Complexity of Planning with Partial Observability

Jussi Rintanen
Albert-Ludwigs-Universität Freiburg, Institut für Informatik
Georges-Köhler-Allee, 79110 Freiburg im Breisgau
Germany

Abstract

We show that for conditional planning with partial observability the existence problem of plans with success probability 1 is 2-EXP-complete. This result completes the complexity picture for non-probabilistic propositional planning. We also give new more direct and informative proofs for the EXP-hardness of conditional planning with full observability and the EXPSPACE-hardness of conditional planning without observability. The proofs demonstrate how lack of full observability allows the encoding of exponential space Turing machines in the planning problem, and how the necessity to have branching in plans corresponds to the move to a complexity class defined in terms of alternation from the corresponding deterministic complexity class. Lack of full observability necessitates the use of beliefs states, the number of which is exponential in the number of states, and alternation corresponds to the choices a branching plan can make.

Introduction

The computational complexity of many forms of AI planning is well known. The most important problem that has been analyzed is that of existence of plans, in some cases existence of plans having a certain success probability or resource consumption. Plan existence for classical planning (deterministic, one initial state) is PSPACE-complete (Bylander 1994). Conditional planning with nondeterministic operators, several initial states and full observability is EXP-complete (Littman 1997). Conditional planning with nondeterministic operators and without observability (conformant planning) is EXPSPACE-complete (Haslum and Jonsson 1999).

Associating probabilities with nondeterministic choices and imposing a lower bound on plan's success probability make the problem more difficult. Without observability the problem of existence of plans with success probability $\geq c$ is undecidable (Madani *et al.* 2003). Both full observability and restriction to exponentially long plan executions make the problem decidable and bring it down to EXPSPACE and below (Littman *et al.* 1998; Mundhenk *et al.* 2000).

The complexity of one important problem has remained unknown until now. For probabilistic planning under partial observability, the special case of success probability 1 is of great importance. The problem is decidable because for reaching the goals with probability 1 the exact probabilities of nondeterministic events do not matter, and a finite discrete belief space can be used. In many applications plans with success probability anywhere strictly below 1 are not interesting. This is so especially in many engineering and manufacturing applications. This is in strong contrast to many optimizations problems typically expressed as MDPs/POMDPs, for which existence of solutions is obvious, and the problem is to find a solution that is optimal or close to optimal.

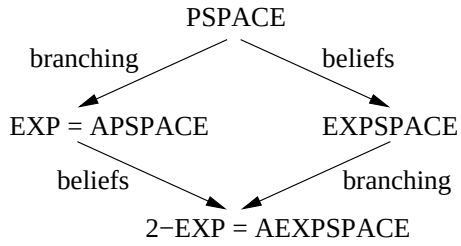

Figure 1: Effect of branching and partial observability on complexity of planning: classical planning is PSPACE-complete, beliefs (partial observability) add space requirement exponentially, and branching (nondeterminism) adds alternation.

In this paper we show that for non-probabilistic (success probability 1) partially observable planning the plan existence problem is 2-EXP-complete. We outline new proofs of the EXP-hardness of conditional planning with full observability and EXPSPACE-hardness of conditional planning without observability, and obtain the 2-EXP-hardness proof as a generalization of both of these two new proofs. The proofs very intuitively explain the problem complexities in terms of the types of Turing machines simulated.

This paper completes the complexity picture of non-probabilistic propositional planning in its most general forms, as summarized in Figure 1. Transition from the state space to the belief space leads to exactly an exponential increase in space complexity. From classical planning to conformant planning this is from PSPACE to EXPSPACE, and from nondeterministic full-information planning to nondeterministic planning with partial observability this is from APSPACE = EXP to AEXPSPACE = 2-EXP. Similarly, transition from the deterministic to the corresponding nondeter-

ministic planning problem[1] means a transition from a deterministic complexity class to the corresponding alternating complexity class; this corresponds to the introduction of branches into the plans. From classical planning to nondeterministic full information planning this is from PSPACE to APSPACE = EXP, and from conformant planning to general partially observable planning this is from EXPSPACE to AEXPSPACE = 2-EXP.

The structure of the paper is as follows. First we define alternating Turing machines and explain the relations between deterministic complexity classes and their alternating counterparts, followed by a definition of the planning problems we address. In the rest of the paper we analyze the computational complexity of the fully observable, unobservable and partially observable planning problems, in first two cases giving a new more direct hardness proof, and in the third case we establish the complexity for the first time. Before concluding the paper we discuss related work.

Preliminaries: Complexity Classes

In this section we define alternating Turing machines and several complexity classes used in this paper.

Definition 1 *An* alternating Turing machine *(ATM) is a tuple* $\langle \Sigma, Q, \delta, q_0, g \rangle$ *where*

- Q *is a finite set of states (the internal states of the ATM),*
- Σ *is a finite alphabet (the contents of tape cells),*
- δ *is a transition function* $\delta : Q \times \Sigma \cup \{|, \square\} \to 2^{\Sigma \cup \{|\} \times Q \times \{L,N,R\}}$,
- q_0 *is the initial state, and*
- $g : Q \to \{\forall, \exists, \mathit{accept}, \mathit{reject}\}$ *is a labeling of the states.*

The symbols | and $\square$ are the left-end-of-tape and the blank symbol, respectively. We require that $s = |$ and $m = R$ for all $\langle s, q', m \rangle \in \delta(q, |)$ and any $q \in Q$, that is, at the beginning of the tape the movement is to the right and | may not be overwritten. For $\langle s', q', m \rangle \in \delta(q, s)$ such that $s \in \Sigma$, we require $s' \in \Sigma$.

A configuration of a TM, consisting of the internal state q and the tape contents, is *final* if $g(q) \in \{\text{accept,reject}\}$.

The acceptance of an input string by an ATM is defined inductively starting from final configurations that are accepting. A final configuration is accepting if $g(q) = \text{accept}$. Non-final configurations are accepting if the state is universal ($\forall$) and all the successor configurations are accepting or if the state is existential ($\exists$) and at least one of the successor configurations is accepting. Finally, an ATM accepts a given input string if the initial configuration with initial state q_0 and the input string on the work tape is accepting.

A nondeterministic Turing machine (NDTM) is an ATM without universal states. A deterministic Turing machine is an NDTM with $|\delta(q, s)| = 1$ for all $q \in Q$ and $s \in \Sigma$.

[1]We can view conformant planning as deterministic planning in the belief space, because the successor belief state uniquely determined by the action and the preceding belief state.

PSPACE is the class of decision problems solvable by deterministic Turing machines that use a number of tape cells bounded by a polynomial on the input length n. Formally,

$$\text{PSPACE} = \bigcup_{k \geq 0} \text{DSPACE}(n^k).$$

Similarly other complexity classes are defined in terms of the time consumption (DTIME($f(n)$), or time and space consumption on alternating Turing machines (ATIME($f(n)$) and ASPACE($f(n)$)) (Balcázar *et al.* 1988; 1990).

$$\begin{aligned}
\text{EXP} &= \bigcup_{k \geq 0} \text{DTIME}(2^{n^k}) \\
\text{EXPSPACE} &= \bigcup_{k \geq 0} \text{DSPACE}(2^{n^k}) \\
\text{2-EXP} &= \bigcup_{k \geq 0} \text{DTIME}(2^{2^{n^k}}) \\
\text{APSPACE} &= \bigcup_{k \geq 0} \text{ASPACE}(n^k) \\
\text{AEXPSPACE} &= \bigcup_{k \geq 0} \text{ASPACE}(2^{n^k})
\end{aligned}$$

There are many useful connections between these classes (Chandra *et al.* 1981), for example

$$\begin{aligned}
\text{EXP} &= \text{APSPACE} \\
\text{2-EXP} &= \text{AEXPSPACE}.
\end{aligned}$$

Preliminaries: Planning

We formally define the conditional planning problem.

Definition 2 *Let A be a set of state variables. A* problem instance *in planning is $\langle I, O, G, V \rangle$ where I and G are formulae on A respectively describing the sets of initial and goal states, O is a set of operators $\langle c, e \rangle$ where c is a formula on A describing the precondition and e is an effect, and $V \subseteq A$ is the set of* observable state variables. *Effects are recursively defined as follows.*

1. *a and $\neg a$ for $a \in A$ are effects.*
2. *$e_1 \wedge \cdots \wedge e_n$ is an effect if $e_1, \ldots, e_n$ are effects (the special case with $n = 0$ is the empty conjunction $\top$.)*
3. *$c \rhd e$ is an effect if c is a formula on A and e is an effect.*
4. *$e_1 | \cdots | e_n$ is an effect if $e_1, \ldots, e_n$ for $n \geq 2$ are effects.*

Above $e_1 \wedge \cdots \wedge e_n$ means that all the effects e_i simultaneously take place. The notation $c \rhd e$ is for conditionality, that is, effect e takes place if c is true in the current state. Nondeterministic effects $e_1 | \cdots | e_n$ mean randomly choosing one of the effects e_i.

Compound observations and observations dependent on the last applied operator can be reduced in polynomial time to the basic model defined above in which the same set V of state variables is observable all the time.

Definition 3 *Let A be a set of state variables. A problem instance $\langle I, O, G, V \rangle$ on A is*

1. fully observable *if $V = A$,*
2. unobservable *if $V = \emptyset$,*
3. partially observable *if no restrictions on V are imposed.*

Plans are directed graphs with nodes of degree 1 labeled with operators and edges from branch nodes labeled with formulae.

Definition 4 *Let $\langle I, O, G, V\rangle$ be a problem instance in planning. A plan is a triple $\langle N, b, l\rangle$ where*

- *N is a finite set of nodes,*
- *$b \in N$ is the initial node,*
- *$l : N \to (O \times N) \cup 2^{\mathcal{L}\times N}$ is a function that assigns each node an operator and a successor node $\langle o, n\rangle \in O \times N$ or a set of formulae and successor nodes $\langle \phi, n\rangle$.*
  Only the observable state variables V may occur in the branch labels ϕ. Nodes with $l(n) = \emptyset$ are terminal.

The definition of plan execution should be intuitively clear. When a successor node of a branch node is chosen, the result is undefined if more than one condition is true.

In this paper we do not discuss plans with loops (cycles.) Loops are needed for some nondeterministic problems instances as a representation of plans with unbounded execution length, like obtaining 12 by throwing dice repeatedly. The complexity of the plan existence problem with or without loops is the same, but simulations of alternating Turing machines become more complicated because existence of looping plans does not exactly correspond to the acceptance criterion of alternating Turing machines.

Planning under Full Observability

Littman (1997) showed that the plan existence problem of probabilistic planning with full observability is EXP-complete. EXP-hardness was shown by reduction from the game G_4 (Stockmeyer and Chandra 1979). The reduction does not rely on exact probabilities, and hence also the non-probabilistic problem is EXP-complete.

Here we sketch a new proof based on the equality EXP = APSPACE. In Theorem 8 we generalize this proof and the EXPSPACE-hardness proof of Theorem 7 to a 2-EXP-hardness proof for the general partially observable problem.

Theorem 5 *Planning with full observability is EXP-hard.*

Proof: Sketch: The proof is a Turing machine simulation like the PSPACE-hardness proof of classical planning (Bylander 1994), but for the more powerful alternating Turing machines, yielding hardness for EXP = APSPACE. The difference is caused by $\forall$ and $\exists$ states of the ATM. For $\forall$ states, the transition is chosen nondeterministically and a branch in the plan chooses how execution continues thereafter, and for $\exists$ states the plan chooses the transition. The ATM accepts if and only if there is a plan that reaches the goal states under all nondeterministic choices. □

Testing plan existence is easily seen to be in EXP. This test is performed by traversing the state space backwards starting from $D_0 = G$ by computing $D_{i+1} = \bigcup_{o\in O} \text{preimg}_o(D_i) \cup D_i$ until $D_n = D_{n+1}$ for some $n \geq 0$. If $I \subseteq D_n$, then a plan exists. The number of states is exponential, and computing D_n is polynomial time in the number of states. The preimage $\text{preimg}_o(B)$ of a set B of states with respect to an operator o is the set of states from which a state in B is always reached by o. Viewing operators $o \in O$ as relations on the set S of states, we define $\text{preimg}_o(B)$ as

$$\{s \in S | sos' \text{ for some } s' \in B, sos' \text{ for no } s' \in S\backslash B\}.$$

Planning without Observability

The plan existence problem in probabilistic planning without observability is undecidable. Madani et al. (2003) prove this by using the close connection of the problem to the emptiness problem of probabilistic finite automata (Paz 1971; Condon and Lipton 1989). Without observability, plans are sequences of actions. The emptiness problem is about the existence of a word with an acceptance probability higher than c. This equals the planning problem when plans are identified with words and goal states are identified with accepting states. The acceptance probability may be increased by increasing the length of the word, and in general from a given c no finite upper bound can be derived and the problem is therefore undecidable.

When $c = 1$ the situation is completely different. Probabilities strictly between 0 and 1 can be identified, which leads to a finite discrete belief space. For any problem instance there is a finite upper bound on plan length, if a plan exists, and repetitive strategies for increasing success probability to 1 do not have to be used and they do not help in reaching 1. Notice that for every other fixed $c \in]0, 1[$ undecidability holds as the general problem can be reduced to the fixed-c problem for any $c \in]0, 1[$ by adding a first action that scales every success probability to the fixed c.

The unobservable planning problem is easily seen to be in EXPSPACE. Haslum and Jonsson (1999) point out this fact and outline the proof which is virtually identical to the PSPACE membership proof of plan existence for classical planning (Bylander 1994) except that it works at the level of belief states instead of states.

Theorem 6 *In the unobservable case, testing the existence of plans with success probability 1 is in EXPSPACE.*

Haslum and Jonsson (1999) also show that an unobservable planning problem similar to ours is EXPSPACE-hard. Their proof is a reduction from the EXPSPACE-hard universality problem of regular expressions with exponentiation (Hopcroft and Ullman 1979).

Our EXPSPACE-hardness proof simulates deterministic exponential-space Turing machines by planning. The main problem to be solved is the simulation of the exponentially long tape. In the PSPACE-hardness proof of classical planning each tape cell can be represented by one state variable, but with an exponentially long tape this kind of simulation is not possible. Instead, we use a randomization technique that forces the tape contents to be faithfully represented in the plan which may have a doubly exponential length.

Theorem 7 *Existence of plans in planning without observability is EXPSPACE-hard.*

Proof: Let $\langle \Sigma, Q, \delta, q_0, g\rangle$ be a deterministic Turing machine with an exponential space bound $e(x)$, and σ an input string of length n. We denote the ith symbol of σ by σ_i.

For encoding numbers from 0 to $e(n) + 1$ we need $m = \lceil \log_2(e(n) + 2)\rceil$ Boolean state variables.

We construct a problem instance in nondeterministic planning without observability for simulating the Turing machine. The size of the instance is polynomial in the size of the TM and the input string.

It turns out that when not everything is observable, instead of encoding all tape cells in the planning problem, it is sufficient to keep track of only one tape cell (which we call the *watched tape cell*) that is randomly chosen for every plan execution.

The set P of state variables consists of

1. $q \in Q$ for the internal states of the TM,
2. w_i for $i \in \{0, \ldots, m-1\}$ for the watched tape cell,
3. $s \in \Sigma \cup \{|, \Box\}$ for contents of the watched tape cell, and
4. h_i, $i \in \{0, \ldots, m-1\}$ for the position of the R/W head.

The uncertainty in the initial state is about which tape cell is the watched. Otherwise the formula encodes the initial configuration of the TM, and it is the conjunction of the following formulae.

1. q_0
2. $\neg q$ for all $q \in Q \backslash \{q_0\}$
3. Contents of the watched tape cell:

$$\begin{array}{rcl} | & \leftrightarrow & (w = 0) \\ \Box & \leftrightarrow & (w > n) \\ s & \leftrightarrow & \bigvee_{i \in \{1,\ldots,n\}, \sigma_i = s} (w = i) \text{ for all } s \in \Sigma \end{array}$$

4. $h = 1$ for the initial position of the R/W head.

So the initial state formula allows any values for state variables w_i and the values of the state variables $s \in \Sigma$ are determined by the values of w_i. The expressions $w = i$ and $w > i$ denote the obvious formulae for integer equality and inequality of the numbers encoded by $w_0, w_1, \ldots$. We also use effects $h := h + 1$ and $h := h - 1$ for incrementing and decrementing the number encoded by variables h_i.

The goal is the following formula.

$$G = \bigvee \{q \in Q | g(q) = \text{accept}\}$$

To define the operators, we first define effects corresponding to all possible transitions.

For all $\langle s, q \rangle \in (\Sigma \cup \{|, \Box\}) \times Q$ and $\langle s', q', m \rangle \in (\Sigma \cup \{|\}) \times Q \times \{L, N, R\}$ define the effect $\tau_{s,q}(s', q', m)$ as $\alpha \wedge \kappa \wedge \theta$ where the effects α, κ and θ are defined as follows.

The effect α describes the change to the current tape symbol. If $s = s'$ then $\alpha = \top$ as nothing on the tape changes. Otherwise, $\alpha = ((h = w) \rhd (\neg s \wedge s'))$ to denote that the new symbol in the watched tape cell is s' and not s.

The effect κ changes the internal state of the TM. Again, either the state changes or does not, so $\kappa = \neg q \wedge q'$ if $q \neq q'$ and $\top$ otherwise. If R/W head movement is to the right then $\kappa = \neg q \wedge ((h < e(n)) \rhd q')$ if $q \neq q'$, and $(h = e(n)) \rhd \neg q$ otherwise. This prevents reaching an accepting state if the space bound is violated: no further operators are applicable.

The effect θ describes the movement of the R/W head.

$$\theta \quad = \quad \begin{cases} h := h - 1 & \text{if } m = L \\ \top & \text{if } m = N \\ h := h + 1 & \text{if } m = R \end{cases}$$

Now, these effects $\tau_{s,q}(s', q', m)$ which represent possible transitions are used in the operators that simulate the DTM. Let $\langle s, q \rangle \in (\Sigma \cup \{|, \Box\}) \times Q$ and $\delta(s, q) = \{\langle s', q', m \rangle\}$.

If $g(q) = \exists$ then define the operator

$$o_{s,q} = \langle ((h \neq w) \vee s) \wedge q, \tau_{s,q}(s', q', m) \rangle.$$

A correct simulation is clearly obtained assuming that when operator $o_{s,q}$ is executed the current tape symbol is indeed s. So assume that some $o_{s,q}$ is the first operator that misrepresents the tape contents. Let $h = c$ for some c. Now there is an execution (initial state) of the plan so that $w = c$. On this execution the precondition of $o_{s,q}$ is not satisfied, and the plan is not executable. Hence a valid plan cannot contain operators that misrepresent the tape contents. □

Planning under Partial Observability

Showing that the plan existence problem for planning with partial observability is in 2-EXP is straightforward. The easiest way to see this is to view the partially observable planning problem as a nondeterministic fully observable planning problem with belief states viewed as states. An operator maps a belief state to another belief state nondeterministically: subset of the image of a belief state with respect to an operator is chosen by observations. Like pointed out earlier, the algorithms for fully observable planning run in polynomial time in the size of the state space. The state space with the belief states as the states has a doubly exponential size in the size of the problem instance, and hence the algorithm runs in doubly exponential time in the size of the problem instance.

The main result of the paper, the 2-EXP-hardness of partially observable planning, is obtained as a generalization of the proofs of Theorems 5 and 7. From the EXPTIME-hardness proof we take the simulation of alternation by nondeterministic operators, and from the EXPSPACE-hardness proof the simulation of the exponentially long working tape. These easily yield a simulation of alternating Turing machines with an exponential space bound, and thereby proof of AEXPSPACE-hardness.

Theorem 8 *Testing existence of plans with success probability 1 is 2-EXP-hard.*

Proof: Let $\langle \Sigma, Q, \delta, q_0, g \rangle$ be any alternating Turing machine with an exponential space bound $e(x)$. Let σ be an input string of length n. We denote the ith symbol of σ by σ_i.

We construct a problem instance in nondeterministic planning with partial observability for simulating the Turing machine. The problem instance has a size that is polynomial in the size of the description of the Turing machine and the input string.

Here we just list the differences to the proof of Theorem 7 needed for handling alternation.

The set of state variables is extended with

1. s^* for $s \in \Sigma \cup \{|\}$ for the symbol last written, and
2. L, R and N for the last movement of the R/W head.

The observable state variables are L, N and R, $q \in Q$, and s^* for $s \in \Sigma$. These are needed by the plan to decide how to proceed execution after a nondeterministic $\forall$-transition.

The initial state formula is conjoined with $\neg s^*$ for all $s \in \Sigma \cup \{|\}$. The goal formula remains unchanged.

Next we define the operators. All the transitions may be nondeterministic, and the important thing is whether the transition is for a $\forall$ state or an $\exists$ state. For a given input symbol and a $\forall$ state, the transition corresponds to one nondeterministic operator, whereas for a given input symbol and an $\exists$ state the transitions corresponds to a set of deterministic operators.

Effects $\tau_{s,q}(s', q', m) = \alpha \wedge \kappa \wedge \theta$ are like in the proof of Theorem 7 except for the following modifications. The effect α is modified to store the written symbols to the state variables s^*. If $s = s'$ then $\alpha = \top$ as nothing on the tape changes. Otherwise, $\alpha = ((h = w) \rhd (\neg s \wedge s')) \wedge s'^* \wedge \neg s^*$. The effect θ is similarly extended to store the tape movement to L, N and R.

$$\theta \quad = \quad \begin{cases} (h := h-1) \wedge L \wedge \neg N \wedge \neg R & \text{if } m = L \\ N \wedge \neg L \wedge \neg R & \text{if } m = N \\ (h := h+1) \wedge R \wedge \neg L \wedge \neg N & \text{if } m = R \end{cases}$$

Now, these effects $\tau_{s,q}(s', q', m)$ which represent possible transitions are used in the operators that simulate the ATM. Operators for existential states $q, g(q) = \exists$ and for universal states $q, g(q) = \forall$ differ. Let $\langle s, q\rangle \in (\Sigma \cup \{|, \Box\}) \times Q$ and $\delta(s, q) = \{\langle s_1, q_1, m_1\rangle, \ldots, \langle s_k, q_k, m_k\rangle\}$.

If $g(q) = \exists$, then define k deterministic operators

$$\begin{array}{l} o_{s,q,1} = \langle((h \neq w) \vee s) \wedge q, \tau_{s,q}(s_1, q_1, m_1)\rangle \\ o_{s,q,2} = \langle((h \neq w) \vee s) \wedge q, \tau_{s,q}(s_2, q_2, m_2)\rangle \\ \vdots \\ o_{s,q,k} = \langle((h \neq w) \vee s) \wedge q, \tau_{s,q}(s_k, q_k, m_k)\rangle \end{array}$$

That is, the plan determines which transition is chosen. If $g(q) = \forall$, then define one nondeterministic operator

$$\begin{array}{lll} o_{s,q} = \langle((h \neq w) \vee s) \wedge q, & (\tau_{s,q}(s_1, q_1, m_1)| \\ & \vdots \\ & \tau_{s,q}(s_k, q_k, m_k))\rangle. \end{array}$$

That is, the transition is chosen nondeterministically.

We claim that the problem instance has a plan if and only if the Turing machine accepts without violating the space bound. If the Turing machine violates the space bound, then $h > e(n)$ and an accepting state cannot be reached because no further operator will be applicable.

From an accepting computation tree of an ATM we can construct a plan, and vice versa. The construction of the plan is recursive: accepting final configurations correspond to terminal nodes of plans, $\exists$-nodes correspond to an operator corresponding to the transition made in the node followed by the plan recursively constructed for the successor configuration, and $\forall$-nodes correspond to the execution of a nondeterministic operator followed by a branch node that selects the plan recursively constructed for the corresponding successor configuration.

Construction of computation trees from plans is similar, but involves small technicalities. A plan with DAG form can be turned into a tree by having several copies of the shared subplans. Branches not directly following a nondeterministic operator causing the uncertainty can be moved earlier so that only plan nodes with a non-singleton belief state immediately follow a nondeterministic operator. With these transformations there is an exact match between plans and computation trees of the ATM.

Because alternating Turing machines with an exponential space bound are polynomial time reducible to the nondeterministic planning problem with partial observability, the plan existence problem is AEXPSPACE=2-EXP-hard. □

Impact of Determinism on Complexity

Our proofs for the EXP-hardness and 2-EXP-hardness of plan existence for fully and partially observable planning use nondeterminism. Also the EXP-hardness proof of Littman (1997) uses nondeterminism. The question arises whether complexity is lower when all operators are deterministic.

Theorem 9 *The plan existence problem with full observability and deterministic operators is in PSPACE.*

Proof: This is because iteration over an exponential number of initial states needs only polynomial space, and testing goal reachability for each initial state needs only polynomial space like in the PSPACE membership proof of classical planning (Bylander 1994). □

Under unobservability determinism does not reduce complexity: proof of Theorem 7 uses deterministic operators only. But for the general partially observable problem determinism does reduce complexity.

Theorem 10 *The plan existence problem with partial observability and deterministic operators is in EXPSPACE.*

Proof: The idea is similar to the EXPSPACE membership proof of planning without observability: go through all possible intermediate stages of a plan by binary search. Determinism yields an exponential upper bound on the sum of the cardinalities of the belief states that are possible after branching and a given number of actions, (see Figure 2), and determinism also makes it unnecessary to visit any belief state more than once. Hence plan executions have doubly exponential length and binary search needs only exponential recursion depth.

Let $\langle C_1, \ldots, C_k\rangle$ be the classes of observationally indistinguishable states. Let S be the set of states. For belief state B and set L of belief states with $\sum_{B' \in L} |B'| \leq |S|$, test reachability with plans of depth 2^i by the following procedure.

procedure reach(B,L,i)
if $i = 0$ **then**
  begin
    for each $j \in \{1, \ldots, k\}$
      if $\text{img}_o(B) \cap C_j \not\subseteq B'$ for all $o \in O$ and $B' \in L$
      and $B \cap C_j \not\subseteq B'$ for all $B' \in L$
      then return false
    end

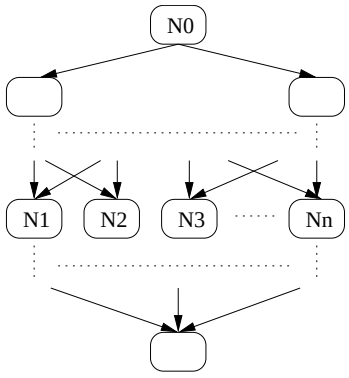

Figure 2: In a deterministic problem instance, the sum of the cardinalities of the possible sets of states at plan nodes $N_1, \ldots, N_n$ cannot exceed the cardinality of the possible set of states at the root node of the plan N_0. Here $N_1, \ldots, N_n$ are plan nodes that intersect the plan in the sense that none of these nodes is an ancestor node of another.

	deterministic	non-deterministic
full observability	PSPACE	EXP
no observability	EXPSPACE	EXPSPACE
partial observability	EXPSPACE	2-EXP

Table 1: Summary of complexities of planning with multiple initial states, deterministic or non-deterministic operators, and different degrees of observability.

```
    return true;
  end
else
  for each L' ⊆ 2^S such that Σ_{B'∈L'} |B'| ≤ |B| and
      for every B' ∈ L', B' ⊆ C_j for some j ∈ {1,...,k}
    if reach(B,L',i − 1) then
      begin
        flag := true;
        for each B'' ∈ L'
          if reach(B'',L,i − 1) = false then flag := false;
        if flag = true then return true
      end
  end
return false
```

We can now test plan existence by calling reach(B,L,$|S|$) for every $B = I \cap C_i, i \in \{1, \ldots, k\}$ and $L = \{G \cap C_i | i \in \{1, \ldots, k\}\}$. The algorithm always terminates, and a plan exists if and only if answer *true* is obtained in all cases.

The space consumption is (only) exponential because the recursion depth is exponential and the sets $L' \subseteq 2^S$ with $\sum_{B' \in L'} |B'| \leq |B|$ have size $\leq |S|$. Sets L' this small suffice because all operators are deterministic, and after any number of actions, independently of how the plan has branched, the sum of the cardinalities of the possible belief states is not higher than the number of initial states. □

The problem complexities are summarized in Table 1. Restriction to only one initial state affects the deterministic unobservable and partially observable planning problems only: they both come down to PSPACE from EXPSPACE.

Related Work and Conclusions

The complexity of constructing and evaluating policies for MDPs with the restriction to finite-horizon performance has been thoroughly analyzed by Mundhenk et al. (2000). They evaluate the computational complexity of policy evaluation and policy existence under different assumptions. While in the fully observable, unobservable and in the general case we have problems complete for EXP, EXPSPACE and 2-EXP, Mundhenk et al. have EXP, NEXP and EXPSPACE for history-dependent POMDPs with problem instances represented as circuits, exponentially long horizons and same observability restrictions. The latter complexities are this low because of the exponential horizon length.

In this paper we have proved that the plan existence problem of propositional non-probabilistic planning with partial observability is 2-EXP-complete. Complexity of classical planning and non-probabilistic propositional planning with other observability restrictions (full observability, no observability) was already known (Bylander 1994; Littman 1997; Haslum and Jonsson 1999), and our result was the most important missing one.

We also gave more direct proofs of EXPSPACE-hardness of planning without observability and EXP-hardness of planning with full observability, shedding more light to results first proved by Haslum and Jonsson and by Littman, and discussed the impact of determinism on complexity.

The results do not address plan optimality with respect to a cost measure, like plan size or plan depth, but it seems that for many cost measures there is no increase in complexity and that this is in many cases easy to prove.

References

José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity I*. Springer-Verlag, Berlin, 1988.

José Luis Balcázar, Josep Díaz, and Joaquim Gabarró. *Structural Complexity II*. Springer-Verlag, Berlin, 1990.

Tom Bylander. The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, 69(1-2):165–204, 1994.

A. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.

Anne Condon and Richard J. Lipton. On the complexity of space bounded interactive proofs (extended abstract). In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 462–467. IEEE, 1989.

Patrik Haslum and Peter Jonsson. Some results on the complexity of planning with incomplete information. In Susanne Biundo and Maria Fox, editors, *Recent Advances in AI Planning. Fifth European Conference on Planning (ECP'99)*, number 1809 in Lecture Notes in Artificial Intelligence, pages 308–318. Springer-Verlag, 1999.

John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company, 1979.

M. L. Littman, J. Goldsmith, and M. Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9:1–36, 1998.

Michael L. Littman. Probabilistic propositional planning: Representations and complexity. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97) and 9th Innovative Applications of Artificial Intelligence Conference (IAAI-97)*, pages 748–754, Menlo Park, July 1997. AAAI Press.

Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 2003. to appear.

Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon Markov decision process problems. *Journal of the ACM*, 47(4):681–720, 2000.

Azaria Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971.

Larry J. Stockmeyer and Ashok K. Chandra. Provably difficult combinatorial games. *SIAM Journal on Computing*, 8(2):151–174, 1979.