A Flexible Approach to Modeling Unpredictable Events in MDPs

Stefan J. Witwicki
INESC-ID and Instituto
witwicki@inesc-id.pt

Francisco S. Melo
Superior Técnico, UTL, Portugal
fmelo@inesc-id.pt

Jesús Capitán
U. of Duisburg-Essen, Germany
jesus.capitan@uni-due.de

Matthijs T.J. Spaan
T.U. Delft, The Netherlands
m.t.j.spaan@tudelft.nl

Abstract

In planning with a Markov decision process (MDP) framework, there is the implicit assumption that the world is predictable. Practitioners must simply take it on good faith the MDP they have constructed is comprehensive and accurate enough to model the exact probabilities with which all events may occur under all circumstances. Here, we challenge the conventional assumption of complete predictability, arguing that some events are inherently *unpredictable*. Towards more effectively modeling problems with unpredictable events, we develop a hybrid framework that explicitly distinguishes decision factors whose probabilities are not assigned precisely while still representing known probability components using conventional principled MDP transitions. Our approach is also flexible, resulting in a factored model of variable abstraction whose usage for planning results in different levels of approximation. We illustrate the application of our framework to an intelligent surveillance planning domain.

1 Introduction

Modeling an intelligent agent acting in an uncertain environment is challenging. For this purpose, researchers have developed elegant mathematical frameworks, such as the Markov Decision Process (MDP), that encode all states of the environment, actions, and transitions, as a dynamical system (Puterman 1994). However, in order to apply these frameworks to agents that interact with the real world, there are inherent obstacles that the practitioner must overcome.

First, it is intractable to model the real world comprehensively or with any extensive level of detail. Instead, the practitioner should choose an appropriate depth of abstraction. Second, the practitioner must select which features to include in the environment state. Not only should these features capture the critical events on which agents should base smart decisions, but they should also comprise a system whose dynamics are self-contained. In particular, the probability of a next state must be an ascertainable function of the previous values of the selected features (and only their latest values, in the case of a Markov model). This means that all modeled events must be strictly predictable (from modeled features) and their probabilities accurately prescribed.

In this paper, we propose an alternative framework that relaxes the conventional assumption of complete predictability. We take the position that, from an agent's perspective, an event may be inherently unpredictable. This could be because the event's underlying causes are prohibitively complex to model as part of the agent's state, or because circumstances surrounding the event reside in a portion of the environment that the agent cannot sense. Yet another reason to label an event as *unpredictable* could be that it is so rare as to preclude an accurate estimate of transition probabilities (neither through collected data nor through expert knowledge).

We contend that the agent should treat the occurrences of unpredictable events with corresponding features that it explicitly distinguishes from the conventional, predictable features using a factored model. We show that, independently of the complexity required to accurately model the dynamics of unpredictable features, equivalently accurate predictions are obtained by a model that depends only on the history of observable features. In constructing such a model, the agent avoids assigning arbitrary probabilities to the occurrences of unpredictable events. Yet it retains the ability to plan for all possible future paths, while accounting for known probability components associated with predictable feature values.

Our framework has several other advantages over conventional modeling options. First, it is simpler for a practitioner to specify the model, since some of the hard-to-estimate probabilities can be avoided. Second, it avoids the computational complexity of modeling additional features that only enable weak prediction of rare events. Third, our modeling approach naturally circumvents errors associated with probabilities assigned to unpredictable events. Our approach is also flexible in the model that it produces. At one end of the dial, the practitioner can specify a dependence on complete histories of observable features, yielding optimality guarantees but at a computational cost. We also contribute a principled approach on how such dependence can be alleviated by varying the order of the history dependence. We expect such approximation, in practical situations, to strike an effective balance in computational performance and the quality of approximation.

2 Motivating Scenario

As a motivating example, we will use a scenario with surveillance activities. A robot moves within the simplified surveillance environment in Fig. 1, corresponding to a floor of the

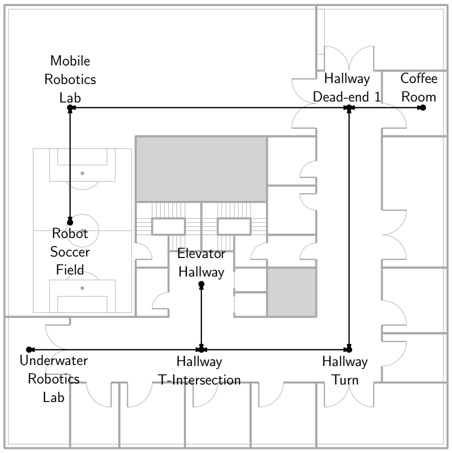


Figure 1: Outline and topological representation of the ISR surveillance environment.

Institute for Systems and Robotics (ISR) in Lisbon. For robot navigation, the position of the robot is defined as its location in a topological map with 8 possible positions (nodes in Fig. 1). In addition to the robot, there is a network of video cameras that is able to detect events such as a *fire* in the Coffee Room and *visitors* at the Elevator Hallway who require assistance.

Thanks to its local sensors and a path planner, the robot can move from location to location by selecting high-level navigation actions $\{N, S, E, W\}$ corresponding to the four cardinal directions. Nevertheless, the underlying machinery is not perfect, sometimes resulting in failed navigation actions, which we can reliably predict using a Markovian probabilistic model. For example, taking the action N at the Hallway T-intersection moves the robot successfully to the Elevator Hallway with a particular probability. The robot is in charge of completing several tasks, namely:

Surveillance of the environment. The robot should maintain under close surveillance the Underwater Robotics Lab and the Robot Soccer Field, where valuable items are stored, and the Coffee Room and the Elevator Hallway, to complement the surveillance network in the task of detecting fire and people arriving.

Fire assistance. If a fire is detected, the robot should head to the Coffee Room to assist in putting out the fire.

Assistance to visitors. If a person arrives at the Elevator Hallway and requires assistance, the robot should head to that location to assist the person.

Associated with each of these tasks is a relative priority value; the objective of the robot is to plan its movement so as to balance its expected completion of tasks given these priorities.

3 Background

In this section, we review the conventional factored MDP framework (Boutilier, Dean, and Hanks 1999) for planning activities. This formalism sets the stage for our approach, but it presents some challenges when modeling events in problems such as our example.

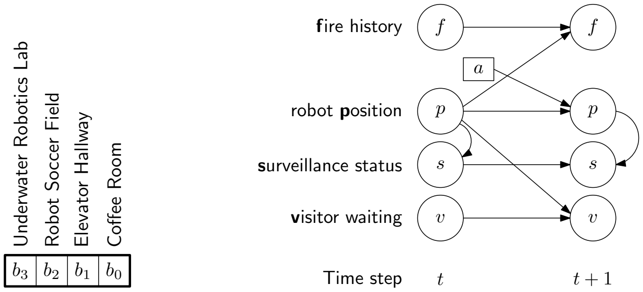


Figure 2: (Left) A binary representation of state-feature X_s, where each bit indicates whether or not the corresponding location has been visited recently. (Right) A DBN representation of the dependencies in the state-transitions of the MDP.

3.1 Conventional (Factored) MDP Model

A Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ can be used to model tasks related to robot planning in a factored manner (Boutilier, Dean, and Hanks 1999). The components are the following: (i) the *state space* is $\mathcal{X}$ (in factored models, this is the Cartesian product of several feature spaces $\mathcal{X}_p \times \mathcal{X}_s \times \cdots \times \mathcal{X}_f$); (ii) the *action space* is $\mathcal{A}$; (iii) the *transition function* (it can be factored) is $\mathsf{P} : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \mapsto \mathbb{R}$, and it encodes the probabilities of next state $X(t+1)$ as a Markov function of current state $X(t)$ and action $A(t)$; (iv) the *reward function* (it could be factored too) is $r : \mathcal{X} \times \mathcal{A} \mapsto \mathbb{R}$, and it encodes the reward obtained given the current state $X(t)$ and action $A(t)$; (v) the *discount factor* is γ, which weighs future rewards when computing the total expected reward accumulation.

To model the example problem introduced in Section 2, we can represent the state space as a set $\mathcal{X} = \mathcal{X}_p \times \mathcal{X}_s \times \mathcal{X}_f \times \mathcal{X}_w$. The state factor, or feature, $X_p(t) \in \mathcal{X}_p$ encodes the position of the robot at time t (which is any of of the labeled graph nodes in Fig. 1). The remaining features encode the statuses of the robot's completion of its various tasks.

We associate with the *surveillance task* feature $X_s(t)$ that indicates which of the target locations have been visited recently (at time t). This feature takes values in $\mathcal{X}_s = \{0, \ldots, 15\}$, which is the decimal representation of a 4-bit sequence corresponding to 4 flags indicating the target locations that have been visited (see Fig. 2, left). When one of the target locations is visited, the corresponding flag is set to 1. Whenever $X_s(t) = 15$, then $X_s(t+1) = 0$, indicating that the robot should repeat its surveillance of all target sites.

We associate with the *fire assistance task* binary feature $X_f(t)$ that indicates whether a fire is identified as active at time t. It is set to 1 when a fire is detected in the Coffee Room. After the robot visits that room, this feature is reset to 0, indicating that the robot has successfully put out the fire.

We associate with the *visitor assistance task* a binary feature $X_v(t)$ that indicates whether there is a person needing assistance at time t. It is set to 1 whenever a person needing assistance is detected in the Elevator Hallway. After the robot visits that place, the flag is reset to 0, indicating that the robot has successfully assisted the person.

Thanks to the factored structure of the model, the transition probabilities may be encoded potentially more compactly using a 2-stage Dynamic Bayesian Network like the one in Fig. 2 (right). In this case, the set of possible actions is $\mathcal{A} = \{N, S, E, W\}$. The factored structure also allows for a compact representation of the transition probabilities. In fact, the transition probabilities associated with a state-factor X_s, for example, can be represented as a conditional probability table (CPT) that represents the probability distribution $\mathbb{P}[X_s(t+1) \mid X_s(t), A(t), X_p(t+1)]$, with one entry for each combination of action and values of $X_s(t)$, $X_p(t+1)$ and $X_s(t+1)$. The transition probabilities associated with each action $a \in \mathcal{A}$ can then easily be obtained from the product of such CPTs.

Similarly, we can also specify a factored reward function that separately represents the priorities of the individual tasks.

$$r(x) = w_s r_s(x) + w_f r_f(x) + w_v r_v(x). \tag{1}$$

Each component r_i encodes the goals of a task, and (1) indicates that the robot should complete all tasks. The weights w_i indicate the relative importance/priority of the different tasks. For concreteness, we define the reward components as

$$\begin{aligned} r_s(x) &= \mathbf{1}_{\{x_s=15\}}(x), \\ r_f(x) &= -\mathbf{1}_{\{x_f=1\}}(x), \\ r_v(x) &= -\mathbf{1}_{\{x_v=1\}}(x), \end{aligned}$$

where the operator $\mathbf{1}$ works as follows: component r_s rewards those states in which the robot recently visited all critical locations (corresponding to $x_s = 15$); component r_f penalizes those states in which a fire is active ($x_f = 1$); and component r_v penalizes those states in which there is a visitor in need of assistance ($x_v = 1$).

3.2 Planning in the Conventional MDP Model

Planning in a conventional MDP model consists of determining a policy (an action selection rule) that maximizes the total reward accumulated by the agent throughout its lifetime. Formally, this amounts to determining the policy $\pi : \mathcal{X} \mapsto \mathcal{A}$ that maximizes the corresponding *value* for every state $x \in \mathcal{X}$,

$$V^\pi(x) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(X(t), A(t)) \mid X(0) = x \right],$$

where the expectation is taken with respect to trajectories $\{X(t), t = 0, \ldots\}$ induced by the actions $A(t)$, which in turn are selected according to the rule $A(t) = \pi(X(t))$. The policy with maximal value is known as the *optimal policy*, and its corresponding value function, V^*, the *optimal value function*. The optimal value function V^* is known to verify

$$V^*(x) = \max_{a \in \mathcal{A}} \mathbb{E}_{Y \sim \mathsf{P}(x,a)} \left[r(x, a) + \gamma V^*(Y) \right],$$

and it is possible to define the *action-value function* Q^* as

$$\begin{aligned} Q^*(x, a) &= \mathbb{E}_{Y \sim \mathsf{P}(x,a)} \left[r(x, a) + \gamma V^*(Y) \right] \\ &= \mathbb{E}_{Y \sim \mathsf{P}(x,a)} \left[r(x, a) + \gamma \max_{b \in \mathcal{A}} Q^*(Y, b) \right]. \end{aligned}$$

The recursive relation above can be used to iteratively compute $Q^*(x, a)$ for all $(x, a) \in \mathcal{X} \times \mathcal{A}$, a dynamic programming method known as *value iteration*. From Q^*, the optimal policy can then be trivially computed as

$$\pi^*(x) = \operatorname*{argmax}_{a \in \mathcal{A}} Q^*(x, a).$$

3.3 Modeling Events

Using a factored model such as the one just described, an event may be modeled by simply associating a boolean state feature with the occurrence of the event (Becker, Zilberstein, and Lesser 2004; Goldman et al. 2007). In this paper, we restrict our consideration to uncontrollable events:

Definition 1. *An* ***uncontrollable event*** *i corresponds to a boolean feature $X_{Ui} \in \{0, 1\}$, such that*

- *i is said to occur when X_{Ui}'s value changes from 0 to 1;*
- *Given current state x, the probability of occurrence at time $t+1$ is independent of the action a taken at time t:*

$$\mathbb{P}[X_{Ui}(t+1) = 1 \mid X(t) = x, A(t) = a] = \mathbb{P}[X_{Ui}(t+1) = 1 \mid X(t) = x].$$

For instance, in our example problem X_v corresponds to the uncontrollable event that a *visitor* needs assistance. From statistical data, the robot may estimate the appearance of such a visitor with a probability $\mathbb{P}[X_v(t+1) = 1] = p_{\text{visitor}}$. As long as we have included features in our current-state representation that together encode sufficient information for predicting the occurrence of the event in the next state, then our Markovian transition model is perfectly suitable.

4 Modeling Unpredictable Events

For some events, however, it may not be possible to accurately prescribe occurrence probabilities. An instance is the *fire* event from our running example, which is represented using feature X_f. For illustrative purpose, in Section 3.3, we have assigned the occurrence probability $\mathbb{P}[X_f(t+1) = 1 \mid X_f(t) = 0]$ as a small constant p_{fire}. However, if the transition model is presumed to have been estimated from real experience of fires in the coffee room, there is simply not enough data to know the true transition probabilities of this feature with a high degree of certainty. Given the rarity of a fire, the assignment of p_{fire} is bound to be arbitrary. Our event model is, at best, an approximation.

Definition 2. *An* ***unpredictable event*** *is an uncontrollable event whose occurrence probability cannot be accurately estimated as a Markov function of the latest state and action.*

We now describe two solution approaches for modeling such events. The first approach augments the conventional model with additional features that effectively render the event predictable. In the second approach, we devise a model wherein unpredictable events are explicitly treated as special factors whose CPTs are not assigned precisely, and provide a formal method for recasting the problem as a bounded-parameter model.

4.1 Expanded Event Model (EEM)

To help the robot to better predict fires, an alternative would be to also model the underlying process of how the fire was caused. In general, an unpredictable event could be made predictable if the factored model were to include additional features with transitions known, and sufficient for predicting that event. In our example problem, we could model the underlying causes of a fire. Figure 3 gives the reader a flavor of what such a model might look like. Restricting consideration to fires caused by faulty wiring in the coffee machine, we have expanded our original model with additional features that serve to predict the *fire* event. Working our way up

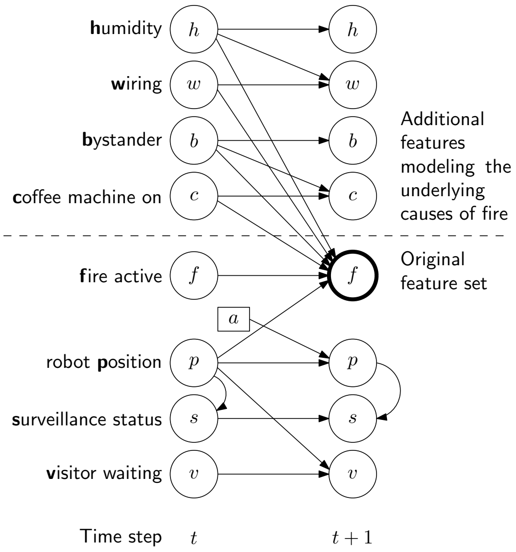


Figure 3: Expanded factored model for predicting X_f.

through these added features, first there is whether or not the coffee machine is on. The fire is much more likely to start with the machine on than off. Whether or not the machine is on is affected by the presence of a person in the coffee room. However, having a bystander in the room also drastically decreases the chances of a fire breaking out, since the bystander would intervene in the case that the coffee machine appears to be malfunctioning. The fire event is also affected by the condition of the wiring and the humidity in the room.

The challenge with this expanded event model (EEM) is that it makes the robot's decision-making problem more complex. We doubled the number of features in our state representation, thereby increasing our state space by an order of magnitude or more. Furthermore, the robot may not be able to directly observe the added features due to sensory limitations of the system. In this case the problem would become a Partially Observable Markov Decision Process (POMDP) (Kaelbling, Littman, and Cassandra 1998), introducing significant additional complexity (besides that caused by the state space expansion).

4.2 Boundedly-Predictable Event Model (BPEM)

Another challenge with the aforementioned modeling approach is that it assumes that we are able to construct a well-specified Markov model of the underlying causes of all events. These may themselves be difficult to model or predict, requiring additional layers of causes that underlie the underlying causes, thereby combinatorially exploding the augmented model. Here we develop a flexible and principled approach for leaving out any or all of these additional factors.

The idea is to treat some features as *external* to the agent. Given a prior distribution over external feature values but a lack of agent observability of these values, these variables can be marginalized out of the EEM. The result is a reduced model that highlights CPT entries (denoting event occurrence probabilities) that are not precise. Instead, these are bounded probability values. Fortunately, the bounds on these parameters may be tightened given knowledge about the dynamics of the process underlying the events.

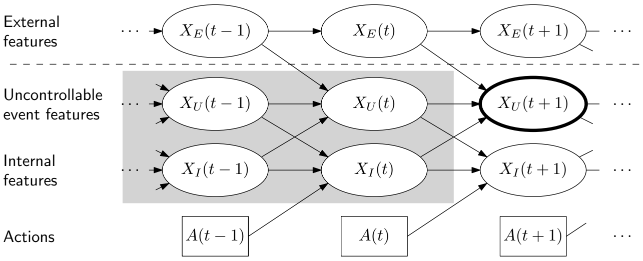


Figure 4: An illustration of the d-separation of $X_U(t{+}1)$ and $\{A(0\dots t)\}$ by the grayed region.

Theorem 1. *If we choose a set of external features X_E that serve to inform predictions of events encoded by feature set X_U, and all remaining internal features X_I, such that:*

1. *the state is factored into $X = \langle X_E, X_U, X_I\rangle$,*
2. *at each time t, the agent is assumed to directly observe $O(t) = \langle X_U(t), X_I(t)\rangle$ but not the external feature values $X_E(t)$,*
3. *the external features are conditionally independent of all else $\mathbb{P}[X_E(t+1)|X(t), A(t)] = \mathbb{P}[X_E(t+1)|X_E(t)]$, and*
4. *the internal features are conditionally independent of the external features $\mathbb{P}[X_I(t{+}1)|X(t), A(t)] = \mathbb{P}[X_I(t{+}1)|X_U(t), X_I(t), A(t)]$,*

then maintaining an alternate state representation

$$X'(t) = \langle X_I(0\dots t), X_U(0\dots t)\rangle, \tag{2}$$

is sufficient for predicting the events:

$$\mathbb{P}[X_U(t{+}1)|A(0\dots t), O(0\dots t)] = \mathbb{P}[X_U(t{+}1)|X_I(0\dots t), X_U(0\dots t)]\,. \tag{3}$$

Proof Sketch. The equality in Equation 3 is proven in two steps. First, consider that, since all internal feature values $X_I(0\dots t)$ are observed, as well as the event feature values $X_U(0\dots t)$, and all external feature values unobserved, we can rewrite the left-hand side as follows:

$$\mathbb{P}[X_U(t{+}1)|A(0\dots t), O(0\dots t)] = \mathbb{P}[X_U(t{+}1)|X_U(0\dots t), X_I(0\dots t), A(0\dots t)]\,. \tag{4}$$

All that remains to reduce Equation 4 to the right-hand side of Equation 3 is to prove that $X_U(t+1)$ is conditionally independent of $\{A(0\dots t)\}$ given evidence $\{X_I(0\dots t), X_U(0\dots t)\}$. This holds as consequence of the *d-separation* relationship (Pearl 1988) shown in Fig. 4. □

Corollary 1. *Given properties 1–4 in Theorem 1, alternate state representation $X'(t) = D(t)$, where $D(t) \subseteq \{X_U(0\dots t), X_I(0\dots t)\}$* d-separates *$X_U(t+1)$ and $\{A(0\dots t), \{X_U(0\dots t), X_I(0\dots t)\}/D(t)\}$, is sufficient for predicting the events.*

The implication of the Theorem and Corollary 1 is that we can capture all the relevant effects of unobservable external variables without explicitly modeling the external variables. Whether or not we have the know-how or capability to model external variables, a model containing only observable event-effectors is sufficient for making predictions. Figure 5 portrays a reduced model for our running example, where only the robot's position history and the fire history suffice to predict future fire status *regardless of* how complex the external fire-generating process is.

Inferring a reduced model. Given a model of event-underlying external variables, we can compact our model without losing predictive power, by simply marginalizing these variables out. Reducing our model becomes an inference problem that updates the conditional probability tables of the affected event features as the affecting external variables are removed. The equations below describe this inference problem as an iterative process that computes three kinds of terms for each decision stage. The first term, which we refer to as the *joint-external-event* distribution, or $\boldsymbol{J}(t)$ at decision stage t, effectively merges the two variables $X_E(t)$ and $X_U(t)$ into one node. The second term is the *marginal* event distribution, $\boldsymbol{M}(t)$, which is induced by marginalizing over the first term. The third term, which we call the *induced-external* distribution $\boldsymbol{IE}(t)$, is used for computing the joint-external-event distribution of the next decision stage.

For stage $t = 0$, $\boldsymbol{IE}(0) \equiv \underbrace{\mathbb{P}[X_E(0)]}$. For stages $t \geq 1$:

$$\boldsymbol{J}(t) \equiv \mathbb{P}[X_E(t), X_U(t) | X_U(0 \dots t{-}1), X_I(0 \dots t{-}1)] = \sum_{X_E(t-1)} \Big(\underbrace{\mathbb{P}[X_U(t) | X_E(t{-}1), X_U(t{-}1), X_I(t{-}1)]} \underbrace{\mathbb{P}[X_E(t) | X_E(t{-}1)]}\, \underbrace{\boldsymbol{IE}(t{-}1)} \Big) \quad (5)$$

$$\boldsymbol{M}(t) \equiv \mathbb{P}[X_U(t) | X_U(0 \dots t{-}1), X_I(0 \dots t{-}1)] = \sum_{X_E(t)} \underbrace{\boldsymbol{J}(t)} \quad (6)$$

$$\boldsymbol{IE}(t) \equiv \mathbb{P}[X_E(t) | X_U(0 \dots t), X_I(0 \dots t)] = \frac{\underbrace{\boldsymbol{J}(t)}}{\overbrace{\boldsymbol{M}(t)}} \quad (7)$$

Upon augmenting the state with the necessary history ($X'(t) = \langle X_U(0 \dots t), X_I(0 \dots t) \rangle$, in the worst case), Equations 5–7 allow us to complete our specification of the reduced model. The marginal distribution $\boldsymbol{M}(t)$ defines the new CPT of the event features $X_U(t{+}1)$ in terms of old CPT entries from the EEM. As we reduce the EEM, the CPTs for all other internal state variables remain unaffected, since they are conditionally independent of past external variables given the event features.

Note that we are replacing the external variables with histories of event features, but that we expect that this is a reasonable trade-off in the case that many external variables can be eliminated. Moreover, depending on the dynamics of the problem, histories of events can often be encoded compactly such as by encoding the past times that a fire occurred and the respective durations, rather than the whole bit sequence. Note also that in modeling problems with our framework, there is flexibility in terms of which and how many variables we choose to marginalize out. Marginalizing out fewer external variables could, for instance, decrease the history dependence (though if these variables are unobservable, we would have a POMDP instead of a history-augmented MDP).

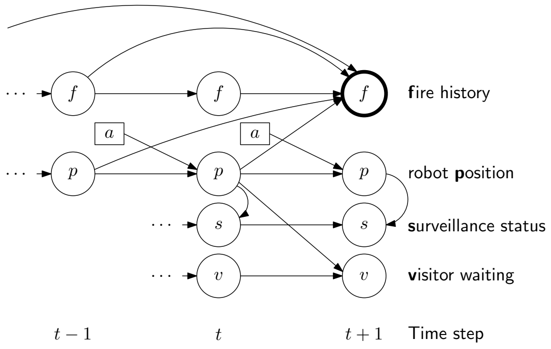


Figure 5: An equivalent reduction of the model in Figure 3.

Propagating Bounds. If the dynamics of some or all of the external variables are not known, or imprecisely specified, then the model reduction above results in a bounded probability distribution for predicting the events. Here, we assume that the CPT entries involving X_E in the EEM are each represented with a lower and upper bound $[p_{lb}, p_{ub}]$. These bounds should in turn be propagated to the reduced model as the external variables are marginalized out. We can still apply Equations 5–7, but we should do it twice, so as to compute an upper and lower bound for each entry in each distribution. In particular, to compute lower (upper) bounds, each term indicated with a faint bracket underneath should be substituted with the lower (upper) bound for that term, and each term indicated with a bracket above should be substituted with the upper (lower) bound.[1]

Some bounds may be tighter than others. For instance, although we may not know the probabilities of faultiness in the coffee-machine wiring or of how likely this is to cause a fire, we can more easily collect data about bystander and coffee machine usage and patterns. Moreover, we may be certain that if the coffee machine is not on, then no fire will break out. This knowledge may improve the bounds that are propagated through marginalization, tightening the bounds on various parameters of our reduced model.

The resulting factored model is an instance of a bounded-parameter MDP (BMDP) (Givan, Leach, and Dean 2000), and so solution methods for BMDPs can be readily applied. In contrast to the general BMDP, our model has the advantage that we have explicitly highlighted which parameters are uncertain. If we were to encode the problem without performing such factorization, and systematically propagating uncertainty, we would be left with the relatively more daunting task of assigning bounds to all parameters for all states in the transition matrix.

Approximating the event distribution. Although we have eliminated the external variables, we are left with a history dependence whose computational consequences may

[1] In the interest of space, we describe the most simplistic method for propagating bounds, but there exist more sophisticated approaches yielding tighter bounds (e.g., Bidyuk and Dechter 2006).

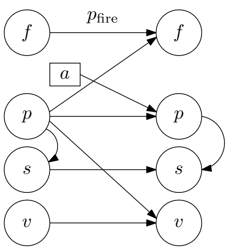


Figure 6: BPEM for the scenario where only p_{fire} is unknown, but the $fire$ dynamics are Markovian.

be undesirable. Moreover, if the horizon of the planning problem is infinite, maintaining a dependence on the entire history is untenable. Fortunately, there are principled methods for approximating such distributions finitely (Begleiter, El-Yaniv, and Yona 2004; Littman, Sutton, and Singh 2002).

One such solution is to predict the events using a k-order Markov model. This amounts to assuming that the event-generating process is k-order stationary, or to assuming that such a process is a sensible approximation of the true process. Under this assumption, we can infer our *reduced model* using exactly the same inference technique described by Equations 5–7, iterating only up to time step k. In particular, the new CPT for event features $X_U(t)$ is:

$$\boldsymbol{M(t)} \equiv \mathbb{P}[X_U(t)|X_U(t-k\ldots t-1), X_I(t-k\ldots t-1)] = \mathbb{P}[X_U(k)|X_U(0\ldots k-1), X_I(0\ldots k-1)] \quad (8)$$

for any given time step $t \geq k$.

This approach flexibly approximates the event prediction model to a desired level of granularity. An appropriate level of k may be selected depending on computational restrictions and on the presumed complexity of the underlying event process. The larger the value of k, the closer the prediction model will be to the underlying process. However, if less is known about the process, a larger k can also lead to looser bounds in the probability parameters of the model. The smaller the value of k the simpler the decision model used to plan. At the extreme, we can model the events as depending on neither history nor on state by approximating the distribution with a single probability denoting the likelihood of the event taking place at any given time step.

5 Illustrative Examples

In this section, we illustrate the application of our modeling approach to different instances of the scenario introduced in Section 2.

Fixed unknown probabilities. In the first set of experiments, we consider the situation in which fires break out with a fixed probability p_{fire} that is known to lie in the interval $[0.0, 0.7]$ but is otherwise unknown. This corresponds to the simplest event model, where the dependence of $X_f(t)$ on the history of observations is only through $X_f(t-1)$ and $X_p(t-1)$.[2] Using the BPEM depicted in Fig. 6, we can now build a standard BMDP that accommodates for the uncertainty in p_{fire}. We solve the BMDP using *interval value iteration* (Givan, Leach, and Dean 2000) and compute two policies, $\pi_{\max}$ and $\pi_{\min}$. These two policies attain, respectively, the best and worst possible performances given the uncertainty in the BMDP parameters.

For comparison, we also computed the optimal policy for an *optimistic MDP*, π_{opt}, that considers $p_{\text{fire}} = 0.0$, and a pessimistic MDP, π_{pes}, that considers $p_{\text{fire}} = 0.7$. Note that, since the only uncertainty in the BMDP model concerns the parameter p_{fire}, we expect the performances of π_{opt} and π_{pes} to match those of $\pi_{\max}$ and $\pi_{\min}$, respectively.

We tested the four policies in our navigation scenario, running each policy for a total of 200 independent Monte Carlo trials. In each trial, a (simulated) robot moved around the environment for a total of 100 time-steps while following the prescribed policy and the total discounted reward accumulated was evaluated.[3] Figure 8(a) presents the results obtained. The solid area corresponds to the empirical estimation of the value bounds obtained from the BMDP. As expected, the values obtained by the two MDP policies match, approximately, the bounds prescribed by the BMDP policy. We also compared the performance of the different policies as a function of the actual value of p_{fire}. The results are depicted in Fig. 8(b) and, as before, the MDP policies closely match the bounds attained by the BMDP solution.[4]

Next, we generalized this experiment to the case where both *fire* and *assistance* events are treated as unpredictable (with $p_{\text{assistance}}$ lying in the interval $[0.0, 1.0]$). The results are depicted in Fig. 7, where we observe again that the MDP policies closely match the bounds attained by the BMDP solution.[5]

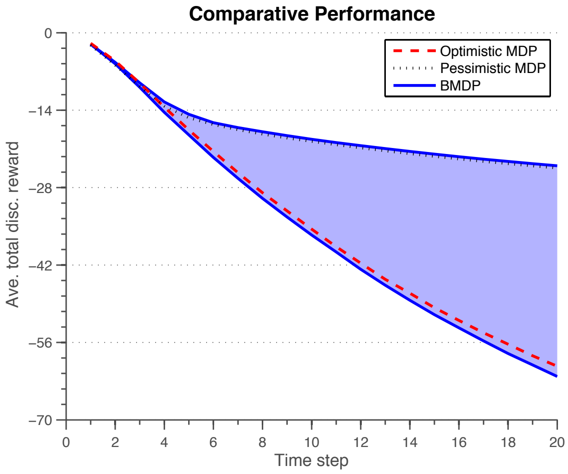


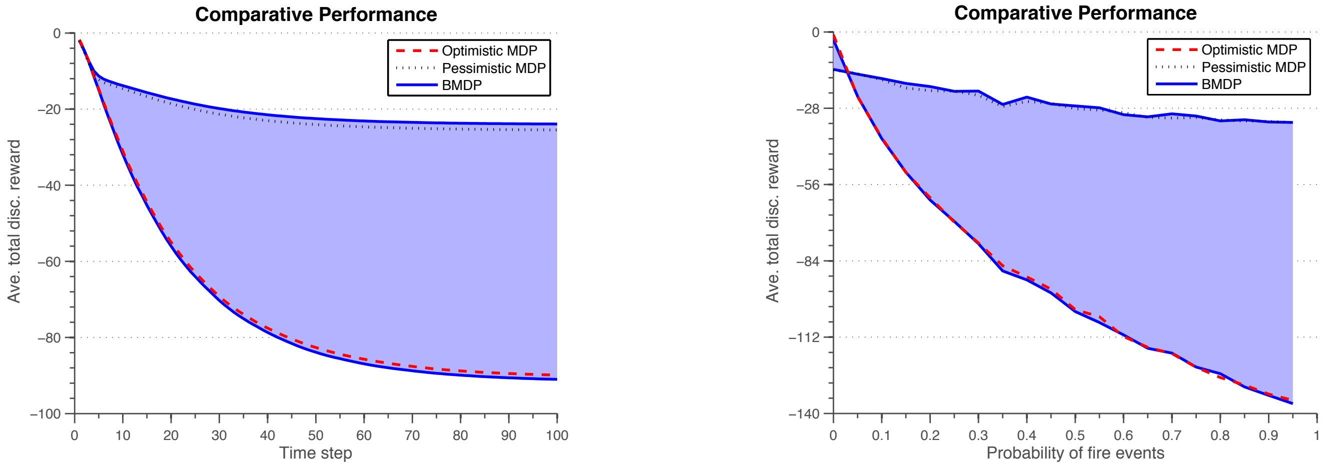
Figure 7: Comparison of BMDP, optimistic MDP and pessimistic MDP policies in a scenario where both p_{fire} and $p_{\text{assistance}}$ are unknown.

[2] Unlike the fire event, the *assistance* event was assumed to be predictable here, with a fixed probability $p_{\text{assistance}} = 0.1$.

[3] The actual value of p_{fire}, unknown to the robot, was 0.4.

[4] Note that in all of these experiments, we have intentionally chosen unrealistically-high fire probabilities to bring out the potentially-significant differences in performance among methods. In a separate experiment (whose detailed results we have omit here to save space) we also tested smaller probabilities (e.g., $p_{\text{fire}} = 0.01$), and observed the same qualitative trends.

[5] Actual values of $p_{\text{assistance}}$ and p_{fire} were both 0.4.



(a) Unknown fire probability, $0 \leq p_{\text{fire}} \leq 0.7$.

(b) Dependence of the performance on the value of p_{fire}.

Figure 8: Comparison of BMDP, optimistic MDP and pessimistic MDP policy in a scenario with constant fire probability p_{fire}.

Non-Markovian fire events. In the second set of experiments, we consider a situation in which a pre-determined number of fires break out at random instants in time during the simulation. This means that the feature $X_f(t)$ is non-Markovian given the state information available to the robot.

In these experiments, predicting the fire would require several additional features (among which the time remaining until the end of the episode), as depicted in Fig. 9. However, as we derived in Section 4.2, this EEM can be reduced to the more compact BPEM in Fig. 5, where uncertainty has been propagated into the dependence of $X_f(t)$ on $\{X_f(0 \ldots t), X_p(0, \ldots, t)\}$. Note that, in this case, the BPEM complexity is unaffected by the number of fires, whereas the EEM complexity is critically dependent, leading to a larger latent state space with every additional fire. In a problem context such as this, the BPEM is a much more sensible alternative due to its scalability.

For illustrative purposes, we adopt the simplest approximation, considering a first-order Markov approximation as described in Section 4.2. The resulting BPEM is thus equivalent to that used in the simpler context of the previous experiments. As before, we tested the four policies in our navigation scenario, running each policy for a total of 200 independent Monte Carlo trials. Figure 10 presents the results obtained.

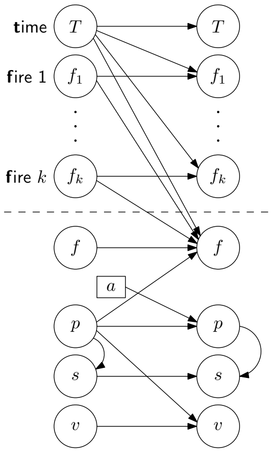


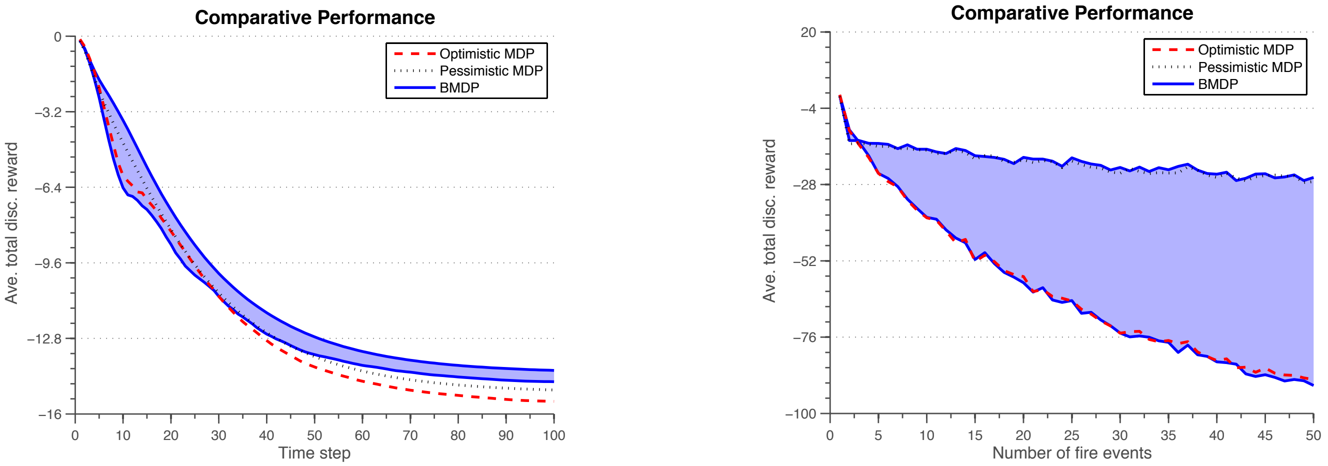
Figure 9: EEM for the second test scenario, requiring a large number of additional features.

Notice that, for a small number of *fire* events (Fig. 10(a)), the MDP policies do not trace the performance bounds of the BMDP approach. However, as the number of *fire* events increases (Fig. 10(b)), this difference in performance disappears. This may be an artifact of the approximation, indicating that the scenario wherein only a small number of fires take place is not adequately captured by a simple MDP model. As events become less rare, the approximation appears to become sufficiently accurate for planning. We note, however, that this preliminary experiment is merely an illustration of the approximate BPEM model, serving as a precursor for more rigorous evaluations of its efficacy.

6 Related Work

The conventional method of reasoning about uncertain events while planning, which we extend here, is to include the events as part of the state and model their occurrence as transition probabilities (Becker, Zilberstein, and Lesser 2004; Witwicki and Durfee 2009). Other work (Cao and Zhang 2008) models an MDP that reasons only about these kind of events instead of states, completely abstracting away from the underlying Markov model. To combine asynchronous events and actions, others have employed Generalized Semi-Markov Decision Processes (GSMDPs) (Younes and Simmons 2004; Rachelson et al. 2008). They consider time to be continuous and model a set of external events that can be triggered at any time affecting the system state. A difficulty with such models is that a time distribution is assumed for each event, which may not be easy to construct in a real problem, especially when considering rare events.

Others have acknowledged that rare events are difficult to model conventionally. There is extensive work on estimating the probabilities of rare events (Juneja and Shahabuddin 2006;



(a) Three fires break out at random times during each trial.

(b) Dependence of the performance on the number of fires.

Figure 10: Comparison of BMDP, optimistic MDP and pessimistic MDP policies in a scenario with non-Markovian fire events.

Rubino and Tuffin 2009). In general, the idea is to simulate the system and apply different sampling techniques in order to derive small probabilities for the events. These techniques are used in a wide spectrum of applications, such as biological systems, queue theory, reliability models, etc. Usually the rare events (e.g., a queue overflow or a system failure) have a very low probability but they can occur if the simulations are properly driven. A challenge is that, given the existence of many hidden variables, fine-tuning an arbitrarily complex model is not always possible.

Our approach is complementary to previous work that introduced bounded and imprecise parameter models and developed solution algorithms, such as MDPs with imprecise probabilities (Harmanec 2002), POMDPs with imprecise probabilities (Itoh and Nakamura 2007), bounded-parameters variations in MDPs (Givan, Leach, and Dean 2000) and POMDPs (Ni and Liu 2008), as well as Factored MDPs with imprecise probabilities (Delgado, Sanner, and de Barros 2011). The modeling framework that we develop describes a principled approach to actually specifying a bounded-parameter model. Thus, our work contributes a useful precursor to, and a formal context for, applying bounded-parameter models.

The idea of reducing a decision model by eliminating unobserved external variables has also been explored in multiagent settings (Oliehoek, Witwicki, and Kaelbling 2012; Witwicki and Durfee 2010), where agents model abstract *influences* from peers rather than the peers' full decision models. Here we show that by modeling environmental influences abstractly, the same principle also facilitates single-agent decision making. Along a similar vein, PSRs (Littman, Sutton, and Singh 2002) also seek to make predictions using a compact representations of histories of observable features.

7 Conclusions and Future Work

In this paper, we have addressed the challenge of modeling unpredictable events for the purposes of intelligent planning and decision-making under uncertainty. In contrast to the majority of work in this area, which assumes a perfectly predictive model of world dynamics, we have acknowledged that an accurate MDP model may be impossible to prescribe. As a precursor to planning in these scenarios, we have developed a framework for accounting for events whose underlying processes are prohibitively complex or unknown.

Our main contribution is a principled modeling approach. We have shown that if (potentially large) portions of the underlying event process can be treated as unobservable external variables, those variables do not need to be included in the decision model. In particular, we have proven that an MDP model *without* the external variables provides the same predictive power as a POMDP model *with* the external variables. And we have formulated how inference techniques can be used to reduce the model through marginalization.

For those situations where there is uncertainty in the underlying process, we have developed a method for systematically propagating the uncertainty to distinguished parameters in our reduced model. And for when knowledge is gained about the underlying process, the same method incorporates that knowledge in the form of tightened probability bounds on event prediction probabilities. The output of our method, the BPEM, is a special type of BMDP, which our principled approach helps the modeling practitioner to specify. Our approach offers the benefit of flexibility in the richness of the BPEM, accommodating different levels of knowledge about event dynamics, and supporting trade-offs in computational complexity and predictive precision.

We have illustrated how our framework can be used to model fire events in a robot planning problem. In the future, we plan to extend our experimental work to carefully analyze the purported trade-offs among event models, in particular comparing computation and quality of EEM planning with approximate BPEM planning. Additionally, the modeling groundwork that we have presented here motivates future work into exploiting our BPEM's factored structure in planning. In particular, we envision algorithms that compute plans more efficiently by leveraging the fact that unpredictability only manifests itself in the CPTs of uncontrollable events.

Acknowledgments

This work was partially supported by the Portuguese Fundação para a Ciência e Tecnologia (FCT) and by Carnegie Mellon Portugal Program and its Information and Communications Technologies Institute under project CMU-PT/SIA/0023/2009 and by the FCT (INESC-ID multiannual funding) under project PEst-OE/EEI/LA0021/2011. Matthijs Spaan was funded by the FP7 Marie Curie Actions Individual Fellowship #275217 (FP7-PEOPLE-2010-IEF).

References

Becker, R.; Zilberstein, S.; and Lesser, V. 2004. Decentralized Markov decision processes with event-driven interactions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 302–309. IEEE Computer Society.

Begleiter, R.; El-Yaniv, R.; and Yona, G. 2004. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research* 22:385–421.

Bidyuk, B., and Dechter, R. 2006. Improving bound propagation. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI)*, 342–346.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research (JAIR)* 11:1–94.

Cao, X., and Zhang, J. 2008. Event-based optimization of Markov systems. *IEEE Trans. Automatic Control* 53(4):1076–1082.

Delgado, K. V.; Sanner, S.; and de Barros, L. N. 2011. Efficient solutions to factored MDPs with imprecise transition probabilities. *Artificial Intelligence* 175(9-10):1498–1527.

Givan, R.; Leach, S.; and Dean, T. 2000. Bounded-parameter Markov decision processes. *Artificial Intelligence* 122(1-2):71–109.

Goldman, R. P.; Musliner, D. J.; Boddy, M. S.; Durfee, E. H.; and Wu, J. 2007. "Unrolling" complex task models into MDPs. In *AAAI Spring Symposium: Game Theoretic and Decision Theoretic Agents*, 23–30.

Harmanec, D. 2002. Generalizing Markov decision processes to imprecise probabilities. *Journal of Statistical Planning and Inference* 105(1):199–213.

Itoh, H., and Nakamura, K. 2007. Partially observable Markov decision processes with imprecise parameters. *Artificial Intelligence* 171(8-9):453–490.

Juneja, S., and Shahabuddin, P. 2006. Rare-event simulation techniques: An introduction and recent advances. In Henderson, S., and Nelson, B., eds., *Handbooks in operations research and management science*, volume 13. Addison Wesley. 291–350.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101:99–134.

Littman, M.; Sutton, R.; and Singh, S. 2002. Predictive representations of state. In *Adv. Neural Information Proc. Systems*, 1555–1561.

Ni, Y., and Liu, Z.-Q. 2008. Bounded-parameter partially observable Markov decision processes. In *Proc. 18th Int. Conf. Automated Planning and Scheduling*.

Oliehoek, F.; Witwicki, S.; and Kaelbling, L. 2012. Influence-based abstraction for multiagent systems. In *Proceedings of the Twenty-Sixth AAAI Conference (AAAI-2012)*.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Puterman, M. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc.

Rachelson, E.; Quesnel, G.; Garcia, F.; and Fabiani, P. 2008. Approximate policy iteration for generalized semi-Markov decision processes: an improved algorithm. In *European Workshop on Reinforcement Learning*.

Rubino, G., and Tuffin, B., eds. 2009. *Rare event simulation using Monte Carlo methods*. John Wiley & Sons, Ltd.

Witwicki, S. J., and Durfee, E. H. 2009. Flexible approximation of structured interactions in decentralized Markov Decision Processes. In *Proceedings of the 4th Worshop on Multi-agent Sequential Decision-Making in Uncertain Domains (MSDM-2009)*, 65–72.

Witwicki, S. J., and Durfee, E. H. 2010. Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS-2010)*, 185–192.

Younes, H., and Simmons, R. 2004. Solving generalized semi-Markov decision processes using continuous phase-type distributions. In *Proceedings of the National Conference on Artificial Intelligence*, 742–748.