

POMDP Planning for High Level UAV Decisions: Search vs. Strike

Doug Schesvold, Jingpeng Tang, Benzir Md Ahmed, Karl Altenburg,
Kendall E. Nygard

Department of Computer Science and Operations Research
North Dakota State University
Fargo, ND 58105, USA
Kendall.Nygard@ndsu.nodak.edu

Abstract

The Partially Observable Markov Decision Process (POMDP) model is explored for high level decision making for Unmanned Air Vehicles (UAVs). The type of UAV modeled is a flying munition with a limited fuel supply. The UAV is destroyed when it strikes a target. When a UAV detects a target, a decision has to be made whether to continue to search for a better target or strike the current target immediately. Many factors influence this decision, including target value, target density, threat levels, and fuel level. POMDP is a suitable model for this battle field situation because of uncertainties due to the stochastic nature of the problem and the imperfect sensors of the UAV. Two POMDP models are presented in this paper. One uses planning horizon to model the fuel level, while the other models the fuel level explicitly in the states.

1 Introduction

The POMDP models proposed in this paper addresses the Search vs. Strike scenario. The scenario consists of a swarm of unmanned air vehicles (UAVs) that search a given area for specified targets to destroy [1]. Given a rectangular search area, the swarm of UAVs flies in a parallel formation in such a way as to ensure complete coverage of the search area. The UAVs use radar-type sensors to scan for potential targets. The UAVs make as many passes across the search area as needed until the entire area is searched. Each UAV maintains its own private list of detected targets. Once a target is detected, a Search vs. Strike decision must be made. In the real battle field, the sensor information is noisy and imperfect. POMDP explicitly models the imperfect information. Fuel level is an important factor that influences the behavior of the UAVs. With the limited fuel, the UAV must accomplish its mission. A high fuel level makes it more attractive for the UAV to search for a better target, while low fuel level makes it more attractive to strike the best target found. There are two POMDP models presented in this paper. One uses planning horizon to model the fuel level, while the other models the fuel level explicitly in the states.

2 Related Work

POMDP modeling has been used in various areas, such as machine maintenance, medical diagnosis, autonomous robots, and many others. POMDP is used for autonomous office navigation [3] enabling a robot to utilize all its sensor information, for hallway robot navigation problem [4] using function-approximation methods for representing the value functions. POMDP also proves its robustness in modeling problems like searching for a moving target [5]. The Linear Programming and POMDP marriage technique [6] is used for solving large-scale allocation problems with partially observable states and constrained action and observation resources. The POMDP framework is also used in medical applications, such as the management of ischemic heart disease [8].

3 POMDP Model

POMDP extends the Markov Decision Process (MDP) model [1] by allowing partial observability of the system state. The POMDP model is defined by a 6-tuple (S, A, Z, T, R, O) . For any system, it defines a finite set of states of the system, S ; a finite set of actions, A ; transition function, $T(s\zeta s, a) = P(s\zeta s, a)$, probability of transitioning from one state s to another state s' after taking an action a in state s ; rewards, $R(s, a)$, for taking an action a in state s ; a finite set of observations, Z ; observation function, $O(s, a, z) = P(z/s, a)$, probability of getting observation z given the resulting state s and action a taken in previous state. It may also consist of the horizon length h , the discount factor γ , and the initial belief state. Belief state is the probability distribution over the states, which is kept to access the current system state. In a POMDP model, the belief state has to be updated from the previous belief state. Hence, the process of maintaining the belief state is Markovian. This means that the POMDP model can be seen as a continuous space MDP as the belief space is continuous. The recursive value iteration algorithm for the MDP model can be used to solve the continuous space MDP to find the optimal policy after a little adaptation. More detailed description of the POMDP model can be found in [6].

4 POMDP Modeling of UAV for Search vs. Strike Decision

The first step in constructing a POMDP model of UAV activity is to divide the search area into a grid. A UAV searches one grid element at each time step. The goal is to destroy the highest valued target. Once at least one target is discovered, a decision must be made at each successive grid element of whether to continue to search or to strike the highest value target found thus far. Assume also that the targets degrade in value according to the time elapsed since their discovery. Furthermore, threats may be encountered which may destroy the UAV. The tradeoffs of the search vs. strike decision are illustrated in the following example. Suppose a medium value target is found early on in the mission. The UAV may decide to forgo the certain reward of striking immediately, in hopes of finding a higher value target. The risk associated with making this decision is that the UAV may subsequently only find targets of low value. Also, in the time it took to search for better targets, the medium value target has degraded to a low value target. Thus, the *continue to search* decision cost the UAV the difference between a medium and low value target. However, if the UAV had subsequently found and then struck a higher value target, the decision would have paid off. Another risk associated with the *continue to search* decision is that the UAV may be destroyed by the enemy before it can destroy a target.

4.1 POMDP Formulation

The set of states is:

$$S = \{V_i T_j, D \mid 0 \leq i \leq m, 0 \leq j \leq n\}$$

Where,

V_i : current highest value target

T_j : current highest threat level

D : absorbing state

m, n : number of distinct target and threat values respectively

The highest value may reflect either a newly discovered target, or one of time degraded value. We are assuming that the target value degrades by one for each time step. The state D is an absorbing state corresponding to striking a target, or being shot down by enemy threat. The set of actions is:

$$A = \{Strike, Search\}$$

The *Strike* action means striking the highest value target found thus far. The *Search* action means searching the next grid element.

The set of observations is:

$$O = \{V_i T_j, D \mid 0 \leq i \leq m, 0 \leq j \leq n\}$$

Where,

V_i : current highest value target

T_j : current highest threat level

The observations correspond directly to the states.

4.1.1 Probability Definitions for the Model

Assume each type (value) of target is equally likely to be found in each grid square. Also assume the value of a discovered target decreases by 1 for each time step. Threats of a given level are also equally likely to be encountered in each grid square. In this model, we assume that the threat level doesn't decay.

Define $pV_i T_j$ ($0 \leq i \leq m, 0 \leq j \leq n$) to be the probability of target of value i being highest and threat j being highest in a particular grid square. These are mutually exclusive and they sum to 1. pK_j is the probability of UAV being destroyed by threat j for each time step.

4.1.2 State Transitions

Equations (1) to (5) describe the transition probabilities of the model.

$$P(D \mid V_i T_j, Strike) = 1 \quad (1)$$

$$P(D \mid V_i T_0, Search) = 0 \quad (2)$$

$$P(V_k T_l \mid V_i T_0, Search) = \sum_{\text{valid } m, n} pV_m T_n \quad (3)$$

$$P(D \mid V_i T_j, Search) = pK_j \quad (4)$$

$$P(V_k T_l \mid V_i T_j, Search) = (1 - pK_j) \sum_{\text{valid } m, n} pV_m T_n \quad (5)$$

For example, assuming $m = 4$, and $n = 1$, the probability of transitioning from state $V_3 T_1$ to state $V_2 T_1$ is given by $(pV_0 T_0 + pV_1 T_0 + pV_2 T_0 + pV_0 T_1 + pV_1 T_1 + pV_2 T_1) * (1 - pK_1)$. The term $pV_3 T_1$ is not valid because the new state would have the highest value of 3 instead of 2.

4.1.3 Observation Probabilities

If sensors give perfect observations, Equation (6) and (7) describe the observation probabilities.

$$P(D \mid D, Strike) = 1 \quad (6)$$

$$P(V_i T_j \mid V_k T_l, Search) = \begin{cases} 1, & \text{if } i = k, j = l \\ 0, & \text{otherwise} \end{cases} \quad (7-1)$$

For imperfect observation,

$$P(V_i T_j \mid V_k T_l, Search) = p, 0 \leq p \leq 1 \quad (7-2)$$

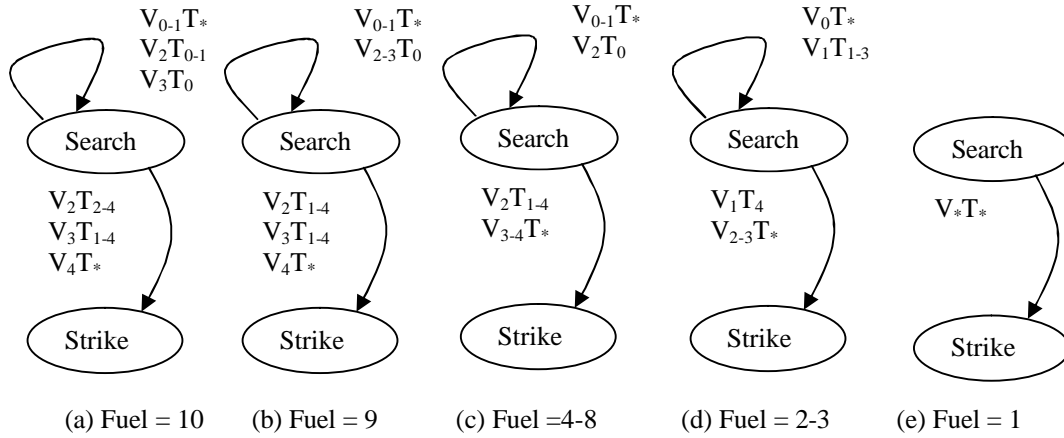


Figure 1. Optimal policy

4.1.4 Reward Function

The reward function is such that no immediate reward is given for the *Search* action. For the *Strike* action, a reward is given that corresponds to the target value of the current state. Equation (8) and (9) describe the reward function.

$$R(V_i T_j, Search) = 0 \quad (8)$$

$$R(V_i T_j, Strike) = i \quad (9)$$

4.1.5 Results

We used POMDP solver written by Anthony Cassandra [????] to solve our POMDP formulations. The exact solver produces output files corresponding to the value function and policy graph of the solution. After experimentation with several data sets modeling imperfect sensors, it was discovered that the exact solver could not produce solutions in a reasonable amount of time. Therefore, we present results for the model that assumes perfect sensors. This reduces the POMDP problem to an MDP. Since modeling imperfect sensors is of high interest, future work will involve the use of heuristic POMDP solvers.

The use of the model is demonstrated with an example. The targets and threats are modeled with the following probabilities for a grid square:

Probability of target values (0-4) being highest: 0.37, 0.19, 0.17, 0.15, 0.12

Probability of threat levels (0-4) being highest: 0.95, 0.02, 0.02, 0.005, 0.005

The $V_i T_j$ probabilities are easily calculated from the above probabilities assuming independence of targets and threats.

Probability of being shot down per time step from threat level (0-4): 0.0, 0.16, 0.2, 0.25, 0.33

Figure 4.1 shows the optimal policy for the given input values. Subscript notation can include ranges and wildcards. The subscript values indicate the observation of target value (V) and threat level (T).

5 POMDP Model with Fuel State Information

Based on the previous model, fuel level is introduced in this model. The system state is described by the combination of the target value, threat level, and fuel level. The state describes two aspects of the environment: external (target value, threat level) and internal (fuel level).

The POMDP model is described as follow:

(1) action : strike, search

(2) states :

$$S = \{V_i T_j F_k, D \mid 0=i=m, 0=j=n, 1=k=q\}$$

Where,

V_i : current highest value target

T_j : current highest threat level

F_k : current fuel level

D : absorbing state

m, n, q : number of distinct target, threat and fuel values respectively

(3) observation: $V_i T_j F_k$

(4) transition function: the only legal transition for fuel level is transfer to one level lower state or stay in the same fuel level state.

(5) observation function:

$$P(V_i T_j F_k | V_l T_m F_n, Search) = p, 0 \leq p \leq 1$$

(10)

The policy graph shown in Figure 5.1 describes the action taken by the UAV based on different observations. For this example, there are $2 \times 2 \times 2 = 8$ possible observations. From the figure we see that the UAV continues searching if the fuel level is high or the target value is low. The UAV will strike if the target value is high and the fuel level is low. Assignment of reward values and target values are arbitrary in our work. In practice, the assignment would reflect the professional judgment of some military officer. The improvement of this model is that fuel level is explicitly described, which is an important feature that reflects the short life of UAV.

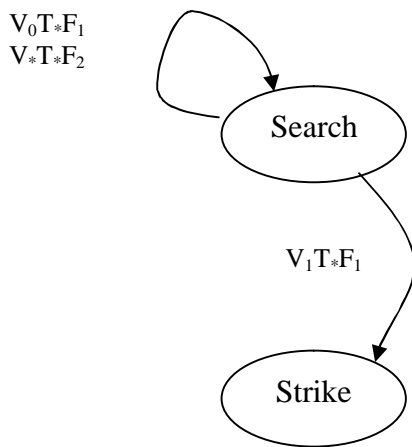


Figure 2. Optimal policy for explicit fuel level

6 Conclusions and Future Work

The POMDP model is explored for high level decision making for Unmanned Air Vehicles (UAVs). Reasonable policies for strike vs search are obtained by the POMDP model. In general, the POMDP model is not scalable. Solutions for all but small problems are intractable. Only problems with very few states and actions can be solved exactly. However, a good heuristic is needed to solve POMDP model for problems of moderate size in reasonable time. Heuristic methods for POMDP is an area of active research

For the scalability problem, to get a policy online fast, building a policy library off-line may be a feasible solution. The UAV can easily and quickly find the suitable policy searching existing policies based on the current situation. Case based reasoning techniques, such as RABIT (Retrieval with Attribute Based Indexing Technique) [9-10], can be used in tracing the policy. The problem with the offline solution is, policy library may not completely reflect all real battle field situations. A better solution may be to use newly acquired information

to improve the best possible offline policy heuristically. A good general purpose heuristic solution is yet to be discovered. One main limitation in using POMDP model is that accurate information required by the POMDP model may not be readily available.

References

- [1] Joseph Schlecht, Karl Altenburg, Benzir Md Ahmed, Kendall E. Nygard, Decentralized Search by Unmanned Air Vehicles using Local Communication
 - [2] Howard, R. A. 1960. *Dynamic Programming and Markov Processes*. MIT Press.
 - [3] Simmons, R., and Koenig, S. 1995. *Probabilistic navigation in partially observable environments*. Fourteenth International Joint Conference on Artificial Intelligence, 1080--1087. Montreal, Canada: Morgan Kaufmann.
 - [4] Cassandra, A.; Kaelbling, L.; and Kurien, J. 1996. *Acting under uncertainty: Discrete bayesian models for mobile-robot navigation*. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems.
 - [5] Eagle, J.N., *The Optimal Search for a Moving Target When the Search Path is Constrained*, Operations Research, Vol. 32, No. 5, September-October 1984, pp. 1107-1115.
 - [6] Yost, K. and Washburn, A.R., *The LP/POMDP Marriage: Optimization with Imperfect Information*, Naval Research Logistics, Vol. 47, No. 8, 2000, pp. 607-619.
 - [7] Cassandra, A. R. 1998. Exact and Approximate Algorithms for Partially Observable Markov Decision Processes. Ph.D. Dissertation, Department of Computer Science, Brown University.
 - [8] M. Hauskrecht. Dynamic decision making in stochastic partially observable medical domains: ischemic heart disease example, Proceedings of AI in Medicine Europe (AIME), Grenoble, France, pp. 296-299, 1997.
- [<http://www.cs.brown.edu/research/ai/pomdp/>]
- [9] Wolfgang Grather 1994. Computing Distances between attribute-value reorientations in an associative memory. Similarity Concepts and Retrieval Methods, volume 13 of Fabel-Report, pages 12-25. GMD, Sankt Augustin, 1994.
 - [10] Bernd Linowski. RABIT: Ein objektorientiertes System zum Retrieval attributbasierter fälle. Febel-Report 34, GND, Sankt Augustin, June 1995, 162 pages.