Complexity issues in Markov decision processes

Judy Goldsmith

Dept. of Computer Science
University of Kentucky
Lexington, KY 40502

Martin Mundhenk

Fachbereich IV – Theoretische Informatik
Universität Trier
Trier, Germany

Abstract

We survey the complexity of computational problems about Markov decision processes: evaluating policies, finding good and best policies, approximating best policies, and related decision problems.

1 Introduction

Partially-observable Markov decision processes (POMDPs) model sequential decision making when outcomes are uncertain and the state of the system cannot be completely observed. They consist of decision epochs, states, observations, actions, transition probabilities, and rewards. At each decision epoch, the process is in some state, from which a "signal" is sent out which can be observed from outside. (Note that different states may send equal signals.) Choosing an action in a state generates a reward or possibly a cost (negative reward) and determines the state at the next decision epoch through a transition probability function. Policies are prescriptions of which action to take under any eventuality (i.e. any sequence of observations made in the previous decision epochs). Decision makers seek policies which maximize the expected sum of rewards after a certain number of decision epochs.

The control theory, mathematics, and operations research literature on MDPs concentrates primarily on algorithms for building the models and for finding "good enough" policies. Partially observable MDPs were developed by Drake [Dra62]. Sondik, in his thesis [Son71] and subsequent papers [SS73], was the first to address and resolve computational difficulties associated with POMDPs. In the subsequent years, different algorithmic methods were designed (see [Lov91] for an overview). The first lower bounds for the complexity of partially-observable MDPs were obtained by Papadimitriou and Tsitsiklis [PT86, PT87]. Their results apply to POMDPs with nonpositive rewards ("costs"), and concern the existence of zero-cost policies.

The maximization problem investigated by Papadimitriou and Tsitsiklis cannot be used in a binary search to determine the maximal performance of a POMDP in general. (The use of such a search depends on both positive and negative rewards.) In fact, it is often the case that the existence problem for policies with an *exact* value k is harder than the problem of whether there exists a policy with value greater than or equal to k. For example, the question of whether there exists a history-dependent policy for a POMDP with nonnegative rewards is NL-complete [MGLA97], as compared to PSPACE-completeness in [PT87, Theorem 6]. Beauquier et al. [BBS95] also considered various optimality criteria for MDPs.

In [MGA97, MGLA97], a systematical investigation of the complexity of Markov decision process decision problems under various parameters continued the work initiated in [PT87]. Methods from [MGA97] were applied in [LGM98a] to show complexity results for planners, as described in the AI literature. The AI/planning community has analyzed the complexity of several variations of MDPs and a wide variety of other related models. Most of these AI papers, especially [Bäc95, Byl94, Cha87, EHN96, ENS95], consider the complexity of *deterministic* control processes [1]. The complexity of evaluation and existence problems for these systems is generally lower than for stochastic processes (except [GLM97, Lit96] and one theorem in [Byl94]).

[1] Note, though, that stochastic systems are now hot; *JACM* has recently added an editor for "Decisions, Uncertainty, and Computation."

Many of the complexity results coming out of the AI community are for succinctly represented systems, described by 2-Phase Temporal Bayes Nets [BDH95, BDG95], sequential-effects-tree representation (ST) [Lit97a], probabilistic state-space operators (PSOs) [KHW95], 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 some 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 [Lit97a]; the PSPACE-completeness was proved independently in [GLM97].)

Sometimes the stochasticity of the system does not add computational complexity. For instance, Bylander [Byl94] 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 [MGLA97], PSPACE-completeness of the stochastic version of this problem is shown.

Whereas several algorithms are known to yield approximations of optimal policies for POMDPs (see e.g. [Lov91] for a survey), the question of how approximable these problems are in general has only recently begun to receive the attention it deserves. In [BdRS96], particular cases of unobservable MDP are shown to be non-approximable, and POMDPs with a bound on the number of different observations are shown to be approximable. In [LGM98b], we showed that most of these problems are not ε-approximable, unless some major collapse occurs.

2 Markov Decision Processes

Markov decision processes have their roots in operations research. An excellent reference is [Put94]. A partially-observable Markov decision process (POMDP) describes a controlled stochastic system by its states and the consequences of actions on the system. The following POMDP example is from [SS73]. Consider a hypothetical manufacturing operation that produces a finished product once an hour at the end of each hour. This machine consists of two identical internal components, each of which must operate once upon the product berfore it is finished. Unfortunately, each component can fail spontaneously. If a component has failed, there is some probability that it will cause the product to be defective. If the machine must be dissassembled in order to examine the status of the internal components, then its internal state is not directly observable. There are several control options during each one-hour production interval. Simplest, the manufacturing process is continued with no examination of the finished product. A second alternative is to observe the quality of the product; if it is defective, the machine is stopped, dissassembled, and the failed components become replaced. Finally, another alternative replaces both components each 5 hours. Each of the alternatives incurs some gains and losses, and we wish to know which is optimal.

Formally, a POMDP 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.[2] If the reward function maps to nonnegative integers, we say it has *nonnegative rewards*.

2.1 Policies and Performances

Policies describe how to act depending on observations. A *history-dependent* policy π selects an action dependent on the full history of observations made after the process was started in its initial state. π can be seen as a function $\pi : \mathcal{O}^* \to \mathcal{A}$, mapping sequences of observations to actions.

Let $M = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{O}, t, o, r)$ be a POMDP. 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 $\text{prob}_\pi(M, \theta)$ of a trajectory $\theta = \sigma_1, \sigma_2, \ldots, \sigma_m$ under policy π is the product of probabilities that state

[2] 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.

$\sigma i+1$ is reached from state σ_i under policy π with the appropriate observations.

The *reward* $\mathrm{rew}_\pi(M,\theta)$ of trajectory θ under π is the sum of its rewards, $\mathrm{rew}_\pi(M,\theta)=\Sigma_{i=1}^{m-1} r(\sigma_i,\pi(\cdot))$. 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. $\sum_{\theta\in\Theta(\sigma,k)} \mathrm{prob}_\pi(M,\theta)\cdot \mathrm{rew}_\pi(M,\theta)$, where $\Theta(\sigma,k)$ is the set of all length k trajectories beginning with state σ. The *value* $\mathrm{val}(M)$ of M for horizon k is M's maximal performance under any policy for horizon k. Note that the value of M may be different for different types of policies.

3 Functional Problems

3.1 Calculating the performance

Our first set of problems considers the complexity of calculating the performance of POMDP M under a policy π after $|M|$ steps. Since performances are rational numbers, we extend the notion of GapL (resp. GapP) to deal with rational numbers. We say that a function f is in GapL, if $f(x)=\frac{n_1(x)-n_2(x)}{d(x)}$ for $\#L$ functions n_1 and n_2, and for an FL function d. Intuitively, complexity properties of this GapL coincide with that originally defined by [FFK94].

Calculating the performance of a policy reduces to matrix powering.

Theorem 3.1 *(follows from [MGA97]) Evaluating a POMDP under a stationary, time-dependent, or history-dependent policy is* GapL*-complete.*

We include the proof, because it gives insight into the class GapL. 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 polynomial in the dimension n of the matrix.

Lemma 3.2 *(cf. [Vin91]) Let T be an $n\times n$ matrix of nonnegative binary integers, each of length n, and let $1\le i,j\le n$, $0\le m\le n$. The function mapping (T,m,i,j) to $(T^m)_{(i,j)}$ is in* #L.

Lemma 3.3 *Evaluating a stationary policy for a POMDP is in* GapL.

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 $\mathrm{perf}_s(\hat{M},s_0,|\mathcal{S}|,\pi)$ – the performance of $\hat{M}$ run from initial state s_0 for $|\mathcal{S}|$ steps under stationary policy π – can be computed in GapL.

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

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

Now, we have to show that the function p is in GapL. Define the weighted sum of positive rewards $p^+(M,i,m)=2^{hm}r^+(i,a)+\sum_{j\in\mathcal{S}} p(M,j,m-1)\cdot 2^h t(i,a,j)$ for $r+(i,a)=\max\{0,r(i,a)\}$, and let $p^-(M,i,m)$ be defined analogously as the weighted sum of negative rewards. Then $p(M,i,m)=p^+(M,i,m)+p^-(M,i,m)$. We show that p^+ and $-p^-$ are #L functions. 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 3.2 we get that $(T^{k-1})_{(i,j)}\in$ #L. The reward function is part of the input. Because #L is closed under multiplication and polynomial summation it follows that $p^+\in$ #L. Analogous arguments show that $p^-\in$ #L. Because GapL functions are differences of #L functions [AO96], it follows that $p\in$ GapL. □

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

Corollary 3.4 *Stationary policies for unobservable and fully observable MDPs can be evaluated in* GapL.

Lemma 3.5 *Evaluating stationary policies for unobservable MDPs is* GapL*-hard.*

Proof Consider $f \in$ GapL. Then there exists a probabilistic logspace Turing machine N corresponding to f and a polynomial p such that each computation of N on x uses at most $p(|x|)$ random decisions [Jun85]. 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 $\text{rew}(\theta) \cdot \text{prob}(\theta)$ equals 1 for trajectories θ corresponding to accepting computations (independent of their length), or -1 for rejecting computations, or 0 otherwise. Since $f(x)$ is the number of accepting computations of $N(x)$ minus the number of rejecting computations, we calculate $f(x)$ by calculating the number of trajectories θ of length $|\mathcal{S}|$ for $M(x)$ with $\text{rew}(\theta) \cdot \text{prob}(\theta) = 1$ minus the number of trajectories θ with $\text{rew}(\theta) \cdot \text{prob}(\theta) = -1$. But this is exactly $\text{perf}(M(x), s_0, |\mathcal{S}|, a)$. □

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. History-dependent policies present other issues. The logarithmic space bound depends in essential ways on the fact that the input history-dependent policy can be used to store the respective sequence of observations. (If the policy is given explicitly, then every possible sequence of observations is listed.) If we consider succint encodings of policies as, for instance, circuits, whose size is bounded by the size of the input POMDP – called *tractable* policies – the complexity of evaluation jumps; we break the limits of logarithmic space but not of polynomial time.

Theorem 3.6 *[Mun97] Evaluating a POMDP under a tractable history-dependent policy is* GapP*-complete.*

For POMDPs with nonnegative rewards, we obtain respective completeness results for #L and for #P.

3.2 Calculating the value

The *value* of a POMDP is its maximum performance under any policy. One of the important results for fully-observable MDPs is that the value under history-dependent policy can be calculated efficiently using Dynamic Programming [Bel57, Der70]. Moreover, the values under time-dependent and history-dependent policies are equal. It turns out that calculation of the value is easiest for fully-observable MDPs and hardest for partially-observable MDPs. (These results follow from [PT87, MGLA97].)

Theorem 3.7 *Calculating the time-dependent or history-dependent value of a fully-observable MDP is* FP*-complete.*

For stationary policies, the respective problem is FP-hard and in FP^{NP}, but a completeness result is not known. For unobservable MDPs, the stationary value is easier to calculate, because in this case each policy consists of exactly one action. This does not reflect the pattern for the other types of policies, for which the value for unobservable MDPs is harder to calculate than for for fully-observable MDPs.

Theorem 3.8 *Calculating the history-dependent value of an unobservable MDP is* FP^{NP}*-complete.*

This immediately implies a lower bound for partial observability.

Theorem 3.9 *Calculating the stationary or time-dependent value of a partially-observable MDP is* FP^{NP}*-complete.*

For history-dependent policies, it is even harder.

Theorem 3.10 *Calculating the history-dependent value of a partially-observable MDP is* FPSPACE*-complete.*

For fully-observable MDPs the time-dependent and history-dependent values are equal; this equality holds for POMDPs as well if $\text{FP}^{\text{NP}} = \text{FPSPACE}$.

3.3 Approximating the value

As we have seen, in many cases, computing the optimal policy for an MDP is not feasible. An obvious question is whether such optimal policies can be approximated in polynomial time. In several cases, it can be shown that there is no ε-approximation of the optimal policy unless some surprising collapse holds.

We distinguish two kinds of approximability. An algorithm A is said to *approximate the value of a POMDP with factor* ε, if A is polynomial-time bounded and on input POMDP M outputs $A(M)$ with $(1-\varepsilon)\cdot \mathrm{val}(M) \leq A(M) \leq \mathrm{val}(M)$ for $\mathrm{val}(M)$ being the value of M with horizon $|\mathcal{S}|$.

Theorem 3.11 *[LGM98b]*

1. *The stationary value of a partially-observable MDP is approximable with factor ε for any $\varepsilon < 1$ iff* P = NP.

2. *The time-dependent value of an unobservable or a partially-observable MDP is approximable with factor ε for any $\varepsilon < 1$ iff* P = NP.

Burago, et al. [BdRS96], considered the class of POMDPs with a bound of m on the number of states corresponding to an observation, where the rewards corresponded to the probability of reaching a fixed set of goal states (and thus was bounded by 1). They showed that for any fixed m, the optimal history-dependent policies for POMDPs in this class can be approximated to within an additive constant k. This means, for every k there exists a polynomial-time algorithm which on input a POMDP with history-dependent value v outputs a number between $v-k$ and v. We call this *additive k-approximability*. It can be shown that any class of POMDPs for which the optimal history-dependent policies can be approximated to within an additive constant ε have polynomial time ε-approximation schemes [LGM98b] *as long as there are no a priori bounds on the rewards.*

Notice, however, that Theorem 3.12 does not give us information about the classes of POMDPs that Burago, et al., considered; because of the restrictions associated with the parameter m, our hardness results do not contradict their result.

Theorem 3.12 *[LGM98b] The history-dependent value of POMDPs is additively k-approximable for any k if and only if* P = PSPACE.

Because one can multiply the expected reward of an MDP by a constant, one can leverage an additive k-approximation into a multiplicative ε-approximation, or vice-versa, by applying the given approximation to an MDP with suitably modified reward. Thus, the following theorem is actually a corollary of Theorem 3.12.

Theorem 3.13 *[LGM98b] The optimal history-dependent value for POMDPs is ε-approximable for any ε if and only if* P = PSPACE.

4 Decision Problems

Decision problems appear when we are interested only in one bit of the above function values, often the highest bit. The *policy evaluation problem* asks whether a given POMDP has positive performance under a given policy. The *policy existence problem* asks whether a given POMDP has positive performance under any policy of the specified type. The complexity of both these problems depends on the type of observability, the type of policy, and the type of encoding.

4.1 Policy evaluation

The policy evaluation problem is strongly related to the performance calculation. Essentially, their complexities are equal (compare Section 3.1). In the case of MDPs with nonnegative rewards, the problem degenerates to finding a trajectory whose probability is greater 0 and on which some reward greater 0 is obtained.

Theorem 4.1

1. *[MGA97] The stationary and time-dependent policy evaluation problems for partially-observable MDPs with nonnegative rewards are* NL-*complete.*

2. *[MGA97] The stationary and time-dependent policy evaluation problems for partially-observable MDPs are* PL-*complete.*

3. *[Mun97] The tractable history-dependent policy evaluation problems for fully-observable or partially-observable MDPs are* PP-*complete.*

4.2 Policy existence

The policy existence problem is strongly related to the value and performance calculation. In general, *guessing* a policy and *checking* its value gives an upper bound on the complexity of policy existence in terms of a nondeterministic machine with an oracle for policy evaluation. In the case of history-dependent policies, this straightforward approach yields an exponential-time nondeterministic machine, because the description length of a history-dependent policy might be exponential in the description length of the respective POMDP. As it turns out, history-dependent policies have useful structures which yield a better upper bound than the straightforward one.

Theorem 4.2 *The history-dependent policy existence problem is*

1. *[PT87]* P-*complete for fully-observable MDPs,*

2. *[MGA97]* NP-*complete for unobservable MDPs,*

3. *[Mun97]* $\mathrm{NP}^{\mathrm{PP}}$*-complete for tractable policies and partially-observable MDPs, and*

4. *[PT87]* PSPACE*-complete for partially-observable MDPs.*

The guess-and-check approach does not give tight bounds here for history-dependent policies, at least for fully-observable and partially-observable MDPs. For stationary policies, guess-and-check is optimal for partially observable, but not unobservable MDPs; for fully-observable MDPs, we do not yet know the exact complexity of the policy existence problem. For time-dependent policies, the only case where guess-and-check is not optimal is for fully-observable MDPs.

Theorem 4.3 *The stationary policy existence problem is*

1. *[MGA97]* PL*-complete for unobservable MDPs,*

2. *[PT87]* P*-hard for fully-observable MDPs, and*

3. *[MGA97]* NP*-complete for partially-observable MDPs.*

Finally, time-dependent policies fit best into the guess-and-check approach.

Theorem 4.4 *The time-dependent policy existence problem is*

1. *[PT87]* P*-complete for fully-observable MDPs, and*

2. *[PT87]* NP*-complete for unobservable and for partially-observable MDPs.*

Note that these results somehow reflect that the type of policy and the type of observability influence the complexity in very different ways. E.g., whereas in case of fully-observable MDPs, the policy existence problem is easier for history-dependent policies than for stationary policies, this is reversed for unobservable MDPs.

As above, for MDPs with non-negative rewards, the problem reduces to a simple reachability problem.

Theorem 4.5 *[MGA97] The policy-existence problem for partially-observable MDPs with nonnegative rewards is* NL*-complete.*

5 Succinctness

Many of the MDPs that arise in practice are both huge and highly structured. For instance, a state may be represented as a sequence of values for a set of parameters, and the transitions may be local, with each parameter dependent on only a small number of possible values for other parameters. The AI practitioners call such a representation "structured" or "propositional." They consider a variety of representations for such systems, most of which are equivalent. (See [BDG95, Byl94, Lit97b, GLM97, Lit96, LGM98a] for further discussion.)

We consider MDPs whose transition and reward functions can be represented by circuits, and describe them as *succinct,* [MGA97], in keeping with the complexity-theory literature [GW83, Wag86]. In particular, a circuit for the transition function takes as input $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$, and i, and outputs the i^{th} bit of the probability $t(s, a, s')$. (Similarly for the reward function.) Because the horizon is bounded by the number of states, it is now exponential in the size of the description of the MDP. Unsurprisingly, the complexities of succinctly represented problems are exponentially higher than those with so-called "flat" representations. (See [BLT92] for conditions that make such increases necessary.) For example, policy existence problems for POMDPs with nonnegative rewards are NL-complete under straightforward descriptions. Using succinct descriptions, the completeness increases to PSPACE. Since the techniques used in these proofs are familiar to many complexity theorists, we won't go into details.

5.1 Influence of parameters

Since 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 a number of steps only polynomial in the size of MDP's the description (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.

One can also use circuits to represent functions in a more straightforward manner. We consider succinct representations where the circuit for the transition function takes as input $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$, and outputs the entire probability $t(s, a, s')$. Necessarily, then, the number of bits in the representation of the probability $t(s, a, s')$ is polynomial (linear or sublinear) in the size of the input, and hence logarithmic in the number of states $|\mathcal{S}|$. We call these representations *compressed.* The issue of bit-counts in transition probabilities arises elswhere, for instance, in [BBS95, Tse90]. Note that probabilities are specified by single bit-strings, rather than as rationals specified by two bit-strings.

Similarly, the reward function can be represented by compact circuits. 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. We will focus, for the rest of this section, on decision problems, namely the policy evaluation and policy existence problems. These allow us to focus on the classes and problems of interest from a complexity theoretic standpoint.

We begin with some notation. 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 the transition probability and reward functions are compressed represented and $|\mathcal{A}| \leq f(|\mathcal{S}|)$, and an α policy π, such that $\mathrm{perf}(M, s_0, g(|\mathcal{S}|), \pi) > 0$.

The policy existence problems are expressed similarly.

The most interesting results are about the class $\mathrm{NP}^{\mathrm{PP}}$. This class sits above the polynomial hierarchy, yet within PSPACE. A surprising number of problems that arise in the study of succinctly represented planning problems with uncertainty turn out to be $\mathrm{NP}^{\mathrm{PP}}$-complete. We include some theorems and proofs related to the class $\mathrm{NP}^{\mathrm{PP}}$, in particular about succinct unobservable MDPs.

For the complexity of policy existence problems for unobservable MDPs, the number of actions of the MDP is important. When we proved [MGLA97] that the stationary policy existence problem is PL-complete for UMDPs, 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 5.1 *The stationary policy existence problem for unobservable compressed MDP*$[\log n]$*s with* $\log n$ *horizon is* PP*-complete.*

Proof Hardness for PP is a straightforward Turing machine simulation, as in the proof of Theorem 3.1. To show containment in PP, we use arguments similar to those in the proof of Lemma 3.3.

Remember that $A \in \mathrm{PP}$ 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 [FFK94]). One can show that the function p, constructed in the proof of Lemma 3.3, is in GapP under these circumstances, because the respective matrix powering is in GapP, and GapP is closed under multiplication and summation. Finally, PP is closed under polynomial-time disjunctive reducibility [BRS95], which completes the proof. □

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

Theorem 5.2 *The stationary policy existence problem for unobservable compressed MDPs with* $\log n$ *horizon is* $\mathrm{NP}^{\mathrm{PP}}$*-complete.*

Proof To see that this problem is in $\mathrm{NP}^{\mathrm{PP}}$, note that the corresponding policy evaluation problem is PP-complete. 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 [Tor91], 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 $\exists y, |y| \leq p(|x|)$). Let M be the PP-machine for $R(x, y)$. One can model the computation of $M(x, y)$ as a $\log n$-horizon policy evaluation problem for an unobservable compressed MDP. The compressed MDP 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 $\mathrm{NP}^{\mathrm{PP}}$-complete. □

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-hardness does not depend on the existence of negative as well as positive rewards nor on probabilistic transitions; the deterministic case with nonnegative rewards is PSPACE-hard as well.

Theorem 5.3 *The stationary policy existence problem for unobservable compressed MDPs is* PSPACE*-complete.*

Proof Hardness for PSPACE follows from the fact that the set of all configurations of a polynomially space bounded Turing machine can be chosen as the MDP's state space, on which a "local" (deterministic) transition function can easily be defined. The MDP's only action is "make the next configuration transition", and a positive reward is obtained when an accepting configuration is reached. Containment in PSPACE follows from a simulation similar to that in the proof of Theorem 3.3 and the fact that PPSPACE = PSPACE [Lad89]. □

As a consequence we obtain that the policy existence problem for flat MDPs with exponential horizon is in PSPACE. This was originally proved in [Lit97b].

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 5.4 *The time-dependent policy existence problems for unobservable compressed MDPs with* $\log n$ *horizon and for unobservable compressed MDP*[$\log n$]*s with* $\log n$ *horizon are* $\mathrm{NP}^{\mathrm{PP}}$*-complete.*

6 Discussion and Open Questions

One general lesson that can be drawn from the 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, it is not always the case that one of these problems is easier than the other.

There are a few problems that have not yet been categorized by completeness. One of those – the stationary policy existence problem for fully-observable MDPs – seems to be a good candidate for a "low" set in NP. 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, for $\mathrm{NP}^{\mathrm{PP}}$-complete problems a first attempt is done using techniques from Boolean satisfiability and dynamic programming in [ML98].

References

[AO96] E. Allender and M. Ogihara. Relationships among PL, #L, and the determinant. *RAIRO Theoretical Informatics and Applications*, 30(1):1–21, 1996.

[Bäc95] C. Bäckström. Expressive equivalence of planning formalisms. *Artificial Intelligence*, 76:17–34, 1995.

[BBS95] D. Beauquier, D. Burago, and A. Slissenko. On the complexity of finite memory policies for Markov decision processes. In *Math. Foundations of Computer Science*, pages 191–200. Lecture Notes in Computer Science #969, Springer-Verlag, 1995.

[BDG95] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *14th International Conference on AI*, 1995.

[BDH95] 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.

[BdRS96] 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.

[Bel57] R. Bellman. *Dynamic programming.* Princeton University Press, 1957.

[BLT92] 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.

[BRS95] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. *Journal of Computer and System Sciences*, 50:191–202, 1995.

[Byl94] T. Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69:165–204, 1994.

[Cha87] D. Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–379, 1987.

[Der70] C. Derman. *Finite state Markovian decision processes.* Academic Press, 1970.

[Dra62] A. Drake. *Observation of a Markov process through a noisy channel.* PhD thesis, Dept. of Electrical Engineering, Massachussets Institute of Technology, 1962.

[EHN96] K. Erol, J. Hendler, and D. Nau. Complexity results for hierarchical task-network planning. *Annals of Mathematics and Artificial Intelligence*, 1996.

[ENS95] K. Erol, D. Nau, and V. S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning. *Artificial Intelligence*, 76:75–88, 1995.

[FFK94] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48(1):116–148, 1994.

[GLM97] J. Goldsmith, M. Littman, and M. Mundhenk. The complexity of plan existence and evaluation in probabilistic domains. In *Proc. 13th Conf. on Uncertainty in AI.* Morgan Kaufmann Publishers, 1997.

[GW83] H. Galperin and A. Wigderson. Succinct representation of graphs. *Information and Control*, 56:183–198, 1983.

[Jun85] H. Jung. On probabilistic time and space. In *Proceedings 12th ICALP*, pages 281–291. Lecture Notes in Computer Science #194, Springer-Verlag, 1985.

[KHW95] N. Kushmerick, S. Hanks, and D.S. Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76:239–286, 1995.

[Lad89] R. Ladner. Polynomial space counting problems. *SIAM Journal on Computing*, 18:1087–1097, 1989.

[LGM98a] M. L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning, 1998.

[LGM98b] C. Lusena, J. Goldsmith, and M. Mundhenk. Most Markov decision process problems aren't approximable. Technical Report 274-98, University of Kentucky Department of Computer Science, 1998.

[Lit96] M.L. Littman. Probabilistic STRIPS planning is EXPTIME-complete. Technical Report CS-1996-18, Duke University Department of Computer Science, November 1996.

[Lit97a] M.L. Littman. Probabilistic propositional planning: Representations and complexity. In *Proc. 14th National Conference on Artificial Intelligence.* AAAI Press/The MIT Press, 1997.

[Lit97b] M.L. Littman. Probabilistic propositional planning: Representations and complexity. In *Proc. 14th National Conference on AI.* AAAI Press / MIT Press, 1997.

[Lov91] W.S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28:47–66, 1991.

[MGA97] M. Mundhenk, J. Goldsmith, and E. Allender. The complexity of the policy existence problem for partially-observable finite-horizon Markov decision processes. In *Proc. 25th Mathematical Foundations of Computer Sciences*, pages 129–138. Lecture Notes in Computer Science #1295, Springer-Verlag, 1997.

[MGLA97] M. Mundhenk, J. Goldsmith, C. Lusena, and E. Allender. Encyclopaedia of complexity results for finite-horizon Markov decision process, 1997.

[ML98] S.M. Majercik and M.L. Littman. MAXPLAN: A new approach to probabilistic planning. Technical Report 1998-01, Duke University, Department of Computer Science, 1998.

[Mun97] M. Mundhenk. The complexity of POMDPs under tractable history dependent policies, 1997.

[PT86] C.H. Papadimitriou and J.N. Tsitsiklis. Intractable problems in control theory. *SIAM Journal of Control and Optimization*, pages 639–654, 1986.

[PT87] C.H. Papadimitriou and J.N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[Put94] M.L. Puterman. *Markov decision processes.* John Wiley & Sons, New York, 1994.

[Son71] E. Sondik. *The optimal control of partially observable Markov processes.* PhD thesis, Stanford University, 1971.

[SS73] R. Smallwood and E. Sondik. The optimal control of partially observed Markov processes over the finite horizon. *Operations Research*, 21:1071–1088, 1973.

[Tor91] J. Torán. Complexity classes defined by counting quantifiers. *Journal of the ACM*, 38(3):753–774, 1991.

[Tse90] P. Tseng. Solving h-horizon, stationary Markov decision problems in time proportional to $\log h$. *Operations Research Letters*, pages 287–297, September 1990.

[Vin91] V. Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *Proc. 6th Structure in Complexity Theory Conference*, pages 270–284. IEEE, 1991.

[Wag86] K. W. Wagner. The complexity of combinatorial problems with succinct input representation. *Acta Informatica*, 23:325–356, 1986.