# The complexity of policy evaluation for finite-horizon partially-observable Markov decision processes *

Martin Mundhenk † Judy Goldsmith ‡ Eric Allender §

February 23, 1997

**Abstract**

A partially-observable Markov decision process (POMDP) is a generalization of a Markov decision process that allows for incomplete information regarding the state of the system. POMDPs are used to model controlled stochastic processes, from health care to manufacturing control processes (see [19] for more examples).

We consider several flavors of finite-horizon POMDPs. Our results concern the complexity of the policy evaluation and policy existence problems, which are characterized in terms of completeness for complexity classes. Although a large body of literature in mathematics, operations research, and engineering deals with optimization and approximation strategies for POMDPs, there has been little work aimed at characterizing the complexity of these problems and proving lower bounds.

We prove a new upper bound of the policy evaluation problem for POMDPs, showing it is Probabilistic Logspace complete. From this, we prove policy existence problems for several variants of unobservable, succinctly represented MDPs to be complete for $\mathrm{NP}^{\mathrm{PP}}$, a class for which not many natural problems are known to be complete.

Our work presents a uniform approach to a wide variety of planning problems, and differs from previous works as follows. Papadimitriou and Tsitsiklis [22] considered a different policy existence problem, to which the problem to determine an optimal policy does not polynomial-time reduce, whereas it does so in our work. Beauquier et al. [5] considered different optimality criteria. Most of the related AI literature, especially [9, 11, 10] consider computationally simpler problems without probabilistic transitions. Thus, the complexity of their problems is generally lower than of ours (except [18, 14] and one theorem in [9]).

## 1 Introduction

Partially observable Markov Decision Processes (POMDPs) are ubiquitous in the world of mathematical modeling, used by ecologists, economists, engineers, and even computer scientists, to name a few. Although they were only introduced in the early 1960's, there is a rich literature on them, spanning the mathematics, operations research, engineering, scientific, and social science communities. This literature concentrates on applications of POMDPs, and algorithms for computing or approximating them. Very little of the literature discusses the inherent computational complexity of the model in its myriad variations, although there is certainly analysis of particular algorithms.

POMDPs model controllable stochastic processes: there is a set of *states;* a controller chooses among some number of *actions,* each of which has an associated probability matrix of state transitions; associated with each state and action pair is a *reward.* Unlike Markov Decision Processes (MDPs), the controller may not get complete information about the system's current state, i.e. the state space is partitioned into *observations*. The basic goal, given such a model, is to find a strategy or *policy* for choosing actions, that maximizes the total expected reward over some fixed time horizon. Policies may consider the current observation only (stationary), the observation and time (time-dependent), or the full history of observations including the current one (history-dependent). We take the following example from [27].

Consider the job of a doctor. Patients arrive, exhibiting or describing certain symptoms; based on those symptoms, the doctor constructs a hypothesis, which can be modeled as a probability distribution

*An earlier version appeared as *University of Kentucky, CS-Dept. TR 269-96*

†Supported in part by the Office of the Vice Chancellor for Research and Graduate Studies at the University of Kentucky, and by the Deutsche Forschungsgemeinschaft (DFG), grant Mu 1226/2-1. Correspondence address: Universität Trier, FB IV – Informatik, D-54286 Trier (Germany). mundhenk@ti.uni-trier.de

‡Supported in part by NSF grant CCR-9315354.

§Supported in part by NSF grant CCR-9509603.

| policy | succinct | reward | horizon | completeness |
|---|---|---|---|---|
| s/t | – | + | $n$ | NL |
| s/t | – | −/+ | $n$ | PL |
| s/t | $n$ | + | $\log n$ | NP |
| s/t | $\log n$ | −/+ | $\log n$ | PP |
| s/t | $n$ | + | $n$ | PSPACE |
| s/t | $n$ | −/+ | $n$ | PSPACE |

Table 1: Completeness results of policy evaluation problems for partially-observable MDPs

| policy | succinct | reward | actions | horizon | completeness |
|---|---|---|---|---|---|
| s/t | – | + | $n$ | $n$ | NL |
| s | – | −/+ | $n$ | $n$ | PL |
| t | – | −/+ | $n$ | $n$ | NP |
| s/t | $n$ | + | $n$ | $\log n$ | NP |
| s | $\log n$ | −/+ | $\log n$ | $\log n$ | PP |
| t | $\log n$ | −/+ | $\log n$ | $\log n$ | $\mathrm{NP}^{\mathrm{PP}}$ |
| s/t | $\log n$ | −/+ | $n$ | $\log n$ | $\mathrm{NP}^{\mathrm{PP}}$ |
| s/t | $n$ | + | $n$ | $n$ | PSPACE |
| s | $n$ | −/+ | $n$ | $n$ | PSPACE |
| t | $n$ | −/+ | $n$ | $n$ | NEXP |

Table 2: Completeness results of policy existence problems for unobservable MDPs

over some (presumably finite) set of states of health. The doctor can take actions in the form of testing or treatment, based on this distribution. She has access to medical literature, which models the progression of diseases and their responses to various therapies probabilistically. Her goal may be to optimize the patient's health, to minimize costs, or some combination thereof. This system can be modeled as a partially observable Markov decision practice: the states are the various diseases or clusters of diseases; the actions are tests or therapies; the rewards are the costs and the value placed on degrees of healing. How easy is it for the doctor to evaluate a policy of testing and treatment, or to find an optimal one?

We consider the complexity of the following decision problems. The *policy evaluation problem* asks whether a given POMDP has positive expected reward under a given policy. The *policy existence problem* asks whether – given a POMDP and type of policy – there exists a policy of the given type under which the POMDP has positive expected reward. All of the problems considered here are for POMDPs that run for a fixed, finite time horizon, equal to the number of states of the system or the size of the representation of the system.

We consider both straightforward and succinct representations of MDPs. Our interest in succinct representations is motivated by work in AI/planning, including [7, 9, 17], and many others.

For a given POMDP and stationary or time-dependent policy, the expected reward is known to be computable in polynomial time using Dynamic Programming. We show that the Dynamic Programming approach for POMDPs can be performed in a parallel manner. Therefore we obtain an algorithm for the policy evaluation problem in probabilistic logarithmic space (PL) and prove completeness for PL. We then apply our technique to succinctly represented POMDPs. This representation allows encoding of e.g. graphs with regularities, which arise in many practical and theoretical areas of computer science, in a compact way (see [13, 26, 3]). As a matter of fact, every compact representation of POMDPs – e.g. Bayes nets used in several AI applications – can be transformed to such a succinct representation without increasing the representation size by more than a (universal) constant. We show that the policy evaluation problem for succinctly encoded POMDPs is complete for polynomial space (PSPACE). We even show completeness for probabilistic polynomial time (PP), if the POMDPs have "regular" transition probabilities. As consequences, we get completeness results for the policy existence problems for unobservable POMDPs, i.e. POMDPs where the controller does not receive any information about the system's state. Our results are summarized in tables 1 and 2. Each line stands for the completeness of some type(s) of MDP problems. The number of states of the MDP is denoted by $n$. There are lots of factors which contribute to the complexity of the problems. In the "policy" column, "s" stands for stationary, "t" for time-dependent. The "succinct" column says whether the encoding of the MDP is not succinct (–), or whether the encoding is succinct and the rewards and transition probabilities have $\log n$ bits (i.e. "regular" transitions) or $n$ bits. The "rewards" column says whether we consider MDPs with nonnegative rewards, or with unrestricted (i.e. rational) rewards. The "horizon" and the number of "actions" may be restricted accordingly.

To the best of our knowledge, the following papers are the only ones that give complexity results for finite horizon POMDP problems. Papadimitriou and Tsitsiklis [21, 22] have considered POMDPs with non-positive rewards, and asked whether there is a policy with expected value of 0. Note that their maximization problem can be stated as minimization problem for POMDPs with nonnegative rewards (see Section 6). As with many other optimization problems, the complexity of the minimization and maximization versions differs strongly. They showed an NP-completeness result for the stationary, finite-horizon case [21]. For time-dependent policies they showed that the policy existence problem for (fully-observable) MDPs is P-complete (for finite or infinite horizon), and for history-dependent policies they proved the policy existence problem for POMDPs to be PSPACE-complete, and that for unobservable MDPs to be NP-complete [22]. Bylander [9] considered a deterministic variant of that problem, showing that the time-dependent policy existence problem for unobservable, succcinct MDPs with nonnegative rewards is PSPACE-complete. In Theorem 6.1 we show PSPACE-completeness even of the stochastic version of this problem. Unobservable MDPs were also considered in [8] under the aspect of approximability. Other work, such as [11, 10], concentrates on computationally simpler problems than those considered here, primarily because they do not use probabilistic transitions.

We give formal definitions for basic POMDPs in Section 2, and proofs of the basic results in Section 3. Succinctly represented POMDPs are defined and analyzed in Sections 4 and 5. Finally, we state our results about POMDPs with nonnegative rewards in Section 6.

## 2 Definitions and Preliminaries

For definitions of standard complexity classes, reductions, and results from complexity theory we refer to [20]. We give a short review of definitions of counting classes we use. The class #L was defined in [2]. It is the class of functions $f$ such that, for some nondeterministic logarithmically space bounded machine $N$, the number of accepting paths of $N$ on $x$ equals $f(x)$. The class GapL (defined in [25]) is the class of functions $f$ such that, for some nondeterministic logarithmically space bounded machine $N$, the number of accepting paths of $N$ on $x$ minus the number of rejecting paths of $N$ on $x$ equals $f(x)$. Note that every logspace computable function is in #L, and #L is contained in GapL. Whereas all #L functions map to nonnegative integers, GapL functions map to all integers. All GapL functions are computable in polynomial time.

Probabilistic logspace, PL, is defined as the class of sets $A$ for which there exists a nondeterministic logarithmically space bounded machine $N$, such that $x \in A$ if and only if the number of accepting paths of $N$ on $x$ is greater than its number of rejecting paths. In apparent contrast to P-complete sets, sets in PL can be decided using very fast parallel computations (see [6]).

Counting classes for polynomial time or space are defined analogously. #P is the class of functions $f$ such that, for some nondeterministic polynomial time bounded machine $N$, the number of accepting paths of $N$ on $x$ equals $f(x)$. GapP and PP are defined accordingly.

For polynomial space bounded computations, the respective definitions can be shown to yield the same class, so PSPACE = PPSPACE, and #PSPACE is the same as the class of polynomial space computable functions (see [16]).

### 2.1 Markov Decision Processes

A partially-observable Markov decision process (POMDP) describes a controlled stochastic system by its states and the consequences of actions on the system. It is denoted as a tuple $M = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{O}, t, o, r)$, where

- $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{O}$ are finite sets of *states*, *actions* and *observations*,
- $s_0 \in \mathcal{S}$ is the *initial state*,
- $t : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the *state transition function*, where $t(s, a, s')$ is the probability to reach state $s'$ from state $s$ on action $a$ (where $\Sigma_{s' \in \mathcal{S}} t(s, a, s') \in \{0, 1\}$ for $s \in \mathcal{S}, a \in \mathcal{A}$),
- $o : \mathcal{S} \to \mathcal{O}$ is the *observation function*, where $o(s)$ is the observation made in state $s$,

- $r : \mathcal{S} \times \mathcal{A} \to \mathbf{Z}$ is the *reward function*, where $r(s, a)$ is the reward gained by taking action $a$ in state $s$.

We distinguish two special types of observability of a POMDP. If states and observations are identical, i.e. $\mathcal{O} = \mathcal{S}$ and $o$ is the identity function (or a bijection), then the POMDP is called a *fully-observable* MDP. It is called *unobservable*, if the set of observations contains only one element or the observation function is constant, i.e. in every state the same observation is made.[1] If the reward function maps to nonnegative integers, we say that the POMDP has *nonnegative rewards*.

### 2.2 Policies and Performances

A policy describes how to act depending on observations. We distinguish two types of policies.

- A *stationary policy* $\pi_s$ (for $M$) is a function $\pi_s : \mathcal{O} \to \mathcal{A}$, mapping each observation to an action.
- A *time-dependent policy* $\pi_t$ is a function $\pi_t : \mathcal{O} \times \mathbf{N} \to \mathcal{A}$, mapping each pair $\langle$observation, time$\rangle$ to an action.

Let $M = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{O}, t, o, r)$ be a POMDP. A *trajectory* $\theta$ *for* $M$ is a finite sequence of states $\theta = \sigma_1, \sigma_2, \ldots, \sigma_m$ ($m \geq 0$, $\sigma_i \in \mathcal{S}$). The probability $\text{prob}_\pi(M, \theta)$ of a trajectory $\theta = \sigma_1, \sigma_2, \ldots, \sigma_m$ under policy $\pi$ is

- $\text{prob}_\pi(M, \theta) = \Pi_{i=1}^{m-1} t(\sigma_i, \pi(o(\sigma_i)), \sigma_{i+1})$, if $\pi$ is a stationary policy,
- $\text{prob}_\pi(M, \theta) = \Pi_{i=1}^{m-1} t(\sigma_i, \pi(o(\sigma_i), i), \sigma_{i+1})$, if $\pi$ is a time-dependent policy, and

The *reward* $\text{rew}_\pi(M, \theta)$ of trajectory $\theta$ under $\pi$ is the sum of its rewards, i.e. $\text{rew}_\pi(M, \theta) = \Sigma_{i=1}^{m-1} r(\sigma_i, \pi(\cdot))$ dependent on the type of policy. The *performance of a policy* $\pi$ for finite-horizon $k$ with initial state $\sigma$ is the expected sum of rewards received on the next $k$ steps by following the policy $\pi$, i.e. $\text{perf}(M, \sigma, k, \pi) = \sum_{\theta \in \Theta(\sigma,k)} \text{prob}_\pi(M, \theta) \cdot \text{rew}_\pi(M, \theta)$, where $\Theta(\sigma, k)$ is the set of all length $k$ trajectories beginning with state $\sigma$. The $\alpha$-*value* $\text{val}_\alpha(M, k)$ of $M$ for horizon $k$ is $M$'s maximal performance under a policy $\pi$ of type $\alpha$ for horizon $k$ when started in its initial state, i.e. $\text{val}_\alpha(M, k) = \max_{\pi \in \Pi_\alpha} \text{perf}(M, s_0, k, \pi)$, where $\Pi_\alpha$ is the set of all $\alpha$ policies.

For simplicity, we assume that the size of a POMDP is determined by the size $|\mathcal{S}|$ of its state space. We assume that there are no more actions than states, and that each state transition probability is given as binary fraction with $|\mathcal{S}|$ bits and each reward is an integer of $|\mathcal{S}|$ bits. This is no real restriction, since adding unreachable "dummy" states allows one to use more bits for transition probabilities and rewards. Also, it is straightforward to transform a POMDP $M$ with non-integer rewards to $M'$ with integer rewards such that $\text{val}_\alpha(M, k) = c \cdot \text{val}_\alpha(M', k)$ for some constant $c$.

## 3 "Flat" POMDPs

There are different methods how an MDP can be represented. In this section, we consider the complexity of policy evaluation and existence problems for problem instances being represented in a straightforward way. Namely an MDP with $n$ states is represented by a set of $n \times n$ tables for the transition function (one table for each action) and a similar table for the reward function. Because we assume that the number of actions and the number of bits needed to store each transition probability or reward does not exceed $n$, such a representation requires $O(n^4)$ bits. In the same way, stationary policies can be encoded as lists with $n$ entries, and time-dependent policies for horizon $n$ as $n \times n$ tables.

The $\alpha$ *policy evaluation problem for partially-observable MDPs* is the set of all pairs $(M, \pi)$ consisting of a POMDP $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$ and an $\alpha$ policy $\pi$ with $\text{perf}(M, s_0, |\mathcal{S}|, \pi) > 0$.

The $\alpha$ *policy existence problem for unobservable MDPs* is the set of all unobservable MDPs $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$ with $\text{val}_\alpha(M, |\mathcal{S}|) > 0$.

### 3.1 Policy evaluation problems

The standard polynomial time algorithm for evaluating a given policy for a given POMDP uses dynamic programming [22, 23]. We show that for POMDPs this evaluation can be performed quickly in parallel. Eventually this yields the policy evaluation problem being complete for PL. We begin with a technical

[1] Making observations probabilistically does not add any power to POMDPs. Any probabilistically observable MDP can be turned into one with deterministic observations with only a polynomial increase in its size.

lemma about matrix powering, and show that each entry $(T^m)_{(i,j)}$ of the $m$th power of a nonnegative integer square matrix $T$ can be computed in #L, if the power $m$ is at most the dimension $n$ of the matrix.

**Lemma 3.1** *(cf. [25]) Let $T$ be an $n \times n$ matrix of nonnegative binary integers, each of length $n$, and let $1 \leq i, j \leq n$, $0 \leq m \leq n$. The function mapping $(T, m, i, j)$ to $(T^m)_{(i,j)}$ is in* #L.

We review the relations between the function class #L and the class of decision problems PL. GapL is the class of functions representable as differences of #L functions, GapL = #L − #L, and PL is the class of sets $A$, for which there is a GapL function $f$ such that for every $x$, $x \in A$ iff $f(x) > 0$ (see [1]). We use these results to prove the following.

**Lemma 3.2** *The stationary policy evaluation problem for partially-observable* MDP*s is in* PL.

**Proof** Let $\hat{M} = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{O}, \hat{t}, o, \hat{r})$ be a POMDP, and let $\pi$ be a stationary policy, i.e. a mapping from $\mathcal{O}$ to $\mathcal{A}$. We show that $\mathrm{perf}_s(\hat{M}, s_0, |\mathcal{S}|, \pi) > 0$ can be decided in PL.

We transform $\hat{M}$ to an unobservable MDP $M$ with the same states and rewards as $\hat{M}$, having the same performance as $\hat{M}$ under policy $\pi$. Since $\pi$ is a stationary policy, this can be achieved by "hard-wiring" the actions chosen by $\pi$ into the MDP. Then $M = (\mathcal{S}, s_0, \{a\}, \mathcal{O}, t, o, r)$ where $t(s, a, s') = \hat{t}(s, \pi(o(s)), s')$ and $r(s, a) = \hat{r}(s, \pi(o(s)))$. Since $M$ has only one action, the only policy to consider is the constant function mapping each observation to that action $a$. It is clear that $\hat{M}$ under policy $\pi$ has the same performance as $M$ under (constant) policy $a$, i.e. $\mathrm{perf}(\hat{M}, s_0, |\mathcal{S}|, \pi) = \mathrm{perf}(M, s_0, |\mathcal{S}|, a)$. This performance can be calculated using a recursive definition of perf, namely $\mathrm{perf}(M, i, m, a) = r(i, a) + \sum_{j \in \mathcal{S}} t(i, a, j) \cdot \mathrm{perf}(M, j, m-1, a)$ and $\mathrm{perf}(M, i, 0, a) = 0$.

The state transition probabilities are given as binary fractions of length $h = |\mathcal{S}|$. In order to get an integer function for the performance, define the function $p$ as $p(M, i, 0) = 0$ and $p(M, i, m) = 2^{hm} r(i, a) + \sum_{j \in \mathcal{S}} p(M, j, m-1) \cdot 2^h t(i, a, j)$. One can show that

$$\mathrm{perf}(M, i, m, a) = p(M, i, m) \cdot 2^{-hm} \quad .$$

Therefore, $\mathrm{perf}(M, s_0, |\mathcal{S}|, a) > 0$ iff $p(M, s_0, |\mathcal{S}|) > 0$.

In order to complete the proof using the characterization of PL mentioned above, we have to show that the function $p$ is in GapL.

Let $T$ be the matrix obtained from the transition matrix of $M$ by multiplying all entries by $2^h$, i.e. $T_{(i,j)} = t(i, a, j) \cdot 2^h$. The recursion in the definition of $p$ can be resolved, and we get

$$p(M, i, m) = \sum_{k=1}^{m} \sum_{j \in \mathcal{S}} (T^{k-1})_{(i,j)} \cdot r(j, a) \cdot 2^{(m-k+1) \cdot h} \quad .$$

Each $T_{(i,j)}$ is logspace computable from the input $\hat{M}$. From Lemma 3.1 we get that $(T^{k-1})_{(i,j)} \in$ #L. The reward function is part of the input too, thus $r$ is in GapL (note that rewards may be negative integers). Because GapL is closed under multiplication and polynomial summation (see [1]), it follows that $p \in$ GapL. ∎

**Lemma 3.3** *The stationary policy evaluation problem for partially-observable MDPs is* PL*-hard.*

**Proof** Consider $A \in$ PL. Then there exists a probabilistic logspace machine $N$ accepting $A$, and a polynomial $p$ such that each computation of $N$ on $x$ uses at most $p(|x|)$ random decisions [15]. Now, fix some input $x$. We construct an (unobservable) MDP $M(x)$ which has only one action and models the behavior of $N$ on $x$. Each state of $M(x)$ is a pair consisting of a configuration of $N$ on $x$ (there are polynomially many) and an integer used as counter for the number of random decisions made to reach this configuration (there are at most $p(|x|)$ many). Also, we add a final "trap" state reached from states containing a halting configuration or from itself. The state transition function of $M(x)$ is defined according to the state transition function of $N$ on $x$, so that each halting computation of $N$ on $x$ corresponds to a length $p(|x|)$ trajectory of $M(x)$ and vice versa. The state transition function is defined so that each halting computation of $N$ on $x$ corresponds to a length $p(|x|)$ trajectory of $M(x)$ and vice versa. The reward function is chosen such that $\mathrm{rew}(\theta) \cdot \mathrm{prob}(\theta)$ equals 1 for trajectories $\theta$ corresponding to accepting computations (independent of their length), or $-1$ for rejecting computations, or 0 otherwise. Since $x \in A$ if and only if the number of accepting computations of $N$ on $x$ is greater than the number of rejecting computations, it follows that $x \in A$ if and only if the number of trajectories $\theta$ of length $|\mathcal{S}|$ for $M(x)$ with $\mathrm{rew}(\theta) \cdot \mathrm{prob}(\theta) = 1$ is greater than the number of trajectories $\theta$ with $\mathrm{rew}(\theta) \cdot \mathrm{prob}(\theta) = -1$, which is equivalent to $\mathrm{perf}(M(x), s_0, |\mathcal{S}|, a) > 0$. ∎

Because the above reduction function constructs an MDP having only one action, the same hardness proof applies for time-dependent policies.

**Theorem 3.4** *The stationary and time-dependent policy evaluation problems for partially-observable MDPs are* PL*-complete.*

**Proof** Completeness of the stationary case follows from the above Lemmas 3.2 and 3.3. Because every stationary policy is a time-dependent policy too, hardness of the time-dependent case follows from Lemma 3.3. It remains to show that the time-dependent case is contained in PL. Let $\hat{M} = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{O}, \hat{t}, \hat{o}, \hat{r})$ be a partially-observable MDP, and let $\pi$ be a stationary policy, i.e. a mapping from $\mathcal{O} \times \{0, \ldots, |\mathcal{S}| - 1\}$ to $\mathcal{A}$. Essentially, we proceed in the same way as in the above proof. The main difference is that the transformation from $\hat{M}$ to $M$ is more involved. We construct $M$ by making $|\mathcal{S}|$ copies of $\hat{M}$ such that all transitions from the $i$th copy go to the $i+1$st copy, and all these transitions correspond to the transition chosen by $\pi$ in the $i$th step. The rest of the proof proceeds as in the proof of Lemma 3.2. ■

### 3.2 Policy existence problems for unobservable MDPs

It turns out that for unobservable MDPs and for stationary policies, the evaluation and the existence problem are of the same complexity, whereas this is not the case for time-dependent policies. For fully-observable and partially-observable MDPs, the policy existence problems have the same complexity for both types of policies.

**Theorem 3.5** *The stationary policy existence problem for unobservable MDPs is* PL*-complete.*

**Proof** First we show that the problem is in PL. Let $M$ be an unobservable MDP with set of actions $\mathcal{A}$. Because $M$ is unobservable, every policy for $M$ is a constant function $a \in \mathcal{A}$. Then there exists a policy under which $M$ has performance greater 0 if and only if for some $a \in \mathcal{A}$, $M$ under $a$ has performance greater 0. Thus the policy-existence problem for stationary unobservable MDPs logspace disjunctively reduces to the stationary policy evaluation problem for unobservable MDPs. From Lemma 3.2 and the closure of PL under logspace disjunctive reductions (see [1]), it follows that the policy existence problem is in PL.

In order to show PL-hardness, note that for MDPs with only one action there is no difference between the complexity of the policy evaluation problem and the policy existence problem. In the proof proof of Lemma 3.3, every PL-computation was logspace reduced to an unobservable MDP with one action. This proofs PL-hardness of the policy existence problem. ■

The time-dependent policy existence problem for unobservable MDPs turns out to be as complex as the one for partially-observable MDPs.

**Theorem 3.6** *The time-dependent policy existence problem for unobservable MDPs is* NP*-complete.*

Papadimitriou and Tsitsiklis proved a similar theorem [22]. Their MDPs had only non-positive rewards, and their formulation of the decision problem was whether there is a policy with reward 0. However, our result can be proven by a proof very similar to theirs showing a reduction from 3SAT. The fact that the problem is in NP follows from the fact that a time-dependent policy can be guessed nondeterministically in polynomial time, and the policy evaluation problem for the given MDP and the guessed policy can be checked in PL.

## 4 Succinctly represented POMDPs

Succinct representations of MDPs arise when the system being modeled has sufficient structure, e.g., when a state is modelled by the states of $n$ variables, and for each action, the state of each variable depends on only a few of the variables. This is true, for instance, when a system is modeled as a Bayes belief net, or when actions are given in the STRIPS model [9, 18, 14].

Changing the way in which POMDPs (and policies) are represented may change the complexities of the considered problems too. We focus on the concept of *succinct representations*, which was introduced independently in [13, 26]. POMDPs having very regular structure arise in many practical and theoretical areas of computer science, and often it is most natural to represent these processes very compactly. For example, the transition tables of a POMDP with $n$ states and actions can be represented by a Boolean circuit $C$ with $4\lceil \log n \rceil$ input bits such that $C(s, a, s', l)$ outputs the $l$-th bit of $t(s, a, s')$. Encodings of those circuits are no larger than the straightforward matrix encodings. But for POMDPs with sufficient structure, the circuit encodings may be much smaller, namely of size order of the logarithm of the state space. In this case, we say that the problem has a succinct representation. It is easy to see that in going from straightforward to succinct encodings, the complexity of the corresponding decision problems increases at most exponentially. More importantly (and less obviously), there are many problems for which this exponential increase in complexity is inherent (see [4] for conditions). For the problems considered here, the completeness proofs for the flat POMDPs translate upward, for the respective succinctly represented POMDPs.

The *$\alpha$ policy evaluation problem for partially-observable* $s$MDP*s* is the set of all pairs $(M, \pi)$ consisting of a succinctly represented partially-observable MDP $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$ and a succinctly represented $\alpha$ policy $\pi$, such that $\mathrm{perf}(M, s_0, |\mathcal{S}|, \pi) > 0$.

The *$\alpha$ policy existence problem for unobservable* $s$MDP*s* is the set of all succinctly represented unobservable MDPs $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$, such that $\mathrm{val}_\alpha(M, |\mathcal{S}|) > 0$.

## 4.1 Policy evaluation problems

Put succinctly, all results from Section 3 translate to succinctly represented POMDPs.

**Lemma 4.1** *Let $T$ be a $2^n \times 2^n$ matrix of nonnegative integers, each consisting of $2^n$ bits, $1 \leq i, j \leq 2^n$, and $0 \leq m \leq n$. Let $T$ be represented by a Boolean circuit $C$ with $2n + \lceil \log n \rceil$ input bits, such that $C(a, b, r)$ outputs the $r$-th bit of $T_{(a,b)}$. The function mapping $(T, m, i, j)$ to $(T^m)_{(i,j)}$ is in* #PSPACE.

As a consequence, we can prove the policy evaluation problems for succinctly represented MDPs is exponentially more complex than the flat ones.

**Theorem 4.2** *The stationary policy evaluation problem for partially-observable* $s$MDP *is* PSPACE-*complete.*

**Proof** In order to show that the problem is in PSPACE, we can use the same technique as in the proof of Lemma 3.2 yielding here that the problem is in PPSPACE (=probabilistic PSPACE). Ladner [16] showed that FPSPACE = #PSPACE, from which it follows that PPSPACE = PSPACE.

Showing PSPACE-hardness is even easier than showing PL-hardness in the proof of Theorem 3.3, because here we deal with a deterministic class. We can transform the computations of a polynomial space bounded machine on input $x$ directly into a succinct representation of a deterministic MDP. ∎

The same complexity holds for time-dependent policies.

**Theorem 4.3** *The time-dependent policy evaluation problem for partially-observable* $s$MDP *is* PSPACE-*complete.*

## 4.2 Policy existence problems for unobservable MDPs

Just as for the policy evaluation problems, the complexity of the policy existence problem makes an exponential jump from the flat to the succinct representation. The arguments used to prove the following results are similar to those used in the proofs of Theorem 3.5 and Theorem 3.6.

**Theorem 4.4** *The stationary policy existence problem for unobservable* $s$MDP *is* PSPACE-*complete.*

**Theorem 4.5** *The time-dependent policy existence problem for unobservable* $s$MDP *is* NEXP-*complete.*

# 5 Succinctly represented "small" POMDPs

A special case of succinctly represented POMDPs are those where the circuits representing the transition tables have many output gates from which on input $s, a, s'$ the transition probability $t(s, a, s')$ can be read. Because the number of output gates is bounded by the size of the circuit, for a POMDP with $n$ states, which is encoded by such circuits of size order of $\log n$, the transition probabilities can use only $\log n$ binary bits to be written down (in their standard representation); i.e. they can be represented by a small number of bits. Our completeness proof techniques from the last section do not apply for these small POMDPs. But we show that for succinctly represented small POMDPs we get completeness results intermediate between those of flat and succinct POMDPs, if we restrict the horizon to be logarithmic in the size of the state space. It turns out that the number of bits needed to represent the actions and rewards contributes to the complexity too. Since there are so many factors now, the problem descriptions are more involved.

The *$\alpha$ policy evaluation problem for partially-observable* $s_{\log}$MDP$[f(n)]$ *with $g(n)$ horizon* is the set of all pairs $(M, \pi)$ consisting of a partially-observable succinctly encoded MDP $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$

where each transition probability and reward takes $\log|\mathcal{S}|$ many bits and $|\mathcal{A}| \le f(|\mathcal{S}|)$, and an $\alpha$ policy $\pi$, such that $\mathrm{perf}(M, s_0, g(|\mathcal{S}|), \pi) > 0$.

The *$\alpha$ policy existence problem for unobservable $s_{\log}$MDP$[f(n)]$ with $g(n)$ horizon* is the set of all unobservable MDP $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$ where each transition probability and reward takes $\log|\mathcal{S}|$ many bits and $|\mathcal{A}| \le f(|\mathcal{S}|)$, such that $\mathrm{val}_\alpha(M, g(|\mathcal{S}|) > 0$.

If $f$ or $g$ is the identity function, we omit it in the problem description.

## 5.1 Policy evaluation problems

Remember that #P is the class of functions $f$ for which there exists a nondeterministic polynomial time bounded Turing machine $N$ such that $f(x)$ equals the number of accepting computations of $N$ on $x$, and that FP $\subseteq$ #P $\subseteq$ FPSPACE. The matrix powering problem for integer matrices with "small" entries is in #P, using the same proof idea as for Lemma 3.1.

**Lemma 5.1** *Let $T$ be a $2^n \times 2^n$ matrix of nonnegative integers, each consisting of $n$ bits, $1 \le i, j \le 2^n$, and $0 \le m \le n$. Let $T$ be represented by a Boolean circuit $C$ with $2n + \lceil \log n \rceil$ input bits, such that $C(a, b, r)$ outputs the $r$-th bit of $T_{(a,b)}$. The function mapping $(T, m, i, j)$ to $(T^m)_{(i,j)}$ is in* #P.

As a consequence, the complexity of the policy evaluation problems can be shown to be between NP and PSPACE.

**Theorem 5.2** *The stationary and time-dependent policy evaluation problems for partially-observable $s_{\log}$MDP$[\log n]$ and $\log n$ horizon are* PP*-complete.*

The proof is similar to that of Theorem 3.4.

## 5.2 Policy existence problems for unobservable MDPs

For the complexity of policy existence problems, the number of actions of the MDP is important. Remember that in the proof of Theorem 3.5, we used the fact that the policy existence problem disjunctively reduces to the policy evaluation problem in the way that for a given MDP $M$ we computed a set of pairs $(M, a)$ for every action $a$. If the set of actions is as large as the set of states, this reduction cannot be performed in polynomial time anymore.

**Theorem 5.3** *The stationary policy existence problem for unobservable $s_{\log}$MDP$[\log n]$ with $\log n$ horizon is* PP*-complete.*

**Proof** Hardness for PP follows from the hardness proof of Lemma 5.2. To show containment in PP, we use arguments similar to those in the proof of Lemma 3.2. We make use of the fact that $A \in$ PP if and only if there exists a GapP funtion $f$ such that for every $x$, $x \in A \leftrightarrow f(x) > 0$ (see [12]). For the function $p$ from the proof of Lemma 3.2, one can show that it is in GapP under these circumstances, because the respective matrix powering is in #P (Lemma 5.1), and GapP is closed under multiplication and summation. Finally, PP is closed under polynomial-time disjunctive reducibility, which completes the proof. ■

Omitting the restriction on the number of actions, the complexity of the problem rises to $\mathrm{NP}^{\mathrm{PP}}$.

**Theorem 5.4** *The stationary policy existence problem for unobservable $s_{\log}$MDP with $\log n$ horizon is $\le_m^p$-complete for* $\mathrm{NP}^{\mathrm{PP}}$.

Following a result from [24], $\mathrm{NP}^{\mathrm{PP}}$ equals the $\le_m^{np}$ closure of PP, which can be seen as the closure of PP under polynomial-time disjunctive reducibility with an exponential number of queries (each of the queries computable in polynomial time from its index in the list of queries). Using this, containment in and hardness for $\mathrm{NP}^{\mathrm{PP}}$ follows using a similar arguments as in the proof of Theorem 3.5.

If the horizon increases to the size of the state space (which may be exponential in the size of the MDP's representation), we get completeness for PSPACE. Note that the same problem for unobservable MDPs with nonnegative rewards is PSPACE-complete too (see Theorem 6.1).

**Theorem 5.5** *The stationary policy existence problem for unobservable $s_{\log}$MDP is* PSPACE*-complete.*

**Proof** Containment in PSPACE follows from Theorem 4.4. Note that in the proof of Theorem 4.2 a polynomially space bounded computation was transformed into a deterministic $s$MDP with nonnegative rewards and only one action. The same proof applies here too. ■

As a consequence we obtain that the policy existence problem for flat MDPs with exponential horizon is in PSPACE.

The complexity gap between the stationary and the time-dependent policy existence problems for $s_{\log}$MDP[$\log n$]s is as big as that for flat MDPs, but the difference no longer depends on the number of actions.

**Theorem 5.6** *The time-dependent policy existence problem for unobservable* $s_{\log}$MDP[$\log n$] *with* $\log n$ *horizon is* $\mathrm{NP}^{\mathrm{PP}}$*-complete.*

**Theorem 5.7** *The time-dependent policy existence problem for unobservable* $s_{\log}$MDP *with* $\log n$ *horizon is* $\mathrm{NP}^{\mathrm{PP}}$*-complete.*

# 6 POMDPs with nonnegative rewards

For POMDPs with nonnegative rewards, the policy evaluation and existence problems reduce in a straightforward way to graph accessibility problems. In order to determine whether an MDP has performance greater than 0 under a given policy, it suffices to find one "path" with positive probability through the given POMDP which is consistent with the given policy and yields reward $> 0$. Moreover, for succinctly represented POMDPs, the transition probabilities and rewards do not need to be calculated exactly. It suffices to find one positive transition probability and a corresponding positive reward.

This yields the following results.

**Theorem 6.1**
1. *The stationary and time-dependent policy evaluation problems for partially-observable MDPs (resp.* $s$MDP*s) with nonnegative rewards are complete for* NL *(resp.* PSPACE*).*
2. *The stationary and time-dependent policy existence problems for unobservable MDPs (resp.* $s$MDP*s) with nonnegative rewards are complete for* NL *(resp.* PSPACE*).*
3. *The stationary and time-dependent policy evaluation problems for partially-observable* $s$MDP*s with nonnegative rewards for horizon* $\log n$ *are complete for* NP*.*
4. *The stationary and time-dependent policy existence problems for unobservable* $s$MDP*s with nonnegative rewards for horizon* $\log n$ *are complete for* NP*.*


# Acknowledgements

We would like to thank Anne Condon, Chris Lusena, Matthew Levy, and especially Michael Littman for discussions and suggestions on this material.


# References


[1] E. Allender and M. Ogihara. Relationships among PL, #L, and the determinant. *Theoretical Informatics and Applications*, 30(1):1–21, 1996.

[2] C. Àlvarez and B. Jenner. A very hard log-space counting class. *Theoretical Computer Science*, 107:3–30, 1993.

[3] J.L. Balcázar. The complexity of searching implicit graphs. *Artificial Intelligence*, 86:171–188, 1996.

[4] J.L. Balcázar, A. Lozano, and J. Torán. The complexity of algorithmic problems on succinct instances. In R. Baeza-Yates and U. Manber, editors, *Computer Science*, pages 351–377. Plenum Press, 1992.

[5] D. Beauquier, D. Burago, and A. Slissenko. On the complexity of finite memory policies for Markov decision processes. In *Mathematical Foundations of Computer Science*, pages 191–200. Lecture Notes in Computer Science, Springer-Verlag, 1995.

[6] A. Borodin, S. Cook, and N. Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. Technical Report 162/83, University of Toronto, Department of Computer Science, 1983.

[7] C. Boutilier and D. Poole. Computing optimal policies for partially observable decision processes using compact representations, 1994.

[8] D. Burago, M. de Rougemont, and A. Slissenko. On the complexity of partially observed Markov decision processes. *Theoretical Computer Science*, 157(2):161–183, 1996.

[9] T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69:165–204, 1994.

[10] K. Erol, J. Hendler, and D. Nau. Complexity results for hierarchical task-network planning. *Annals of Mathematics and Artificial Intelligence*, 1996.

[11] K. Erol, D. Nau, and V. S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning. *Artificial Intelligence*, 76:75–88, 1995.

[12] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48(1):116–148, 1994.

[13] H. Galperin and A. Wigderson. Succinct representation of graphs. *Information and Control*, 56:183–198, 1983.

[14] J. Goldsmith, M. Littman, and M. Mundhenk. The complexity of plan existence and evaluation in probabilistic domains. Technical report, Duke University Department of Computer Science, 1997.

[15] H. Jung. On probabilistic time and space. In *Proceedings 12th ICALP*, pages 281–291. Lecture Notes in Computer Science, Springer-Verlag, 1985.

[16] R. Ladner. Polynomial space counting problems. *SIAM Journal on Computing*, 18:1087–1097, 1989.

[17] M.L. Littman. Probabilistic propositional planning: Representations and complexity. Technical Report CS-1996-18, Department of Computer Science, Duke University, 1996.

[18] M.L. Littman. Probabilistic STRIPS planning is EXPTIME-complete. Technical Report CS-1996-18, Duke University Department of Computer Science, November 1996.

[19] W.S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28:47–66, 1991.

[20] C.H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[21] C.H. Papadimitriou and J.N. Tsitsiklis. Intractable problems in control theory. *SIAM Journal of Control and Optimization*, pages 639–654, 1986.

[22] C.H. Papadimitriou and J.N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[23] M.L. Puterman. *Markov decision processes.* John Wiley & Sons, New York, 1994.

[24] J. Torán. Complexity classes defined by counting quantifiers. *Journal of the ACM*, 38(3):753–774, 1991.

[25] V. Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *Proc. 6th Structure in Complexity Theory Conference*, pages 270–284. IEEE, 1991.

[26] K. W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Informatica*, 23:325–356, 1986.

[27] R. Washington. Incremental Markov-model planning.
http://www.cs.washington.edu/homes/hanks/projects/prob-planning.html