# Uncertain Probabilistic Roadmaps with Observations

**Richard Dearden**
School of Computer Science
University of Birmingham
Birmingham, B15 2TT, UK
*rwd@cs.bham.ac.uk*

**Michael Kneebone**
School of Computer Science
University of Birmingham
Birmingham, B15 2TT, UK
*mlk@cs.bham.ac.uk*

### Abstract

Probabilistic roadmaps (PRMs) are a commonly used approach to path planning in continuous spaces with obstacles. We examine the case where the obstacle locations are not known with certainty but can be observed during execution of the plan. We abstract the problem to one of traversing a graph where some edges (referred to as uncertain edges) may or may not be present, and where noisy observations of these edges can be made from some of the vertices of the graph. We show that this problem can be represented as a POMDP, and then use the structure in the problem to derive a number of MDP approximations to the POMDP. We show that using these approximations we can solve larger PRMs efficiently while producing policies that are close to optimal for many problems, and that we can produce optimal solutions for PRMs with smaller numbers of uncertain edges.

## Introduction

Probabilistic Roadmaps (PRM) are a popular technique for path planning in high dimensional spaces. They are applicable to many situations including robot arm motion planning and path planning for mobile agents. PRMs work by generating a random graph with vertices representing reachable robot poses and edges representing motion from one pose to another. This graph is then searched for a path from the initial state to the goal. Their success is due to the fact that they reduce search in a large continuous space into search in a graph. One of the limitations of most PRM approaches is that the planner must be aware of all obstacles in the environment prior to building the roadmap graph or plotting a route. There are many instances where this is not the case. Our approach allows the agent to be initially uncertain about obstacle locations and hence about which graph edges may be blocked. The system builds a plan to reach the goal that makes observations of these *uncertain edges* to determine which are blocked and which it can use.

We will assume that most edges in the graph can be traversed with certainty, and that there are only a relatively small number of these uncertain edges that have a non-zero probability of being blocked. We can think of this as a decision-theoretic model identification problem where there are $m$ uncertain edges, hence $2^m$ possible models. The problem is to select actions that help identify which model is the true one, at least for the edges we may wish to visit on the way to the goal. The decision over which route to take is then not only based on the agent's belief about which edges are usable, but also upon the value of being able to make better observations about obstacles in the world.

In the next section we outline the basics of PRM graph construction and introduce our representation of uncertainty in PRMs. We then show how this can be represented as a partially observable Markov decision process (POMDP) where a series of observations alters the agent's belief about the true model of the world. Due to the complexity of the POMDPs, this generates problems which are infeasible to solve in reasonable time. We develop three Markov decision problem (MDP) approximations to the POMDP model and discuss methods for solving them efficiently. Finally, we evaluate the effectiveness of these approaches and discuss future research directions.

## Probabilistic Roadmaps

Probabilistic Roadmaps (PRM) planners are built around the idea of plotting paths for robotic agents by searching over a graph of possible configurations for that agent. PRM-based approaches have been shown to be scalable (Kavraki and Latombe 1998) to agents featuring high dimensionality (i.e. robots with large degrees of freedom (d.o.f.)). This is an advantage over other deterministic path planners that, while efficient in two or three dimension spaces, struggle to cope with the exponential growth that occurs as the quantity of dimensions increases. By sampling random values for each d.o.f. of the agent to create a complete configuration, a PRM planner is capable of "exploring" the entire configuration space (C-space) for a given agent. Not all configurations in the space will be usable due to obstacles or self-collisions, so the n-dimensional space is conceptually divided into two non-contiguous areas named $\mathrm{C}_{free}$ and $\mathrm{C}_{obst}$ for free and obstructed configurations respectively. A PRM graph is built in the preprocessing phase of the planner by using a sampling algorithm which randomly generates configurations (poses) for the agent according to some criteria and then either accepts or rejects the generated poses. Each accepted pose becomes a node on the graph. The simplest sampler uniformly generates random poses for the agent and

accepts poses that exist completely in $C_{free}$ and reject poses in $C_{obst}$; more complex samplers have been created that are biased towards sampling in narrow corridors for instance (Hsu et al. 2003). The graph is completed by taking each sampled node and attempting to create $k$ edges to neighbouring nodes by checking for a collision-free route between the two nodes. In the query phase of PRM, a completed graph can then be searched rapidly for paths between two arbitrary poses with standard graph searching algorithms.

## Uncertainty in PRM

The use of uncertainty in PRM has not received as much attention as other issues surrounding motion planning. Missiuro and Roy (2006) describe the use of uncertain maps to allow robots to navigate through worlds without exact knowledge of obstacle locations. They use sampling methods that incorporate the agent's confidence in a sample being located in $C_{free}$ before accepting it as a point in the roadmap. A minimum cost route planner with a bias towards low risk routes is also described. Burns and Brock (2006) also explore map uncertainty, but use(s) a lazy approach to roadmap construction which builds roadmaps as queries are evaluated. Roadmap refinement techniques are employed that increase the detail of sensing in tricky areas if a generated route falls below a preset confidence threshold. The idea of associating a "success probability" with edges is used by several implementations and is similar to the first stage of processing used in (Nielsen and Kavraki 2000). Lazy PRM (Bohlin and Kavraki 2000) is an approach used by many PRM based planners when dealing with dynamic obstacles such as the one described by (Jaillet and Simeon 2004) which utilises a lazy roadmap construction algorithm including a local reconnection strategy. The local planner reconnects points in the roadmap which become broken when obstacles change position such as a door closing or opening. Dynamic and moving obstacles can be accounted for in PRM by planning in the state $\times$ time space as in (Hsu et al. 2000). Roadmaps are constructed lazily and new points on the map are generated by altering robot control inputs and using the configuration reached a short interval of time later. Moving obstacles are planned around, but the locations and trajectories must be known a priori.

Our approach is to generate a PRM graph in advance in which some edges may intersect the uncertain obstacles. We then to build a path plan that traverses the graph optimally, given the fact that observations will be made as we get closer to the obstacles that will allow us to determine which edges are obstacle-free. This contrasts with the previous approaches because we explicitly reason about the information we may receive while executing the plan. The benefit of is demonstrated in the following section.

## Model Formulation

To formulate this uncertain PRM in a tractable way, we represent the obstacle locations only in terms of their effect on the uncertain edges, and consider the problem as one of efficiently traversing the PRM graph.

To look at solution methods for this problem, we now abstract out a lot of the details. Since we're looking at PRM for path planning, we will assume we can reason directly about the traversability of the PRM graph, so we abstract away details of the location of the obstacles and assume the problem is as follows:

Let $G = \{V, E\}$ be a graph where the vertices in the graph, $V = \{v_1, ..., v_n\}$ represent robot poses, and the edges $E = \{e_1, ...e_k\}$, where $e_i = \langle v, v' \rangle$, represent paths between poses (we assume that if there is a path $\langle v, v' \rangle$, then there is a corresponding path $\langle v', v \rangle$). We also identify locations $v_S$ and $v_G$, the start and goal poses. Each edge $e$ has a cost $c_e$ of traversing the edge in either direction associated with it.

Most existing PRM algorithms assume that the locations of all the obstacles in the space are exactly known a priori. If this isn't the case (for example when the movement is over long distances, or obstacles may be obscured behind others), we may not know immediately which edges are collision-free and which are not when creating the graph. However, as the robot moves through the space, more information may be obtained as obstacles become visible and their locations can be measured more accurately.

When we visit a vertex of the graph, we make observations of all the uncertain edges from our new location, and we receive information about whether the edges are obstacle-free[1]. We will assume that for each edge $e_i$ we observe either $f_i$, meaning that we observe the edge to be collision-free, or $b_i$ if the edge appears blocked. These observations are uncertain, in that if the edge is collision-free, we may still observe $b_i$ some of the time, and vice versa. Assume there are $m$ uncertain edges. We write $\overline{o} =< o_1, \ldots, o_m >$ for an observation of each of the uncertain edges, where $o_i$ is either $f_i$ or $b_i$.

Although we can't observe them, there is in fact a true state of the world in which each edge is either collision-free or blocked. Let $W = \{w_1, \ldots, w_{2^m}\}$ be the set of all such worlds. We write $P(\overline{o}|v_i, w_j)$ for the probability of making observation $\overline{o}$ from location $v_i$ if the true state of the world is $w_j$. While we've abstracted away the obstacle locations in this representation, the fact that the traversability of the edges is based on obstacle locations is important because it implies that for close edges, the probability that the edges are blocked may not be independent. If the edges are independent, then the likelihood that world $w$ is the true state of the world is simply the product of the likelihoods of each edge being collision-free or blocked. If this is not true (for example, where one obstacle is likely to intersect two edges), then the edge probabilities are dependent, and the probability of each world must be maintained separately in the belief state.

To find an optimal path in a PRM graph with no uncertain edges, we can represent the problem as an MDP where the state space $S = V$, the action space $A = \{\text{goto-}v : v \in V\}$ (note that not all actions in this model are applicable in every state), and the reward and transition functions are given by:

$$R(s, \text{goto-}v) = \begin{cases} -c_e & \text{if } e = \langle s, v \rangle \text{ and } v \neq v_G \\ -\infty & \text{otherwise} \end{cases}$$

[1] If not all edges are visible, we assume we receive an observation that gives no information.

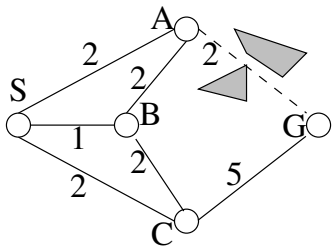


Figure 1: A small example PRM graph. In this example there are five vertices and one uncertain edge, which initially has $P(\text{blocked}) = 0.5$ (the dotted line indicates the uncertain edge). The state of the edge can be observed with certainty from vertices $A$ and $B$.

$$P(s_i, \text{goto-}s_j, s_j) = \begin{cases} 1 & \text{if } \langle s_i, s_j \rangle \in E \\ 0 & \text{otherwise} \end{cases}$$

where $v_G$ is a goal node. Intuitively, the vertices correspond to states, edges correspond to actions, all the transition probabilities are 0 except for those that correspond to edges in the graph, and rewards correspond to costs of traversing edges. The objective is to find a minimal-cost path to the goal state, so we assume the goal is an absorbing state, and we can solve the problem as a finite- or infinite-horizon decision problem. Note that this formulation is only valid if there is no uncertainty about whether edges are collision-free.

If the PRM graph includes edges that are uncertain, the planning problem is to find the shortest cost path from the initial state to the goal given the uncertainty about which edges are collision-free and the information we can gather while moving through the graph. Figure 1 illustrates the problem on a very small PRM with only one uncertain edge. For this example we assume that points A and B are close enough to the obstacle to observe with certainty if the edge is blocked. If the edge is collision-free, the optimal path is to move from S to A to G. If the edge is blocked, then S to C to G is optimal, but if our current belief is that the edge is blocked with $p = 0.5$, the optimal behaviour is to move to B, observe the edge from there, and then move to A or C as appropriate. The important observation is that the optimal policy under uncertainty is different from the best policies in each of the possible worlds. B only looks optimal if the planner can reason about the information it will gain, otherwise going to A or C appears a better choice.

We can think of this as a model identification problem. It is a Markov decision problem in the sense that we select actions to move from state to state, our movement depends only on our current state, and we always know which state we are in. However, the problem is that we don't know exactly which MDP we are moving in, and the challenge is to select actions to maximise long-term reward, but this depends on the model, so we must identify (some of) the model in order to do this.

## POMDP Representation

We represent the model identification problem as a partially observable MDP. A POMDP is a tuple $< S, A, O, T, H, R >$ where $S$ is the set of possible system states, $A$ is the set of possible actions, $O$ is the set of possible observations, $T = P(s, a, s')$ is the transition function that governs how an action changes the state of the system, $H = P(o|s, a, s')$ is the observation function that governs how likely each observation is given a state, action and resulting state, and $R = r(s, a)$ is the reward function which specifies the immediate utility (or cost) of doing a particular action in a state. This is an MDP model with the addition of observations.

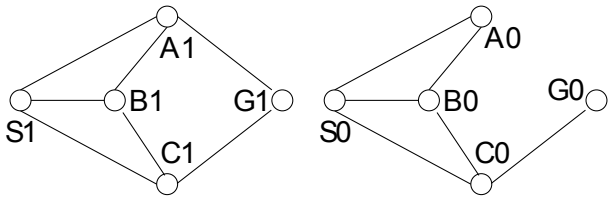


Figure 2: The POMDP model of the graph in Figure 1. The states and actions are shown, the rewards are as before. At each vertex, we observe the vertex (letter) we are at. We additionally observe A→G as free at $A1$ and $B1$ and as blocked at $A0$ and $B0$. Note that this makes $S0$ and $S1$ indistinguishable.

Intuitively, to translate our model identification problem into a POMDP, we make a copy of the PRM graph for each of the possible models (the set $W$ above). For figure 1, the two models are shown in figure 2. These models have no uncertainty about their edges. The POMDP state space is the union of the state spaces of all these MDPs, giving $n2^m$ states. In the example, $m = 1$ and $n = 5$ giving the 10 states shown. The action set consists of one action for each node: 5 in this case and the rewards and transition functions are the unions of the set of MDPs.

We define the function valid$(v, v', w)$ to represent the fact that there is a collision-free edge in world $w$ between $v$ and $v'$. Formally, valid$(v, v', w)$ is true if $\exists e \in E : e = \langle v, v' \rangle$ and $e$ is collision-free in $w$.

We can now formally define the model identification POMDP as follows:

- $S = V \times W$ as described above. Each state is a combined $\langle v \in V, w \in W \rangle$ pair.
- $A = \{\text{goto-}v : v \in V\}$ as above (there are obviously other ways to encode the actions which lead to smaller POMDP formulations, but for clarity's sake, we will use this encoding here).
- $O = \mathcal{P}(\bar{o})$. Each observation consists of the vertex that we are at, plus an observation of all the uncertain edges. $\bar{o} = \langle o_1, o_2, \ldots, o_m \rangle$ where $o_i$ is an observation of uncertain edge $i$.
- $T$ is defined as follows:

$$P(\langle v, w \rangle, \text{goto-}v', \langle v'', w' \rangle) = \begin{cases} 1 & \text{if } w = w',\ v' = v'', \text{ and valid}(v, v', w) \\ 0 & \text{otherwise} \end{cases}$$

$T$ defines the probability of a transition from one state to another. All transitions have probability zero except transitions where the PRM states being moved between are in the same world, and there is a collision-free edge in that world between those two states.

- $H$ is defined as follows:

$$P(\overline{o}|\langle v, w\rangle, \text{goto-}v', \langle v', w\rangle) = \begin{cases} P(\overline{o}|v', w) & \text{if valid}(v, v', w) \\ 0 & \text{otherwise} \end{cases}$$

For brevity we have relaxed the notation compared to the definition of $T$. $H$ is the probability of an observation given a state, action, and resulting state, and in this case is defined solely by the resulting state. Whenever we perform an action in the POMDP, we get an observation of whether each uncertain edge is collision-free or blocked.

- $R$ is defined as in the MDP model above by:

$$R(\langle v, w\rangle, \text{goto-}v') = \begin{cases} -c_e & \text{if } e = \langle v, v'\rangle, \text{valid}(v, v', w) \\ & \text{and } v' \neq v_G \\ -\infty & \text{otherwise} \end{cases}$$

The belief state is only a distribution over the possible worlds, so we can represent the fact that e.g. two edges always are in the same state (collision-free or blocked) by making all worlds in which one is blocked and the other is not have prior probability zero. The agent is then forced to believe those worlds are impossible. If the probabilities that two edges are blocked are not independent, this is represented in the POMDP model using the belief state. If two edges are independent, then the belief state should be such that the probability of the model(s) in which both edges are blocked is the product of the probabilities that each edge is blocked. Dependencies are encoded in the belief state, so the choice of initial belief state is crucial — it specifies the edge dependencies to the agent in the world. We use various initial belief states for the experiments to setup edges as being dependent or not as appropriate.

We can now solve this POMDP to find an optimal policy under our uncertainty about which model we are in. All the reachable belief states when executing this (or any other) policy will consist of non-zero probabilities for (some of) the $2^m$ POMDP states corresponding to the current graph vertex, and zero probability for all other states. The non-zero probabilities indicate the probability, given the actions and observations so far, for each of the possible models. The optimal policy for the problem in 1 is shown in Figure 3 where the possible belief states are shown as circles with the PRM node and the belief that the edge from $A$ to $G$ is collision-free. The optimal policy is shown by the arrows, and is to go to B, discover whether the edge is blocked, and then to go to either A if it is collision-free, or C otherwise.

## A Continuous MDP Model

Solving even very small POMDPs is computationally hard. In this case, there is a lot of structure in the POMDP, in particular in the kinds of belief states that are possible. We would like to find ways to exploit this structure to more efficiently solve the problem. There are a number of general POMDP solvers that exploit structure in the belief space, for example point based value iteration (Pineau, Gordon, and Thrun 2003) or belief compression (Roy, Gordon, and Thrun 2005). We would expect that these approaches could exploit the structure we see, but since these are general approaches (they can exploit many kinds of structure in belief states), we should be able to do significantly better by using algorithms specific to the type of structure present.

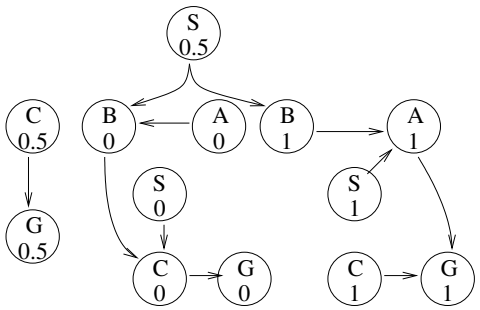


Figure 3: The optimal policy for the example graph in Figure 1. The policy shown is for starting from $S$ with a probability of 0.5 that the uncertain edge is blocked. The split arrow from $S$ indicates that the resulting state is unknown.

In the case of our domain, the only belief states that are possible are ones that are certain about which graph vertex they are in, but have probabilities over the which of the $2^m$ possible worlds is the true one. We now develop an approximate MDP model that explicitly represents this structure.

Consider a problem such as the one in Figures 1 and 2 in which there is only a single uncertain edge. In this case, every possible belief state consists of an underlying MDP state plus a single probability of being in the obstacle-free world. Thus we can solve this problem using an MDP solver for continuous states (or approximately by discretising the continuous variable). When we generalise this approach to higher numbers of uncertain edges, since the edge probabilities could be dependent on one another, the $m$ uncertain edges produce $2^m$ possible worlds, leading to an MDP with states with $2^m - 1$ continuous variables.

To formulate this model as an MDP, we discretise the belief space over each continuous variable (the probability of each model). Since zero and one are important to distinguish in these problems as they lead to smaller branching factors in the MDP, we discretise into $d + 1$ values as follows: $D = \{0, 1/d, 2/d, \ldots, (d-1)/d, 1\}$ where every value in the range $[i/d - 1/(2d), i/d + 1/(2d))$ is discretised to $i/d$, and 0 and 1 correspond to the ranges $[0, 1/(2d))$ and $[1 - 1/(2d), 1]$ respectively. Given such a discretisation, and a set of $2^m$ possible worlds, let $\widetilde{P}(w_i)$ be the probability of world $i$, discretised according to $D$. Let $\widetilde{P}_W = \{\widetilde{P}(w_1), \ldots, \widetilde{P}(w_{2^m})\}$ such that $\sum_i \widetilde{P}(w_i) = 1$ be a discretised assignment of probability to every possible world. Let $\widetilde{P} = \mathcal{P}(\widetilde{P}_W)$ be the set of all such assignments.

We can now define the discretised MDP formulation of

the problem as follows:

- $S = V \times \widetilde{P}$ is the set of states. It represents the cross product of the set of vertices in the PRM graph with the set of all possible discretised beliefs about which model is the true one.
- $A = \{\text{goto-}v : v \in V\}$ as before.
- $R$, the reward function, is defined as follows:

$$R(\langle v, \widetilde{P}_W \rangle, \text{goto-}v') = \sum_w \widetilde{P}(w) R(\langle v, w \rangle, \text{goto-}v') \quad (1)$$

  where $R(\langle v, w \rangle, \text{goto-}v')$ is defined as in the POMDP formulation above. Intuitively, the reward for doing a particular action in a belief distribution over the worlds is the expected reward for the action in the POMDP where the expectation is over the belief distribution, because it depends on the state that results from the action.
- $T$, the transition function, is somewhat complex to define as we have to specify both the transition probabilities, and the states that result. When we make a transition in the POMDP formulation above, the actual transition is deterministic in the underlying states, but the observation (which is stochastic) moves us from one belief state to another. In the MDP formulation, the transitions with non-zero probability correspond to all the possible observations that could be made, and the resulting states are the discretised belief states that would result from making each observation. This is expressed formally as follows:

$$P(\langle v, \widetilde{P}_W \rangle, \text{goto-}v', \langle v', \widetilde{P}'_W \rangle) = p$$

  We first consider the cases where $p$ is non-zero. For this to be true, $\langle v, v' \rangle \in E$. Suppose we make an observation $\overline{o}$, then the belief state we move to is:

$$\widetilde{P}'_W = \{P_{v'}(w_1|\overline{o}, \widetilde{P}_W), \ldots, P_{v'}(w_{2^m}|\overline{o}, \widetilde{P}_W)\}$$

  where $P_{v'}(w_i|\overline{o}, \widetilde{P}_W)$ is the new probability of being in model $i$ after observing $\overline{o}$, which is:

$$\begin{aligned} P_{v'}(w_i|\overline{o}, \widetilde{P}_W) &= \frac{P(\overline{o}|v', w_i) P(w_i)}{P(\overline{o}|v', \widetilde{P}_W)} \\ &= \frac{P(\overline{o}|v', w_i) P(w_i)}{\sum_j P(\overline{o}|v', w_j) P(w_j)} \end{aligned} \quad (2)$$

  Where $P(\overline{o}|v, w)$ is defined in the PRM graph (it is the probability of seeing observation $\overline{o}$ from vertex $v$ if the true world is $w$), and $P(w)$ given by the prior world probabilities from $\widetilde{P}_W$. We now discretise $P_{v'}(w_i|\overline{o}, \widetilde{P}_W)$ to produce the new belief state after the action.

  It only remains to compute $p$, the probability of reaching these belief states, and this is given by:

$$\begin{aligned} p &= P(\overline{o}|v', \widetilde{P}_W) \\ &= \textstyle\sum_j \widetilde{P}(w_j) P(\overline{o}|v', w_j) \end{aligned} \quad (3)$$

  For all states other than those identified above, $p = 0$.

As before, if two uncertain edges are correlated, this is reflected in the belief state over the possible worlds, so affects only the initial state in the MDP.

Having defined the MDP, we can now solve it to get an optimal (modulo the discretisation) policy for the POMDP, and hence for the original PRM problem. We will refer to this MDP formulation as the *dependent MDP* as it allows us to represent belief states where there are arbitrary dependencies between the probabilities that edges are collision-free. The dependent MDP formulation is significantly quicker to solve than the POMDP formulation of the problem.

## Belief State Reachability

As was mentioned in the foregoing discussion, there is a lot of structure in the belief space. This structure is exploited by the MDP but more information is available that is not utilised. The start state specifies the agent's prior belief over which world is the correct one, yet the MDP computes a policy for all states. Many MDP states are never visited from the starting state under any sensible policy. An obvious improvement therefore, is to only consider the value of states reachable from the start state. Several methods for this exist such as RTDP (Barto, Bradtke, and Singh 1993) or envelope methods (Dean et al. 1995). Since the MDPs we are solving represent stochastic shortest path problems, which, barring uncertainty, are optimally solved by the classic A* search algorithm, it seems natural to use LAO* (Hansen and Zilberstein 2001), a generalisation of AO* tree search algorithm (Nilsson 1986) for AND/OR trees with loops. LAO* is a heuristic search algorithm which builds a graph from the start state and expands leaf nodes according to a heuristic akin to how A* explores a graph. A "best partial solution" graph is maintained which consists of the states reachable from the start state according to the current best policy. Value iteration updates (backups) are applied to those states in the best partial solution graph. LAO* terminates when there are no remaining unexpanded nodes in the best solution graph (BSG) and all nodes in the BSG have converged values under value iteration.

We apply LAO* to our problem to achieve substantially increased efficiency in finding the optimal policy. Our problem domain allows admissible heuristics to be easily devised for the algorithm. We use the shortest-path cost to the goal assuming all edges are free as the basis for the heuristic. This is efficiently computed using Dijkstra's algorithm on the PRM graph. Using the same MDP model formulation as before and applying it to LAO* we have an effective algorithm for solving problems on a realistic scale. LAO* exhibits large gains in performance compared to a standard MDP solver that uses value iteration over the full state space.

For small problems, we can compare value iteration to LAO* directly. On the problem in figure 4 a discretisation of $d = 0.1$ creates an MDP of 7,986 states. The MDP solver took 205.8 seconds to generate the optimal policy, while LAO* produced a solution in 1.5 seconds from a start state where all three uncertain edges were blocked with probability 0.5 and generated a state graph of just 376 nodes. To evaluate the policy LAO* found, we implemented a simulator which randomly samples 'true' world states according to the initial belief distribution and executes the policy in each one. The total incurred cost is recorded when the goal is reached or a maximum number of actions have been taken.

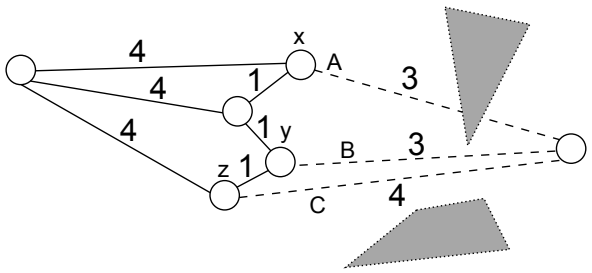


Figure 4: A PRM graph with dependent edges. The uncertain edges are marked with dotted lines, and the top two could be blocked by a single obstacle (shown in grey), so are dependent, while the bottom edge is independent of the other two.

## State Space Reduction

Although the MDP approach is significantly faster than the POMDP solver applied to the same problem, the MDPs it produces are too large to be practically applied to many problems. The problem is the size of the state space: for a PRM graph with $n$ vertices, $m$ uncertain edges, and a discretisation into $d$ discrete values, the total number of valid (belief distributions summing to one) MDP states is:

$$n\frac{(d+2^m-2)!}{(d-1)!(2^m-1)!} \tag{4}$$

For a problem with 20 vertices in the graph, four uncertain edges, and a discretisation of only 0.1 (11 values), this results in around 65 million states.

Examining Equation 4, we see that the major contributors to the number of states are the discretisation and the number of worlds. We can reduce the discretisation to make the MDP smaller, but this produces insufficient discretisation to represent the probabilities accurately. At coarse levels, small changes in the discretisation can lead to large changes in agent behaviour. A better approach is to reduce the number of continuous variables needed to represent the belief state. The easiest way to do this is to assume that the likelihood of each edge being blocked is independent of all others. With this assumption, we don't need to maintain a belief distribution over all the possible worlds, but instead we keep independent distributions for each edge. This reduces the number of states from that given in Equation 4 to $nd^m$, so our example above can be represented using under 300 thousand discrete states.

The changes that need to be made to the MDP formulation for this independent model are to redefine the set of discretised belief states, $\widetilde{P}$. Rather than being the set of possible probability distributions over all possible worlds, this is now the set of products of individual distributions for each edge, so $\widetilde{P} = \mathcal{P}(\widetilde{P}_E)$ where $\widetilde{P}_E = \{\widetilde{P}(e_1), \ldots, \widetilde{P}(e_m)\}$ where $\{e_1, \ldots e_m\}$ is the set of uncertain edges. This then changes the definition of the states space $S$, the reward function $R$, and the transition function.

For the reward function and transition function, the change that needs to be made is in the definition of $\widetilde{P}(w)$ for a world w. This is now defined (by a slight abuse of notation) as:

$$\widetilde{P}(w) = \prod_{e \in w} \widetilde{P}(e) \prod_{e \notin w} (1 - \widetilde{P}(e)) \tag{5}$$

where by $e \in w$ we mean all the uncertain edges that are collision-free in w. The rest of the MDP definition remains the same, with Equation 5 substituted into Equations 1, 2, and 3.

While the independent MDP formulation has a very significant advantage in terms of the number of states and hence the size of problems that can be solved, the policies it produces can be significantly worse than the general MDP formulation. Consider a problem such as that shown in Figure 4 where two of the uncertain edges are precisely correlated so that worlds in which only one of the two is blocked are impossible. We assume we get a reliable observation of edge $B$ from point Y, but no observation of $A$. Consider what would happen if the initial belief state was uninformed about $A$ or $B$, and then reached point Y and observed that $B$ was blocked. The dependent MDP would reason that $A$ would also be blocked and would move to Z. The independent MDP might choose to move towards X, expecting there to be a probability of 0.5 that $A$ was collision-free.

## Clustering Edges

In the PRM formulation of the problem, the obstacles cause the edges to be blocked. This means that if two edges are very close to one another in the PRM graph and are close to the same obstacle, it is very likely that their probabilities of being blocked are dependent. On the other hand, two edges that are far apart in the PRM graph are almost certainly independent. This observation leads us to a second approach to state space reduction that tries to minimise the error in the discovered policy. The idea is to cluster together edges that are close to one another, while leaving far away edges independent. Intuitively this makes the space of possible worlds the cross product of a set of smaller clusters of edges.

As an example, consider again the graph in Figure 4. There are three uncertain edges, but $A$ and $B$ are influenced by a single obstacle, while $C$ is influenced by a different one. If we cluster $A$ and $B$ together independently of $C$, we can represent the graph with three continuous variables for the top two edges, plus one more for the bottom edge, producing an MDP with 18,876 states when discretised at 0.1. This compares with the dependent MDP, which has 116,688 states, and the independent MDP, which has 7,986. If the top two edges are truly independent of the bottom edge, then the clustered MDP will find exactly the same policy for these states as the dependent MDP, but value iteration runs on this problem in 474 seconds on average, compared with 2,703 seconds for the dependent MDP, and 206 seconds for the independent MDP.

The clustered formulation allows the agent to infer information about edges it hasn't directly received an observation about via dependence in the belief state, but still to benefit from the computational advantages of ignoring dependencies for independent edges. With clustering, the quality of

the policy doesn't suffer to the same extent as with an assumption of total independence between edges.

For brevity, we omit the definition of the clustered MDP here, but intuitively, it is analogous with the independent case above. For each cluster of $n$ uncertain edges, there is a continuous variable for each of the $2^n$ worlds, but these are independent of the other clusters.

**Theorem 1.** *The optimal policy for the dependent MDP is at least as good as the optimal policy for the clustered and independent MDPs (neglecting any effects of the discretisation).*

We prove this by observing that since every state in the independent (or clustered) MDP is also represented in the dependent MDP, and since the action space is the same, if a state in the independent MDP has a better action than the corresponding state in the dependent MDP, that action would also be available in the dependent MDP. For it to be better in the independent MDP, it must have higher value, but then it would have a higher value in the dependent MDP than the optimal action, which is a contradiction.

**Corollary 1.** *The optimal policy for the clustered MDP is at least as good as the optimal policy for the independent MDP.*

## Experiments

To demonstrate the power of the MDP approximation, we first solve the POMDP version of the problem using Cassandra's POMDP-solve software[2], and the MDP formulations using a standard implementation of value iteration, in both cases using discounting over an infinite horizon. To compare running times, we run both the POMDP and dependent MDP solvers on the small example in Figure 1 and a slightly larger problem with six vertices in the PRM graph and two uncertain edges. For the graph in Figure 1, with five vertices and one uncertain edge, the POMDP has ten states and five actions and is solved by incremental pruning in 299ms, while the MDP model at a discretisation of 0.1 has 55 states and is solved by value iteration in 8ms. For the two-uncertain edge problem, the POMDP formulation has 24 states and six actions while the dependent MDP formulation has 1716 states. The POMDP is solved by incremental pruning in 63 minutes, and by a grid-based belief compression algorithm in 47 seconds while the full space MDP is solved by value iteration in 158ms. All approaches produce the same policies for the discretised states in the MDP. For larger problems, the POMDP solver cannot be run, and even value iteration soon struggles simply to hold the full MDP in memory, so we can only apply LAO* to these problems.

To investigate the effectiveness of the clustered and independent MDP approximations, we generated five random PRM graphs with 40 vertices and a maximum of four edges per vertex using the standard PRM algorithm. Each graph had five uncertain edges, and since all edges were generated to be outside the mean positions of the obstacles, the prior probabilities of the uncertain edges were always at least 0.5. Figure 5 shows an example of one of these graphs. Unlike

[2]Available from http://www.cs.brown.edu/research/ai/pomdp/

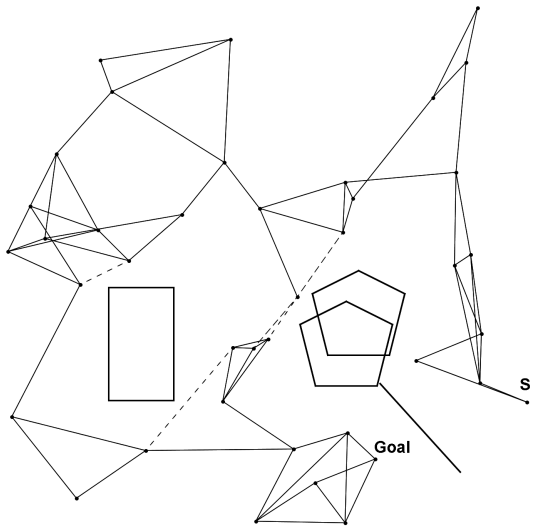


Figure 5: Example of randomly generated PRM graph.

in many other applications, here random graphs are good indicators of real world performance because the PRM algorithm works with random graphs. Unfortunately, the randomness (in this case mostly due to the random positioning of obstacles) leads to widely varying performance, making a statistical analysis of performance difficult.

LAO* was used to compare the dependent, clustered and independent MDP representations on the random graphs. Each graph had a number of edges where the blocked probabilities were correlated, and the clustering was chosen so that not all correlated edges were clustered together (otherwise the clustered and dependent MDPs would certainly have produced the same optimal policy). The first table below shows the policy generation times (averaged over five runs) on each random graph and the second shows the performance of the policies (cost to reach the goal).

| MDP Repn. | Ex 1 | Ex 2 | Ex 3 | Ex 4 | Ex 5 |
|---|---|---|---|---|---|
| Independent | 127 | 91 | 209 | 65 | 342 |
| Clustered | 12867 | 95 | 9257 | 56 | 3083 |
| Dependent | - | 1372 | - | 465 | - |

| MDP | Ex 1 | Ex 2 | Ex 3 | Ex 4 | Ex 5 |
|---|---|---|---|---|---|
| Independent | 1055 | 1475 | 983 | 1184 | 1502 |
| Clustered | 1019 | 1477 | 912 | 1184 | 1428 |
| Dependent | Fail | 1470 | Fail | 1183 | Fail |

All experiments were run on a 3GHz Pentium IV with 500M of memory allocated to the process, and with $d = 1 \times 10^{-5}$ and discount factor $\gamma = 0.999$. Costs were averaged over 50,000 simulated executions of the policy. As the results show, there is considerable difference between the graphs. On Examples 2 and 4, the execution times were very similar for the clustered and independent MDPs and the solution qualities were almost identical for all methods (the difference between the independent and clustered costs in the second table are within the error margin of the simulator). This appears to be because the optimal policies didn't

need to use the information about edge dependencies as either the dependent edges weren't used, or the optimal policy visited vertices that gave information about both uncertain edges. In the other three examples we see that there is a significant advantage to being able to reason about the correlated edges, but that the reachable state space of the dependent MDP is so much larger that the solver ran out of memory before finding a solution.

The solution quality table illustrates the advantages of the clustered model over the independent. Being able to represent knowledge of edge dependencies in the former model enables the creation of better policies. The dependent model is even closer to optimal, but as our results show, even with five uncertain edges the reachable state space is too large to solve reliably. We have been able to solve larger dependent problems by using heuristic weighting (Pearl 1984, section 3.2). This allows us to solve all five example PRM graphs, but at a significant cost in optimality (we can solve Example 1 in only 20ms, but the policy found has an expected cost of 1034). We are currently investigating the costs and benefits of this approach.

The overall conclusion we draw from the experiments is that the dependent and clustered representations are frequently quite close in terms of performance to the dependent MDP, and are generally significantly faster to compute. The advantage of the clustered representation is that it can be customised to the particular graph—if there are uncertain edges where a significant benefit could be gained by treating them as correlated, they can be clustered while the others can be left independent. One area of future work is to look at how this decision could be made automatically without having to compute the policies.

## Conclusions

We have shown that the problem of planning in PRM graphs with uncertain obstacle locations can be thought of as a model identification problem, and can be represented as a POMDP. We have shown that this POMDP has a very structured belief space and that it can be represented and solved efficiently using an MDP approximation. To solve the POMDP efficiently, we developed three MDP approximations. The dependent MDP produces policies that only differ from the POMDP due to the discretisation of the belief space. Using a clustered approach retains the advantages of a fully dependent model while being more scalable to more complex examples. When dependencies between edges are weak, the dependent policy doesn't gain much advantage since there is no benefit to representing the additional states.

At present the algorithm appears to behave in two different ways on different PRM graphs. We are currently doing more experiments to try to understand this better. As well as the work on weighted heuristics mentioned above, we are looking at other ways of scaling the approach to larger problems. One approach is to take advantage of the fact that the agent's belief about the uncertain edges only changes when it visits a graph vertex that allows an observation.

The approach we have discussed here fixes the PRM graph and then explores what can be done in that particular graph structure. Another approach we are looking at is to use a Monte-Carlo sampling approach for the object locations to allow the agent to obtain distributions over object positions as opposed to world models.

## References


Barto, A. G.; Bradtke, S. J.; and Singh, S. P. 1993. Learning to act using real-time dynamic programming. Technical Report UM-CS-1993-002, University of Massachusetts, Amherst MA 01003.

Bohlin, R., and Kavraki, L. 2000. Path planning using lazy PRM. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, 521–528.

Burns, B., and Brock, O. 2006. Sampling-based motion planning using uncertain knowledge. Technical report, University of Massachusetts Amherst.

Dean, T.; Kaelbling, L. P.; Kirman, J.; and Nicholson, A. 1995. Planning under time constraints in stochastic domains. *Artificial Intelligence* 76(1–2):35–74.

Hansen, E. A., and Zilberstein, S. 2001. LAO * : A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence* 129(1-2):35–62.

Hsu, D.; Kindel, R.; Latombe, J.; and Rock, S. 2000. Randomized kinodynamic motion planning with moving obstacles. In *Workshop on the Algorithmic Foundations of Robotics*.

Hsu, D.; Jiang, T.; Reif, J.; and Sun, Z. 2003. The bridge test for sampling narrow passages with probabilistic roadmap planners. In *IEEE International Conference on Robotics and Automation*, 4420–4426.

Jaillet, L., and Simeon, T. 2004. A PRM-based motion planner for dynamically changing environments environments. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*.

Kavraki, L., and Latombe, J. 1998. *Probabilistic roadmaps for robot path planning*. John Wiley, West Sussex, England. 33–53.

Missiuro, P., and Roy, N. 2006. Adapting probabilistic roadmaps to handle uncertain maps. In *Proceedings 2006 IEEE International Conference on Robotics and Automation*, 1261–1267.

Nielsen, C., and Kavraki, L. 2000. A two level fuzzy PRM for manipulation planning. Technical Report TR2000365, Rice University.

Nilsson, N. J. 1986. *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers, Inc. San Francisco, California.

Pearl, J. 1984. *Heuristics: Intelligent search strategies for computer problem solving*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Pineau, J.; Gordon, G.; and Thrun, S. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1025 – 1032.

Roy, N.; Gordon, G.; and Thrun, S. 2005. Finding approximate pomdp solutions through belief compression. *Journal of Artificial Intelligence Research* 23:1–40.