid world, the POMDP definition is as follows:

A. State Space

$$S = \{[1,1], [2,1], [3,1], ..., [N,N], \text{found}\} \tag{1}$$

Each coordinate in the state space corresponds to a grid position, where [1,1] is the bottom left position and [N,N] is the top right position. The 'found' state is a terminal state in which the victim has been found.

B. Action Space

$$\begin{aligned} A = \{&[1,0],[-1,0],[0,1],[0,-1],[1,1],[1,-1],\\ &[-1,1],[-1,-1],\text{look},\text{dig}\} \end{aligned} \tag{2}$$

Each of the eight vectors in the action space is a movement from a grid position to an adjacent grid position. The 'look' action is reading a measurement from the transceiver. The 'dig' action is digging for the victim at the current grid position.

C. Reward Function

Let s_{victim} be the grid position of the victim.

$$R(s,a) = \begin{cases} 100 & \text{if } s = s_{victim},\quad a = \text{dig} \\ -2 & \text{if } a = \text{look} \\ -50 & \text{if } a = \text{dig} \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

Digging at an incorrect grid position is extremely costly as it jeopardizes finding the victim alive.

D. Transition Function

Let S_{cliff} be the set of all grid positions where a cliff exists. Let S_{adj} be the set of all grid positions adjacent to s, and $|S_{adj}|$ is the length of the set.

$$T(s'|s,a) = \begin{cases} 1 & \text{if } s = s_{victim},\quad a = \text{dig},\quad s' = \text{found} \\ 1 & \text{if } s = \text{found},\quad s' = s \\ 1 & \text{if } a \in \{\text{look,dig}\},\quad s' = s \\ 0.7 & \text{if } s' = s + a,\quad s' \notin S_{cliff} \\ \frac{0.3}{|S_{adj}|} & \text{if } s' \neq s + a,\quad s \in S_{adj} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The transition probability distribution over the adjacent states of s captures the realistic risk of uncoordinated steps when trudging through the snow.

E. Observation Space

The searcher observes the environment by looking at its transceiver, which provides a reading of distance and direction along a flux line. We simplify the transceiver's physics by assuming perfectly circular flux lines as visualized in Fig. 2. For the searcher in state s, let arc_s be the minimum distance along a flux line from the searcher to the target. Let dir_s be the direction action in A that is nearest to the tangent of the flux line. Consider a single flux line:

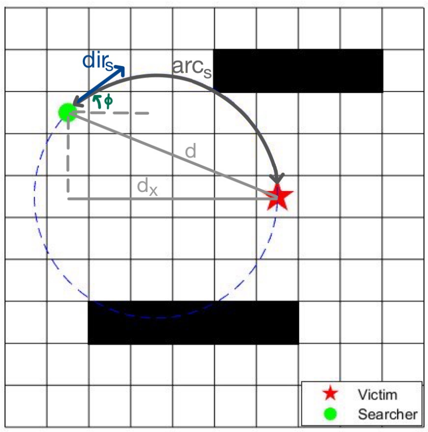

Fig. 3. Transceiver Reading

As illustrated in Fig. 3, let d be the distance between the searcher state and victim state, let d_x be the horizontal component of d, and let ϕ be the angle between the tangent of the flux line at the searcher state and the horizontal. To calculate arc_s and ϕ:

$$\text{arc}_s = \left[\pi - 2cos^{-1}(\frac{d_x}{d})\right] \frac{d^2}{2dx} \tag{5}$$

$$\phi = 2cos^{-1}(\frac{dx}{d}) - \frac{\pi}{2} \tag{6}$$

dir_s is then derived as the nearest direction in the action space corresponding to the angle ϕ. Hence:

$$O = \{(\text{arc}_s, \text{dir}_s) | s \in S\} \tag{7}$$

F. Observation Function

We define the function flux(s) which returns the transceiver reading $(\text{arc}_s, \text{dir}_s)$. Again, S_{adj} is the set of all grid positions adjacent to s. Let $S_{adj,rec}$ be the set of all grid positions adjacent to the grid position where the most recent "look" action was taken. Let S_{rem} be the set of all remaining states which are neither equal to s' nor $\in S_{adj}$.

$$Z(o|a = \text{look}, s') = \begin{cases} 0.9 & \text{if } o = \text{flux}(s') \\ \frac{0.09}{|S_{adj}|} & \text{if } o = \text{flux}(s), \quad s \in S_{adj} \\ \frac{0.01}{|S_{rem}|} & \text{if } o = \text{flux}(s), \quad s \in S_{rem} \end{cases} \tag{8}$$

$$Z(o|a \neq \text{look}, s') = \begin{cases} 0.2 & \text{if } o = \text{flux}(s') \\ \frac{0.7}{|S_{adj,rec}|} & \text{if } o = \text{flux}(s) \quad s \in S_{adj,rec} \\ \frac{0.01}{|'_{rem}|} & \text{if } o = \text{flux}(s), \quad s \in S_{rem} \end{cases} \tag{9}$$

Effectively, when the searcher looks at the transceiver, the reading is 90% accurate. When the searcher takes any other action, it discerns its probable state by considering its previous transceiver reading.

G. Discount

$$\gamma = 0.9 \tag{10}$$

The discount factor enforces the urgency of the rescue mission. It is still beneficial to eventually find the victim, but the searcher must get there as quickly as possible to increase the chance of survival.

IV. Solution Approach

In order to implement our avalanche rescue simulation, we programmed our problem into a QuickPOMDP model using the off the shelf packages POMDPs.jl and QuickPOMDPs.jl in Julia. The objective was then to calculate an optimal policy that maximizes the accumulated discounted reward defined above. To solve the POMDP, we unleashed various offline and online algorithms, namely QMDP, SARSOP, and POMCP. We implemented QMDP from scratch and utilized the SARSOP and POMCP solvers available in the POMDPTools.jl package for Julia. For POMCP in particular, we employed 100 tree queries to keep simulation time down and an exploration constant, c, of 100, in order to attempt to encourage the exploration of the state space. Within POMCP, for evaluation of values at leaf nodes throughout the operation of the solver, we used the value iteration solver from the DiscreetValueIteration.jl package as a simple rollout solution.

Additionally, we developed a heuristic policy that imitates real world operation of a transceiver in a rescue scenario: the searcher exclusively follows flux lines unless bypassing obstacles. We sought to use this set of solvers as well as our heuristic policy for benchmarking of the performance of varied common approaches to solving a problem such as the one we had designed, in order to see whether the space lent itself more readily to an online or offline solution approach, as well as to observe promising features and drawbacks of each solver when applied to a simplified avalanche search scenario.

V. Results

Upon simulating our environment with searchers controlled by QMDP, SARSOP, POMCP, and a simple heuristic policy one hundred times each, we found the average accumulated discounted reward of every searcher (defined as ”score”) as well as their scores' standard deviations and the solvers' processing time (time it took for the solver to calculate an optimal policy). Their performance can be seen in the table below:

TABLE I
PERFORMANCE COMPARISON OF SOLVERS

Solver	Score	Standard Deviation	Runtime
QMDP	7.85	1.58	68.141 s
SARSOP	13.84	1.28	2.154 s
POMCP	1.28	1.41	326.654 ms
Heuristic	-3.69	1.30	N/A

Notably, SARSOP performed the best by yielding the maximum score and minimum standard deviation. Interestingly, both offline solvers, SARSOP and QMDP, outperformed the online solver, POMCP, which shows that our problem space appears to favor offline solutions over online ones. While our specific problem dynamics may better suit the offline solvers, an online method, however, would be more pragmatic in an actual scenario as the searcher would need to interact with the environment in order to learn about it, unless a detailed map were to be provided ahead of time. This detailed map would present additional problems, though, as real world environments tend to be highly complex and variable, and the curse of dimensionality would cause runtimes for offline solvers to oftentimes be excessive, since they were significantly worse than that of online solvers, even in our fairly restrained state space. The most extreme example of offline solver's significant runtime disadvantage was that of QMDP, with an observed runtime over 30 times worse than SARSOP. This highlights the impractical cap on realistic space complexity that can be applied to certain solvers.

Evidently, the ”always follow flux lines” heuristic policy yielded the lowest score. This result is expected and can be clearly intuited when considering the following simulated searcher paths produced by the SARSOP and heuristic policies:

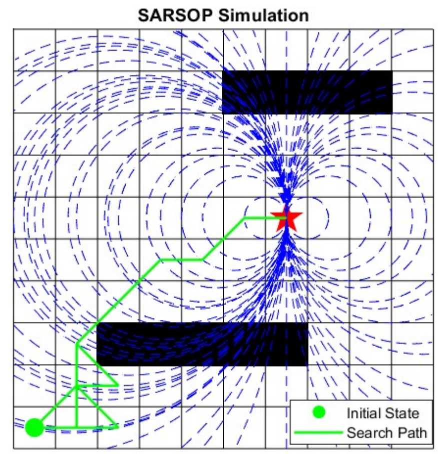

Fig. 4. SARSOP Policy Rescue Simulation

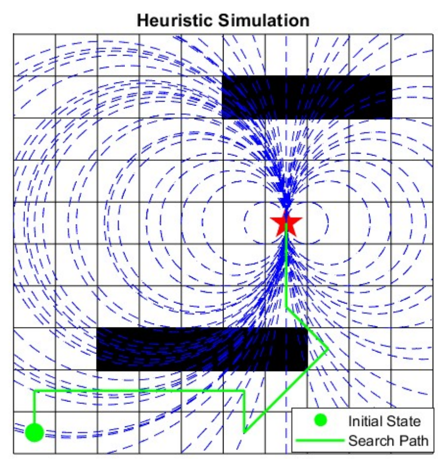

Fig. 5. Heuristic Policy Rescue Simulation

While the search path in Fig. 5 takes a roundabout journey to the victim by following flux lines, the searcher in Fig. 4 exploits its belief about the state of the environment in order to shortcut a path to the victim. Consequently, the results convey that the optimal policy in a rescue mission is combine transceiver observations with a developed sense of where the victim is in order to create a more direct path to the victim.

VI. Conclusion

Throughout this experiment, we observed both the strengths and drawbacks of using POMDPs as a problem formulation to approximate a highly complex real world scenario like an avalanche rescue problem. A discretized POMDP space such as the one implemented in this paper, as expected, proves to have some difficulty displaying the range of possibility that would be contained within a real search and rescue scenario. However, it encapsulates many of the core mechanisms needed to serve as a good benchmark for real world implementation of algorithms. Notably, it has proven that the algorithms benchmarked have the capacity to be used in conjunction with human judgement to optimize search and rescue behaviors in the future, as they were capable of finding optimal paths to the victim state that did more than just follow the flux lines output by the avalanche transceiver, as a human in such a situation would do alone. This presents an interesting possibility of POMDP algorithmically enhanced sensor solutions that could help first responders be more efficient.

Additionally, our results demonstrated that our POMDP solution was best suited for offline solvers, which would necessitate moderately comprehensive knowledge of the environment and significant pre-computational power when considering larger, more realistic state spaces. It did also demonstrate, however, that even in a simplified 10 by 10 representation of a state space, less robust algorithms such as QMDP, even while finding moderately optimal solutions, struggled to run fast enough to be able to break the "curse of dimensionality" that plagues POMDP problems. More viable options for future applications can be seen with SARSOP and POMCP, whose optimal solutions and extremely short comparative runtime respectively solidified them as promising options in use in possible future applications in similar simultaneous localization and mapping problems and search and rescue scenarios.

CONTRIBUTIONS

Yarden derived the arc length and direction for the flux line measurements. He also coded the POMDP definition and assisted in running simulations.

Andrew helped with the testing and debugging of the environment throughout the design process, and he helped in benchmarking the performance of the solvers used.

RELEASE

The authors grant permission for this report to be posted publicly.

REFERENCES

[1] H. Kurniawati, "Partially Observable Markov Decision Processes (POMDPs) and Robotics," arXiv:2107.07599v1 [cs.RO], Jul. 2021.

[2] D. Ognibene, L. Mirante, and L. Marchegiani, "Proactive intention recognition for joint human-robot search and rescue missions through Monte Carlo planning in POMDP environments," in Social Robotics: 11th International Conference, ICSR 2019, Madrid, Spain, November 26–29, 2019, Proceedings, vol. 11. Springer International Publishing, 2019.

[3] M. Lauri, D. Hsu, and J. Pajarinen, "Partially observable Markov decision processes in robotics: A survey," IEEE Transactions on Robotics, vol. 39, no. 1, pp. 21-40, 2022.

[4] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in Advances in Neural Information Processing Systems 23 (NIPS 2010).

[5] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces," in Robotics: Science and Systems IV, MIT Press, 2009, pp. 65-72.

[6] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A Framework for Sequential Decision Making under Uncertainty," Journal of Machine Learning Research, vol. 18, no. 26, pp. 1-5, 2017.

[1] is a general overview of the benefits and drawbacks of formulating problems as POMDPs in robotics decision making scenarios, and their current status in terms of real world practicality and applicability. It provides overviews of common algorithms used in POMDP solutions, including POMCP and SARSOP. It also looks at applications to real-life robots, all of which were motivators of our project.

[2] explores the interesting concept using POMDP style formulations and Monte-Carlo online solvers for human-robot interactions in search and rescue applications, which was within the realm of search and rescue, where we were focusing our simulation.

[3] is another general survey of the strengths, weaknesses, and current as well as future applications of POMDPs as a structure for decision making in autonomous agents. It has interesting explorations of use cases in SLAM, ie navigation and localization, for all common algorithms, which related to our implementation case.

[4] is the original paper introducing POMCP and benchmarking its performance as a POMDP solver in gridworlds like rocksample, which was applicable to our project, since we also were implementing POMCP.

[5] is the original paper introducing and benchmarking SARSOP in tasks like underwater navigation, which we found quite useful due to our implementation of SARSOP in our solution.

[6] is the paper associated with the POMDPs.jl github repository, which was invaluable in both our formulation and our solution to this avalanche search and rescue simulation problem.

Our code is located here: DMU Final Project Code