Possibilistic Planning: Representation and Complexity

Célia Da Costa Pereira[1], Frédérick Garcia[1]
Jérôme Lang[2], and Roger Martin-Clouaire[1]

[1] INRA/BIA, Auzeville
BP 27, 31326 Castanet Tolosan cedex France
[2] IRIT, Université Paul Sabatier
118 Route de Narbonne, 31062 Toulouse cedex France

Abstract. A possibilistic approach of planning under uncertainty has been developed recently. It applies to problems in which the initial state is partially known and the actions have graded nondeterministic effects, some being more possible (normal) than the others. The uncertainty on states and effects of actions is represented by possibility distributions.
The paper first recalls the essence of possibilitic planning concerning the representational aspects and the plan generation algorithms used to search either plans that lead to a goal state with a certainty greater than a given threshold or optimally safe plans that have maximal certainty to succeed. The computational complexity of possibilistic planning is then studied, showing quite favorable results compared to probabilistic planning.

1 Introduction

In a "classical" planning problem, it is assumed that actions are deterministic, the initial state is known and the goal is defined by a set of final states; a solution plan is then an unconditional sequence of actions that leads from the initial state to a goal state. However, most practical problems do not satisfy these conditions of complete and deterministic information. These strict assumptions can be relaxed in several directions including the following:

- Enabling the representation of *uncertainty about the initial state and/or the possible effects of actions*. This issue has lead to several approaches to *probabilistic planning* including ([19], [11], [31], [9], [13], [21]), and in particular, the Kushmerick et al.'s BURIDAN planner [25] for which the effects of actions as well as the initial state description are represented by means of probabilistic state operators that are a probabilistic extension of STRIPS' operators. A solution plan is there an unconditional sequence of actions leading to a goal state with a probability not less than a given threshold.
- Enabling the *flexible representation of goals*, replacing the set of goal states of classical planning by a preference ordering on goal states ([2] [17]) or a utility function ([22] ...) thus embedding the representation of the planning problem into decision theory.

– Taking account of different assumptions concerning *observability*. In a *fully observable* decision process, the current state is always known before the agent has to act so that an adequate solution consists in a conditional plan mapping each possible state at each time point to an actual decision. Such an assumption leads naturally to conditional planning, and in a decision-theoretic perspective to fully observable Markov decision processes (FOMDP) [3]. The other extreme case is *non-observability*, where the agent never gets any feedback from the process, which entails that one looks for an unconditional plan. Between these two extreme cases, *partially observable* decision processes enable the agent to gather some further information about the current state by performing tests (see for instance [7] [5]).
– Enabling the agent to interrupt the planner at any time after it has been launched ("anytime planning"), assuming thus that at any step of the planning process, the planner maintains a solution whose quality increases with execution time, eventually leading to an optimal solution.

This article considers the class of planning problems in which, firstly, the environment is *static* (which means that all changes that take place result from actions specified in the plan given by the agent), secondly, the goals are not flexible (i.e. defined by a set of goal states) and, thirdly, the environment is assumed to be *unobservable* during plan execution, thus requiring the search of non-conditional plans that must be robust to uncertainty. It outlines (see [10] for a detailed presentation) a possibilistic counterpart[1] of the Kushmerick et al.[25]'s approach in which possibility distributions [14] are used to represent the uncertainty both on the initial and subsequent states and on the outcomes of the execution of the context-dependent actions. Two notions of solution plans are used: γ-*acceptable plans* that lead to a goal state with a certainty greater than a given threshold γ, and *optimally safe plans* that lead to a goal state with maximal certainty.

An essential contribution expected from the possibility theory framework concerns the ability to represent more qualitatively and, thus more faithfully, what is known about the initial state and the possible effects of actions; the possibilistic approach is likely to be less sensitive to a lack of precision in the assessment of uncertainty. Using a model in which actions have possibilistic effects is particularly well-suited for cases in which the probabilities of the resulting effects of actions are not available, not very reliable, or hard to obtain, that is, in situation of partial or total ignorance about the immediate consequence of applying an action. Moreover, the notion of action with possibilistic effects properly generalizes the notion of nondeterministic actions by enabling the representation

[1] An alternative approach has been proposed by Chrisman [9] and further developed by Doan [13] which relies on a belief function represention that in essence amounts to express uncertainty by a set of probability distributions. The possibilistic representation used in this paper could be seen as a particular case of working with a set of probability distributions instead of a single one since a possibility (resp. necessity) measure is an upper (resp. lower) probability envelope [15].

of ordinal grading in the uncertainty that characterize the uncontrollable choice process through which the real effect of an action will be determined. What is represented is simply that one or several effects are normal in essence (nothing prevents them from occurring) and that some are more normal (less exceptional) than others, that is, some may be considered more plausible than others in the absence of any further information.

Besides its representational adequacy the possibilistic planning framework, as shown in this paper, has interesting properties of computational complexity compared to probabilistic planning. Indeed possibility theory is an *ordinal* model of uncertainty and the only operations needed in our framework are min, max and order reversal $(1 - .)$. This ordinal aspect of a possibilistic representation (contrarily to probabilistic representations which uses sum and product operators instead of max and min) can significantly reduce the complexity of plan generation: a core property is that the search of a γ-acceptable plan amounts to solve a planning problem strait-forwardly derived from the original possibilistic one and constituted only of pure (non graded) nondeterministic actions. This principle has led to the definition of three partial order planning algorithms, called NDP, POSPLAN and POSPLAN*.

Sections 2 and 3 provide a general description of the possibilitic planning approach. Section 4 presents complexity results for possibilitic planning and for pure nondeterministic planning that is encompassed as a special case.

2 Domain Representation in Possibilistic Planning

In this section we define the basic components of a possibilistic planning problem, and two different notions for a solution plan.

2.1 States and Actions

The facts or properties that need to be talked about in the application domain are represented in a finite propositional language by expressions that are conjunctions of atomic sentences (symbols) in either positive or negative form, i.e. conjunctions of literals. For convenience, we shall also occasionally represent an expression as the set of literals involved in the conjunction, an empty set representing no specification at all. A state is a complete description of the world at a time point, that is, a particular expression in which all atomic sentences of the language appear exactly once in a positive or negative form. A state is said to be satisfied by an expression ϵ (denoted $s \models \epsilon$) if and only if each literal of ϵ is in s. We define the set of states satisfied by an expression ϵ as $\mathcal{S}(\epsilon) = \{s \in S / s \models \epsilon\}$.

Definition 1 (uncertain states).
Let S denotes the set of all conceivable states. The uncertainty about the current state of the world is represented by a possibility distribution π over the set S of states such that $\max_{s \in S} \pi(s) = 1$. *The initial state is described by the possibility distribution π_{init}.*

π conveys what is known about the actual state of the world. $\pi(s)$ expresses to what extent it is possible that the real world state is s ; in particular, $\pi(s) = 0$ means that s is surely not the real world state, and $\pi(s) = 1$ means that nothing prevents s from being the real state. Note that there may be several states s such that $\pi(s) = 1$.

2.2 Possibilistic Actions

The actions considered here can be executed in any world state and their effect depends both on the execution-time state (context-dependent effect) and on chance (nondeterministic effect). The feasible nondeterministic results of the application of an action can be specified by a possibility distribution that enables a ranking of the possible outcomes on the scale of *normality* (i.e. non-exceptionality). More formally a possibilistic action is defined as follows.

Definition 2 (possibilistic actions).
A possibilistic action, denoted a, is a set of possibilistic effects $a = \{ep_i, i = 1, \ldots, m\}$, *in which ep_i is the i-th possible effect defined by:*

$$ep_i =< t_i, (\pi_{i1}, e_{i1}), \ldots, (\pi_{in_i}, e_{in_i}) >$$

where $\forall i, j$, t_i and e_{ij} are expressions, $\pi_{ij} \in]0, 1]$, such that

- *for all state s, there is a single i such that $s \models t_i$;*
- *for all i, $\max_{1 \leq j \leq n_i} \pi_{ij} = 1$.*

The ep_i's, the t_i's and the e_{ij}'s are called *possibilistic effects*, *discriminants* and *elementary consequences* respectively. The e_{ij}'s play the role of Add/Delete lists of the STRIPS action model: the state resulting from the change on a state s caused by e_{ij} is defined by:

$$Res(e_{ij}, s) = e_{ij} \cup \{l \in s \mid \bar{l} \notin e_{ij}\} \ .$$

If the context defined by t_i is verified before the execution of a, then it is possible at degree π_{ij} that effect e_{ij} is verified after the execution. If π_{ij} is equal to 1, then e_{ij} is a normal effect (i.e. nothing prevents it from occuring), else the smaller π_{ij} the more exceptional e_{ij}. For a given discriminant, the elementary consequences together with their associated degrees of possibility constitute a possibility distribution over the changes to the world.

Definition 3 (effect of a possibilistic action).
The result of executing action a on s is given by a possibility distribution on S defined by:

$$\pi[s'|s, a] = \begin{cases} \max_k \pi_{ik} \ \textit{if} \ s \in S(t_i) \ \textit{and} \ s' = Res(e_{ik}, s) \\ 0 \ \textit{otherwise} \end{cases} \ .$$

If the initial state is described by a possibility distribution π_{init} over S, then the effect of executing a is defined by the following possibility distribution:

$$\pi[s'|\pi_{init}, a] = \max_{s_0} \min(\pi[s'|s_0, a], \pi_{init}(s_0)) \ .$$

We will say that action a is *nondeterministic* if and only if $\forall s, s' \in S,\ \pi[s'|s,a] \in \{0,1\}$.

2.3 Plans and Possibilistic Planning

We classically define a sequential plan as a totally ordered set of actions $\langle a_i \rangle_{i=0}^{N-1}$. A partially ordered plan is a pair $\mathcal{P} = (A, O)$ where A is a set of actions and O is a set of ordering constraints between these actions. A completion of $\mathcal{P}$ is a sequential plan $\mathcal{CP} = \langle a_i \rangle_{i=0}^{N-1}$ such that $A = \{a_0, \ldots, a_{N-1}\}$ and the total ordering $a_0 < \cdots < a_{N-1}$ is consistent with O.

The possibility to reach a given state s_N by executing a sequential plan of possibilistic actions $\langle a_i \rangle_{i=0}^{N-1}$ starting in a state s_0 is defined by:

$$\pi[s_N | s_0, \langle a_i \rangle_{i=0}^{N-1}] = \max_{\langle s_1 \ldots s_{N-1} \rangle} \min_{i=0\ldots N-1} \pi[s_{i+1} | s_i, a_i] \ .$$

Let $Goals \subseteq S$ the set of the goal states, and π_{init} a possibility distribution over S that describes the initial state. The possibility and necessity measures to reach a goal state after the execution from π_{init} of the sequential plan $\langle a_i \rangle_{i=0}^{N-1}$ are then given by:

$$\begin{aligned}
\Pi[Goals|\pi_{init}, \langle a_i \rangle_{i=0}^{N-1}] &= \max_{s_0 \in S, s_N \in Goals} \min(\pi[s_N | s_0, \langle a_i \rangle_{i=0}^{N-1}], \pi_{init}(s_0)) \\
N[Goals|\pi_{init}, \langle a_i \rangle_{i=0}^{N-1}] &= \min_{s_0 \in S, s_N \in \overline{Goals}} \max(1 - \pi_{init}(s_0), 1 - \pi[s_N | s_0, \langle a_i \rangle_{i=0}^{N-1}]) \ .
\end{aligned}$$

Definition 4 (possibilistic planning problem).
A possibilistic planning problem Δ is a triplet $\langle \pi_{init}, \epsilon_{Goals}, \mathcal{A} \rangle$ where π_{init} is the possibility distribution associated to the initial state, ϵ_{Goals} is an expression defining the set of goal states Goals and $\mathcal{A}$ is the set of available possibilistic actions.

Given a possibilistic planning problem Δ, two criteria may be considered to define a *solution plan*: we say that a partially ordered plan is a **γ-acceptable plan** for Δ if $N[Goals|s_0, \mathcal{CP}] \geq \gamma$ for all totally ordered completion $\mathcal{CP}$ of $\mathcal{P}$; $\mathcal{P}$ is an **optimally safe plan**, or **optimal plan**, if $N[Goals|s_0, \mathcal{CP}]$ is maximal among all possible sequential plans for all totally ordered completion $\mathcal{CP}$ of $\mathcal{P}$.

3 Generation of Solution Plans

The algorithms we have developped [10] for solving a possibilistic planning problem are based on the equivalence between the search of γ-acceptable plans and the resolution of a derived planning problem that has only pure nondeterministic actions.

Definition 5 (from possibilistic to nondeterministic planning).
Let $\Delta = \langle \pi_{init}, \epsilon_{Goals}, \mathcal{A} \rangle$ *a possibilistic planning problem and* $\gamma \in]0,1]$*. The nondeterministic planning problem* $\Delta_{1-\gamma}$ *constructed from* Δ *is defined by* $\Delta_{1-\gamma} = \langle \pi_{init1-\gamma}, \epsilon_{Goals}, \mathcal{A}_{1-\gamma} \rangle$ *where* $\pi_{init1-\gamma}(s) = 1$ *if* $\pi_{init}(s) > 1-\gamma$ *and* 0 *else,* $\mathcal{A}_{1-\gamma} = \{a_{1-\gamma} \mid a \in \mathcal{A}\}$ *such that if* $a = \{\langle t_i, \ldots (\pi_{ij}, e_{ij}) \ldots \rangle\}$ *then* $a_{1-\gamma} = \{\langle t_i, \ldots (1, e_{ij}) \ldots \rangle \mid \pi_{ij} > 1-\gamma\}$.

The distribution $\pi_{init1-\gamma}$ represents the set of initial states that have a possibility greater than $1-\gamma$. The nondeterministic action $a_{1-\gamma} \in \mathcal{A}_{1-\gamma}$ is the result of transforming the action a in $\mathcal{A}$ by retaining only the effects having a possibility greater than $1-\gamma$.

The following key result is the core of the possibilistic planning algorithms we outline next (the proofs of the following propositions are presented in [10]) :

Proposition 1 (equivalence).
A partial plan $\mathcal{P}$ *is* $\gamma-$*acceptable for* Δ *if and only if it is* 1*-acceptable for* $\Delta_{1-\gamma}$*.*

3.1 NDP: a Planning Algorithm for Nondeterministic Actions

NDP is a planning algorithm we have developed to solve nondeterministic planning problems. Like most of the classical planning algorithms, NDP explores a search tree of partial plans whose root is the null plan and the branches represent refinements of the current plan in order to establish subgoals or to confront threats on some already established causal links. The search stops when the current plan is recognized as a solution plan.

The main function of a planning algorithm like NDP is the refinement which transforms a partial plan $\mathcal{P}$ into a new partial plan $\mathcal{P}'$, by eliminating *flaws*. A flaw in $\mathcal{P}$ can be a subgoal not yet established, or a threat on a subgoal already established, that prevents $\mathcal{P}$ from being a solution plan.

Like in SNLP [27], UCPOP [29] or BURIDAN [25] NDP establishes a subgoal proposition p by adding a *causal link* between the effect of an operator that adds p and the action of which p is a precondition. More formally, if $a_j : t_k : p$ is a subgoal of $\mathcal{P}$ where $p \in t_k$ and t_k a discriminant of a_j in $\mathcal{P}$, and if $p \in e_{lm}$ where e_{lm} is an elementary consequence of a nondeterministic effect of the action a_i, then NDP can add the causal link $a_i : e_{lm} \xrightarrow{p} a_j : t_k$. Consequently, each proposition $q \in t_l$ becomes a new subgoal to establish. To each partial plan in the search graph is associated a set of subgoals $\mathcal{SG}$ and a set of causal links $\mathcal{CL}$. They are initialized with $\mathcal{SG} = \{end : \epsilon_{Goals} : q, \quad \forall q \in \epsilon_{Goals}\}$ and $\mathcal{CL} = \emptyset$.

A threat in a partial plan $\mathcal{P} = (A, O)$ is a pair $\langle a; l \rangle$ where l is a causal link $a' : e \xrightarrow{p} a'' : t$ in $\mathcal{CL}$ establishing a proposition p, and a is an action in A that potentially can delete p, i.e. one of its elementary consequences contains $\neg p$. In a classical propositional planner like SNLP, there are two ways to confront such a threat: you can add the constraint $a'' < a$ (*promotion*) or add the constraint $a < a'$ (*demotion*). NDP naturally keeps these two techniques. Due to the specificity of the problem of planning actions with context-dependent effects, the refinement function of NDP must in addition use another trick to protect a causal link as it is done in the two planners UCPOP and BURIDAN. In the NDP planner, we retain the *confrontation* form of BURIDAN. If $a : e$ is a threatening effect for a causal link l of $\mathcal{CL}$, NDP will try to make another non threatening effect $a : e'$ occur. This is done by adding to the discriminant of the causal link consumer a *safety proposition sp* unique to this threat, which becomes a new subgoal to achieve, and by adding to each non threatening elementary consequence $a : e'$ the proposition sp.

In classical planning with STRIPS-like operators, a polynomial truth criterion can be used to check whether or not a partial plan is a solution plan[8, 23]. In such a case the causal link structure is a way to simplify the management of this truth criterion. The problem of planning actions with context-dependent effects like in UCPOP is not as simple. It has been shown that the assessment problem for actions with context-dependent effect was NP-hard [8, 12]. We exhibit in Sect. 4 a similar result for non-deterministic action representation. Consequently, like in BURIDAN, we propose to check if a partial plan $\mathcal{P}$ is a safe plan by directly executing each of its totally ordered completions from each possible initial state.

3.2 Generating γ-Acceptable Plans and Optimal Plans

POSPLAN and POSPLAN* are the two planning algorithms we have developed to solve possibilistic planning problems.

Given a possibilistic planning problem Δ the possibilistic planning algorithm POSPLAN transforms it into a nondeterministic planning problem $\Delta_{1-\gamma}$. Then the nondeterministic planning algorithm NDP takes $\Delta_{1-\gamma}$ as input, and generates a partially ordered safe plan, which is γ-acceptable for Δ.

The link between γ-acceptable plans and optimal plans is clear: $\langle a_i \rangle_{i=1}^{N-1}$ is optimal if and only if it is γ-acceptable and $\forall \gamma' > \gamma$, there is no γ'-acceptable plan. Therefore a meta-algorithm for computing an optimal plan might consist in searching γ-acceptable plans with well-chosen successive values of γ. Several strategies can be thought of; they all require that the state transition degrees (i.e. the possibility degrees involved in the actions and the initial state) be ordered beforehand. Assume that $\{\alpha_i, i = 0, n\}$ is the set of possibility degrees named such that $0 = \alpha_0 < \alpha_1 < \alpha_2 < \ldots < \alpha_n < 1$ or, equivalently, $1 = \gamma_0 > \gamma_1 > \ldots > \gamma_{n-1} > \gamma_n > 0$ with $\gamma_i = 1 - \alpha_i$.

The POSPLAN* algorithm for generating optimal plans implements an increasing acceptability method. The principle is to consider successively the γ-acceptability levels γ_n, γ_{n-1}, ... and to solve the corresponding non-deterministic planning problems $\Delta_{1-\gamma_i}$, until no solution can be found. The main characteristic of POSPLAN* is to avoid replanning from scratch at each iteration $\gamma_{i+1} \rightarrow \gamma_i$. Instead, POSPLAN* reuses the search tree developed so far and only realizes the updating of the current plan that is required to cope with the newly added elementary effects that have a possibility $\pi > 1 - \gamma_i$.

Let $\mathcal{P}$ be the current partial plan in the search tree, and γ_i the current acceptability level associated with $\mathcal{P}$. If $\mathcal{P}$ is not a solution plan for $\Delta_{1-\gamma_i}$, then $\mathcal{P}$ is classically refined into the new partial plan $\mathcal{P}'$, by considering the refinement method of NDP for the $\Delta_{1-\gamma_i}$ problem. In that case, the only action effects considered are the ones with a possibility greater than $1-\gamma_i$. Conversely, if $\mathcal{P}$ appears to be a solution plan for $\Delta_{1-\gamma_i}$, then the current acceptability level of $\mathcal{P}$ becomes γ_{i-1} and γ_{max} is set to the value γ_i if necessary (see [10] for further details).

In other words, the first iteration amounts to search a plan made of actions restricted to their normal effects only. The successive iterations consist in incorporating the more and more exceptional effects.

The search by increasing acceptability has two worth-mentioning computational properties: POSPLAN* is "anytime" in the sense that it can supply a solution at any time (provided at least one exists and enough time has been allocated to its generation) and the supplied solution is all the better as the algorithm runs longer; the planning problems generated at each iteration are more and more difficult; consequently, the first iteration should be faster.

POSPLAN* can be considered as a hierarchical version of POSPLAN, and relies on an abstraction principle, similar to the ones that have been recently developed. In [32, 24, 18], hierarchal planning exploits a partitioning of the precondition propositions into abstraction level, such that within a given abstraction level only a subset of the subgoals are considered. The abstraction level definition is based on threatening constraints between operators, in order to minimize interaction between subgoals in different levels. In [4], abstractions for Markov decision processes are generated by eliminating not very relevant propositions, in regard to the utility function.

The soundness and completeness properties of NDP, POSPLAN and POSPLAN* have been established in [10]. In particular, the following proposition holds (an essential solution plan is a totally ordered solution plan such that no subplan can be a solution plan).

Proposition 2 (completeness of POSPLAN and POSPLAN*).
Let $\Delta = \langle \pi_{init}, \epsilon_{Goals}, \mathcal{A} \rangle$ be a possibilistic planning problem. If for $\gamma \in]0,1]$ there exists an essential solution plan $\langle a_i \rangle_{i=0}^{N-1}$ such that $N[Goals|\pi_{init}, \langle a_i \rangle_{i=0}^{N-1}] \geq \gamma$ then POSPLAN(Δ, γ) will generate a partial solution plan $\mathcal{P}$ such that $\langle a_i \rangle_{i=0}^{N-1}$ is one of its completions. If γ_{max} is the greater γ_i such that there exists an essential solution plan $\langle a_i \rangle_{i=0}^{N-1}$ with $N[Goals|\pi_{init}, \langle a_i \rangle_{i=0}^{N-1}] = \gamma_i$ then POSPLAN will generate a partial solution plan $\mathcal{P}$ with $N[Goals|\pi_{init}, \mathcal{P}] = \gamma_{max}$.*

4 Computational Complexity

In this section we determine the complexity classes in which the various problems associated with possibilistic planning fall into. The methodology we use is inspired from recent results on probabilistic planning [20] [26] and on well-known results on planning with structured action representations [1] [6] [16].

Proposition 3.
Nondeterministic plan verification (NPV) *is* co-NP*-complete.*

Proof.
(a) Membership of the complementary problem $\overline{\text{NPV}}$ to NP is showed by the following nondeterministic algorithm which has as input $\langle \mathcal{A}, s_0, \epsilon_{Goals}, \langle a_0 \ldots a_n \rangle \rangle$:

1. guess a sequence $\langle s_1 \ldots s_n s_{n+1} \rangle$;
2. $\forall i$, verify that (s_i, a_i, s_{i+1}) is allowed by $\mathcal{A}$;
3. verify that $s_{n+1} \notin Goals$.

Steps 2 and 3 can be done in polynomial time, hence membership to NP.
(b) As to completeness, we give a polynomial reduction from the validity problem for DNF formulas (which is co-NP-complete) to NPV. Let $\varphi = D_1 \vee \ldots \vee D_m$ be a DNF formula on the alphabet $\{a_1, \ldots a_n\}$. We associate to φ the planning problem defined by the alphabet $\{d_1 \ldots d_m, g\}$, $s_0 = \{d_1, \ldots, d_m, g\}$, $\mathcal{A} = \{\texttt{fix}{-}a_1, \ldots \texttt{fix}{-}a_n, \texttt{look}\}$, $\epsilon_{Goals} = \{g\}$; action effects are defined by:

$\texttt{fix}{-}a_i : \{\langle \emptyset, \{\overline{d_j} | a_i \in D_j\}, \{\overline{d_j} | \neg a_i \in D_j\} \rangle\}$;

$\texttt{look} : \{\langle \{\overline{d_1}, \ldots, \overline{d_m}\}, \{\overline{g}\} \rangle\}$.

Then, φ is valid iff $\langle \texttt{fix}{-}a_1, \ldots, \texttt{fix}{-}a_n, \texttt{look} \rangle$ is a good plan. Intuitively, the meaning of d_j is "disjunct D_j has not been violated yet by the current partial interpretation". g is set to `false` iff the interpretation violates *all* disjuncts (action `look` thus examines all disjuncts) and remains `true` otherwise; action `fix-`a_i nondeterministically chooses a truth value for a_i; if this value is `true` (resp. `false`) then all disjuncts containing $\neg a_i$ (resp. a_i) are (i.e., become or remain) violated. Initially all disjuncts are not violated. It is rather easy to check that all executions of $\langle \texttt{fix}{-}a_1, \ldots, \texttt{fix}{-}a_n, \texttt{look} \rangle$ produce all models for the alphabet $a_1 \ldots a_n$ and that g remains true in all possible executions iff all models satisfy at least one of the disjuncts, i.e., if φ is valid. □

As to plan existence, things are trickier. Classical STRIPS plan existence [6] is PSPACE-complete, so we cannot expect nondeterministic-STRIPS planning to be at a lower complexity level. It seems that it is actually *not* in PSPACE – all we can say is that it is in EXPTIME[2].

This point is left for further study. However, by adding the restriction that the plan length should be polynomially bounded (polynomial in the size of the input), we obtain nontrivial results:

Proposition 4.
Nondeterministic polynomial-length plan existence (NPPE) *is* Σ_2^P*-complete*[3].

[2] Note that unrestricted probabilistic plan existence [26] is EXPTIME-complete.
[3] $\Sigma_2^P = \mathsf{NP}^{\mathsf{NP}}$ is the class of decision problems that can be decided in polynomial time on a nondeterministic Turing machine using NP-oracles (see [30] [28]). The canonical

Proof.
(a) membership follows from the following algorithm:

1. guess a plan $\langle a_0 \dots a_l \rangle$;
2. verify that it is a good plan for $\langle s_0, \mathcal{A}, \epsilon_{Goals} \rangle$.

Step 2 is done by a NP-oracle, and the rest of the algorithm is polynomial, because the length l of the plan is bounded by a polynomial function of the size of the input[4] hence NPPE $\in \mathsf{NP}^{\mathsf{NP}} = \Sigma_2^P$.
(b) completeness is proved by the following polynomial reduction from 2-QBF to NPPE. Let $\exists a_1 \dots \exists a_m \forall b_1 \dots \forall b_p F$ be an instance of 2-QBF where F is under DNF, i.e., $F = D_1 \vee \dots \vee D_m$. The associated instance of NPPE is defined by
$s_0 = \{\neg a_1-\mathtt{set}, \dots \neg a_m-\mathtt{set}, \neg b_1-\mathtt{set}, \dots \neg b-\mathtt{set}, d_1, \dots d_n, g\}$;
$\epsilon_{Goals} = \{a_1-\mathtt{set}, \dots a_m-\mathtt{set}, b_1-\mathtt{set}, \dots, b_p-\mathtt{set}, g\}$;
$\mathcal{A} = \{a_1, \overline{a_1}, \dots, a_m, \overline{a_m}, \mathtt{fix}-b_1, \dots, \mathtt{fix}-b_p, \mathtt{look}\}$, with

$a_i : \{\langle \overline{a_i-\mathtt{set}}, a_i-\mathtt{set} \cup \{\overline{d_j} | \neg a_i \in D_j\}\rangle\}$;
$\overline{a_i} : \{\langle \overline{a_i-\mathtt{set}}, a_i - set \cup \{\overline{d_j} | a_i \in D_j\}\rangle\}$;
$\mathtt{fix}-b_i : \{\langle \overline{b_i-\mathtt{set}}, b_i-\mathtt{set} \cup \{\overline{d_j} | b_i \in D_j\} \cup \{\overline{d_j} | \neg b_i \in D_j\}\rangle\}$;
$\mathtt{look} : \{\langle \{\overline{d_1}, \dots, \overline{d_m}\}, \{\bar{g}\}\rangle\}$

Actions a_i and $\overline{a_i}$ (i = 1 ...m) are deterministic and fix the truth value of a_i to `true` and `false` respectively. $\mathtt{fix}-b_i$ and `look` are as in the proof of Proposition 4 (plus the extra variables $b_i-\mathtt{set}$, needed here to ensure that all b_i's will be assigned a truth value in the plan). If G is attained by a plan, then necessarily this plan contains all $\mathtt{fix}-b_j$'s (in any order) and for all i, either a_i or $\overline{a_i}$; then, it is a good plan iff the corresponding truth assignment of the a_i's satisfies the 2-QBF instance. □

These results on nondeterministic planning can be immediately extended to possibilistic planning:

Proposition 5.
Deciding that $N(G|s_o, \langle a_o \dots a_n \rangle) \geq \alpha$, $\alpha \in]0,1]$ (ΠPV$\geq$) is co-NP*-complete.*

Proposition 6.
Deciding that there exists a plan P with polynomial length such that $N(G|s_o, P) \geq \alpha$, $\alpha \in]0,1]$ (ΠPE$\geq$) is Σ_2^P-complete.

Proof. These are immediate corollaries from Propositions 4 (resp. 5) and 1. □

Σ_2^P-complete problem is 2-QBF (for "quantified Boolean formula") [30]: show that $\exists a_1 \dots \exists a_m \forall b_1 \dots \forall b_p F(\boldsymbol{a}, \boldsymbol{b})$, where the a_i's and b_j's are Boolean variables and F a formula built on them. The problem remains complete if F is under DNF.

[4] Here is why this result does not work without the restriction that plan length should be polynomially bounded: in the general case, the shortest good plan may have up to 2^n actions where n is the number of propositional variables ([6] for the deterministic case).

Proposition 7.
Deciding that $N(G|s_o, \langle a_o \dots a_n \rangle) = \alpha$, $0 < \alpha < 1$ (ΠPV=) *is* DP-*complete*[5] [6].

Proof. It is based on the following lemma:
for $\beta < 1$, *deciding that* $N(G|s_o, \langle a_o \dots a_n \rangle) > \beta$ *is* co-NP-*complete.*

The proof of the lemma is very similar to that of Proposition 4: indeed, we know that only a finite number of possibility values is used and we can thus define β^+ as the value immediately above β in the set $\{1-x \mid \exists s, a, s' \; s.t. \; \pi(s'|a, s) = x\}$. It is clear that $N(G|s_o, \langle a_o \dots a_n \rangle) > \beta$ iff $N(G|s_o, \langle a_o \dots a_n \rangle) \geq \beta^+$, which (with Proposition 4) proves the lemma.
(a) membership: obvious since $N(G|s_o, \langle a_o \dots a_n \rangle) = \alpha$ iff $N(G|s_o, \langle a_o \dots a_n \rangle) \geq \alpha$ (NP) and not ($N(G|s_o, \langle a_o \dots a_n \rangle) > \alpha$ (co-NP).
(b) completeness: immediate from Propositions 6 and 7. □

Conclusions are twofold. First, it appears that possibilistic planning has (up to a linear factor) the same orders of complexity as nondeterministic planning. Second, these orders of complexity are far below those for the corresponding problems in probabilistic planning (also with compact representations): contrarily to possibilistic planning, probabilistic planning basically needs examining all paths (in the worst case), or at least to count them, to compute if the probability to reach the plan is higher than a given threshold. Goldsmith et al. [20] show that probabilistic plan evaluation with succinct representations is PP-complete (while the similar problem for possibilistic planning is co-NP-complete), and polynomial size probabilistic plan existence with succinct representations is $\mathsf{NP}^{\mathsf{PP}}$-complete (while the similar problem for possibilistic planning is Σ_2^P-complete). Since co-NP $\subseteq$ PP (the inclusion being likely to be strict), possibilistic planning appears to be easier than probabilistic planning.

5 Conclusions

The main goal of this paper was to present and analyse a possibilistic approach of planning under uncertainty. It was inspired by the work done on the BURIDAN[25] planner that relies on a probabilistic representation of uncertainty. In practice, it seems more natural and easier to see actions in terms of normal and more or less exceptional effects rather than probable ones.

Besides its representational adequacy, the possibilistic approach has interesting computational properties since the search for γ-acceptable or optimally safe plans amounts to solve induced planning problems that have only crisp nondeterministic actions (i.e. each action having then only normal effects). Moreover,

[5] DP is the set of all languages which are the intersection of a language in NP and a language in co-NP. The canonical DP-complete problem is SAT-UNSAT: given two propositional formulas φ, ψ, decide that φ is satisfiable and ψ is not [28].

[6] Obviously, the similar problems for $\alpha = 1$ and $\alpha = 0$ are respectively co-NP-complete (Proposition 4) and co-NP-complete (complementary of the problem studied in the lemma).

in the case of optimal plan generation, the proposed sound and complete POSPLAN* algorithm is an anytime least-commitment planner; it possesses the additional feature of iteratively solving derived planning problems that are progressively more complex and exploits at each iteration the partial plans developed in the previous ones. The POSPLAN* planner and, consequently, the NDP and POSPLAN algorithms have been implemented in Common Lisp reusing part of BURIDAN's code (in particular its SNLP basis).

The complexity results of the paper show that possibilistic planning is easier than probabilistic planning. They should however be relativized by the fact that the kind of problems suitable for the probabilistic and the possibilistic settings are very different in nature, hence the difficulty to compare these results on a single reference problem. Moreover, there may be efficient and much less complex approximations of probabilistic planning problems.

Acknowledgements The authors are grateful to an anonymous reviewer for pointing out the relevant works of Chrisman, Doan and Haddawy et al. (1995).

References

1. C. Bäckström. *Computational complexity of reasoning about plans.* PhD thesis, University of Linköping, 1992.
2. B. Bonet and H. Geffner. Arguing for decisions: a qualitative model for decision making. In *Proc. of UAI'96*, 1996.
3. C. Boutilier, T. Dean, and S. Hanks. *New Directions in AI Planning*, chapter Planning under uncertainty: structural assumptions and computational leverage, pages 157–172. IOS Press, Amsterdam, 1996.
4. C. Boutilier and R. Dearden. Using abstractions for decision-theoretic planning with time constraints. In *Proceedings AAAI*, pages 1016–1022, 1994.
5. C. Boutilier and D. Poole. Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings AAAI'96*, pages 1168–1175, 1996.
6. T. Bylander. The computational complexity of propositional strips planning. *Artificial Intelligence*, 69:161–204, 1994.
7. A.R. Cassandra, L.P. Kaelbling, and M.L. Littman. Acting optimally in partially observable stochastic domains. In *Proceedings AAAI*, pages 1023–1028, 1994.
8. D. Chapman. Planning for Conjunctive Goals. *Artificial Intelligence*, 32, 1987.
9. L. Chrisman. Abstract probabilistic modeling of action. In *Proceedings AAAI*, pages 28–36, 1992.
10. C. Da Costa Pereira, F. Garcia, J. Lang, and R. Martin-Clouaire. Planning with graded nondeterministic actions: a possibilistic approach. *Int. J. of Intelligent Systems*, 12(12), 1997.
11. M. De Glas and E. Jacopin. An algebric framework for uncertain strips planning. In *Proceedings AIPS*, 1994.
12. T. Dean and M. Boddy. Reasoning about Partially Ordered Events. *Artificial Intelligence*, 36:375–399, 1988.
13. A. Doan. Modeling probabilistic actions for practical decision-theoretic planning. In *Proc. of AIPS'96*, 1996.

14. D. Dubois and H. Prade. *Possibility theory - an approach to computerized processing of uncertainty.* Plenum Press, 1988.
15. D. Dubois and H. Prade. Fuzzy sets and probability: misunderstanding, bridges and gaps. In *Proceeding of 2nd IEEE Int. Conf. on Fuzzy Systems*, pages 1059–1068, 1993.
16. H. Erol, D. Nau, and V.S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning. *Artificial Intelligence*, 76:75–88, 1995.
17. H. Fargier, J. Lang, and R. Sabbadin. Towards qualitative approaches to multistage decision making. In *Proc. of IPMU'96*, 1996.
18. F. Garcia and P. Laborie. *New Directions in AI Planning*, chapter Hierarchisation of the Search Space in Temporal Planning, pages 217–232. IOS Press, Amsterdam, 1996.
19. R. Goldman and M. Boddy. Epsilon-safe planning. In *Proc. of UAI'94*, pages 253–261, 1994.
20. J. Goldsmith, M. Littman, and M. Mundhenk. The complexity of plan existence and evaluation in probabilistic domains. In *Proc. of UAI'97*, pages 182–189, 1997.
21. P. Haddawy, A. Doan, and R. Goodwin. Efficient decision-theoretic planning: Techniques and empirical analysis. In *Proc. of UAI'95*, pages 229–236, 1995.
22. P. Haddawy and P. Hanks. Issues in decision-theoretic planning: utility functions for deadline goals. In *Proc. of the 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 71–82, 1992.
23. S. Kambhampati and D.S. Nau. On the nature and role of modal truth criteria in planning. *Artificial Intelligence*, 1996.
24. C. Knoblock. Automatically generating abstractions for planning. *Artificial Intelligence*, 68:243–302, 1994.
25. N. Kushmerick, S. Hanks, and D.W Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76:239–286, 1995.
26. M.L. Littman. Probabilistic propositional planning: representations and complexity. In *Proceedings AAAI*, pages 748–754, 1997.
27. D. McAllester and D. Rosenblitt. Systematic nonlinear planning. In *Proceedings AAAI*, pages 634–639, 1991.
28. C.H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.
29. J.S. Penberthy and D. Weld. UCPOP: A sound, complete, partial-order planner for ADL. In *Proc. of the 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'92)*, pages 103–114, 1992.
30. L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.
31. S. Thiébaux, J. Hertzberg, W. Shoaff, and M. Schneider. A stochastic model of actions and plans for anytime planning under uncertainty. *Int. J. of Intelligent Systems*, 10(2):155–183, 1995.
32. Q. Yang and J. D. Tenenberg. Abstracting a nonlinear least commitment planner. In *Proceedings AAAI*, pages 204–209, 1990.