Nearly Deterministic Abstractions of Markov Decision Processes

Terran Lane and **Leslie Pack Kaelbling**
MIT Artificial Intelligence Laboratory
200 Technology Square
Cambridge, MA 02139
{terran,lpk}@ai.mit.edu

Abstract

We examine scaling issues for a restricted class of compactly representable Markov decision process planning problems. For one stochastic mobile robotics package delivery problem it is possible to decouple the stochastic local-navigation problem from the deterministic global-routing one and to solve each with dedicated methods. Careful construction of macro actions allows us to effectively "hide" navigational stochasticity from the global routing problem and to approximate the latter with off-the-shelf combinatorial optimization routines for the traveling salesdroid problem, yielding a net exponential speedup in planning performance. We give analytic conditions on when the macros are close enough to deterministic for the approximation to be good and demonstrate the performance of our method on small and large simulated navigation problems.

Introduction

Imagine a robot that runs errands in a large office building. At any given time, it has a set of pending requests to deliver items, pick up printer output, and so on. Perhaps it also acts as security guard, with the task of keeping certain areas under surveillance by visiting them periodically. It must also ensure that its batteries never completely run down by periodically visiting a charging station. If the robot's actions were entirely deterministic, the only uncertainty in the domain would be in the arrival of errand requests. However, there is always a certain amount of unreliability in a robot's actions.

Markov decision processes (MDP) have been popular models for uncertain planning problems, such as this one. They handle uncertainty effectively, but are computationally too complex for such large domains. While this domain can be represented quite compactly, traditional solution methods are intractable for it. If we would like to solve such large domains, we will have to give up complete optimality for a reduction in computation time. Many promising approaches to doing so, via abstraction and factorization, have been suggested in the literature. These techniques are general-purpose and it is, therefore, difficult to characterize the degree to which the behavior they generate is suboptimal. Furthermore, recent complexity results indicate that even for compactly representable MDPs, the exact and approximate planning problems for MDPs are intractable on *general* domains (Littman 1997; Mundhenk *et al.* 2000; Lusena, Mundhenk, & Goldsmith 2001). Thus, no single algorithm can be both efficient and accurate for *all* MDPs.

In this paper we argue that it may be useful to step back from general purpose algorithms. Rather, we should seek additional special structure in our MDP domains (beyond the transition factorability that leads to compactly expressible models) and exploit it with special-purpose algorithms. For the same reason that we do not, in practice, employ a single, all-purpose graph algorithm or combinatorial optimization algorithm, we believe that we can make significant progress on MDP planning by examining restricted classes of problems.

As an example of this approach, in this work we study a particular, quite restricted, class of MDPs, and provide an approximation method with bounded error. The problem class includes a robot running errands in an office, though, in its current form, it does not extend to problems of surveillance or battery maintenance. We provide a formal description of the problem, a formulation of the domain in terms of special-purpose macro actions, and an algorithm for efficiently deriving approximately optimal solutions to the resulting semi-MDP. We give a bound on the error on the approximation as a function of properties of the domain and conclude with empirical results in small and large simulated delivery scenarios.

Formal Development

In this section, we give a brief background on Markov decision process theory, give our notation, and formally describe the package delivery domain and our planning method.

Markov Decision Processes and Options

A Markov decision process is a model of a finite, discrete, synchronous control process with noisy control actions (Puterman 1994). Formally an MDP $\mathcal{M}$ is specified by four components: a *state space*, $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$, of cardinality $|\mathcal{S}| = N$; a set of primitive (or atomic) *actions*, $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$; a *transition function*, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$; and a *reward function*, $R : \mathcal{S} \to \mathbb{R}$. An agent acting in an MDP is, at any time step, located at a single state $s \in \mathcal{S}$. The agent chooses an action $a \in \mathcal{A}$ and is relocated to a new

state, s', determined by the transition probability distribution $T(s'|s,a)$, whereupon it receives reward $R(s')$. In this paper we are concerned with *cost to move* frameworks in which each atomic action incurs some movement cost $c < 0$ and there exists one or more zero-cost, absorbing "goal" states. We will assume here that $R(s') = c$ for all non-goal states.

In many useful domains, the state space is best expressed as a product of state variables $\mathcal{S} = \mathcal{V}_1 \times \ldots \times \mathcal{V}_m$, and the cardinality of the total state space is exponential in m. Such MDPs can often be compactly represented by exploiting structure in their transition and reward functions—writing the former as a dynamic Bayes net (DBN) and the latter as a decision tree (Boutilier, Dearden, & Goldszmidt 2000).

The output of an MDP planning algorithm is a *policy* $\pi : \mathcal{S} \rightarrow \mathcal{A}$ (or *plan* in this work) that specifies an action for the agent to take for any state in the state space. Our goal is to locate a plan that maximizes the expected aggregate reward received by the agent over its lifetime. In addition, because we are interested in plans for exponentially large state spaces, we will seek implicitly represented plans that specify actions only for subsets of the state space.

For our model of macro actions, we adopt the *options framework* of Sutton, Precup, and Singh (Sutton, Precup, & Singh 1999; Precup 2000). An option (or *macro* in this paper) is an abstraction that allows us to treat an entire MDP policy as a single action. In this model, an option o specifies three elements: a set $\mathcal{I} \subseteq \mathcal{S}$ from which it can be initiated, a probability mapping $\beta : \mathcal{S} \rightarrow [0,1]$ giving the probability that the macro terminates upon reaching state s, and a policy π defining the agent's actions while the option is in force.[1] An agent acting in an MDP $\mathcal{M}$ with option set $\mu = \{o_1 \ldots o_k\}$ at each step can take any applicable option as a discrete action choice. Upon invocation of macro o_i, the agent executes actions according to π_i until the option terminates probabilistically according to β_i. The combination of such macros with an MDP yields a semi-Markov decision process (SMDP): a decision process in which actions are considered to have temporal extent with stochastic distribution.

Domain Description

This work was motivated by navigational problems arising in mobile robotics domains. We take as an example a simple package delivery problem in which an agent navigates through a building with stochastic movement commands and attempts to deliver packages to fixed locations.[2] We encode the robot's state with the state variables x and y, denoting physical location in the world, and k "indicator" bits, $b_1, \ldots, b_k$, which record the delivery status of each of k packages ($b_i = 1$ iff package i has been successfully delivered). The goal of the agent is to attain a state in which all packages have been delivered ($b_i = 1$ for all i) in the fewest steps possible. The robot has four primitive actions available, corresponding to movement in the cardinal directions. The actions are stochastic with dynamics defined by a dynamic Bayes net of the topology displayed in Figure 1.

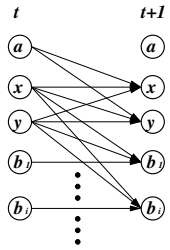

Figure 1: DBN topology for the package delivery domain. The nodes are the agent's choice of action, its x and y coordinates, and the settings of the indicator bits $b_1 \ldots b_k$.

The conditional probability table interrelating the x and y variables describes the physical geography of the world—locations of walls and doors—as well as the robot's movement dynamics. For this paper, we take a simple model in which movement in the cardinal directions succeeds with some fixed probability $p_{\text{trans}} < 1$. When movement fails, it returns the agent to its original $\langle x, y \rangle$ location or deposits it in one of the accessible adjacent grid cells with uniform probability. Walls are impenetrable except at doorways, which allow free movement. The relation between $\langle x, y \rangle$ and b_i defines the notion of package delivery—when the robot reaches the delivery location for package i, denoted $\text{loc}(i)$, the package is delivered and b_i is set to 1 with probability 1. Thereafter, b_i can never be reset to 0 (i.e., a package cannot later be wrenched away from its recipient). Importantly, the package bits are independent of each other given the robot's location—delivery of one package does not prevent delivery of another, nor does it change the dynamics of the robot's movement. This independence will later allow us to decompose the model into sub-problems corresponding to the tasks of delivering individual packages.

The reward function encodes our notion of goals for the robot. Here we consider only the simplest possible reward function: we reward the agent only for successfully completing the entire task (i.e., delivering all the packages) and we wish to minimize the total number of steps taken. This can be modeled in a infinite-horizon undiscounted model with the reward and value functions:

$$\begin{aligned} R(s) &= c < 0 \quad \text{iff some } b_i = 0 \text{ in state } s \\ R(s) &= 0 \quad \text{otherwise} \\ V^{\pi}(s) &= \mathrm{E}\left[\sum_{t=0}^{\text{final delivery}} R(s_t)\right] \end{aligned} \tag{1}$$

This is a negative model whose optimal value function is finite (Puterman 1994). Extension to prioritized goals is not difficult, but yields a more complex deterministic optimization problem (the minimum latency path problem (Goemans & Kleinberg 1996; Arora & Karakostas 1999)). We discuss this case fully in the extended version of this paper.

[1] Technically, we are using only *flat, pure Markov options* here.

[2] We may be able to address arbitrary locations by exploiting Moore et al.'s equivalent of an all-pairs shortest-paths data structure for MDPs (Moore, Baird, & Kaelbling 1999).

Optimality Criterion

While the general optimality criterion for MDPs is extremely expressive, we gain a great deal of leverage in the package delivery domain by observing that the structure of its reward function leads to a special form of optimality criterion. In particular, we can rewrite the value function of Equation 1 to reflect the underlying optimization problem:

$$\begin{aligned} V^{\pi}(s) &= \mathrm{E}\left[c(\#\text{ steps to deliver all packages})\right] \\ &= c\,\mathrm{E}\left[\sum_{i=1}^{k} n(i-1,i)\right] \\ &= c\sum_{i=1}^{k} \mathrm{E}[n(i-1,i)] \\ &= c\sum_{i=1}^{k} d_{\tau(i)}(\tau(i-1)) \end{aligned} \tag{2}$$

where the expectation is over trajectories through $\mathcal{M}$ under policy π and $n(i,j)$ is the random variable representing the number of steps taken between the deliveries of the i^{th} and j^{th} *previously undelivered* packages on the trajectory (delivery of package 0 is taken to be the start state of the robot). The step from the third to fourth line removes the "previously undelivered" caveat from the distance function into an ordering variable, τ. τ is a permutation of locations: $\tau(i) = j$ indicates that package j is the i^{th} delivered. The variable $d_i(j)$ gives the expected number of atomic steps between the location of package j and the location of package i (regardless of whether either has yet been delivered).

Thus, there are two quantities that we must address simultaneously to achieve an optimal policy: the number of steps between delivery locations and the order in which to move among locations. The independence of movement dynamics (x and y variables) from the settings of the package indicator bits ensures that we can optimize the two quantities separately. We can optimize paths between locations ($d_i(j)$) without regard to the settings of the bits, and we locate the best ordering of package deliveries (τ) without considering how to get from one location to another. We will address each of these in the next section, but the reader may observe that the first quantity can be represented with carefully constructed macros while the second (given by the sum in Equation 2) is simply the traveling salesdroid problem (TSP) optimization criterion.

Planning with Semi-Stochastic Macros

Figure 2 summarizes our decomposition and planning approach. We proceed in two phases. In the first, off-line, phase, we develop macros for achieving limited sub-goals and characterize their performance in terms of their expected accrued rewards for invocation and their transition distributions. Formally, we construct the k sub-MDPs $\mathcal{M}_i$, $i = 1 \ldots k$, on the state spaces $\mathcal{S}_i = \langle x, y, b_i \rangle$ by removing the extraneous variables from the transition DBN of Figure 1. In doing so, we violate no dependencies between x, y, and b_i and the resulting model is a valid MDP corresponding to the task of delivering package i in isolation. $\mathcal{S}_i$ is exponentially smaller than $\mathcal{S}$ and, for the purposes of this paper, we will assume that it is tractable for classical MDP planning techniques.[3] This yields a policy π_i that expresses the best way to deliver package i alone from any $\langle x, y \rangle$ coordinate. We now construct the option $o_i = \langle \mathcal{I}_i, \pi'_i, \beta_i \rangle$ where $\mathcal{I}_i = \{s \in \mathcal{S} : b_i = 0\}$, $\pi'_i(s') = \pi_i(s)$ whenever $\langle x', y', b'_i \rangle = \langle x, y, b_i \rangle$, and $\beta_i(s) = 1$ whenever $\langle x, y \rangle = \mathrm{loc}(j)$ for any j such that $b_j = 0$. This option expresses the notion "whenever package i is undelivered, you can deliver it by following π_i until you reach *any* delivery location for an undelivered package." Each macro terminates with probability one at one of at most k $\langle x, y \rangle$ locations.

1. **Preprocessing (Macro construction):**
   (a) Decompose full model $\mathcal{M}$ into sub-models $\mathcal{M}_1, \ldots, \mathcal{M}_k$ corresponding to individual sub-goals
   (b) Solve sub-models using, e.g., value iteration
   (c) Construct macros $o_1, \ldots, o_k$ for sub-goals
   (d) Solve for macro rewards and transition distribution
2. **Per-Episode (Macro integration):**
   (a) Construct goal graph from episode package set, s_0, and (fixed) macro characteristics
   (b) Solve deterministic graph problem (via TSP solver)
   (c) Convert graph solution to macro policy in original $\widehat{\mathcal{M}}$
   (d) Execute macro plan

Figure 2: Summary of decomposition and planning method.

The set of options $\mu = \{o_1, \ldots, o_k\}$ represents a set of actions for a semi-Markov decision process $\widehat{\mathcal{M}}$ over the reduced state space $\widehat{\mathcal{S}} = \{\mathrm{loc}(i)\} \times b_1 \times \ldots \times b_k$. In principle, this process can be solved exactly with standard techniques, but it is still exponentially large in k. In the second, per-episode planning phase, we treat this instead as a deterministic graph and solve it with a TSP planner. To do so, we need estimates of the mean reward (cost) accrued by each macro and its distribution over next states. The first quantity, d_i, is related to the mean absorption time of the chain induced by π_i on $\mathcal{M}_i$, while the second is just the probability of absorption into each goal location (Kemeny & Snell 1976). Both can be calculated from the macros and sub-models in time polynomial in $|x|$ and $|y|$. We will use the latter quantity to determine whether the macro can be reasonably approximated as deterministic.

Although $\widehat{\mathcal{M}}$ is technically a semi-MDP, it can be treated as an MDP because we are working in an undiscounted model. This means that the transit times from state to state affect the reward received on the transition, but have no effect on the future value of the resulting state. Thus, if we let the reward depend on both the start and end states of a transition, we can stay within the MDP framework. We will define $c(s, s'|a)$ to be the expected cost of making a transition from state s to s' under macro action a.

[3] If the remaining spatial problem is itself too large for direct solution, we can resort to further decompositions, such as hierarchical spatial partitioning, or to other scaling techniques.

On each planning episode, we are presented with a set of packages to be delivered, represented as a configuration of package delivery bits b_i, and a starting state s_0. We construct a deterministic graph G that approximates $\widehat{\mathcal{M}}$ under the assumptions that every macro reaches its nominal goal (i.e., that o_i terminates at $\langle x, y\rangle = \mathrm{loc}(i)$) and that it takes exactly its expected duration in doing so. Formally, $G = \langle V, E\rangle$ where $V = \{\mathrm{loc}(i) : b_i = 0 \text{ in the episode description}\} \cup \mathrm{loc}(0)$ and $e(i,j) = d_j(i)$ where $\mathrm{loc}(0)$ is the location of the starting state. (We omit goals irrelevant to the current episode from the graph.) Given a starting state s_0, the minimal tour τ over G is the basis for the implicit policy for this episode.[4] Finding τ is, of course, still NP-hard, but there are very effective practical heuristic methods; good approximations for systems with hundreds of thousands of nodes can often be found in seconds (Johnson & McGeoch 2001).

Finally, we map τ back into a policy over macros in $\widehat{\mathcal{M}}$. At any $s \in \widehat{\mathcal{S}}$, the agent chooses option $o_{\tau(i)}$ for $i = \min_j\{j : b_{\tau(j)} = 0\}$ (i.e., it attempts to deliver the first as-yet-undelivered package on tour τ). This is an implicit representation of a total policy—it is defined at all states of $\widehat{\mathcal{S}}$—but it was not created to deal with circumstances such as the agent accidentally wandering into an unexpected delivery location. In the next section we will give bounds on when this willful ignorance still yields acceptable plans.

Analysis of Algorithm

In this formulation of the problem, we are making two approximation steps. The first is the move from the underlying MDP $\mathcal{M}$ to the semi-MDP $\widehat{\mathcal{M}}$, induced by the macro actions. In doing so, we are likely to introduce some error, with the macros no longer allowing us to express the true optimal policy for $\mathcal{M}$. We are currently unable to argue formally that this loss is small; however, we expect that it is, for the intuitive reason that the macros are derived expressly for the purpose of achieving subgoals that are *strictly required* in order to achieve the overall goal of the domain. However, using the macros forces the agent to choose an ordering on the subgoals (or at least, to choose a first subgoal), and does not allow it to be "agnostic"—taking a few actions to see what happens, then pursuing the subgoal that turns out to be nearer, for example. Although we cannot guarantee that such a situation does not occur in our target problems, Parr has developed a test that can discriminate whether a specific, fixed set of macros is sufficient to express a nearly optimal policy (Parr 1998a; 1998b). In practice, we could test a set of "goal-seeking" macros for near-optimality offline before proceeding to the SMDP solution phase.

The second approximation step is treating $\widehat{\mathcal{M}}$ as if it were the deterministic model G. We can provide a bound on the loss due to this approximation, given in the following theorem. The bound will depend on the degree of determinism of the original model, characterized by the parameter p.

[4] Strictly speaking, we are not seeking a full tour, as we do not require return to $\mathrm{loc}(0)$, but we can add synthetic, zero-cost "return to start" arcs to G and find a full tour over the resulting graph.

Let p be the minimum, over all states $s \in \widehat{\mathcal{S}}$ and actions $a \in \mu$ of the maximum, over all $s' \in \widehat{\mathcal{S}}$, of $\Pr(s'|s,a)$, and let $\delta = 1 - p$. In addition, the bound depends on $\Delta_c = c_{\max} - c_{\min}$, the difference between the largest and smallest transition costs in $\widehat{\mathcal{M}}$.

Theorem 1 *For every stationary policy π defined on state space $\widehat{\mathcal{S}}$ and macro action space μ, if the non-determinism of the macro actions is bounded by*

$$\delta \leq \frac{k - \sqrt{k^2 + \frac{2\epsilon}{\Delta_c}(1-k)}}{k^2 - k}$$

then at every state $s \in \widehat{\mathcal{S}}$, the value of state s under policy π in $\widehat{\mathcal{M}}$, $V_\pi(s)$, differs from the value of s under π in G, $D_\pi(s)$, by at most ϵ.

Proof: First note that, by construction, each macro terminates only upon delivering *some* package — its "intended package", with probability $\geq p$, or any of the other previously undelivered packages, with total probability $\leq \delta = 1 - p$. Thus, an agent started at s_0 with k packages outstanding reaches the terminal state of the episode in exactly k macro actions. A policy π on $\widehat{\mathcal{M}}$ can be coupled to G to produce some deterministic tour of cost $D_\pi(s_0) \geq kc_{\min}$ (corresponding to the path of "expected outcomes" in $\widehat{\mathcal{M}}$). We will call the trajectory of "expected outcome" states in $\widehat{\mathcal{M}}$ the *nominal path*, having *expected* cost equal to $D_\pi(s_0)$. The agent, acting according to π in $\widehat{\mathcal{M}}$, will complete the nominal path with probability p^k. If some macro terminates at an unexpected state (total probability $1-p^k$), the agent can still complete the task with, at worst, $kc_{\max}$ cost. Thus, the true value of s_0 under π, $V_\pi(s_0)$, differs from the deterministic approximation, $D_\pi(s_0)$ by at most $\epsilon = (1-p^k)k\Delta_c$. Rearranging and noting that, for small δ, a second-order approximation to the binomial expansion of $(1-\delta)^k$ gives a good upper bound to p^k, yields the desired polynomial relation between ϵ and δ. □

Finally, we note that every policy on $\widehat{\mathcal{M}}$ yields a nominal path (because every macro action has an expected outcome) and every nominal path has a corresponding path in G. Thus, given "reasonably deterministic" macros, the optimal policy for the approximate model G will have value within ϵ of the truly optimal policy for $\widehat{\mathcal{M}}$.

Empirical Investigation

We have implemented this planner for the package delivery domain and examined its performance in a number of synthetic floorplan scenarios: a set of small, randomly generated maps and one larger map roughly corresponding to one floor of a real office building. The random floorplans are, by design, small enough that exact solution is feasible, allowing direct comparison between hybrid and optimal solutions, while the office building simulation is large enough to be intractable to direct solution (2^{30}–2^{55} states) and serves to demonstrate the scalability of our method.

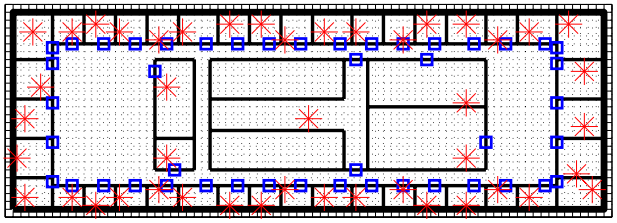

Figure 3: Office building floorplan. Asterisks mark the goal locations and small squares denote doorways.

Comparison to Exact Solution

In our first experiment, we generated a number of small delivery problems populated with randomly selected walls, doors, and goals (for an example of a similar, but non-random, floorplan, see Figure 3). The movement dynamics are those described in the "Domain Description" section, with $p_{\text{trans}} = 0.9$. We constructed floorplans varying in size between 400 and 900 locations with between 4 and 6 delivery locations, yielding MDPs with between 6400 $(400 \cdot 2^4)$ and 57,600 $(900 \cdot 2^6)$ states, the latter near the limit of our ability to solve the system directly. In these worlds we could find the optimal TSP path directly with brute-force search, so the only sources of suboptimality are from the use of macro rather than atomic actions and the deterministic approximation of $\widehat{\mathcal{M}}$.

For each of 54 such maps, we constructed both the exact atomic policy and the atomic expansion of the TSP+macro policy and evaluated both policies analytically. To strictly adhere to the terms of Theorem 1, we should evaluate a macro policy that takes the best TSP tour from *every* SMDP state (i.e., replans the TSP tour from every s in $\widehat{\mathcal{M}}$), but that code was not ready at the time of submission. We will present those results in the extended version of this work, but for this paper we evaluated the policy consisting of a *single* TSP tour—this policy does not try to replan when it falls off the tour, but simply attempts to return to the tour.

At the states corresponding to the semi-MDP states on the optimal tour, the TSP+macro planner achieved a value within 5.8% of the optimal on average. In states substantially off of the tour (e.g., non-SMDP states across the grid from $\text{loc}(0)$ with all packages undelivered), the TSP plan deviates substantially from the optimal—over 70% in some cases—but this is not unexpected, as the TSP system has explicitly neglected such states under the assumption that they are reached with very low probability. This assumption is validated by the small net influence that these states have on the values at the on-tour states. We expect replanning at off-tour SMDP states to dramatically improve these values, as will careful early termination of macros (Precup 2000).

Scaling to Large Scenarios

Our larger simulation is a set of delivery scenarios in the floorplan pictured in Figure 3, roughly modeled on one floor of a large office building. This map is quantized into 75 x and 25 y coordinates and contains 45 delivery locations in offices (and one near the elevator shafts in the center of the map) for a total of roughly 2^{55} states. The dynamics of the world are the same as those of the floorplans of the previous section; the larger map differs only in scale and geography.

In this world, we performed thirty "delivery episodes" with different random subsets of between 20 and and all 45 of the potential delivery locations. These scenarios yield MDPs with far too many states for explicit tabular solution, but the corresponding TSP instances are trivial to approximate. In a preprocessing step, we constructed and cached 45 goal-seeking macros corresponding to the 45 potential goal locations in the world. For each episode, we generated the TSP macro solution using a minimum-spanning-tree heuristic TSP planner[5] and evaluated the plan's performance by averaging over twenty sampled trajectories from the atomic model. Over the thirty episodes, the mean deviation between the projected and sampled trajectory lengths was only 0.38% and the projected length was always within one standard deviation of the mean sampled length. Furthermore, in the tours selected by the TSP planner, $1 - p$ is on the order of 10^{-6}, so the probability of failing to correctly complete the tour is $1 - p^k \approx 10^{-5}$. Unsurprisingly, all of the the six hundred sampled trajectories successfully completed the projected TSP tour without encountering an unexpected delivery location. Together, we take these results to indicate that the deterministic graph is a good approximation of the true semi-MDP in this domain.

Related Work

Our planning method is perhaps closest in spirit to envelope methods (Dean *et al.* 1995; Baum & Nicholson 1998) which attempt to restrict the planner's attention to only a highly probable subset of the state space, either by discarding states or by suppressing dimensions. Our approach can be thought of as a two-phase envelope method: in the first phase, we use structural knowledge about the transition function to discard most dimensions and apply classical stochastic planning techniques. In the second phase, the envelope consists of only the previously discarded dimensions and we employ a deterministic planner to handle the scalability question. In general, this can be dangerous, as it explicitly assumes that the agent *won't* leave the envelope, but we provide analytic sufficient conditions (available after the first phase) on when it is reasonable to make this assumption.

A closely related class of techniques clusters groups of similar or related states together according to their behavior under the global value function (Boutilier, Dearden, & Goldszmidt 2000) or their transition distributions (Dean & Givan 1997). Planning takes place on a model constructed from the "meta states." These approaches typically begin with a coarse clustering of the state space and successively split it when necessary to maintain homogeneity within state clusters. These methods are extremely general and can converge to exact or bounded approximations of the optimal plan, but the state partitioning may, in the worst case, explode to an exponentially large set of singleton clusters. Function ap-

[5]Much more sophisticated TSP heuristics are available, but our interest is in the applicability of deterministic planning in general rather than in the quality of the TSP solution per se.

proximation methods (Koller & Parr 2000), on the other hand, use a bounded space representation for the MDP's value function and, thus, policy, but do not necessarily yield near-optimal plans. Both classes of methods are intended to address general compact MDPs. We instead seek to provide compact and near-optimal policies for only a restricted class of compact MDPs by exploiting additional structure in the model beyond that used to factor the transition function.

Our use of options for macros puts this method into the class of *temporal abstractions* as well (Sutton, Precup, & Singh 1999; Precup 2000). Macros have previously been used to speed up MDP planning and reinforcement learning (McGovern, Sutton, & Fagg 1997) and for knowledge reuse; our contribution is using them to "hide" stochasticity and render the semi-MDP nearly deterministic. Similar macro formulations can be used to partition state variables (rather than suppressing entire dimensions as we do), for example, to hierarchically decompose a physical space into regions such as Voronoi regions or rooms and corridors (Kaelbling 1993; Hauskrecht *et al.* 1998; Parr 1998a; 1998b; Guestrin & Ormoneit 2001). Macro integration again involves a meta-planning process which treats macros as primitive actions. The difficulty is in constructing a complete basis set of macros sufficient to respond to all possible reward scenarios—how you choose to act in one region may depend on apparent rewards in adjacent regions. In general, an exponential number of macros may be required even for a single fixed region.

Our approach, however, need not be exclusive of these other methods for scaling MDP planning. The sub-problems $\mathcal{M}_i$ resulting from the initial model decomposition (or any nondeterministic component of the original MDP in general) could still be intractably large and it may be profitable to apply one of these other techniques to them.

The analysis of our algorithm was inspired by the MDP Simulation Lemma (Kearns & Koller 1999) which demonstrates a notion of similarity between two MDPs. We develop such a similarity between stochastic and deterministic semi-MDPs (i.e., distance graphs), using the finite horizon model and bounded branching factor in place of mixing time and model parameter cardinality.

Conclusions

In this paper, we have demonstrated a method for efficient planning in a restricted class of mixed, "semi-stochastic, semi-deterministic" MDPs. By carefully selecting macros to "hide" the stochasticity of the navigational component of our package delivery problem, we are able to attack the deterministic routing problem directly with special-purpose methods. The combination of macros with the atomic MDP yields a semi-MDP corresponding to the routing problem, and we have given a tractable evaluable relation between the optimal plans for the semi-MDP and its deterministic approximation. Finally, we demonstrated that the two models are close on a large simulated domain and that the value of the optimal deterministic plan is close to that of the optimal atomic solution on some small domains.

In general, we believe that this *type* of approach holds great promise for stochastic planning. MDPs can encapsulate a wide variety of deterministic optimization problems for which good solutions are available; by carefully exploiting the structure of such MDPs and constructing appropriate macro actions, we could harness those solutions directly into the stochastic planning framework. While we have presented our techniques in terms of the TSP optimization problem for mobile robot navigation domains, we believe that this work will extend to related problems like shortest-path, location monitoring, battery maintenance, or vehicle routing. In the extended version of this paper, we address the prioritized packages version of package delivery which yields the more complex *prize-collecting minimum latency path* optimization problem (Goemans & Kleinberg 1996; Arora & Karakostas 1999).

We have assumed here that all macros run to termination (i.e., encountering a delivery location), but it is known that policies can be improved by terminating macros prematurely (Precup 2000). In general this requires knowing the true value of the current *full* state with respect to each available macro. We have avoided computing this term, but we can give sufficient conditions for premature termination of macros given only the *local* information of the agent's current value with respect to each macro's individual sub-goal.

In this work we have employed extensive domain knowledge to decompose the model and identify the underlying optimization problem. One of the most interesting outstanding question is how to automatically identify these quantities, especially for model-free systems. We believe these questions to be difficult, but recent advances in model structure identification and algorithm identification may provide useful insights.

Acknowledgements

The authors would like to thank Luke Zettlemoyer, Natalia H. Gardiol, and Mike Benjamin for valuable discussions on the analysis of the algorithm and comments on early drafts of this paper, and David Karger and Maria Minkoff, for insights on the underlying optimization problem. This work was supported in part by DARPA contract #DABT63-99-1-0012 and in part by NASA award #NCC2-1237.

References

Arora, S., and Karakostas, G. 1999. Approximation schemes for minimum latency problems. In *ACM Symposium on Theory of Computing*, 688–693.

Baum, J., and Nicholson, A. E. 1998. Dynamic non-uniform abstractions for approximate planning in large structured stochastic domains. In *Proceedings of the 5th Pacific Rim International Conference on Artificial Intelligence*, 587–598.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 2000. Stochastic dynamic programming with factored representations. *Artificial Intelligence* 121(1–2):49–107.

Dean, T., and Givan, R. 1997. Model minimization in Markov decision processes. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 106–111. Providence, RI: AAAI Press/MIT Press.

Dean, T.; Kaelbling, L. P.; Kirman, J.; and Nicholson, A. 1995. Planning under time constraints in stochastic domains. *Artificial Intelligence* 76.

Goemans, M., and Kleinberg, J. 1996. An improved approximation ratio for the minimum latency problem. In *SODA: Proceedings of the Seventh ACM-SIAM Symposium on Discrete Algorithms*.

Guestrin, C., and Ormoneit, D. 2001. Robust combination of local controllers. In Breese, J., and Koller, D., eds., *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-01)*, 178–185. Seattle, WA: Morgan Kaufmann.

Hauskrecht, M.; Meuleau, N.; Boutilier, C.; Kaelbling, L. P.; and Dean, T. 1998. Hierarchical solution of Markov decision processes using macro-actions. In Cooper, G. F., and Moral, S., eds., *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*. Morgan Kaufmann.

Johnson, D. S., and McGeoch, L. A. 2001. *The Traveling Salesman Problem and its Variations*. Kluwer Academic Publishers. chapter Experimental Analysis of Heuristics for the STSP. To appear.

Kaelbling, L. P. 1993. Hierarchical learning in stochastic domains: Preliminary results. In *Proceedings of the Tenth International Conference on Machine Learning*, 167–173.

Kearns, M., and Koller, D. 1999. Efficient reinforcement learning in factored MDPs. In Dean, T., ed., *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, 740–747. Stockholm, Sweden: Morgan Kaufmann.

Kemeny, J. G., and Snell, J. L. 1976. *Finite Markov Chains*. Undergraduate Texts in Mathematics. New York: Springer-Verlag.

Koller, D., and Parr, R. 2000. Policy iteration for factored MDPs. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI 2000)*. Morgan Kaufmann.

Littman, M. 1997. Probabilisitic propositional planning: Representations and complexity. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 748–754. Providence, RI: AAAI Press/MIT Press.

Lusena, C.; Mundhenk, M.; and Goldsmith, J. 2001. Nonapproximability results for partially observable Markov decision processes. *Journal of Artificial Intelligence Research* 14:83–103.

McGovern, A.; Sutton, R. S.; and Fagg, A. H. 1997. Roles of macro-actions in accelerating reinforcement learning. In *Proceedings of the 1997 Grace Hopper Celebration of Women in Computing*, 13–18.

Moore, A. W.; Baird, L. C.; and Kaelbling, L. 1999. Multi-value-functions: Efficient automatic action hierarchies for multiple goal MDPs. In Dean, T., ed., *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*. Stockholm, Sweden: Morgan Kaufmann.

Mundhenk, M.; Goldsmith, J.; Lusena, C.; and Allender, E. 2000. Complexity of finite-horizon markov decision process problems. *Journal of the ACM* 47(4):681–720.

Parr, R. 1998a. Flexible decomposition algorithms for weakly coupled Markov decision problems. In Cooper, G. F., and Moral, S., eds., *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*. Morgan Kaufmann.

Parr, R. E. 1998b. *Hierarchical Control and Learning for Markov Decision Processes*. Ph.D. Dissertation, University of California at Berkeley.

Precup, D. 2000. *Temporal Abstraction in Reinforcement Learning*. Ph.D. Dissertation, University of Massachusetts, Amherst, Department of Computer Science.

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

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112:181–211.