DIMACS Technical Report 97-47
September 1997

Encyclopaedia of Complexity Results for Finite-Horizon Markov Decision Process Problems[1]

by

Martin Mundhenk [2]
Universität Trier
FB IV – Informatik
D-54286 Trier, Germany
`mundhenk@ti.uni-trier.de`

Judy Goldsmith [3]
Dept. of Computer Science
University of Kentucky
Lexington KY 40506-0046
`goldsmit@cs.engr.uky.edu`

Christopher Lusena [4]
Dept. of Computer Science
University of Kentucky
Lexington KY 40506-0046
`lusena@cs.engr.uky.edu`

Eric Allender[5,6]
Department of Computer Science
Rutgers University
Piscataway, NJ 08855, USA
`allender@cs.rutgers.edu`

[1]A preliminary version of some of this work appeared as [27].

[2]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.

[3]Supported in part by NSF grant CCR-9315354 and CCR-9610348.

[4]Supported in part by NSF grant CCR-9315354 and CCR-9610348.

[5]Permanent Member

[6]Supported in part by NSF grant CCR-9509603. Portions of this work were performed while at the Institute of Mathematical Sciences, Chennai (Madras), India, and at the Wilhelm-Schickard Institut für Informatik, Universität Tübingen (supported by DFG grant TU 7/117-1).

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs, Bellcore, and Bell Labs.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

The computational complexity of finite horizon policy evaluation and policy existence problems are studied for several policy types and representations of Markov decision processes. In almost all cases, the problems are shown to be complete for their complexity classes; classes range from nondeterministic logarithmic space and probabilistic logarithmic space (highly parallelizable classes) to exponential space. In many cases, this work *shows* that problems that already were widely believed to be hard to compute are probably intractable (complete for NP, $\mathrm{NP}^{\mathrm{PP}}$, or PSPACE), or provably intractable (EXPTIME-complete or worse). The major contributions of the paper are to pinpoint the complexity of these problems; to isolate the factors that make these problems computationally complex; to show that even problems such as median-policy or average-policy evaluation may be intractable; and the introduction of natural $\mathrm{NP}^{\mathrm{PP}}$-complete problems.

1 Introduction

Markov decision processes are used in the natural and social sciences and engineering to model a vast array of systems. (Puterman's book [32] has excellent discussions of applications.) They are used in a variety of applied computer science fields, including control theory, robotics, expert systems, learning theory, and data mining to model assembly lines, robot location problems, medical diagnosis and treatment experts, and many other systems.

Markov Decision Processes (MDPs) model controlled stochastic processes. Such a process has states, an external agent or controller that can choose actions (here assumed to be chosen at discrete time intervals) which have probabilistic effects, and rewards associated with each state-action pair. The process is assumed to run (in the cases considered here) for a fixed finite time, approximately equal to the size of the system or its representation.

There are several computational questions associated with the use of MDPs. We do not address the questions that arise in designing MDP models of stochastic processes. Rather, we address the computational complexity, given an MDP and specifications for a controller, of finding a good or optimal controller. Goodness is measured in terms of the expected reward, and the specifications on the system and the controller include the amount of feedback that the controller gets from the system (observability), the length of time that the process will run (horizon), the amount of memory the controller has, and the form of the input and output. Is the system fully specified, or is it represented by some sort of function (here, circuits) that computes the probabilities as needed? Similarly for the controller itself.

For instance, partially observable MDPs (POMDPs) model everything from robot location problems to medical treatment [38]. Unfortunately, the optimal policies for POMDPs are generally expressible as functions from initial segments of the history of a process (a series of observations, giving partial or full information about the state at previous and current times, and — implicitly — of actions taken). This is unfortunate because (as practitioners already suspected, and we prove), the problem of finding optimal policies for POMDPs is PSPACE-hard.

In this paper, we primarily consider the *policy evaluation problem* (given an MDP and a policy, what is the expected reward accrued by that policy over the given horizon), and the *policy existence problem* (given an MDP, is there a policy with expected reward greater than 0).

Linear programming has been used to decide these two problems for certain forms of MDPs and certain types of policies since 1960 [15]; in 1987, Papadimitriou and Tsitsiklis [30] showed that in fact certain policy existence problems in these cases are P-complete. The complexity of these problems is quite sensitive to details in the specification. We show that a seemingly similar problem, based on maximization rather than minimization, is complete for nondeterministic logspace (Corollary 5.2), whereas a more analogous

	Evaluation	Existence	Evaluation (nonneg)	Existence (nonneg)
Partially-observable				
Stationary	PL 4.5	NP 6.4	NL 5.1	NP 5.4
Time-dependent	PL 4.5	NP 6.5	NL 5.1	NL 5.3
History-dependent		PSPACE 6.6		NL 5.3
Fully-observable				
Stationary	PL 4.5	?	NL 5.1	NL 5.2
Time-dependent	PL 4.5	P 6.1	NL 5.1	NL 5.2
Unobservable				
Stationary	PL 4.5	PL 6.2	NL 5.1	NL 5.2
Time-dependent	PL 4.5	NP 6.3	NL 5.1	NL 5.2

Table 1: Completeness results for Flat MDPs

P-completeness result can be found in Theorem 6.1.

In most other cases, however, we show that the decision problems considered here are complete for classes thought or known to be above P, such as NP, PSPACE, and even as high as EXPSPACE. Thus, the contribution this paper makes to the fields of control theory, economics, medicine, etc., are largely negative: there are unlikely to be any efficient algorithms for finding *the optimal* policies for the general MDP problems.

However, there is good news in this bad news. We *prove* what many practitioners already suspected. This gives strength to the call for approximations, and the search for special cases that are in fact computationally simpler than the general case.

For instance, there is interest in the AI community in so-called "structured," or succinctly represented MDPs [10, 13, 25]. These representations arise when one can make use of structures in the state space to provide small descriptions of very large systems [10]. Whereas those systems are not tractable by classical methods, there is some hope – expressed in different algorithmic approaches – that many special cases of these structured POMDPs can be solved more efficiently. Boutillier, et al., in [10] conjecture that finding optimal policies for structured POMDPs is infeasible. We prove this conjecture by showing that in many cases the complexity of our decision problems increases exponentially if succinct descriptions [19, 37] for POMDPs are considered. For example, policy existence problems for POMDPs with nonnegative rewards are NL-complete under straightforward descriptions. Using succinct descriptions, the completeness increases to PSPACE. We also consider a new intermediate notion of compressed representations and get intermediate complexity results, e.g. NP-completeness for the above example.

The results of this paper are summarized in Tables 1–3. The numbers next to complexity classes indicate the relevant theorems. The history-dependent cases for fully-observable and unobservable MDPs have been omitted, since those cases are identical to the time-dependent ones.

	Evaluation	Existence	Evaluation (nonneg)	Existence (nonneg)
Partially-observable				
Stationary	PSPACE 4.7	NEXP 6.8	PSPACE 5.5; NP 5.6*	NEXP 5.10
Time-dependent	PSPACE 4.8	NEXP 6.8	PSPACE 5.5; NP 5.6	PSPACE 5.11
History-dependent		EXPSPACE 6.8		PSPACE 5.11
Fully-observable				
Stationary	PSPACE 4.7	?		PSPACE 5.9
Time-dependent	PSPACE 4.8	EXP 6.7		PSPACE 5.9
Unobservable				
Stationary	PSPACE 4.7	PSPACE 6.9	NP 5.7	PSPACE 5.8
Time-dependent	PSPACE 4.8	NEXP 6.10	NP 5.7	PSPACE 5.8

Table 2: Completeness results for Succinct MDPs

* The difference depends on whether the horizon is polynomial in the size of the input or in the number of states.

	Evaluation	Existence	Evaluation (nonneg)	Existence (nonneg)
Partially-observable				
Stationary	$\mathrm{NP}^{\mathrm{PP}}$ 6.18; PP 4.10*	NEXP 6.11		NP 5.14
Time-dependent	PP 4.10	NEXP 6.11		NP 5.14
History-dependent		PSPACE 6.15		
Fully-observable				
Stationary	PP 4.10	PSPACE 6.14		NP 5.12
Time-dependent	PP 4.10	PSPACE 6.14		NP 5.12
Unobservable				
Stationary	PP 4.10	PSPACE 6.20		NP 5.13
Time-dependent	PP 4.10	$\mathrm{NP}^{\mathrm{PP}}$ 6.21		NP 5.13

Table 3: Completeness results for Compressed MDPs

* The differences here depend on $|\mathcal{A}|$.

Given the widespread use and consideration of MDPs, there has been surprisingly little work on the computational complexity of finding good policies. There have certainly been algorithms presented and analyzed, but we will not generally survey them here. For that, we recommend [26, 32] as a starting place.

The complexity of partially-observable MDPs was first considered by Papadimitriou and Tsitsiklis [29, 30]. They considered POMDPs with nonpositive rewards ("costs") and investigated the complexity of the decision problem: given a POMDP, does there exist a time-dependent or history-dependent policy with expected sum of rewards *equal to* 0? Their maximization problem can be stated as a minimization problem for POMDPs with nonnegative rewards.

They showed an NP-completeness result for the stationary, finite-horizon case [29, p. 645]. For time-dependent policies they showed that the policy existence problem for fully-observable MDPs is P-complete (for finite or infinite horizon). 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 [30].

Note that the minimization problem investigated by Papadimitriou and Tsitsiklis cannot be used in a binary search to determine the maximal performance of a POMDP. (The use of such a search depends on both positive and negative rewards.) Our work extends theirs significantly, both in the problems considered, and in the scope of the results. Because of the subtle differences between the decision problems, our complexity results on similar problems differ greatly from the results in [30]. For example, our decision problem for POMDPs with nonnegative rewards under history-dependent policy is NL-complete. (Cf. Corollary 5.3, compare to PSPACE-completeness in [30, Theorem 6].) Unobservable MDPs were also considered in [12]. Beauquier et al. [6] considered different optimality criteria, and thus their results are not directly comparable with ours.

Most of the related AI papers, especially [3, 13, 14, 16, 18], consider computationally simpler problems without probabilistic transitions. Thus, the complexity of their problems is generally lower than of ours (except [20, 24] and one theorem in [13]). Many of the results are for succinctly represented systems, described by 2-Phase Temporal Bayes Nets [9, 10], sequential-effects-tree representation (ST) [25], probabilistic state-space operators (PSOs) [22], or other related representations. For these problems, they consider the complexity of finding policies that are of size polynomial in either the size of the representation, or the number of states — leading to different complexities, in certain cases. (For instance, Littman showed that the general plan existence problem is EXP-complete, but if plans are limited to size polynomial in the size of the planning domain, then the problem is PSPACE-complete [25]; the PSPACE-completeness was proved independently in [20].) Sometimes the stochasticity of the system does not add computational complexity. For instance, Bylander [13] showed that the time-dependent policy existence problem for unobservable, succinct MDPs with nonnegative rewards is PSPACE-complete, even if the transitions are restricted to be deterministic. In Theorem 5.8 we show PSPACE-completeness of the stochastic version of this problem.

Definitions of Markov decision processes and policies appear in Section 2. The following section addresses computational issues, especially the representation of MDPs. In each of the next three sections, we consider flat, then succinctly represented, then compressed MDPs: Section 4 presents the complexity of policy evaluation problems for general MDPs; Section 5 concerns both policy evaluation and policy existence problems for MDPs with nonnegative rewards; Section 6 is on policy existence problems for general MDPs. In Section 7, we also consider the question, does a given MDP have a policy with expected reward *exactly* r? Section 8 considers the complexity of testing the median and average policies of an MDP.

2 Definitions and Preliminaries

2.1 Complexity Theory: Counting Classes

For definitions of complexity classes, reductions, and standard results from complexity theory we refer to [28]. In the interest of completeness, in this section we give a short description of the probabilistic and counting complexity classes we use in this work.

The class #L [2] 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 #P is defined analogously as 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 $f(x)$.

Probabilistic logspace, PL, is 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 are decidable using very fast parallel computations [21]. Probabilistic polynomial time, PP, is defined analogously. A classic PP-complete problem is MAJSAT: given a Boolean formula, do the majority of assignments satisfy it?

For polynomial-space-bounded computations, PSPACE equals probabilistic PSPACE, and #PSPACE (defined analogously to #L and #P) is the same as the class of polynomial-space-computable functions [23]. (Note that some functions in #PSPACE produce output of exponential length.)

For any complexity classes $\mathcal{C}$ and $\mathcal{C}'$ the class $\mathcal{C}^{\mathcal{C}'}$ consists of those sets that are $\mathcal{C}$-*Turing reducible* to sets in $\mathcal{C}'$, i.e., sets that can be accepted with resource bounds specified by $\mathcal{C}$, using some problem in $\mathcal{C}'$ as a subroutine (oracle) with instantaneous output. For any class $\mathcal{C} \subseteq$ PSPACE, it is the case that $\mathrm{NP}^{\mathcal{C}} \subseteq$PSPACE, and therefore $\mathrm{NP}^{\mathrm{PSPACE}}$=PSPACE; see Papadimitriou's textbook [28].

Another useful result is that the class $\mathrm{NP}^{\mathrm{PP}}$ equals the "$\leq_m^{\mathrm{NP}}$" closure of PP [34], 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).

2.2 Markov Decision Processes

A Markov decision process (MDP) 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*,
- $t : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ is the *state transition function*, where $t(s,a,s')$ is the probability that state s' is reached from state s on action a (where $\Sigma_{s' \in \mathcal{S}} t(s,a,s') \in \{0,1\}$ for every $s \in \mathcal{S}, a \in \mathcal{A}$),
- $o : \mathcal{S} \rightarrow \mathcal{O}$ is the *observation function*, where $o(s)$ is the observation made in state s,
- $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbf{Z}$ is the *reward function*, where $r(s,a)$ is the reward gained by taking action a in state s.

If states and observations are identical, i.e. $\mathcal{O} = \mathcal{S}$ and o is the identity function (or a bijection), then the MDP is called *fully-observable*. Another special case is *unobservable MDPs*, where the set of observations contains only one element, i.e. in every state the same observation is made and therefore the observation function is constant. Without restrictions on the observability, an MDP is called *partially-observable*.[1]

2.3 Policies and Performances

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

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

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

- A *history-dependent policy* π_h is a function $\pi_h : \mathcal{O}^* \to \mathcal{A}$, mapping each finite sequence of observations to an action.

Notice that, for an unobservable MDP, a history-dependent policy is equivalent to a time-dependent one. Thus, our theorems do not address the case of history-dependent policies for unobservable MDPs.

Let $M = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{O}, t, o, r)$ be an MDP. A *trajectory θ 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 $\mathrm{prob}_\pi(M, \theta)$ of a trajectory $\theta = \sigma_1, \sigma_2, \ldots, \sigma_m$ under policy π is

- $\mathrm{prob}_\pi(M,\theta) = \Pi_{i=1}^{m-1} t(\sigma_i, \pi(o(\sigma_i)), \sigma_{i+1})$, if π is a stationary policy,
- $\mathrm{prob}_\pi(M,\theta) = \Pi_{i=1}^{m-1} t(\sigma_i, \pi(o(\sigma_i), i), \sigma_{i+1})$, if π is a time-dependent policy, and
- $\mathrm{prob}_\pi(M,\theta) = \Pi_{i=1}^{m-1} t(\sigma_i, \pi(o(\sigma_1) \cdots o(\sigma_i)), \sigma_{i+1})$, if π is a history-dependent policy.

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

For simplicity, we assume that the size of an MDP is determined by the size n 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 n bits and each reward is an integer of at most n 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 an MDP M with non-integer rewards to M' with integer rewards such that $\mathrm{val}_\alpha(M,k) = c \cdot \mathrm{val}_\alpha(M',k)$ for some constant c.

3 Representations and Computational Questions

3.1 Flat MDPs

There are various ways an MDP can be represented. We begin by considering problem instances that are 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. We assume that the number of actions and the number of bits needed to store each transition probability or reward does not exceed n, so 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 *α policy evaluation problem for MDPs* is the set of all pairs (M, π) consisting of a POMDP $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$ and an α policy π with performance $\mathrm{perf}(M, s_0, |\mathcal{S}|, \pi) > 0$.

The *α policy existence problem for MDPs* is the set of all MDPs $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$, for which there exists an α policy π with performance $\mathrm{perf}(M, s_0, |\mathcal{S}|, \pi) > 0$.

Notice that there are $|\mathcal{O}|^h$ possible histories for an MDP with time horizon h. If $|\mathcal{O}| > 1$, this means that one cannot fully specify a history-dependent policy in space polynomial in the size of $\mathcal{S}$. Therefore, we do not generally consider the history-dependent policy evaluation problem in this paper. Since we generally assume that $h \leq |\mathcal{S}|$, time-dependent policies do not cause such problems.

3.2 Succinct MDPs

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 [13, 20, 24].

Changing the way in which MDPs (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 [19, 37]. 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 $\mathcal{O}(\log n)$, i.e., the logarithm of the size 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 [5] for conditions). For most of the problems considered here, the completeness proofs for the flat POMDPs translate upward to the respective succinctly represented POMDPs.

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

The *α policy existence problem for partially-observable sMDPs* is the set of all succinctly represented POMDPs $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$, for which there exists an α policy

π, such that $\text{perf}(M, s_0, |\mathcal{S}|, \pi) > 0$.

3.3 Compressed MDPs

In this section, we introduce what we believe is a better model of the type of "succinct" MDPs that are of interest in practice. This model, which we call "compressed MDPs," is intermediate between the flat representation and the succinct representation introduced in the preceding subsection. Note for instance that the transition probabilities $t(s, a, s')$ in the "succinct" representation potentially have exponentially-many bits of precision. It is hard to think of situations when this amount of precision would be known, and thus it is not a serious restriction to consider only succinctly represented MDPs where the circuits representing the transition tables have multiple output gates from which on input s, a, s' the transition probability $t(s, a, s')$ can be read. (Thus, in particular, only polynomially-many bits of precision are considered. That is, because the number of output gates is bounded by the size of the circuit, for an MDP with n states, encoded by a circuit of size $c \log n$, the transition probabilities must be written using roughly $\log n$ bits (in their standard representation).) The issue of bit-counts in transition probabilities has arisen before; it occurs, for instance, in [6, 35]. It is also important to note that our probabilities are specified by single bit-strings, rather than as rationals specified by two bit-strings.

Since the "succinct" MDPs considered in practice frequently arise in the context of planning, it is much more likely that one will be interested in the reward after only a polynomial number of steps (since there will not be time to run the policy for an exponential number of steps). This motivates restricting 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, the problem descriptions are more involved than in the flat or succinct cases.

The *α policy evaluation problem for partially-observable compressed MDP[$f(n)$] with $g(n)$ horizon* is the set of all pairs (M, π) 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 $|\mathcal{A}| \leq f(|\mathcal{S}|)$, and an α policy π, such that $\text{perf}(M, s_0, g(|\mathcal{S}|), \pi) > 0$.

The policy existence problems are expressed similarly.

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

4 Policy Evaluation Problems

4.1 Flat MDPs

The standard polynomial time algorithm for evaluating a given policy for a given POMDP uses dynamic programming [30, 32]. 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 lemma about matrix powering, and show that each entry $(T^m)_{(i,j)}$ of the mth 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 4.1 *(cf. [36]) 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 4.2 *The stationary policy evaluation problem for partially-observable MDPs 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 π 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 π. Since π is a stationary policy, this can be achieved by "hard-wiring" the actions chosen by π 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 π 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 4.1 we get that $(T^{k-1})_{(i,j)} \in \#\mathrm{L}$. 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 $p \in \mathrm{GapL}$. □

Since unobservable and fully observable MDPs are a special case of POMDPs, we get the following corollary.

Corollary 4.3 *The stationary policy evaluation problems for unobservable and fully observable MDPs are in* PL*.*

Lemma 4.4 *The stationary policy evaluation problem for unobservable MDPs is* PL*-hard.*

Proof Consider $A \in \mathrm{PL}$. 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 [21]. Now, fix some input x. We construct an unobservable MDP $M(x)$ with only one action, which 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 a counter for the number of random moves 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 θ 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 θ of length $|\mathcal{S}|$ for $M(x)$ with $\mathrm{rew}(\theta)\cdot\mathrm{prob}(\theta) = 1$ is greater than the number of trajectories θ 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, and also for partially and fully observable MDPs.

Theorem 4.5 *The stationary and time-dependent policy evaluation problems for partially-observable, fully observable, and unobservable MDPs are* PL*-complete.*

Proof Completeness of the stationary case follows from the above Lemmas 4.2 and 4.4. Because every stationary policy is a time-dependent policy too, hardness of the time-dependent case follows from Lemma 4.4. 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 π 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 ith copy go to the $i+1$st copy, and all these transitions correspond to the transition chosen by π in the ith step. The rest of the proof proceeds as in the proof of Lemma 4.2. □

4.2 Succinct MDPs

Put succinctly, all results from Section 4.1 translate to succinctly represented MDPs. The first result which we translate is Lemma 4.1. In Lemma 4.1, we considered the problem of matrix powering for $m \times m$ matrices; here, we consider powering a succinctly-represented matrix of size $\mathcal{O}(2^m)$. In space logarithmic in the size of the matrix, we can apply the the algorithm posited in Lemma 4.1; since the input is size $\mathcal{O}(m)$, this means that the problem is in #PSPACE.

Lemma 4.6 *Let T be a $2^m \times 2^m$ matrix of nonnegative integers, each consisting of 2^m bits, $1 \leq i, j \leq 2^m$, and $0 \leq s \leq m$. Let T be represented by a Boolean circuit C with $2m + \lceil \log m \rceil$ input bits, such that $C(a, b, r)$ outputs the r-th bit of $T_{(a,b)}$. The function mapping (T, s, i, j) to $(T^s)_{(i,j)}$ is in* #PSPACE.

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

Theorem 4.7 *The stationary policy evaluation problems for unobservable, fully, and partially observable sMDPs are* 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 4.2 yielding here that the problem is in PPSPACE (=probabilistic PSPACE). Ladner [23] 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 4.4, 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 4.8 *The time-dependent policy evaluation problem for unobservable, fully, and partially observable sMDPs are* PSPACE*-complete.*

4.3 Compressed MDPs

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 $\mathrm{FP} \subseteq \#\mathrm{P} \subseteq \mathrm{FPSPACE}$. The matrix powering problem for integer matrices with "small" entries is in #P, using the same proof idea as for Lemma 4.1.

Lemma 4.9 *Let T be a $2^m \times 2^m$ matrix of nonnegative integers, each consisting of m bits, $1 \leq i, j \leq 2^m$, and $0 \leq s \leq m$. Let T be represented by a Boolean circuit C with $2m + \lceil \log m \rceil$ input bits, such that $C(a, b, r)$ outputs the r-th bit of $T_{(a,b)}$. The function mapping (T, s, i, j) to $(T^s)_{(i,j)}$ is in* #P.

As a consequence, the complexity of the policy evaluation problems can be shown to be between NP and PSPACE. (Remember that n represents the size of the state space, not the size of the input.)

Theorem 4.10 *The stationary and time-dependent policy evaluation problems for unobservable, fully, and partially-observable compressed MDP*$[\log n]$ *and* $\log n$ *horizon are* PP*-complete.*

The proof is similar to that of Theorem 4.5.

5 MDPs with Nonnegative Rewards

The policy evaluation and existence problems for MDPs with nonnegative rewards are generally easier than for MDPs with unrestricted rewards. In fact, these 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 that is consistent with the given policy and yields reward > 0. Moreover, the transition probabilities and rewards do not need to be calculated exactly.

Note that earlier work, for instance Papadimitriou and Tsitsiklis' [30] considered a different problem than we do here, with strikingly different complexity. They considered MDPs with nonpositive rewards, and asked whether there was a policy with expected reward 0. In order for such a policy to exist, there must be *no* trajectory consistent with that policy that has non-zero reward. Thus, their question has a "for all" flavor to it, whereas ours has a "there exists" flavor; in order for one of our policies to have expected positive reward (when there are no negative rewards), there must be *some* trajectory consistent with that policy that has positive expected reward.

5.1 Flat MDPs

Theorem 5.1 *The stationary and time-dependent policy evaluation problems for partially-observable, fully-observable, and unobservable MDPs with nonnegative rewards are complete for* NL.

Proof An MDP M (any observability) and a stationary policy π can be interpreted as a directed graph: the vertices are states of the MDP, and an edge exists from s_i to s_j iff $t(s_i, \pi(o(s_i)), s_j > 0$. Then $\text{perf}(M, \pi) > 0$ iff the set of vertices $R = \{s_i : r(s_i, \pi(o(s_i))) > 0\}$ is reachable from s_0. If π is time-dependent, then the vertices of the graph are state-time pairs.

Given a graph reachability problem, it can easily be interpreted as a existence problem for an MDP with only one action (all edges from a given vertex have equal probability). The MDP has only one policy, and that policy has positive expected value iff the sink node is reachable in the original graph. Since there is only one possible policy, observability is irrelevant. □

Corollary 5.2 *The stationary, time-dependent and history-dependent policy existence problems for fully-observable or unobservable MDPs with nonnegative rewards are* NL-*complete.*

Proof For fully-observable or unobservable MDPs with nonnegative rewards the existence problem is as easy as the evaluation problem. As we've seen, the expected reward for a policy π is greater than 0 if and only if at least one trajectory $\sigma_0, \ldots, \sigma_m$ consistent with π exists for which the action chosen by π for σ_m yields a reward greater than 0. It is clear that each such π can be changed to a policy for which there exists a consistent trajectory with positive reward, where no state appears twice in that trajectory. The latter bounds the length of the trajectories to be considered by the number of the states of the MDP. Therefore, the policy existence problems can be shown to be equivalent to graph reachability problems, from which we can conclude NL-completeness. □

Corollary 5.3 *The time-dependent and history-dependent policy existence problems for partially-observable MDPs with nonnegative rewards are* NL-*complete.*

For POMDPs the same argument works for time- or history-dependent policy existence problems. In order to create a consistent trajectory, at any step of the process the next action can be chosen without taking former observations and actions into account; each partial trajectory has its own unique history.

For stationary policies, the same action must be chosen whenever the same observation is made. Logspace is not sufficient to handle this problem. On the other hand, we can use observations to store information. This allows us to express NP-complete problems.

Theorem 5.4 *The stationary policy existence problem for POMDPs with nonnegative rewards is* NP*-complete.*

Proof Membership in NP is straightforward, because a policy can be guessed and evaluated in polynomial time. To show NP-hardness, we reduce the NP-complete satisfiability problem ϕ to it. Let $\phi(x_1,\ldots,x_n)$ be such a formula with variables $x_1,\ldots,x_n$ and clauses $C_1,\ldots,C_m$, where clause $C_j = (l_{v(1,j)} \vee l_{v(2,j)} \vee l_{v(3,j)})$ for $l_i \in \{x_i, \neg x_i\}$. We say that variable x_i *appears in* C_j *with signum* 0 *(resp.* 1*)* if $\neg x_i$ (resp. x_i) is a literal in C_j. Without loss of generality, we assume that every variable appears at most once in each clause. The idea is to construct an POMDP $M(\phi)$ having one state for each appearance of a variable in a clause. The set of observations is the set of variables. Each action corresponds to an assignment of a value to a variable. The transition function is deterministic. The process starts with the first variable in the first clause. If the action chosen in a certain state satisfies the corresponding literal, the process proceeds to the first variable of the next clause, or with reward 1 to the final state, if all clauses were considered. If the action does not satisfy the literal, the process proceeds to the next variable of the clause, or with reward 0 to the final state. The partition of the state space into observation classes guarantees that the same assignment is made for every appearance of the same variable. Therefore, the value of $M(\phi)$ equals 1 iff ϕ is satisfiable.

Formally, from ϕ, we construct a POMDP $M(\phi) = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{O}, t, o, r)$ with

$$
\begin{aligned}
\mathcal{S} &= \{(i,j) \mid 1 \le i \le n, 1 \le j \le m\} \cup \{F, T\} \\
s_0 &= (v(1,1),1), \qquad \mathcal{A} \;=\; \{0,1\}, \qquad \mathcal{O} \;=\; \{x_1,\ldots,x_n,F,T\} \\
t(s,a,s') &= \begin{cases}
1, & \text{if } s=(v(i,j),j), s'=(1,j+1), j<m, 1\le i\le 3, \\
 & \qquad\qquad \text{and } x_{v(i,j)} \text{ appears in } C_j \text{ with signum } a \\
1, & \text{if } s=(v(i,m),m), s'=T, 1\le i\le 3, \\
 & \qquad\qquad \text{and } x_{v(i,m)} \text{ appears in } C_m \text{ with signum } a \\
1, & \text{if } s=(v(i,j),j), s'=(v(i+1,j),j), 1\le i<3, \\
 & \qquad\qquad \text{and } x_{v(i,j)} \text{ appears in } C_j \text{ with signum } 1-a \\
1, & \text{if } s=(v(3,j),j), s'=F, \\
 & \qquad\qquad \text{and } x_{v(3,j)} \text{ appears in } C_j \text{ with signum } 1-a \\
1, & \text{if } s=s'=F \text{ or } s=s'=T \\
0, & \text{otherwise}
\end{cases} \\
r(s,a) &= \begin{cases} 1, & \text{if } t(s,a,T)=1, s\ne T \\ 0, & \text{otherwise} \end{cases}, \qquad o(s) \;=\; \begin{cases} x_i, & \text{if } s=(i,j) \\ T, & \text{if } s=T \\ F, & \text{if } s=F \end{cases}.
\end{aligned}
$$

Note that all transitions in $M(\phi)$ are deterministic, and every trajectory has value 0 or 1. There is a correspondence between policies for $M(\phi)$ and assignments of values to the variables of ϕ, such that policies under which $M(\phi)$ has value 1 correspond to satisfying assignments for ϕ, and vice versa. $\square$

5.2 Succinct MDPs with nonnegative rewards

Note that we take our finite horizon length relative to the size of the state space, which may be exponential in the size of the representation of the POMDP. The following theorems follow from our observations on the relationship of policy evaluation problems to graph reachability, and facts about succinctly represented graphs.

Theorem 5.5 *The stationary and time-dependent policy evaluation problems for partially-observable sMDPs with nonnegative rewards are* PSPACE*-complete.*

The proof follows from that of Theorem 5.1.

Theorem 5.6 *The* $\log n$*-horizon stationary and time-dependent policy evaluation problems for partially-observable sMDPs with nonnegative rewards are* NP*-complete.*

Proof Notice that the horizon is roughly the size of the input, so that a trajectory can be guessed in polynomial time, and then checked for consistency, positive probability, and positive reward.

The proof of NP-hardness is a reduction from SAT. Given a Boolean formula ϕ with v variables, we create a partially-observable MDP with $2^v + 1$ states and one action, a. The state s_0 represents the null assignment, and each of the next 2^v states represents an assignment; $t(s_0, a, s_i)$ is either $1/2^v$, if $0 < i < 2^v + 2$, or 0 otherwise. The reward $r(s_i, a)$ equals 1 iff the i^{th} truth assignment satisfies ϕ. The unique policy π has expected reward > 0 iff there is a satisfying assignment for ϕ. □

Corollary 5.7 *The stationary and time-dependent policy evaluation problems for unobservable sMDPs with nonnegative rewards for horizon* $\log n$ *are complete for* NP.

Because the policy existence problem is so close to the the policy evaluation problem in the absence of both positive and negative rewards, we get the following results.

Theorem 5.8 *The stationary and time-dependent policy existence problems for unobservable sMDPs with nonnegative rewards are complete for* PSPACE.

This also follows from Corollary 5.2.

Theorem 5.9 *The policy existence problems for stationary, time-dependent, or history-dependent policies for fully-observable sMDPs with nonnegative rewards are* PSPACE*-complete.*

This follows from Corollary 5.2, or alternatively from Theorem 5.5.

Theorem 5.10 *The stationary policy existence problem for succinct POMDPs with nonnegative rewards is* NEXP*-complete.*

Proof One can use a succinct 3SAT input like that in [28, pp 493–5] to construct a succinct version of the POMDP used in the reduction for Theorem 5.4. □

Theorem 5.11 *The time- or history-dependent policy existence problems for succinct POMDPs with nonnegative rewards are* PSPACE*-complete.*

This follows from Corollary 5.3.

5.3 Compressed MDPs with nonnegative rewards

Theorem 5.12 *The* $\log n$*-horizon stationary, time-dependent and history-dependent policy existence problems for fully-observable compressed MDPs with nonnegative rewards are* NP*-complete.*

Proof Membership in NP follows using similar arguments as those used for Theorem 5.2. Hardness for NP follows from the fact that an NP computation tree can be represented as an MDP with only one action, where nondeterministic computation steps are probabilistic transitions between states. Reaching an accepting state yields a positive reward. □

Examination of the proof shows that it carries over in the unobservable case, since the MDP has only one action.

Corollary 5.13 *The* $\log n$*-horizon stationary and time-dependent policy existence problems for compressed unobservable MDPs with nonnegative rewards are* NP*-complete.*

The argument used for Theorem 5.6 works for compressed POMDPs with nonnegative rewards as well as for succinct POMDPs.

Theorem 5.14 *The* $\log n$*-horizon stationary, time-dependent and history-dependent policy existence problems for compressed POMDPs with nonnegative rewards are* NP*-complete.*

Proof NP-hardness follows from Theorem 5.12, (since fully-observable MDPs are POMDPs), or alternatively from the fact that the reduction in the proof of Theorem 5.6 was in fact to a compressed POMDP. Membership in NP follows as in the proof of Theorem 5.6. □

6 Policy Existence Problems

6.1 Flat MDPs

To decide the performance of a policy for a POMDP with positive and negative rewards, a full "tree" of trajectories has to be evaluated, instead of one trajectory, and the weighted average of their values computed.

The computation of optimal policies for fully-observable MDPs is a well-studied optimization problem. The maximal performance of any infinite-horizon stationary policy for a fully-observable Markov decision process can be solved by linear programming techniques in polynomial time. Dynamic programming can be used for the finite horizon time-dependent policy case.

Theorem 6.1 *The time-dependent and history-dependent policy existence problems for fully-observable MDPs are* P*-complete.*

Because the proof is a straightforward modification of a proof in [30], we omit it here.

The exact complexity of the policy existence problem for stationary, fully-observable MDPs with finite horizon is not known. From [30] it follows that it is P-hard, and it is easily seen to be in NP. But completeness results are not known.

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 stationary and time-dependent policies.

Theorem 6.2 *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 than 0 if and only if for some $a \in \mathcal{A}$, M under a has performance greater than 0. Thus the policy-existence problem for stationary unobservable MDPs logspace disjunctively reduces to the stationary policy evaluation problem for unobservable MDPs. From Lemma 4.2 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 4.4, every PL computation was logspace reduced to an unobservable MDP with one action. This proves PL-hardness of the policy existence problem. □

The time-dependent policy existence problem for unobservable MDPs turns out to be NP-complete.

Theorem 6.3 *The time-dependent policy existence problem for unobservable MDPs is* NP*-complete.*

Papadimitriou and Tsitsiklis proved a similar theorem [30]. 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.

Proof That it is in NP follows from the fact that a policy with performance > 0 can be guessed and checked in polynomial time. NP-hardness follows from the following reduction from 3SAT. At the first step, a clause is chosen randomly. At step $i+1$, the assignment of variable i is determined. Because the process is unobservable, it is guaranteed that each variable gets the same assignment in all clauses. If a clause was satisfied by this assignment, it will gain reward 1, if not, the reward will be $-m$, where m is the number of clauses of the formula. Therefore, if all clauses are satisfied, the time-dependent value of the MDP is positive, otherwise negative.

We formally define the reduction. Let ϕ be a formula with n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$. We say that x_i appears in C_j with signum 1, if $x_i \in C_j$, and with signum 0, if $\neg x_i \in C_j$. Define the unobservable MDP $M(\phi) = (\mathcal{S}, s_0, \mathcal{A}, t, r)$ where

$$
\begin{aligned}
\mathcal{S} &= \{(i,j) \mid 1 \le i \le n, 1 \le j \le m\} \cup \{s_0, T, F\} \\
\mathcal{A} &= \{0,1\} \\
t(s,a,s') &= \begin{cases} \frac{1}{m}, & \text{if } s = s_0, a = 0, s' = (1,j), 1 \le j \le m \\ 1, & \text{if } s = (i,j), s' = T,\ x_i \text{ appears in } C_j \text{ with signum } a \\ 1, & \text{if } s = (i,j), s' = (i+1,j), i < n,\ x_i \text{ doesn't appear in } C_j \text{ with signum } a \\ 1, & \text{if } s = (n,j), s' = F,\ x_n \text{ doesn't appear in } C_j \text{ with signum } a \\ 1, & \text{if } s = s' = F \text{ or } s = s' = T, a = 0 \text{ or } a = 1 \\ 0, & \text{otherwise} \end{cases} \\
r(s,a) &= \begin{cases} 1, & \text{if } t(s,a,T) > 0 \text{ and } s \ne T \\ -m, & \text{if } t(s,a,F) > 0 \text{ and } s \ne F \\ 0, & \text{otherwise} \end{cases} .
\end{aligned}
$$

The correctness of the reduction follows by the above discussion. □

For POMDPs with unrestricted rewards we obtain completeness results for all policy existence problems. Contrary to the results for time- and history-dependent policies, surprisingly, the complexity of the existence problem for stationary partially-observable MDPs does not depend on whether the rewards are nonnegative or unrestricted. Instead, POMDPs seem to obtain their expressive power by history-dependent policies.

Corollary 6.4 *The stationary policy existence problem for POMDPs is* NP*-complete.*

Proof Hardness follows directly from Theorem 5.4, by generalization since a restricted reward MDP is also an unrestricted reward MDP. Membership in NP follows because one can, in polynomial time, guess a policy, then evaluate it in PL $\subseteq P$. □

Corollary 6.5 *The time-dependent policy existence problem for POMDPs is* NP*-complete.*

This follows by generalization on Theorem 6.3, since an unobservable MDP is also a POMDP.

Whereas for fully-observable MDPs the existence problems for time-dependent and history-dependent policies are equally hard to solve, for POMDPs the complexity is different.

Theorem 6.6 *The history-dependent policy existence problem for POMDPs is* PSPACE*-complete.*

The proof of Theorem 6.6 is a straightforward modification of the proof by Papadimitriou and Tsitsiklis [30, Theorem 6], where the rewards for reaching the satisfying resp. unsatisfying final state are changed appropriately.

6.2 Succinct MDPs

Note that we still take the size of the state space as our finite horizon, which may be exponential in the size of the representation of the MDP.

Theorem 6.7 *The time- or history-dependent policy existence problems for fully-observable sMDPs are* EXP*-complete.*

Theorem 6.8 *The stationary and time-dependent policy existence problems for partially-observable sMDPs are* NEXP*-complete. The history-dependent policy existence problem for succinct POMDPs is* EXPSPACE*-complete.*

The arguments used to prove the following results are direct translations of those used in the proofs of Theorem 6.2 and Theorem 6.3. In fact, the proof of the following theorem refers back to the proof of Theorem 4.7.

Theorem 6.9 *The stationary policy existence problem for unobservable sMDPs is* PSPACE*-complete.*

Theorem 6.10 *The time-dependent policy existence problem for unobservable sMDPs is* NEXP*-complete.*

6.3 Compressed MDPs

Theorem 6.11 *The* $\log n$*-horizon stationary and time-dependent policy existence problems for compressed POMDPs are* NEXP*-complete.*

Proof Membership in NEXP follows from Theorem 6.8. To show NEXP-hardness, we sketch a reduction from the succinct version of 3SAT, shown to be NEXP-complete in [31], to the stationary case. The reduction is similar to that of Theorem 6.3. where a formula was transformed into a POMDP which "checks" all clauses of the formula in parallel, works here. We put all states which stand for appearances of the same variable into one observation class and leave out considerations of variables in a clause where they do not appear. Then each trajectory of length at least 4 checks a clause, which satisfies the horizon bound. □

Theorem 6.12 *The* $\log n$*-horizon stationary, time-dependent and history-dependent policy existence problems for fully-observable compressed MDPs are* PSPACE*-hard.*

Proof We consider the case for stationary policies at first. To prove hardness, we show a polynomial time reduction from QBF, the validity problem for quantified Boolean formulae.

Informally, from a formula Φ with n variables we construct a fully-observable MDP with $2^{n+1} - 1$ states, where every state represents an assignment of Boolean values to the first i variables ($0 \leq i \leq n$) of Φ. Transitions from state s can reach the two states representing assignments that extend s by assigning a value to the next unassigned variable. If this variable is bound by an existential quantifier, then the action taken in s assigns a value to that variable; otherwise the transition is random and independent of the action. Reward 1 is gained for every action after a state representing a satisfying assignment for the formula is reached. If a state representing an unsatisfying assignment is reached, reward $-(2^n)$ is gained. Then the value of this MDP is positive iff the formula is true. A succinct representation of the MDP can be computed in time polynomial in the size of the formula.

This construction proves the same lower bound for the other types of policies, since every state except the last one appears at most once in every (consistent) trajectory. □

Theorem 6.13 *The* $\log n$*-horizon history-dependent policy existence problem for fully-observable compressed MDPs is in* PSPACE.

Proof We show that this problems is in NPSPACE = PSPACE. In PSPACE, an entire history-dependent policy cannot be specified, even for horizon $\log n$, since $\log n$ is roughly the size of the input. However, a policy can be guessed.

The set of possible histories (i.e., sequences of length $\leq \log n$ of observations) forms a tree; a policy can be specified by labeling the edges of that tree with actions. Such a policy can be guessed one branch at a time; in order to maintain consistency, one need only keep track of the current branch.

The number of observations is bounded by the number, n, of states of the MDP. Therefore, there are $n^{\log n}$ branches in this tree.

To evaluate a policy, for each branch through the policy tree, one must evaluate each possible sequence of states; if a sequence is consistent with the sequence of observations and actions, then its probability and its expected reward are calculated. The probability of each transition can be represented with $\log n$ bits; the product of $\log n$ such requires $(\log n)^2$ bits. Rewards require $\log n$ bits, and are accumulated (potentially) at each transition. The total reward for a given trajectory, therefore, requires at most $(\log n)^3$ bits.

There are at most $n^{\log n}$ trajectories, each of which is considered for each of the $n^{\log n}$ branches of the policy tree. Therefore, the total expected reward is the sum of $n^{2\log n}$ numbers, each represented by at most $(\log n)^3$ bits. Thus, the sum can be represented by at most $2(\log n)^5$ bits. This is polynomial in the size of the input (i.e., $\log n$). $\square$

Corollary 6.14 *The* $\log n$*-horizon stationary, time-dependent, and history-dependent policy existence problems for fully-observable compressed MDPs are* PSPACE*-complete.*

Theorem 6.15 *The* $\log n$*-horizon history-dependent policy existence problem for compressed POMDPs is* PSPACE*-complete.*

The exact complexity of the time-dependent policy existence problem for compressed POMDPs remains open.

Theorem 6.16 *The* $\log n$*-horizon time-dependent policy existence problem for compressed POMDPs is* PSPACE*-hard and is in* NEXP.

Proof Hardness follows from Theorem 6.12. Containment in NEXP follows from the standard guess-and-check approach. $\square$

For the complexity of policy existence problems for unobservable MDPs, the number of actions of the MDP is important. Remember that in the proof of Theorem 6.2, we used the fact that the policy existence problem disjunctively reduces to the policy evaluation problem: 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 can no longer be performed in time polynomial in $\log n$.

Theorem 6.17 *The stationary policy existence problem for unobservable compressed MDP*$[\log n]$*s with* $\log n$ *horizon is* PP*-complete.*

Proof Hardness for PP follows from the hardness proof of Lemma 4.10. To show containment in PP, we use arguments similar to those in the proof of Lemma 4.2. It is convenient to use the class GapP, which is the class of functions that are the difference of two #P functions. We make use of the fact that $A \in$ PP if and only if there exists a GapP function f such that for every x, $x \in A$ if and only if $f(x) > 0$ (see [17]). One can show that the function p from the proof of Lemma 4.2, is in GapP under these circumstances, because the respective matrix powering is in GapP (see the proofs of Lemmas 4.9 and 4.1), and GapP is closed under multiplication and summation. Finally, PP is closed under polynomial-time disjunctive reducibility [7], which completes the proof. $\square$

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

Theorem 6.18 *The stationary policy existence problem for unobservable compressed MDP with* $\log n$ *horizon is* $\leq_m^p$*-complete for* $\mathrm{NP}^{\mathrm{PP}}$.

Proof To see that this problem is in $\mathrm{NP}^{\mathrm{PP}}$, remember that the corresponding policy evaluation problem is PP-complete, by Theorem 4.10. For the existence question, one can guess a (polynomial-sized) policy, and verify that it has expected reward > 0 by consulting a PP oracle.

To show $\mathrm{NP}^{\mathrm{PP}}$-hardness, one needs that $\mathrm{NP}^{\mathrm{PP}}$ equals the $\leq_m^{np}$ closure of PP [34], 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).

Let $A \in \mathrm{NP}^{\mathrm{PP}}$. Then there is some PP relation, R, such that $x \in A \iff \exists^P y\ R(x,y)$ (where $\exists^P y$ means there is some polynomial p such that $\exists y \wedge |y| \leq p(|x|)$). Let M be the PP-machine for $R(x,y)$. Then one can model the computation of $M(x,y)$ as a $\log n$-horizon policy evaluation problem for an unobservable sMDP, as in the proof of Theorem 6.17. The sMDP depends only on M and x; the input itself supplies the policy. For a fixed x, then, each y specifies a policy. Thus $x \in A$ if and only if there is a y such that $R(x,y)$, i.e., such that the policy specified by x and y has expected value > 0. Thus the policy existence problem is complete for $\mathrm{NP}^{\mathrm{PP}}$. $\square$

From Toda's result that $\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{PP}}$ [33] and Theorem 6.18 we have the following.

Corollary 6.19 *The stationary policy existence problem for unobservable sMDPs and* $\log n$ *horizon is* $\leq_m^p$*-hard for* PH.

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 PSPACE-completeness does not depend on the existence of negative as well as positive rewards; the nonnegative rewards case is PSPACE-complete as well (see Theorem 5.8).

Theorem 6.20 *The stationary policy existence problem for unobservable compressed MDPs is* PSPACE*-complete.*

Proof Containment in PSPACE follows from Theorem 6.9. Note that in the proof of Theorem 4.7 a polynomially space bounded computation was transformed into a deterministic sMDP 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. This was originally proved in [25].

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

Theorem 6.21 *The time-dependent policy existence problem for unobservable compressed MDPs with* $\log n$ *horizon is* $\mathrm{NP}^{\mathrm{PP}}$*-complete.*

Hardness follows from Theorem 6.18, and membership in $\mathrm{NP}^{\mathrm{PP}}$ from Theorem 4.10.

Theorem 6.22 *The time-dependent policy existence problem for unobservable compressed MDP*[$\log n$]*s with* $\log n$ *horizon is* $\mathrm{NP}^{\mathrm{PP}}$*-complete.*

7 Exact Policy Existence Problems

The *exact α policy existence problem for β MDPs* is the set of all pairs (M, k) of a β MDP $M = (\mathcal{S}, \mathcal{O}, s_0, \mathcal{A}, o, t, r)$ and a rational k, for which there exists an α policy π with performance $\mathrm{perf}(M, s_0, |\mathcal{S}|, \pi) = k$.

There are several notable instances where the *exact* policy existence problem has a different complexity than the general policy existence problem. For instance, we have been unable to determine the complexity of the finite horizon stationary policy existence problem for fully-observable MDPs. (The infinite horizon case is known to be P-complete [30]; as noted in the introduction, the finite horizon problem is only known to be P-hard and in NP.) However, we can determine the complexity of the *exact* policy existence problem, as follows.

Theorem 7.1 *The exact stationary policy existence problem for fully-observable (flat) MDPs is* NP*-complete.*

Proof It is clear that the problem is in NP. In order to show its completeness, we give a reduction from the NP-complete problem PARTITION. The instances of PARTITION are functions $f : [n] \to \mathbf{N}$ for arbitrary $[n] = \{1, 2, \ldots, n\}$. Such an instance belongs to PARTITION iff for a subset $A \subseteq [n]$ it holds that $2\sum_{a\in A} f(a) = \sum_{b\in[n]} f(b)$.

Let $g : [m] \to \mathbf{N}$ be an instance of PARTITION. From g, we construct a fully-observable MDP M with state set $\mathcal{S} = [m{+}1]$, initial state $s_0 = 1$, actions $\mathcal{A} = \{in, out\}$, transition function

$$t(s,a,t) = \begin{cases} 1, & \text{if } t = \min\{s+1, m+1\},\ a \in \{in, out\} \\ 0, & \text{otherwise,} \end{cases}$$

and reward function

$$r(s,a) = \begin{cases} g(s), & \text{if } a = in \text{ and } s \leq m \\ 0, & \text{if } a = out \text{ or } s = m+1. \end{cases}$$

Every stationary policy π for M determines a subset $A = \{i \mid \pi(i) = in\}$ of $[m]$, such that $\mathrm{perf}(M, s_0, |\mathcal{S}|, \pi) = \sum_{a \in A} g(a)$. Therefore, $g \in$ PARTITION iff M has a stationary policy with reward exactly $\frac{1}{2}\sum_{b \in [n]} g(b)$. □

Notice that, in the MDP constructed in this proof, transitions depend only on time, not on the action chosen. Thus, the proof holds for unobservable and partially observable MDPs, and also for time-dependent policies.

Corollary 7.2 *The exact stationary and time-dependent policy existence problems partially- and fully-observable (flat) MDPs and the time-dependent policy existence problem for unobservable MDPs are* NP*-complete.*

8 Median and Average Performances

We now ask whether *most* of the policies for an MDP (in fact, we focus on POMDPs) have positive performance, and also, whether we can expect to get a positive performance if we pick a policy randomly. The first of these problems, called the policy median problem, is intuitively harder than the policy existence problem, because we need to count policies with positive performance. We show that it is complete for PP both for stationary and time-dependent policies. The second problem (the policy average problem) turns out to be harder than the policy existence problem for stationary policies, but easier for time-dependent policies.

Theorem 8.1 *The stationary policy median problem for POMDPs is* PP*-complete.*

Proof We show that the problem is in PP. Given a POMDP M, guess a policy, calculate the value of M under that policy, and accept iff that value is greater than 0. Because every policy is guessed with the same probability, this nondeterministic computation has more accepting than rejecting paths iff the median policy of M is greater than 0.

To show PP-completeness, we give a reduction from the PP-complete set MAJSAT. A Boolean formula ϕ is in MAJSAT iff more than half of all assignments satisfy ϕ. Given ϕ, we construct a POMDP $M(\phi)$ as in the proof of Theorem 5.4. Because of the bijection between stationary policies and assignments, we obtain that ϕ is in MAJSAT iff $M(\phi)$

has value 1 under more than half of all policies. Since $M(\phi)$ has value 0 or 1 under any policy, we obtain that the median of $M(\phi)$ is greater than 0 iff ϕ is in MAJSAT. □

Theorem 8.2 *The stationary policy average problem for POMDPs is* PP*-complete.*

Proof In order to prove PP-hardness, we apply the same technique as in the proof of Theorem 8.1, but we change the reward function of the POMDP $M(\phi)$ constructed from the formula ϕ in a way such that M has value 1 under every stationary policy corresponding to a satisfying assignment, and value -1 (instead of 0) under every other policy. Then the average of $M(\phi)$ is greater than 0 iff ϕ is in MAJSAT.

Lemma 4.2 shows that for every POMDP M and policy π there exists a logspace polynomial-time nondeterministic computation such that the number of accepting paths minus the number of rejecting paths in this computation equals the value of M under π multiplied by some constant dependent only on M. To show that the average problem is in PP, instead of generating one computation path for each policy, we now generate such a computation tree for every policy. (Note that it is necessary to write down each policy, requiring polynomial rather than logarithmic space.) Then this whole computation has more accepting than rejecting paths iff the average value of all policies is greater than 0. □

Theorem 8.3 *The time-dependent policy median problem for POMDPs is* PP*-complete.*

Proof Containment in PP can be shown using the same idea as in the stationary case. To show PP-hardness, we use a similar argument as for the stationary case. We showed how to reduce 3SAT to the existence problem for time-dependent policies for POMDPs by giving a transformation from formulae ϕ to POMDP $M(\phi)$ such that ϕ is satisfiable iff there exists a time-dependent policy under which $M(\phi)$ has performance greater than 0. Because we have a bijective correspondence between time-dependent policies and assignments such that $M(\phi)$ has positive value under a policy iff ϕ is satisfied by the correponding assignment, it follows that ϕ is in MAJSAT iff the median of $M(\phi)$ under time-dependent policy is positive. □

The time-dependent policy average problem for POMDPs has complexity even lower than the existence problem – we locate it in PL. The difference between time-dependent and stationary policies is that it is not necessary, in a trajectory for a time-dependent policy, to remember previous choices. Thus, the space requirements drop dramatically.

Theorem 8.4 *The time-dependent policy average problem for POMDPs is* PL*-complete.*

Proof We start with showing that the problem is in PL. Let M be a POMDP. Imagine M as a graph with vertices labeled by states of M and edges labeled by actions between vertices corresponding to the state transition function of M. Every path $\sigma_1 \xrightarrow{a_1} \sigma_2 \xrightarrow{a_2}$

$\cdots \xrightarrow{a_{n-1}} \sigma_n$ with $t(\sigma_i, a_i, \sigma_{i+1}) > 0$ $(i = 1, \ldots, n-1)$ is consistent with some time-dependent policy. Actually, every such path is consistent with exactly $|\mathcal{A}|^{(|\mathcal{O}|-1)(n-1)}$ time-dependent policies (since a time-dependent policy is simply a sequence of $n-1$ functions from $\mathcal{O}$ to $\mathcal{A}$). Because every policy has equal probability, the average performance can be computed by computing the average performance of all paths as above. Let $av(i,m)$ denote the average performance of all paths through the graph for M starting from state i after m steps. This can be calculated as

$$av(i,m) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \left(r(i,a) + \sum_{j \in \mathcal{S}} t(i,a,j) \cdot av(j, m-1) \right) .$$

Because we are interested only in whether $av(i,m)$ is greater than 0, we can multiply both sides with $|\mathcal{A}|2^{|\mathcal{S}|m}$ without changing the sign of the average. Then we obtain

$$av(i,m) \cdot |\mathcal{A}|2^{|\mathcal{S}|m} = \sum_{a \in \mathcal{A}} \left(2^{|\mathcal{S}|m} r(i,a) + \sum_{j \in \mathcal{S}} 2^{|\mathcal{S}|} t(i,a,j) \cdot 2^{|\mathcal{S}|(m-1)} av(j, m-1) \right) .$$

Note that the right hand side of the equation now only deals with integer values. We argue that $av(i,m)$ is in GapL (defined in [36]), i.e. the class of integer functions f for which exist logspace nondeterministic Turing machines N_f such that the number of accepting paths minus the number of rejectin paths N_f on input x equals $f(x)$. The functions r and t are part of the input, and GapL is closed under all operations used on the right hand side of the equation [1]. Therefore it follows that $av(i,m) \cdot |\mathcal{A}|2^{|\mathcal{S}|m}$ is computable in GapL. Thus the problem whether $av(i,m) \cdot |\mathcal{A}|2^{|\mathcal{S}|m}$ is greater than 0 is decidable in PL. (This follows using a characterization of PL by GapL from [1]).

Hardness for PL follows using the standard simulation technique, where rejecting computations have reward -1 and accepting computations have reward 1 (weighted depending on the length of the computation). $\square$

Note that the time-dependent policy average problem for fully-observable MDPs can be shown to have the same complexity using the same arguments as above.

9 Discussion and Open Questions

One general lesson that can be drawn from our results is that there is no simple relationship among the policy existence problems for stationary, time-dependent, and history-dependent policies. Although it is trivially true that if a good stationary policy exists, then good time- and history-dependent policies exist, is is not always the case that one of these problems is easier than the other. (For instance, in Theorem 6.6 and Corollaries 6.4 and 6.5, we see that the policy existence problem for history-dependent policies can be more difficult than for stationary and time-dependent policies, and in Theorems 6.2

and 6.3, the time-dependent case is more difficult than the stationary case. This contrasts with the situtation in Corollary 5.3 and Theorem 5.4 and in Theorems 6.11 and 6.15, where the policy-existence problem is more difficult for stationary policies than for history-dependent policies.)

There are a few problems that have not yet been categorized by completeness. However, the major open questions are of the form: What now? Now that we have proved that these problems are difficult to compute, heuristics are needed in order to manage them. In particular, there are no known heuristics for $\mathrm{NP}^{\mathrm{PP}}$-complete problems; surprisingly, these problems arise frequently, at least in the AI/planning literature. (See [20] for more examples.)

Acknowledgements

We would like to thank Anne Condon, Andy Klapper, 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] C. Bäckström. Expressive Equivalence of Planning Formalisms. *Artificial Intelligence*, 76:17–34, 1995.

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

[5] 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.

[6] 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 Vol. 969, Springer-Verlag, 1995.

[7] R. Beigel, N. Reingold and D. Spielman, PP is closed under intersection. *Journal of Computer and System Sciences,* 50:191-202, 1995.

[8] A. Borodin, S. Cook, and N. Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58:113–136, 1983.

[9] C. Boutilier, T. Dean, and S. Hanks. Planning under uncertainty: Structural assumptions and computational leverage. In *Proceedings of the Second European Workshop on Planning*, 1995.

[10] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *14th International Conference on AI*, 1995.

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

[12] 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.

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

[14] D. Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–379, 1987.

[15] F. D'Epenoux. A probabilistic production and inventory problem. *AManagement Science*, 10:98-108, 1963.

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

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

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

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

[20] J. Goldsmith, M. Littman, and M. Mundhenk. The complexity of plan existence and evaluation in probabilistic domains. *Proc. 13th Annual Conference on Uncertainty in Artificial Intelligence*, 182–189, 1997.

[21] H. Jung. On probabilistic time and space. In *Proceedings 12th ICALP*, pages 281–291. Lecture Notes in Computer Science, Vol. 194, Springer-Verlag, 1985.

[22] N. Kushmerick, S. Hanks, and D.S. Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76:239–286, 1995.

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

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

[25] M.L. Littman. Probabilistic propositional planning: Representations and complexity. *Proc. 14th National Conference on Artificial Intelligence*, AAAI Press/The MIT Press, 1997.

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

[27] M. Mundhenk, J. Goldsmith, and E. Allender. The complexity of policy evaluation for finite-horizon partially-observable Markov decision processes. *Proc. MFCS '97*, 1997.

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

[29] C.H. Papadimitriou and J.N. Tsitsiklis. Intractable problems in control theory. *SIAM Journal of Control and Optimization*, 24, pages 639–654, 1986.

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

[31] C.H. Papadimitriou and M. Yannakakis. A note on succinct representations of graphs. *Information and Control*, 71:181–185, 1986.

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

[33] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20:865–877, 1991.

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

[35] P. Tseng. Solving h-horizon, stationary Markov decision problems in time proportional to $\log h$. *Operations Research Letters*, 9, pages 287–297, September 1990.

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

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

[38] R. Washington. Incremental Markov-model planning. http://www.cis.upenn.edu/~rwash/papers/inc-markov.html (submitted for review).