Bounded Branching and Modalities in Non-Deterministic Planning

Blai Bonet
Departamento de Computación
Universidad Simón Bolívar
Caracas, Venezuela
bonet@ldc.usb.ve

Abstract

We study the consequences on complexity that arise when bounds on the number of branch points on the solutions for non-deterministic planning problems are imposed as well as when modal formulae are introduced into the description language. New planning tasks, such as whether there exists a plan with at most k branch points for a fully (or partially) observable non-deterministic domain, and whether there exists a no-branch (a.k.a. conformant) plan for partially observable domains, are introduced and their complexity analyzed. Among other things, we show that deciding the existence of a conformant plan for partially observable domains with modal formulae is 2EXPSPACE-complete, and that the problem of deciding the existence of plans with bounded branching, for fully or partially observable contingent domains, has the same complexity of the conformant task. These results generalize previous results on the complexity of non-deterministic planning and fill a slot that has gone unnoticed in non-deterministic planning, that of conformant planning for partially observable domains.

Introduction

Consider the game of Mastermind played by a codemaker and a codebreaker. The game starts with the codemaker that picks a secret code, a sequence of 4 colors from 6 available colors. The task of the codebreaker is to reveal the secret code by questioning the codemaker and evaluating his answers. Each question has the form of a *guess*, a sequence of 4 colors, that is answered by the codemaker with two pieces of information: the first is the number of exact matches in the guess, i.e. the number of right colors in right positions, and the second is the number of near matches in the guess, i.e. the number of right colors in wrong positions. To be precise, if $\langle x_1, x_2, x_3, x_4\rangle$ and $\langle y_1, y_2, y_3, y_4\rangle$ are the secret code and guess respectively, the codemaker's answer is the number of times $x_i = y_i$ for $1 \leq i \leq 4$, and the number of times that $x_i = y_j$, for $1 \leq i \leq 4$ and some $j \neq i$.

Since the game proceeds in guess-answer stages, each subsequent guess depends on the previous sequence of guesses and answers, i.e. the information acquired up to that moment. A winning strategy can be represented as a tree such that internal nodes, that represent sets of possible secret codes and are labeled with guesses, have children nodes that represent the sets of codes compatible with the possible answers. The leaves of the tree are nodes that represent single secret codes. Fig. 1, for example, shows an *optimal* strategy for a game of Mastermind with 3 positions and 3 colors. Although the labels on the edges that tell the possible answers are not shown, the important thing to note is the *form* of the solution, i.e. its tree-like structure.

The game of Mastermind can be thought as a non-deterministic planning problem with partial observability, and hence a solution can be obtained with an appropriate planner.[1] Indeed, subsets of possible states (secret codes in Mastermind) are called *belief states*, and solutions like Fig. 1 are called contingent plans in belief space or contingent plans with partial observability. In general, if b is a belief state in the solution graph, o an operator applicable in b, and $z_1, \ldots, z_n$ the collection of possible observations obtained after applying o in b, then the children of b in the solution graph are $b_o^{z_1}, \ldots, b_o^{z_n}$ where $b_o^{z_i}$ is the belief state that results of applying o in b and observing z_i.

An interesting variation of Mastermind, which has become known as *static* Mastermind, was introduced by (Chvatal 1983). In this variation, the codebreaker is asked to give ahead a complete sequence of guesses such that the secret code can be determined from the answers upon such guesses. For example, for a game with 3 positions and 3 colors, the sequence

$$\text{guess}(2,0,0)\ ,\ \text{guess}(2,1,0)\ ,\ \text{guess}(2,2,1)\ . \qquad (1)$$

is guaranteed to succeed independently of the chosen secret code, and its length is minimum among all such sequences.

We call a solution like Eq. 1 a *conformant plan* for a non-deterministic planning problem with partial observability. Conformant planning has been studied before in the context of non-deterministic planning with full observability (Goldman & Boddy 1996; Smith & Weld 1998), yet its general treatment in the context of partial observability is novel to the best of our knowledge.

From the example, it is clear that the potential benefits of a conformant plan over a contingent plan are its simplicity of execution and storage. A conformant plan is a linear

[1] Yet, its formalization requires modal formulae in order to define the goal (see below).

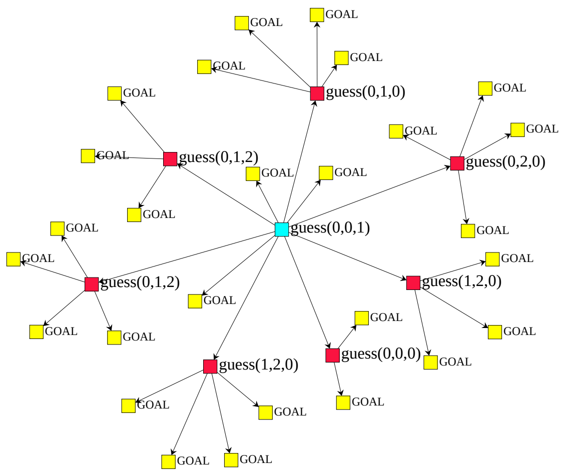

Figure 1: Optimal contingent plan for Mastermind of 3 pegs and 3 colors.

sequence of operators that can be applied effortless by an execution unit and, at the same time, is easier to store and in some cases more succinct than contingent plans.

We can think of Fig. 1 and Eq. 1 as the extreme points of a discrete yet infinite range of solution forms. In one extreme, Eq. 1 is a solution with no branch points whatsoever while Fig. 1 is a solution with an unrestricted number of branch points, that happens to be 2 in the example. Thus, in the general case, it is reasonable to ask whether there is a solution with at most 1 branch point, a solution with at most 2 branch points, and so on. Note however that certain planning tasks might not admit a particular solution form, or none altogether. A similar situation arises in planning with full observability where some tasks have contingent plans but no conformant ones.

Meuleau and Smith (2003) consider the case of bounded branching for non-deterministic problems with full observability. This seems to be the first explicit treatment of the problem in the planning community. They use a formalization based on Partially Observable Markov Decision Processes (POMDPs) for computing plans with bounded branching. Although Meuleau and Smith focus in computing plans as opposed to decide plan existence, their technique might not be the most appropriate since solving POMDPs requires double exponential time while the existence of plans with bounded branching for fully observable problems can be decided in exponential space.

In this paper, we study the task of deciding the existence of plans with a bounded number of branch points in the context of non-deterministic planning with either partial or full observability. We provide complexity results for such tasks as well as decision algorithms. Additionally, we consider planning languages extended with modalities and the effects of such modalities on the complexity of decision problems. As it will be shown, deciding if a planning task with partial observability and modal formulae has a conformant plan or a plan with at most k-branch points is 2EXPSPACE-complete, and deciding if a planning task with full observability has a conformant plan or a plan with at most k-branch points is EXPSPACE-complete.[2] We also study the complexity of restricted classes of problems as whether there are plans of polynomial length for fully observable problems, and whether there are plans for partially observable problems without modal formulae. In the latter case, for example, we show that the complexity of the task decreases from 2EXPSPACE to EXPSPACE.

These results extend the current knowledge on the complexity of non-deterministic planning in two directions: by studying novel plan-existence decision problems, and by considering richer representation languages.

The paper is organized as follows. We first define the planning formalism and the different decisions problems, and give a brief background on complexity theory. Then, algorithms for conformant and bounded-branching planning are given to establish upper bounds on the complexity of such tasks. The lower bounds on complexity, i.e. hardness results, are shown using ideas and techniques from (Haslum & Jonsson 1999) reviewed in a section on regular expressions and automata. Then, we show how to compactly represent automata with belief states and give the lower bounds. The paper ends with a section devoted to special cases of decision problems, and with a discussion on future work, open problems and conclusions.

Planning Problems, Solutions and Modalities

We deal with planning problems of the form $P = \langle D, I, G, O, Z\rangle$ where D is a set of propositional symbols (propositions), I and G are propositional formulae describing the initial and goal situations, O is a set of operators, and $Z \subseteq D$ is the subset of observable variables. Operators are pairs $\langle c, e\rangle$ where c, called the precondition of the operator, is a propositional formula and e is an effect. We allow conditional and non-deterministic effects, defined as:

1. $\top$ (the null effect) is an effect,
2. p and $\neg p$ are effects for $p \in D$,
3. if $e_1, \ldots, e_n$ are effects, so they are $(e_1 \wedge \cdots \wedge e_n)$ (parallel effect) and $(e_1 \oplus \cdots \oplus e_n)$ (non-deterministic effect),
4. if c is a propositional formula over D and e an effect, then $(c \rhd e)$ is an effect (conditional).

States are valuations for all propositions in D, and are usually described by the subset of propositions that hold true in the state. An operator $o = \langle c, e\rangle$ is applicable in state s iff $s \models c$, and in such case the result of o on s is $Res(o, s) \stackrel{\text{def}}{=} Appl(\mathit{Eff}(e, s), s)$ where $\mathit{Eff}(e, s)$ is the set of *effects* induced by e on s, and $Appl(E, s)$ is the application of the effects in E over s. If the operator is not applicable the result is undefined. The set $\mathit{Eff}(e, s)$ is defined as:

1. if $e = \top$, then $\mathit{Eff}(e, s) = \{\emptyset\}$,
2. if $e = l$ for literal l, then $\mathit{Eff}(e, s) = \{\{l\}\}$,
3. if $e = (e_1 \wedge \cdots \wedge e_n)$, then $\mathit{Eff}(e, s) = \{\cup_{i=1}^{n} E_i : E_i \in \mathit{Eff}(e_i, s)\}$,

[2] The EXPSPACE-completeness of conformant planning for fully observable problems is from (Haslum & Jonsson 1999).

4. if $e = (e_1 \oplus \cdots \oplus e_n)$, then $\mathit{Eff}(e,s) = \bigcup_{i=1}^n \mathit{Eff}(e_i,s)$,
5. if $e = (c \rhd e')$, then $\mathit{Eff}(e,s) = \mathit{Eff}(e',s)$ if $s \models c$ and $\mathit{Eff}(e,s) = \{\emptyset\}$ otherwise.

The application of effects is defined as:

$$Appl(E,s) \stackrel{\text{def}}{=} \{e \cup s' : e \in E, s' = s \setminus \cup_{p \in e}\{p, \neg p\}\}.$$

When there are no conditional neither non-deterministic effects, and I determines a unique state, P is an ADL planning problem (Pednault 1989). Further, if G and all preconditions are conjunctions of positive literals, P is a STRIPS planning problem (Fikes & Nilsson 1971). If the set of observables equals D, P is a fully observable non-deterministic planning problem which can be thought as a non-deterministic Markov Decision Process (MDP) (Puterman 1994; Bertsekas 1995). If a solution with no branch points is sought for a fully observable non-deterministic planning problem, we call problem a conformant planning problem with full observability (Goldman & Boddy 1996; Smith & Weld 1998).[3] If $Z \neq D$ then P is a contingent planning problem which can be thought as a non-deterministic POMDP (Kaelbling, Littman, & Cassandra 1999).

The form of valid plans vary with the nature of the problem. For ADL and conformant planning problems with full observability, a plan is a linear sequence of operators that achieves the goal no matter what is the initial state and the non-determinism involved, and hence plans can be recovered by search in state space for ADL problems (McDermott 1996; Bonet & Geffner 2001; Hoffmann & Nebel 2001) or search in belief space for conformant problems (Bonet & Geffner 2000; Rintanen 2004b). For fully observable or contingent problems, valid plans are functions that map states into operators for MDPs, and belief states into operators for POMDPs. In these cases, a valid plan can be recovered through different means: dynamic programming over state space or belief space (Puterman 1994; Bertsekas 1995; Sondik 1978; Cassandra, Littman, & Kaelbling 1994), AND/OR search in state or belief space (Hansen & Zilberstein 2001; Bonet & Geffner 2003),and other techniques as well, e.g. (Majercik & Littman 1998).

We are interested in checking whether there exists a valid plan for an input planning problem P, the so-called plan-existence decision problem. Since different subclasses of planning problems have different properties, it is better to divide the general plan-existence task into subtasks. Thus, there are decision problems for ADL (or STRIPS) planning, for conformant planning, and so on.

In the rest of this section, we give formal definitions for belief states, contingent and conformant plans, and the propositional modal logic to be used. The plan existence problems together with a brief summary of complexity theory are presented in the next section.

[3] Traditionally, conformant planning problems are associated with problems with null observability, i.e. $Z = \emptyset$. We think however that ‘conformance’ is a property of the solutions (plans) and not of the planning problem itself. As we will see, this difference only surfaces when conformant plans for partially observable problems with modal formulae are required.

Belief States and Plans

A belief state is a subset of states that represent a possible situation at a given moment. If b denotes a belief state, an operator $o = \langle c, e\rangle$ is said to be applicable in b if it’s applicable in all $s \in b$. The result of applying o in b is the belief state $b_o \stackrel{\text{def}}{=} Res(o,b) \stackrel{\text{def}}{=} \cup_{s\in b} Res(o,s)$. A belief state b is said to be a goal belief if every state in b is a goal state.

A sequence of operators $\langle o_0,\ldots,o_n\rangle$ is said to be applicable in belief b_0 if o_0 is applicable in b_0, and if $b_{i+1} \stackrel{\text{def}}{=} Res(o_i,b_i)$ then o_{i+1} is applicable in b_{i+1} for all $0 \leq i < n$. The result of applying such sequence in b_0 is $Res(\langle o_0,\ldots,o_n\rangle, b_0) \stackrel{\text{def}}{=} b_{n+1}$.

Let $P = \langle D, I, G, O, Z\rangle$ be a planning problem with full observability, i.e. $Z = D$. Define b_I as the set of initial states, i.e. $b_I \stackrel{\text{def}}{=} \{s : s \models I\}$. A sequence $\pi = \langle o_1,\ldots,o_n\rangle$ is said to be a *conformant plan* for P iff π is applicable in b_I, and $Res(\pi, b_I)$ is a goal belief.

A plan with bounded branching for P has a tree-like structure. A conformant plan is a plan with 0 branch points or 0-plan. A k-plan for b_0, a valid plan with at most k branch points, is a rooted, labeled and directed tree $T = \langle V, E, r, \ell\rangle$, where $r \in V$ is the root and ℓ a labeling of nodes and edges, defined inductively as:

1. Every directed path in π has length $\leq k$,
2. $\ell(n)$ is a sequence of operators for all $n \in V$,
3. $\ell(e)$ is a state for all $e \in E$,
4. the sequence $\ell(r)$ is applicable in b_0,
5. if $(r,n) \in E$ then $\ell(r,n) \in Res(\ell(r), b_0)$,
6. for every $s \in Res(\ell(r), b_0)$ there is a node $n_s \in V$ such that $(r,n_s) \in E$ and $\ell(r,n_s) = s$, and
7. for every n such that $(r,n) \in E$, the subtree rooted at n is a $(k-1)$-plan for the belief state $\{\ell(r,n)\}$.

That is, the labels on nodes give sequence of actions to be applied after which the true state of the system is observed and a branch is taken. The labels on the edges give the possible states that might arise at each branch.

Conditions 1–7 characterize valid plans with at most k branch points for planning problems with full observability. However, for our hardness proofs, we require that k-plans include an *explicit* branch operator o_{branch} such that the plan ‘branches’ on and only on them, i.e. we require

8. $\ell(r)$ is non-empty with last element equal o_{branch}.

Branch operators are regular operators in the sense that might have preconditions and effects and thus are not just a simple technical convenience. Indeed, with the proper use of preconditions and effects, a problem specification can impose constraints on when branch points are allowed.

In partially observable domains, i.e. $Z \neq D$, the application of an operator o in a belief state b is accompanied by an observation z. The belief state b_o^z that results after applying o and obtaining z is defined as the subset of states in b_o that are compatible with z. The set of observations *obtainable* after applying o in b is denoted as $Z_{b,o} \stackrel{\text{def}}{=} \{z : \exists(s \in b_o)[s \models z]\}$.

Since there can be more than one observation after the application of an operator, multiple observations generate multiple successor beliefs and hence branching in belief space.

A conformant plan for a partially observable problem is a sequence of operators $\pi = \langle o_0, \ldots, o_n \rangle$. The difference with respect to full observability is that π must map the initial belief state into a *subset* of goal belief states. Formally, define $B_0 = \{b_I\}$, and

$$B_{i+1} \stackrel{\text{def}}{=} Res(o_i, B_i) \stackrel{\text{def}}{=} \{b^z_{o_i} : b \in B_i, z \in Z_{b,o_i}\}$$

if o_i is applicable in each $b \in B_i$. The sequence π is said to be applicable in B_0 if each o_i is applicable in B_i. Then, π is a *conformant plan* for P if it is applicable in B_0 and each $b \in B_{n+1}$ is a goal belief state.

The k-plans $T = \langle V, E, r, \ell \rangle$ for B_0 are defined similarly as before:

1′. Every directed path in π has length $\leq k$,
2′. $\ell(n)$ is a sequence of operators for all $n \in V$,
3′. $\ell(e)$ is a belief state for all $e \in E$,
4′. the sequence $\ell(r)$ is applicable in B_0,
5′. if $(r, n) \in E$ then $\ell(r, n) \in Res(\ell(r), B_0)$,
6′. for every $b \in Res(\ell(r), B_0)$ there is a node $n_b \in V$ such that $(r, n_b) \in E$ and $\ell(r, n_b) = b$,
7′. for every n such that $(r, n) \in E$, the subtree rooted at n is a $(k-1)$-plan for the set $\{\ell(r, n)\}$, and
8′. $\ell(r)$ is non-empty with last element equal o_{branch}.

Propositional Modal Logic

We extend the standard propositional logic with three modalities $\Box$, $\Diamond$ and $*$. Well formed formulae are built up recursively using the standard logic connectives plus $\Box\varphi$, $\Diamond\varphi$ and φ^* for formula φ.

As usual in modal logic, $\Box$ represents necessity and $\Diamond$ represents possibility. The $*$ is used to evaluate formulae with respect to *previous frames of reference*.

Modal formulae are interpreted (evaluated) in states s belonging to a belief state b. The interpretation is defined with respect to triplets (s, b, σ) where s is a state, b is a belief state, and σ is a *sequence* of states from b used as the frames of reference. Standard logic connectives are interpreted in the usual way, propositions and modalities as:

I1. $(a, b, \sigma) \models p$ iff $s \models p$,
I2. $(s, b, \sigma) \models \Box\varphi$ iff $(s', b, s\sigma) \models \varphi$ for all $s' \in b$,
I3. $(s, b, \sigma) \models \Diamond\varphi$ iff $(s', b, s\sigma) \models \varphi$ for some $s' \in b$,
I4. $(s, b, s'\sigma) \models \varphi^*$ iff $(s', b, \sigma) \models \varphi$, and
I5. $(s, b, \langle\rangle) \not\models \varphi^*$ for all φ.

Observe that σ is used as an stack of contexts such that $\Box$ and $\Diamond$ push contexts into σ while $*$ pops them out.

A formula φ holds in a state s with respect to b iff $(s, b, \langle\rangle) \models \varphi$, and it holds in belief b iff it holds in all $s \in b$ (with respect to b).

Modal logic extends the expressivity in planning since it allows, for example, to express preconditions or goals of the form 'know φ', which holds in b when all states in b agree on the interpretation of φ, and 'possibly φ' which holds in b when some state in b satisfies φ. Indeed, 'know φ' is equivalent to $\Box\varphi \vee \Box\neg\varphi$, and 'possibly φ' is equivalent to $\Diamond\varphi$.

The $*$ is less standard but it can be related to a restricted form of quantification when combined with $\Box$ (universal) and $\Diamond$ (existential) since $*$ provides a "handle" into the appropriate quantified variables. For example, $\Box\Diamond(p^* \leftrightarrow q)$ holds in belief state b iff for all $s \in b$ there is $s' \in b$ such that the value of p in s coincides with the value of q in s'.

Since a belief state is an explicit representation of models, the truth of $b \models \varphi$ can be decided in polynomial space, in $|b|$ and $|\varphi|$, using a stack and definitions I1–I4.

In the Mastermind game, the codebreaker's goal is achieved when it reaches a belief state that is a singleton; that is, when all the propositions that denote the colors of the code are known. Formally, if C is the set of propositions that denote the colors of the pegs, the goal in Mastermind can be coded as $\bigwedge_{p \in C} \Box p \vee \Box\neg p$.

Complexity and Decision Problems

The reader is referred to any book on complexity theory for basic definitions; e.g. (Papadimitriou 1993; Du & Ko 2000). In this section, we introduce some notation and known results, and formally define the plan-existence decision tasks.

We only consider Turing Machines (TM) with semi-infinite tape that halt on all inputs. The language accepted by a TM M is denoted by $L(M)$. Henceforth, DTM and NTM will denote a deterministic and non-deterministic TM respectively. DTIME$(t(n))$ (resp. NTIME$(t(n))$) is the class of all languages accepted by DTMs (resp. NTMs) with time bound $t(n)$. DSPACE$(s(n))$ (resp. NSPACE$(s(n))$) is the class of all languages accepted by DTMs (resp. NTMs) with space bound $s(n)$. The following complexity classes are standard:

$$\begin{aligned}
\text{PSPACE} &\stackrel{\text{def}}{=} \cup_{k\geq 0}\text{DSPACE}(n^k)\,, \\
\text{EXPTIME} &\stackrel{\text{def}}{=} \cup_{k\geq 0}\text{DTIME}(2^{n^k})\,, \\
\text{EXPSPACE} &\stackrel{\text{def}}{=} \cup_{k\geq 0}\text{DSPACE}(2^{n^k})\,, \\
\text{2EXPTIME} &\stackrel{\text{def}}{=} \cup_{k\geq 0}\text{DTIME}(2^{2^{n^k}})\,, \\
\text{2EXPSPACE} &\stackrel{\text{def}}{=} \cup_{k\geq 0}\text{DSPACE}(2^{2^{n^k}})\,.
\end{aligned}$$

Similarly for non-deterministic classes, e.g. NEXPTIME $\stackrel{\text{def}}{=} \cup_{k\geq 0}$NTIME$(2^{n^k})$. Some well-known result are that the non-deterministic 'space' classes are equal to their deterministic counterparts, e.g. EXPSPACE = NEXPSPACE, and that the 'space' classes are closed under complementation, e.g. EXPSPACE = co-EXPSPACE.

A decision problem P is a language. P is in a complexity class $\mathcal{C}$ if there is a TM $M \in \mathcal{C}$ such that $P = L(M)$, P is $\mathcal{C}$-hard if for every decision problem $P' \in \mathcal{C}$ there is a polynomial-time many-one reduction f such that $f(\omega) \in P$ iff $\omega \in P'$, and P is $\mathcal{C}$-complete iff $P \in \mathcal{C}$ and P is $\mathcal{C}$-hard. We say that $\mathcal{C}$ is an upper bound (resp. lower bound) on the complexity of P if $P \in \mathcal{C}$ (resp. P is $\mathcal{C}$-hard).

Alternation

An alternating TM (ATM) is a TM whose states are partitioned into existential states and universal states. A TM is an ATM where all states are existential. An ATM M accepts input ω if there is an *accepting computation tree* rooted at the initial configuration such that the existential configurations have one child, the universal configurations have all their children, and all leaves are accepting; see (Chandra, Kozen, & Stockmeyer 1981). An ATM M is $a(n)$-alternation bounded if the number of alternations, i.e. transitions between existential and universal states or vice versa, of any computation path is bounded by $a(|\omega|)$ for all input ω. The following is due to to A. Borodin (Chandra, Kozen, & Stockmeyer 1981).

Theorem 1 (Borodin) *If M is an $s(n)$-space bounded and $a(n)$-alternation bounded ATM with $s(n) \geq \log n$, then $M \in$ DSPACE$(a(n)s(n) + s(n)^2)$.*

Decision Problems in Planning

We consider the following decision problems:

- PLAN-STRIPS: Let P be a STRIPS planning problem. Does P have a valid plan?
- PLAN-ADL: Let P be an ADL planning problem. Does P have a valid plan?
- PLAN-FO-CONT: Let P be a planning problem with full observability. Does P have a valid contingent plan?
- PLAN-FO-CONT-k: Let P be a planning problem with full observability. Does P have a valid contingent plan with at most k branch points?
- PLAN-FO-CONF: Let P be a planning problem with full observability. Does P have a valid conformant plan?
- PLAN-PO-CONT: Let P be a planning problem with partial observability and modalities. Does P have a valid contingent plan?
- PLAN-PO-CONT-k: Let P be a planning problem with partial observability and modalities. Does P have a valid contingent plan with at most k branch points?
- PLAN-PO-CONF: Let P be a planning problem with partial observability and modalities. Does P have a valid conformant plan?

PLAN-STRIPS and PLAN-ADL are known to be PSPACE-complete (Bylander 1994), PLAN-FO-CONT is known to be EXPTIME-complete (Rintanen 2004a),[4] PLAN-FO-CONF is known to be EXPSPACE-complete (Haslum & Jonsson 1999), and PLAN-PO-CONT is known to be 2EXPTIME-complete (Rintanen 2004a) (yet the case of modal formulae isn't studied by Rintanen). The classes PLAN-FO-CONT-k, PLAN-PO-CONT-k and PLAN-PO-CONF are novel; the first decision problem will be shown to be EXPSPACE-complete while the latter two to be 2EXPSPACE-complete.

Upper Bounds on Complexity

We begin with the easy direction, the upper bounds on PLAN-PO-CONT with modalities, and PLAN-FO-CONT-k, PLAN-PO-CONT-k and PLAN-PO-CONF.

[4] The complexity remains EXPTIME-complete even for testing the existence of plans that reach the goal with probability $\geq t$ for probabilistic problems with full observability (Littman 1997).

Observe that with n propositional symbols, there are $\leq 2^n$ planning states and hence a set of states can be represented with $n2^n$ bits, i.e. in exponential space. Also, the truth of a modal formula can be decided in polynomial space in the size of a belief state. Therefore, PLAN-PO-CONT can be solved with an ATM using exponential space such that the solution computation trees of the ATM corresponds to the contingent plans; see proof of Theorem 3 for a hint. Since an ATM with an exponential space bound can be simulated with a DTM with a double exponential time bound (see Chandra, Kozen, & Stockmeyer 1981), we have

Theorem 2 *Deciding the existence of a contingent plan for planning problems with partial observability, with or without modal formulae, is in 2EXPTIME. Since completeness hold for the restricted case of problems without modal formulae, then PLAN-PO-CONT is 2EXPTIME-complete.*

The membership of PLAN-FO-CONT-k in EXPSPACE is shown with an ATM that makes at most k alternations. Note that each branch point in the plan corresponds to determining the current state of the system and then planning thereafter. Since a different plan must be found for each possible state, the branch can be simulated with a transition from a universal state of the ATM. A final simulation of the ATM with a DTM shows the membership in EXPSPACE.

Theorem 3 *PLAN-FO-CONT-k is in EXPSPACE.*

Proof: Let $P = \langle D, I, G, O, D \rangle$ be a contingent planning problem with full observability with $|D| = n$. The following ATM, with $\leq k$ alternations, decides if there is a k-plan:

1. $K := k$;
2. $b := \{s : I \models s\}$;
3. $steps := 0$;
4. **if** $\forall(s \in b)[s \models G]$ **then** ACCEPT;
5. $\exists$-branch: **choose** either APPLY or BRANCH;
6. **if** BRANCH **then**
7.   **if** $K = 0$ **then** REJECT;
8. $K := K - 1$;
9. $\forall$-branch: **for each** $s \in b$ **do** $b := \{s\}$;
10. **else if** APPLY **then**
11.   **if** $steps = 2^{2^n}$ **then** REJECT;
12. $\exists$-branch: **choose** operator $\langle c, e \rangle$ s.t. $\forall(s \in b)[s \models c]$;
13. $b := \bigcup_{s \in b} Appl(Eff(s, e), s)$;
14. $steps := steps + 1$;
15. **end**
16. **goto** 4.

The ATM is in EXPSPACE since there are at most 2^n states so a subset of states can be stored in $O(n2^n)$ bits. Use now Theorem 1 and the fact that k is constant to get the result. $\square$

The same idea works for the membership of PLAN-PO-CONT-k in 2EXPSPACE yet a subset of belief states is explicitly stored instead of a belief state, and thus the ATM requires $O(n2^n2^{2^n})$ bits.

Theorem 4 *Deciding the existence of a contingent k-plan for planning problems with partial observability, with or without modal formulae, is in 2EXPSPACE.*

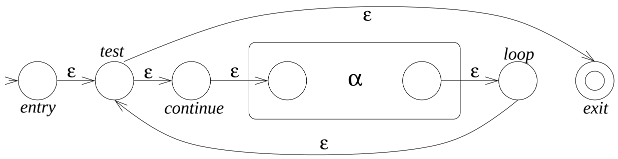

Figure 2: Non-deterministic Finite Automaton with Counters for α^n.

Finally, the membership of PLAN-PO-CONF in 2EXPSPACE is easily shown with an NTM that chooses applicable operators non-deterministically and keeps track of the subset of belief states. The machine accepts if all beliefs in the subset are goal and rejects when the number of steps is bigger than $2^{2^{2^n}}$.

Theorem 5 *Deciding the existence of a conformant plan for planning problems with partial observability, with or without modal formulae, is in 2EXPSPACE.*

Following the work of Haslum and Jonsson (1999), we use regular expressions with exponentiation and non-deterministic finite automata with counters to establish lower bounds on the complexity of decision problems.

Regular Expressions and Automata

A regular expression with exponentiation (REE) is built up recursively from atomic regular expressions with the usual concatenation, sum and Kleene star operations plus an additional exponentiation operation of the form α^n, where n is a positive integer *written in binary*, that denotes the language $\{\sigma_1 \ldots \sigma_n : \sigma_i \in \alpha\}$. As usual, the length of a regular expression is equal to the number of symbols in it.

Theorem 6 (Hopcroft & Ullman 1979) *Let α be an REE and Σ_α the alphabet of α. Deciding if $\alpha = \Sigma_\alpha^*$ is EXPSPACE-hard.*

Haslum and Jonsson use non-deterministic automata with counters (NFAC) to represent regular expressions with exponentiation. A NFAC is a non-deterministic finite automaton augmented with a set of bounded counters C. Each counter $c \in C$ is associated with five states $entry_c$, $test_c$, $continue_c$, $exit_c$ and $loop_c$, and with a bound $bound_c$ *written in binary*. Initially, all counters are set to zero. Upon visiting $entry_c$ the counter is initialized to $bound_c$ and a transition is made to $test_c$; this is the only possible transition from $entry_c$. In $test_c$, the machine tests $c > 0$ and makes a transition to $continue_c$ if it holds and to $exit_c$ otherwise; these are the only transitions from $test_c$ and into $continue_c$ and $exit_c$. Upon visiting $loop_c$, the machine decrements c and makes a transition to $test_c$; the only transitions to $test_c$ are either from $entry_c$ or $loop_c$.[5] Fig. 2 shows the NFAC for α^n.

Let Q be the set of states of an NFAC and $F \subseteq Q$ the set of its accepting states. An instantaneous description (ID) of an NFAC is a tuple in $Q \times \mathsf{X}_{c\in C}[bound_c]$ where $[bound_c]$ denotes the set $\{0, \ldots, bound_c\}$; an ID is *accepting* if its state is final. We denote with Θ the set of all IDs, with Θ_a the set of all accepting IDs, and $\Theta_n = \Theta \setminus \Theta_a$. Formally, an NFAC is a tuple $M = \langle Q, \Sigma, \delta, q_0, F, C\rangle$ where $\delta : Q \times \Sigma \to 2^Q$ is the transition function and C is a set of tuples $\langle c, bound_c, entry_c, test_c, continue_c, exit_c, loop_c\rangle$. The size of M is defined as $|\delta| + \sum_{c\in C} |bound_c|$ where $|bound_c|$ is the number of bits in the binary representation of $bound_c$.

Theorem 7 (Haslum & Jonsson 1999) *For any REE α there exists an NFAC M_α of polynomial size in $|\alpha|$ such that $\omega \in \alpha$ iff $\omega \in L(M_\alpha)$.*

As is standard in NFAs, the domain of definition of δ can be extended from $\delta : Q \times \Sigma \to 2^Q$ into $\overline{\delta} : \Theta \times \Sigma^* \to 2^\Theta$ such that $\omega \in L(M)$ iff $\overline{\delta}(\theta_0, \omega)$ contains an accepting ID where $\theta_0 \stackrel{\text{def}}{=} \langle q_0, 0, \ldots, 0\rangle$ is the *initial* ID.

Haslum and Jonsson proof's of EXPSPACE-hardness for conformant planning with full observability is as follows. Given an REE α, they construct a planning problem $P = \langle D, I, G, O, D\rangle$ that *simulates* the transition function $\overline{\delta}$ of the NFAC M_α in a way that states in P corresponds to IDs of M_α, and that (non-deterministic) planning operators o_a correspond to transitions in M_α upon $a \in \Sigma \cup \{\epsilon\}$ such that

$$Res(\langle o_{a_1}, \ldots, o_{a_n}\rangle, I) \;=\; \overline{\delta}(\theta_0, a_1 \ldots a_n)\,.$$

Hence, if G is defined as being in a non-accepting ID then P has a valid plan $\langle o_{a_1}, \ldots, o_{a_n}\rangle$ iff $a_1 \ldots a_n \notin L(M_\alpha)$. This argument shows that deciding the existence of a conformant plan is co-EXPSPACE-hard. The result then follows with an application of EXPSPACE $=$ co-EXPSPACE.

Compact Representation of Counters

In this section, we show how to encode a counter of exponential length with the belief states of a planning problem with a polynomial number of propositional symbols. This encoding is the main tool behind the 2EXPSPACE-hardness proof of conformant planning for partially observable domains.

Let c be a counter of exponential length with bits c_i for $0 \le i < 2^n$, and consider a planning problem P with n propositional "bit markers" $m_{c,k}$ for $0 \le k < n$. The value of counter c can be represented with the positions of the bits set to 1, i.e. with the set $\{i : c_i = 1\}$. Since each position can be encoded with n bits, the value of c can be represented with the belief state[6]

$$b_c \;=\; \{\{m_{c,k} : (k\text{th bit in } [i]_{\text{bit}}) = 1\} : c_i = 1\}\,.$$

For example, $\{\emptyset\}$ represents the value $1 = 00000001_{\text{bin}}$, $\{\{m_{c,0}\}, \{m_{c,2}\}\}$ the value $18 = 00010010_{\text{bin}}$, and $\{\emptyset, \{m_{c,0}, m_{c,1}, m_{c,2}\}\}$ the value $129 = 10000001_{\text{bin}}$. Note however that 0 is represented by the empty belief state, yet since empty beliefs are not allowed, the value 0 cannot be encoded. Therefore, all counter values will be *shifted* by 1.

Let b_c be a belief state representing counter c, we will need to test whether the value of c is equal to or greater than

[5] Our NFACs are slightly different from Haslum and Jonsson's that allow for simpler proofs.

[6] The notation $[k]_{\text{bit}}$ stands for the integer k written in binary.

1. The former holds iff $b_c = \{\emptyset\}$ while the latter iff b_c has a state satisfying some $m_{c,k}$; these are abbreviated as

$$c = 1 \ : \ \bigwedge_{k=0}^{n} \neg m_{c,k} \quad \text{and} \quad c > 1 \ : \ \Diamond \bigvee_{k=0}^{n} m_{c,k} \, .$$

Similarly, we will need two operations. The first, denoted by $c+k$, is to set all 1-bits in the binary expansion of k to 1. It is implemented with the effect:

$$c + k \ : \ \bigwedge \{ m_{c,i} : (i\text{th bit in } [k]_{\text{bit}}) = 1 \} \, .$$

The second operation, denoted by dec_c, is to decrement the value of the counter c when $c > 1$. For example, it must change the value $72 = 01001000_{\text{bin}}$ represented by

$$b_{72} \ = \ \{\{m_{c,0}, m_{c,1}\}, \{m_{c,1}, m_{c,2}\}\}$$

to the value $71 = 01000111_{\text{bin}}$ represented by

$$b_{71} \ = \ \{\emptyset, \{m_{c,0}\}, \{m_{c,1}\}, \{m_{c,1}, m_{c,2}\}\} \, .$$

That is, the set $\{m_{c,0}, m_{c,1}\}$ should be replaced by the three sets $\emptyset$, $\{m_{c,0}\}$ and $\{m_{c,1}\}$. The general principle here is to replace the subset for the *least significant bit* (the first 1 from right to left in the binary expansion) with the collection of subsets that correspond to all bits of lesser significance. We thus need a formula to identify the subset to replace, and an effect to generate the replacement subsets. It is not hard to see that the subset to replace is the only one that satisfies $\Box\phi_{c,n}$ where

$$\phi_{c,k} \ : \ [(m_{c,k} \wedge \neg m_{c,k}^{*}) \vee ((m_{c,k} \leftrightarrow m_{c,k}^{*}) \wedge \phi_{c,k-1})] \, ,$$
$$\phi_{c,0} \ : \ [(m_{c,0} \wedge \neg m_{c,0}^{*}) \vee (m_{c,0} \leftrightarrow m_{c,0}^{*})] \, .$$

Indeed, $\Box\phi_{c,n}$ holds in $s \in b$ iff for all $s' \in b$, either $s = s'$ or there is k such that $\neg m_{c,k} \in s$, $m_{c,k} \in s'$ and $m_{c,l} \in s$ iff $m_{c,l} \in s'$ for all $l > k$.

Once the subset s to replace is identified, all replacements can be generated by non-deterministically processing all bits $m_{c,k}$ as follows. If $m_{c,k} \in s$, then it's cleared and all bits $m_{c,j}$, $0 \leq j < k$, are flipped non-deterministically, or else all bits $m_{c,j}$, $0 \leq j \leq n$, are set to zero. The decrement effect is thus defined as

$$\begin{aligned} \text{dec}_c \ : \ & \Box\phi_{c,n} \rhd \\ & \bigoplus_{k=0}^{n} \left[(m_{c,k} \rhd \neg m_{c,k} \wedge \bigwedge_{j=0}^{k-1} (m_{c,j} \oplus \neg m_{c,j})) \wedge \right. \\ & \qquad \left. (\neg m_{c,k} \rhd \bigwedge_{j=0}^{n} \neg m_{c,j}) \right] . \end{aligned}$$

In the example, where $s = \{m_{c,0}, m_{c,1}\}$, the processing of $m_{c,0}$ generates $\{m_{c,1}\}$, the processing of $m_{c,1}$ generates $\emptyset$ and $\{m_{c,0}\}$, and the processing of the other bits generates $\emptyset$, hence the effect of dec_c on the belief b_{72} is the belief b_{71}.

As a final remark, observe that multiple counters can be encoded simultaneously by taking the "cross-product" of the encodings for the individual counters.

Lower Bounds on Complexity

The hardness proofs are similar to Haslum and Jonsson's. However, for the case of 1-plans, observe that determining the current planning state amounts to determining the current ID of the automaton M_α. Thus, there is a valid 1-plan for P iff there exists $\omega \in \Sigma^*$ such that

$$\forall(\theta \in \overline{\delta}(\theta_0, \omega)) \exists (w' \in \Sigma^*) [\overline{\delta}(\theta, w') \subseteq \Theta_n] \, ,$$

yet this condition is strictly weaker than $\alpha \neq \Sigma_\alpha^*$ and thus a naive reduction would fail in one direction.

We fix this problem by inserting effects into the branch operators. Indeed, let α be an REE with alphabet Σ and M be an NFAC such that $L(M_\alpha) = \alpha$. As done in the proof of Haslum and Jonsson, construct a planning problem P with full observability such that the states of P encode the IDs of M: the operators of P, of the form o_a for $a \in \Sigma \cup \{\epsilon\}$, simulate the transitions of M with *conditional effects*. Extend P into P' with a binary counter $branch \in \{0, \ldots, k\}$ and with an o_{branch} operator with precondition $branch < k$ and a single effect of increasing the value of $branch$. Since the range of $branch$ is bounded by a constant, the size of o_{branch} is constant. The descriptions I and G are extended such that the unique initial state, that corresponds to θ_0, makes $branch = 0$, and that the goal states are those that correspond to non-accepting IDs and $branch = k$. Finally, add the precondition $branch = 0$ to all operators o_a with $a \in \Sigma \cup \{\epsilon\}$.

We claim that P' has a k-plan, with exactly k branch points, iff $\alpha \neq \Sigma_\alpha^*$. First, assume there is a word $w = a_1 \ldots a_n \notin \alpha$, i.e. $\overline{\delta}(\theta_0, w) \subseteq \Theta_n$. Therefore, the plan $\langle o_{a_1}, \ldots, o_{a_n} \rangle$ is a conformant plan for P and hence generates in P' a set of states associated with non-accepting IDs and $branch = 0$. Extend this plan with k applications of o_{branch} to get a k-plan for P'.

On the other hand, assume P' has a k-plan T with root r. Let $\tau = \langle o_1, \ldots, o_{n+k} \rangle$ be the sequence of operators associated with a maximal directed path in T (a trajectory in the execution of plan T). Since τ makes $branch = k$, it contains k branch operators that must be all consecutive given that o_{branch} increases the counter and that all operators o_a have precondition $branch = 0$. Therefore, by definition of T, the first n operators from τ are of the form $\langle a_1, \ldots, a_n \rangle$, with $a_i \in \Sigma \cup \{\epsilon\}$, and will appear in any trajectory; i.e. $\ell(r) = \langle o_{a_1}, \ldots, o_{a_n}, o_{branch} \rangle$. Since the branch operators doesn't change the state of the automata M, we have that $\ell(r)$ except the last o_{branch} is a plan for P and hence the word $a_1 \ldots a_n \notin \alpha$. We have shown

Theorem 8 *Deciding the existence of a contingent k-plan for planning problems with full observability is EXPSPACE-hard, and hence EXPSPACE-complete.*

The main result of the paper is the 2EXPSPACE-hardness of conformant planning for partially observable domains. The proof idea is essentially the same except that we have to deal with counters of exponential length. To understand the proof, let's revise the proof of Theorem 6. Given a DTM M with an exponential space bound and a word ω, an REE $\alpha = \alpha(M, \omega)$ of polynomial length in $|M| + |\omega|$ is constructed such that $\alpha = \Sigma_\alpha^*$ iff $\omega \notin L(M)$. A closer look at α reveals that if M has a double exponential space bound then α would be the same except that the exponents are of exponential length (Hopcroft & Ullman 1979). Therefore, the associated NFAC have counters of double exponential capacity that must be encoded compactly.

Theorem 9 *Deciding the existence of a conformant plan for problems with partial observability and modal formulae is 2EXPSPACE-hard, and hence 2EXPSPACE-complete.*

Proof: Let M be a DTM with a double exponential space bound $d(x)$ and $\omega \in \Sigma^*$. Consider the REE $\alpha = \alpha(M, \omega)$ given in the proof of Theorem 6. The size of α is exponential yet, if the sizes of the exponents are not measured, $|\alpha|$ is polynomial in $|M| + |\omega|$. Therefore, if $M_\alpha = \langle Q, \Sigma, \delta, q_0, F, C\rangle$ is the NFAC associated to α, the size $|\delta|$ is polynomial in $|M| + |\omega|$ while $\sum_{c\in C} |bound_c|$ is exponential. We are going to build a conformant planning problem with partial observability of polynomial length to simulate the NFAC. The main idea is to encode the IDs with beliefs states.

Let $n = poly(|\omega|)$ be such that $d(|\omega|) < 2^{2^n}$ and define the planning problem $P_\alpha = \langle D, I, G, O, Z\rangle$ with propositional symbols:

1. machine state symbols $\{q : q \in Q\}$,
2. bits marker symbols $\{m_{c,k} : c \in C, 0 \le k \le n\}$,
3. set symbols $\{s_{c,k} : c \in C, 0 \le i \le \lceil \log(|\omega| + 1)\rceil\}$,
4. initialize symbols $\{u_c : c \in C\}$, and
5. decrement symbols $\{d_{c,k} : c \in C, 0 \le k \le 2|\omega| + 1\}$.

The observable symbols are the state symbols, i.e. $Z = \{q : q \in Q\}$. The descriptions I and G reflect the initial ID and all non-accepting IDs respectively,

$$I \stackrel{\text{def}}{=} q_0 \wedge \bigwedge_{q\neq q_0} \neg q \wedge \bigwedge_{c,k} \neg u_c \wedge \neg m_{c,k} \wedge \neg s_{c,k} \wedge \neg d_{c,k}\,,$$

$$G \stackrel{\text{def}}{=} \bigvee_{q\in Q\setminus F} q\,.$$

That is, I encodes the ID with state q_0 and the value of all counters set to 1 (which means 0 since the counters values are shifted by 1).

Let $q \in Q \setminus \{entry_c, test_c, loop_c : c \in C\}$ and $a \in \Sigma \cup \{\epsilon\}$. The transition $\delta(q, a)$ is modeled with a non-deterministic operator

$$o_{q,a} \stackrel{\text{def}}{=} \langle q, \neg q \wedge \textstyle\bigoplus_{p\in\delta(q,a)} p\rangle\,.$$

The transition from $test_c$ is deterministic either to $continue_c$ or $exit_c$ whether $c > 1$; i.e.

$$o_{test_c,\epsilon} \stackrel{\text{def}}{=} \langle test_c, \neg test_c \wedge (c > 1 \rhd continue_c) \wedge (c = 1 \rhd exit_c)\rangle$$

The transition from $loop_c$ is deterministic that changes state to $test_c$ and decrements the counter c; i.e.

$$o_{loop_c,\epsilon} \stackrel{\text{def}}{=} \langle loop_c, \neg loop_c \wedge test_c \wedge \text{dec}_c\rangle\,.$$

The transition from $entry_c$ must set $c = 1 + bound_c$ and change state into $test_c$. In general, we cannot set a counter of double exponential capacity to an arbitrary value using a polynomial number of operators each of polynomial length. However, the exponents in α are in $\{1, \ldots, |\omega| + 1, 2^{2^n} - |\omega| - 1, 2^{2^n} - 1, 2^{2^n} + 1, 2^{2^n} + |\omega|\}$ (Hopcroft & Ullman 1979), and these values can be set with polynomially long operators. Indeed, define the entry operator as

$$o_{entry_c,\epsilon} \stackrel{\text{def}}{=} \langle entry_c, \neg entry_c \wedge u_c \wedge c = 1\rangle$$

which sets the counter to 1 and changes state to a temporary state u_c from which the counter is initialized to the value $1 + bound_c$.

Counters with an initial value of polynomial size, i.e. those with $1 \le bound_c \le |\omega| + 1$ can be set directly with the operators

$$o_{c,u} \stackrel{\text{def}}{=} \langle u_c, \neg u_c \wedge \textstyle\bigoplus\{s_{c,k} : (k\text{th bit of } [1 + bound_c]_{\text{bit}}) = 1\}\rangle\,,$$

$$o_{c,s} \stackrel{\text{def}}{=} \langle \textstyle\bigvee_k s_{c,k}, test_c \wedge \bigwedge_k (s_{c,k} \rhd \neg s_{c,k} \wedge (c + k))\rangle\,.$$

The first operator creates a set in the belief state for each 1-bit in the value of the counter, and the second operator sets the marker bits appropriately in each state. Observe that these two operators must be executed in such order right after the application of $o_{entry_c,\epsilon}$.

Counters with an initial value of exponential size, i.e. those with $bound_c \in \{2^{2^n} - |\omega| - 1, 2^{2^n} - 1, 2^{2^n} + 1, 2^{2^n} + |\omega|\}$, are first set to $2^{2^n} + |\omega| + 1$ and then decremented $2^{2^n} + |\omega| - bound_c$ times (which is a polynomial number of decrements):

$$o_{c,u} \stackrel{\text{def}}{=} \langle u_c, \neg u_c \wedge \big[m_{c,n} \oplus \textstyle\bigoplus\{s_{c,k} : (k\text{th bit of } [1 + |\omega|]_{\text{bit}}) = 1\}\big]\rangle\,,$$

$$o_{c,s} \stackrel{\text{def}}{=} \langle m_{c,n} \vee \textstyle\bigvee_k s_{c,k},\ d_{c,2^{2^n}+|\omega|-bound_c} \wedge \bigwedge_k (s_{c,k} \rhd \neg s_{c,k} \wedge (c + k))\rangle\,,$$

$$o_{c,d_k} \stackrel{\text{def}}{=} \langle d_{c,k}, \neg d_{c,k} \wedge d_{c,k-1} \wedge \text{dec}_c\rangle\,,$$

$$o_{c,d_0} \stackrel{\text{def}}{=} \langle d_{c,0}, \neg d_{c,0} \wedge test_c\rangle\,.$$

For example, to set $c = 2^{2^n} - |\omega|$, the counter is set to $1 + 2^{2^n} + |\omega|$ and then decremented $2|\omega| + 1$ times. The representation is faithful in the sense that P_α has a valid conformant plan iff $\alpha \neq \Sigma_\alpha^*$. □

We remark that the simulation of the NFAC works only for certain REEs and won't work in general; the problem being the initialization of counters' values. Whether there is a reduction from the problem $\alpha = \Sigma_\alpha^*$ for general REEs with exponents of exponential length to conformant planning problems with partial observability is an open problem.

Finally, combining the simulation of counters of exponential length with branch operators, we have

Theorem 10 *Deciding the existence of a contingent k-plan for planning problems with partial observability and modal formulae is 2EXPSPACE-hard, and hence 2EXPSPACE-complete.*

Special Cases

Turner (2002) studies the complexity of deciding the existence of plans of polynomial length using quantified boolean formulae (QBFs). He shows that deciding the existence of conformant plans of polynomial length for fully observable problems is Σ_3^p-complete, and that deciding the existence of contingent plans of polynomial length is PSPACE-complete for problems with either full or partial observability.

We take a similar approach here and study the complexity of deciding the existence of plans of polynomial length[7] for fully observable problems, and the existence of plans for partially observable domains without modalities.

Let us consider a planning problem P with full observability, a fixed planning horizon n, and the codification of P into propositional logic. The codification uses symbols o_i for operators and f_i for fluents tagged with time indices such that o_i (resp. f_i) stands for an operator (resp. fluent) at time $0 \leq i \leq n$. For lack of space, we only present the main results without proofs.

Turner shows that P has a conformant plan of length n iff a QBF of the form

$$\exists o_0 \cdots \exists o_{n-1} \forall f_0 \cdots \forall f_n \exists f'_1 \cdots \exists f'_n . \Psi \tag{2}$$

is valid. Therefore, if n is polynomial then (2) is in Σ_3^p. The existence of a k-plan with branch points at times $0 < t_1 < t_2 < \cdots < t_k < n$ can be verified with a QBF of the form

$$\begin{aligned}&\exists o_0 \cdots \exists o_{t_1 - 1} \forall f_0 \cdots \forall f_{t_1}\\ &\exists o_{t_1} \cdots \exists o_{t_2 - 1} \forall f_{1+t_1} \cdots \forall f_{t_2} \cdots\\ &\exists o_{t_k} \cdots \exists o_{n-1} \forall f_{1+t_k} \cdots \forall f_n \exists f'_0 \cdots \exists f'_n . \Psi\,.\end{aligned} \tag{3}$$

If n is polynomial, then (3) is in Σ_{2k+3}. Since the positions of the branch points can be chosen non-deterministically, the decision problem can be solved with an (oracle) TM in $NP(\Sigma_{2k+3}^p) = \Sigma_{2k+4}^p$.

Theorem 11 *Deciding the existence of a contingent k-plan of polynomial length for planning problems with full observability is in Σ_{2k+4}^p.*

The other special case considered is that of partially observable problems without modalities. Let's begin with the existence of conformant plans. Let B be a subset of belief states, as defined in the Sect. on Belief States and Plans, and o an operator. By definition, o is applicable on B if it is applicable in all $b \in B$. Therefore, since o contains no modalities, o is applicable on B if it is applicable on all states in $\bigcup_{b \in B} b$. Similarly, B is a goal set if all $b \in B$ is a goal belief, and thus, since there are no modalities, B is a goal set if $\bigcup_{b \in B} b$ is a goal belief. In conclusion, the existence of a conformant plan for problems without modalities can be established by considering belief states instead of subsets of belief states.[8]

Theorem 12 *Deciding the existence of a conformant plan for planning problems with partial observability without modal formulae is EXPSPACE-complete. Deciding the existence of a conformant plan of polynomial length for planning problems with partial observability without modal formulae is Σ_3^p-complete.*

The case of k-plans for partially observable problems without modalities is similar to the case of full observability. If the set of fluents f is partitioned into the observables f^Z and the non-observables f^N, the existence of a k-plan of length n can be checked with a QBF of the form

$$\begin{aligned}&\exists o_0 \cdots \exists o_{t_1 - 1} \forall f_0^Z \cdots \forall f_{t_1}^Z\\ &\exists o_{t_1} \cdots \exists o_{t_2 - 1} \forall f_{1+t_1}^Z \cdots \forall f_{t_2}^Z \cdots\\ &\exists o_{t_k} \cdots \exists o_{n-1} \forall f_{1+t_k} \cdots \forall f_n \forall f_0^N \cdots \forall f_{t_k}^N \exists f'_0 \cdots \exists f'_n . \Psi\end{aligned} \tag{4}$$

where $0 < t_1 < t_2 < \cdots < t_k < n$ are the branch points.

Theorem 13 *Deciding the existence of a contingent k-plan of polynomial length for planning problems with partial observability without modal formulae is in Σ_{2k+4}^p.*

Discussion

We have introduced novel decision tasks for checking the existence of conformant plans for partially observable domains, and the existence plans of bounded branching for problem with either full or partial observability. The complexity of the former task is shown to be 2EXPSPACE-complete for problems with modal formulae and EXPSPACE-complete for problems without modal formulae. The complexity of deciding the existence of plans with a bounded number of branch points is EXPSPACE-complete for fully observable domains, and 2EXPSPACE-complete for partially observable domains. If plans of polynomial length are sought, then all decision problems can be reduced to QBFs and hence are in PSPACE.

The 2EXPSPACE-hardness for conformant tasks is shown with a reduction from a subclass of the decision problem of checking the totality of regular expression with exponentiation. Rintanen (2004b) makes a direct simulation of TMs with an exponential space bound to show the EXPSPACE-hardness of conformant planning for fully observable problems. A similar simulation of TMs with a double exponential space bound is also possible with partially observable conformant planning problems.

Our hardness results for plans of bounded branching make the assumption that a branch operator appears explicitly in the plans. An open problem is to remove this assumption in the proofs.

We haven't addressed the task of computing plans and/or optimal plans. Since the size of such plans can be fairly large, e.g. triple exponential for the case of conformant plans with partial observability, the complexity of such tasks will increase.

In the future, we expect to develop heuristic search methods for computing plans for some of the tasks. In particular, we are interested in finding plans of bounded branching for fully observable domains and conformant plans for partially observable domains. We think that AND/OR search algorithm in belief space might be good candidates for the former task, while an A* algorithm in the space of subsets of belief states might be good candidate for the latter task. Additionally, algorithms based on QBFs should be good candidates for computing polynomially long plans.

Acknowledgements

I am in gratitude to Patrik Haslum and Héctor Geffner for illuminating discussions and comments on early drafts. Thanks also to the anonymous reviewers.

[7] A k-plan for problem P has polynomial length if there is a polynomial p such that every path in the plan has length $\leq p(|P|)$; i.e. if k is polynomially bounded.

[8] A similar result appears in (Giacomo & Vardi 1999).

References

Bertsekas, D. 1995. *Dynamic Programming and Optimal Control, (2 Vols)*. Athena Scientific.

Bonet, B., and Geffner, H. 2000. Planning with incomplete information as heuristic search in belief space. In Chien, S.; Kambhampati, S.; and Knoblock, C., eds., *Proc. 6th International Conf. on Artificial Intelligence Planning and Scheduling*, 52–61. Breckenridge, CO: AAAI Press.

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence* 129(1–2):5–33.

Bonet, B., and Geffner, H. 2003. Faster heuristic search algorithms for planning with uncertainty and full feedback. In Gottlob, G., ed., *Proc. 18th International Joint Conf. on Artificial Intelligence*, 1233–1238. Acapulco, Mexico: Morgan Kaufmann.

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

Cassandra, A.; Littman, M.; and Kaelbling, L. 1994. Acting optimally in partially observable stochastic domains. In Hayes-Roth, B., and Korf, R., eds., *Proc. 12th National Conf. on Artificial Intelligence*, 1023–1028. Seattle, WA: AAAI Press / MIT Press.

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

Chvatal, V. 1983. Mastermind. *Combinatorica* 3(3–4):325–329.

Du, D., and Ko, K. 2000. *Theory of Computational Complexity*. New York, NY: Wiley-Interscience.

Fikes, R., and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 1:27–120.

Giacomo, G. D., and Vardi, M. Y. 1999. Automata-theoretic approach to planning for temporally extended goals. In Biundo, S., and Fox, M., eds., *Proc. 5th European Conf. on Planning*, 226–238. Durham, UK: Springer: LNCS 1809.

Goldman, R. P., and Boddy, M. S. 1996. Expressive planning and explicit knowledge. In Drabble, B., ed., *Proc. 3rd International Conf. on Artificial Intelligence Planning Systems*. Edinburgh, Scotland: AAAI Press.

Hansen, E., and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence* 129:35–62.

Haslum, P., and Jonsson, P. 1999. Some results on the complexity of planning with incomplete information. In Biundo, S., and Fox, M., eds., *Proc. 5th European Conf. on Planning*, 308–318. Durham, UK: Springer: LNCS 1809.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14:253–302.

Hopcroft, J., and Ullman, J. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Kaelbling, L.; Littman, M.; and Cassandra, A. 1999. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101:99–134.

Littman, M. 1997. Probabilistic propositional planning: representations and complexity. In Kuipers, B., and Webber, B., eds., *Proc. 14th National Conf. on Artificial Intelligence*, 748–754. Providence, RI: AAAI Press / MIT Press.

Majercik, S., and Littman, M. 1998. Maxplan: A new approach to probabilistic planning. In Simmons, R.; Veloso, M.; and Smith, S., eds., *Proc. 4th International Conf. on Artificial Intelligence Planning Systems*, 86–93. Pittsburgh, PA: AAAI Press.

McDermott, D. 1996. A heuristic estimator for means ends analysis in planning. In Drabble, B., ed., *Proc. 3rd International Conf. on Artificial Intelligence Planning Systems*, 150–157. Edinburgh, Scotland: AAAI Press.

Meuleau, N., and Smith, D. 2003. Optimal limited contingency planning. In Meek, C., and Kjaerulff, U., eds., *Proc. 19th Conf. on Uncertainty in Artificial Intelligence*. Acapulco, Mexico: Morgan Kaufmann.

Papadimitriou, C. 1993. *Computational Complexity*. Addison-Wesley.

Pednault, E. 1989. ADL: Exploring the middle ground between Strips and the situation calculus. In Brachman, R.; Levesque, H.; and Reiter, R., eds., *Proc. 1st International Conf. on Principles of Knowledge Representation and Reasoning*, 324–332. Toronto, Canada: Morgan Kaufmann.

Puterman, M. 1994. *Markov Decision Processes – Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc.

Rintanen, J. 2004a. Complexity of planning with partial observability. In Zilberstein, S.; Koenig, S.; and Koehler, J., eds., *Proc. 14th International Conf. on Automated Planning and Scheduling*, 345–354. Whistler, Canada: AAAI Press.

Rintanen, J. 2004b. Distance estimates for planning in the discrete belief space. In McGuinness, D. L., and Ferguson, G., eds., *Proc. 19th National Conf. on Artificial Intelligence*, 525–530. San Jose, California: AAAI Press / MIT Press.

Smith, D., and Weld, D. 1998. Conformant graphplan. In Mostow, J., and Rich, C., eds., *Proc. 15th National Conf. on Artificial Intelligence*, 889–896. Madison, WI: AAAI Press / MIT Press.

Sondik, E. 1978. The optimal control of partially observable Markov decision processes over the infinite horizon: discounted costs. *Operations Research* 26(2).

Turner, H. 2002. Polynomial-length planning spans the polynomial hierarchy. In Flesca, S., and Giovambattista, I., eds., *Proc. 8th European Conf. on Logics in Artificial Intelligence*, 111–124. Cosenza, Italy: Springer: LNCS 2424.