Width and Complexity of Belief Tracking in Non-Deterministic Conformant and Contingent Planning

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

Hector Geffner
ICREA & Universitat Pompeu Fabra
08003 Barcelona, SPAIN
hector.geffner@upf.edu

Abstract

It has been shown recently that the complexity of belief tracking in *deterministic* conformant and contingent planning is exponential in a width parameter that is often bounded and small. In this work, we introduce a new width notion that applies to *non-deterministic* conformant and contingent problems as well. We also develop a belief tracking algorithm for non-deterministic problems that is exponential in the problem width, analyze the width of non-deterministic benchmarks, compare the new notion to the previous one over deterministic problems, and present experimental results.

Introduction

Planning with incomplete information can be formulated as a search problem problem in belief space (Bonet and Geffner 2000). This is the approach adopted in most recent conformant and contingent planners that have to address then two problems: keeping track of beliefs, and searching for a goal belief. These two tasks are intractable in the worst case and are normally handled by belief representations based on SAT, OBDDs, or regression techniques, and search algorithms that appeal to domain-independent heuristics (Bertoli et al. 2001; Brafman and Hoffmann 2004; Bryce, Kambhampati, and Smith 2006; Shani and Brafman 2011).

Recently, the complexity of belief tracking in *deterministic* conformant and contingent planning has been shown to be exponential in a *width* parameter that is often bounded and small (Palacios and Geffner 2009; Albore, Palacios, and Geffner 2009). Such a bound follows from a family of translations developed for compiling planning problems with incomplete information over beliefs into planning problems with complete information over states. The translations are exponential in the problem width, and for deterministic conformant problems result in problems that can be solved by classical planners.

The aim of this work is to provide an alternative formulation of the notion of *width* for bounding the complexity of belief tracking in *non-deterministic* conformant and contingent problems. The result is a belief representation scheme that is *polynomial* for domains with bounded width. We will show that many interesting non-deterministic planning domains have bounded width and exploit this property in simple conformant and on-line contingent planners that combine this belief representation with simple heuristics.

The work is related to other proposals for tractable forms of belief tracking in logical and probabilistic frameworks (Doucet et al. 2000; Amir and Russell 2003), yet there are two key differences. One is that we are after an *exact* account that can be used to determine with *certainty* whether the goal has been achieved or an action is applicable. The second is that an exact account of belief tracking in *planning* does not have to be *complete over all formulas*. In order to have a sound and complete planner, only the beliefs over action preconditions and goals are required. This is important because the action preconditions and goals are given, and the structure of the actions and goals can be exploited to track those beliefs more efficiently.

The paper is organized as follows. We review first non-deterministic conformant and contingent planning. We then consider the basic belief tracking algorithm and a factored algorithm. We close by reporting experiments, discussing related work, and summarizing the main ideas.

Conformant Planning

Conformant planning is planning with incomplete information and *no sensing* where a goal is to be achieved with certainty in spite of uncertainty in the initial situation or action effects (Goldman and Boddy 1996; Smith and Weld 1998). The model for conformant planning is characterized by a tuple $\mathcal{S} = \langle S, S_0, S_G, A, F\rangle$ where

- S is a finite state space,
- S_0 is a non-empty set of possible initial states $S_0 \subseteq S$,
- S_G is a non-empty set of goal states $S_G \subseteq S$,
- A is a set of actions where $A(s)$ denotes the set of actions applicable in state $s \in S$, and
- F is a non-deterministic state-transition function such that $F(a, s)$ denotes the non-empty set of possible successor states that follow action a in s, $a \in A(s)$.

A solution to a conformant model is an action sequence that maps each possible initial state into a goal state. More precisely, $\pi = a_0, \ldots, a_{n-1}$ is a conformant plan if for each possible sequence of states $s_0, s_1, \ldots, s_n$ such that $s_0 \in S_0$ and $s_{i+1} \in F(a_i, s_i)$, $i = 0, \ldots, n-1$, each action a_i is applicable in s_i and s_n is a goal state.

Conformant planning can be cast as a path finding problems over *beliefs* defined as the sets of states that are deemed possible at any one point (Bonet and Geffner 2000). The initial belief b_0 is S_0, and the belief b_a that results from an action a in belief state b is:

$$b_a = \{s' \mid \text{there is a } s \in b \text{ such that } s' \in F(a,s)\}, \quad (1)$$

where it is assumed that action a is applicable in each state s in b. In this formulation, a conformant plan is an action sequence that maps the initial belief b_0 into a goal belief b_G; i.e., a set of goal states $s \in S_G$.

Syntactically, conformant planning problems are expressed in *compact form* through a set of *state variables*, which for convenience we assume to be *multivalued*. More precisely, a conformant planning problem is a tuple $P = \langle V, I, A, G \rangle$ where V stands for the problem variables X, each one with a finite and discrete domain D_X, I is a set of clauses over the V-literals $X = x$ and $X \neq x$ defining the initial situation where $x \in D_X$, A is set of actions, and G is a set of V-literals defining the goal. Every action a has a precondition $Pre(a)$ given by a set of V-literals, and a set of conditional effects $C \rightarrow E_1 | \ldots | E_n$, where C and each E_i is a set (conjunction) of V-literals. The conditional effect is *non-deterministic* if $n > 1$; else it is deterministic.

A conformant planning problem $P = \langle V, I, A, G \rangle$ defines a conformant model $\mathcal{S}(P) = \langle S, S_0, S_G, A, F \rangle$, where S is the set of possible valuations over the variables in V, S_0 and S_G are the set of valuations that satisfy I and G respectively, $A(s)$ is the set of operators whose preconditions are true in s, and $F(a, s)$ is the non-deterministic transition function that results from collecting the successor states that may follow from a by selecting one effect E_i from each conditional effect $C \rightarrow E_1 | \ldots | E_n$ whose body C is true in s. A conformant plan for P is a conformant plan for $\mathcal{S}(P)$.

Contingent Planning

Contingent planning is planning with both uncertainty and feedback. The model for contingent planning is the model for conformant planning extended with a *sensor model*. A sensor model is a function $O(s, a)$ mapping state-action pairs into observations tokens o. The expression $o \in O(s, a)$ means that token o is a possible observation when s is the true state of the system and a is the last action done. The observed token o provides partial information about the hidden state of the system as the same token may be possible in different states. When two different tokens o_1 and o_2 belong to $O(s, a)$, then either one can be observed in s when a is the last action. Sensing is *deterministic* or *noiseless* when $O(s, a)$ contains one token, else it is *non-deterministic* or *noisy*. The contingent model is thus similar to POMDPs but with uncertainty encoded through sets rather than probability distributions.

Executions in the contingent or partially observable setting are sequences of action-observation pairs $a_0, o_0, a_1, o_1, \ldots$. If $b = b_i$ is the belief state when the action a_i is applied, and o_i is the token that is observed, then the belief b_a after the action $a = a_i$ is given by Equation (1), and the belief $b_{i+1} = b_a^o$ that follows from observing then token o is given by:

$$b_a^o = \{s \mid s \in b_a \text{ and } o \in O(s,a)\}. \quad (2)$$

In *off-line* contingent planning, an action selection strategy is sought that ensures that *all possible executions* end up in a goal belief. In *on-line* contingent planning, an action selection strategy is sought that ensures that the *single execution* that results from the interaction with the real system or suitable simulator, ends up in a goal belief. In both cases, the action selection strategy can be expressed as a *partial* function π over beliefs, called a *policy*, such that $\pi(b)$ is the action to do in belief b. In the off-line setting, the function π has to be defined over the initial belief b_0 and the non-goal beliefs b that can be reached with π from b_0. In the on-line setting, the function π represents an action selection mechanism that is computed for the current belief.

Syntactically, contingent problems are represented by extending the syntactic representation of conformant problems with a compact encoding of the *sensor model*. For this, we assume a set V' of observable multivalued variables Y not necessarily disjoint from the state variables V (i.e., some state variables may be observable), and formulas $W_a(Y = y)$ over a subset of state variables for each action a and each possible value y of each observable variable Y. The formula $W_a(Y = y)$ encodes the states over which the observation $Y = y$ is possible when a is the last action. The formulas $W_a(Y = y)$ for the different y values in D_Y must be logically *exhaustive*, as every state-action pair must give rise to some observation $Y = y$. If in addition, the formulas $W_a(Y = y)$ for the different y values are logically *exclusive*, every state-action pair will give rise to a single observation $Y = y$, so that the sensing over Y will be deterministic. If a state-variable X is observable, then $W_a(X = x)$ is assumed to be given by the formula $X = x$.

As an illustration, if X encodes the position of an agent, and Y encodes the position of an object that can be detected by the agent when $X = Y$, we can have an observable variable $Z \in \{Yes, No\}$ with model formula $W_a(Z = Yes)$ given by $\vee_{l \in D}(X = l \wedge Y = l)$, and model formula $W_a(Z = No)$ given by the negation of $W_a(Z = Yes)$. Here D is the set of possible locations and a is any action. This will be a deterministic sensor. A non-deterministic sensor could be used if, for example, the agent cannot detect the presence of the object at certain locations $l \in D'$. For this, it suffices to set $W_a(Z = No)$ to the disjunction of the previous formula and $\vee_{l \in D'}(X = l \wedge Y = l)$. The result is that the observation $Z = No$ becomes possible in the states where $X = Y = l$ holds for $l \in D'$, where $Z = Yes$ is possible too.

A contingent problem is a tuple $P = \langle V, I, A, G, V', W \rangle$ that defines a contingent model that is given by the conformant model $\langle S, S_0, S_G, A, F \rangle$ determined by the first four components in P, and the sensor model $O(a, s)$ determined by the last two components, where $o \in O(a, s)$ iff o is a valuation over the observable variables $Y \in V'$ such that $Y = y$ is true in o only if the formula $W_a(Y = y)$ in W is true in s for $y \in D_Y$.

Flat Belief Tracking

Conformant planning can be constructed as a special case of contingent planning where there is just one observation token o. From a syntactic point of view, this occurs when there is one observable variable Y with singleton domain $D_Y = \{y\}$ and formula $W_a(Y = y) = true$ for all actions a. We just focus then on contingent planning; the results easily generalize to conformant planning through this reduction.

Given an execution $a_0, o_0, a_1, o_1, \ldots$ the problem of *belief tracking* for conformant and contingent planning where action preconditions and goals must be achieved with certainty is the following:

Definition 1 Belief tracking *in planning is the task of determining the goals and action preconditions that are true in the belief that results from any possible execution $a_0, o_0, a_1, o_1, \ldots a_n, o_n$ over a given problem P.*

The plain solution to this task is given by Equations 1 and 2 where the states, actions, transition function, and observations, are obtained from the syntactic representations. We call the resulting algorithm *flat belief tracking*. The complexity of flat belief tracking is exponential in the number of state variables $|V|$. Yet, often some variables in V are always known with certainty and such variables do not add to the complexity of tracking beliefs. Semantically, a state variable X is *always known* when the value of X is known in the initial belief state b_0 and in any belief state reachable from b_0. Likewise, X is known in a belief state b if it has the same value in all the states in b. Syntactically, a sufficient and common condition for a state variable X to be known is that X is observable, or that some literal $X = x$ is known to be true in the initial situation and that no action makes the value of X uncertain. The latter happens when X is in the head of a non-deterministic effect, or when X is in the head of deterministic effect whose body involves variables that are not known. If we let V_K stand for the set of state variables in V that are always known, the complexity of belief tracking can be expressed as:

Theorem 2 *Flat belief tracking is exponential in $|V_U|$, where $V_U = V \setminus V_K$ and V_K is the set of state-variables in V that are always known with certainty.*

Below we develop a notion of width that improves this complexity bound.

Width

For a variable X in the problem, whether a state variable, an observation variable, or both, we define the *direct causes* of X as follows:

Definition 3 *A variable X is a* direct cause *of a variable X' in a contingent problem P, written $X \in Ca(X')$, iff $X \neq X'$, and either X occurs in the body C of a conditional effect $C \to E_1 | \cdots | E_n$ such that X' occurs in a head E_i, $1 \leq i \leq n$, or X occurs in a formula $W_a(X' = x')$ for an observable variable X' and some $x' \in D_{X'}$.*

Basically, X is a direct cause of X' when uncertainty about X may affect the uncertainty about X' directly, not through other variables. Note that X is not necessarily a direct cause of X' if X appears as the precondition of an action that affects X'. This is because preconditions, unlike conditions, must be known with certainty, and hence do not propagate uncertainty to their effects. The notion of *causal relevance* is given by the transitive closure of direct causation:

Definition 4 *X is* causally relevant *to X' in P if $X = X'$, $X \in Ca(X')$, or X is causally relevant to a variable Z that is causally relevant to X'.*

In order to test whether a given literal $Z = z$ is known after a certain execution $a_0, a_1, \ldots$ in the *conformant setting*, it is possible to show that one can just progress the state over the variables X that are causally relevant to Z:

Proposition 5 *Belief tracking in the deterministic or* non-deterministic conformant *setting is exponential in the max number of variables that are all relevant to a variable appearing in an action precondition or goal.*

This bound is closely related to the bound obtained by Palacios and Geffner in the deterministic setting. Indeed, if we refer to the number of variables $Z \in V_U$ that are causally relevant to X, as the *conformant width* of X, and set the width of P as the max conformant width over the variables X that appear in action preconditions or goals, Proposition 5 simply says that belief tracking for a *non-deterministic* problem is exponential in the problem width. This width notion, however, is not equivalent to the notion of Palacios and Geffner when used in the deterministic setting as it is defined over *variables* rather than *literals*. We will say more about this distinction below. In general, however, the two accounts yield similar widths over most deterministic benchmarks.

In the contingent setting, there are variables that are *not* causally relevant to a given variable Z and yet whose uncertainty may affect Z. The situation is similar to the one arising in Bayesian Networks (Pearl 1988), where relevance flows both *causally*, in the direction of the arrows, or *evidentially*, from the *observations* against the direction of the arrows.

Definition 6 *X is* relevant *to X' if X is causally relevant to X', both X and X' are causally relevant to an observable variable Z, or X is relevant to a variable Z that is relevant to X'.*

Relevance is thus a transitive relation such that X is relevant to X' if X is causally relevant to X', or if both X' and X are causally relevant to an observable variable Z. Unlike Bayesian Networks, the relation is not symmetric though, as for example, X being causally relevant to X' does not imply by itself that X' is relevant to X. From a Bayesian Network perspective, the notion of relevance encodes 'potential dependency' given that some variables are observable and others are not.

As an example of these definitions, from an action effect involving a variable X in the body and variables Y and Z in the head, whether deterministic or not, it follows that X is relevant to Y, while Z is not. However, if in addition, Z is causally relevant to an observable variable, Z will be relevant to X, as both X and Z are causally relevant to the same observable variable, and hence Z will be relevant to Y too, by transitivity.

The *context* of a variable, $Ctx(X)$, is then defined as the set of variables in the problem that are relevant to X:

Definition 7 *The context $Ctx(X)$ of X denotes the set of state variables in the problem that are relevant to X.*

Likewise, the width of a variable is the number of variables in its context that are not known:

Definition 8 *The* width *of a state variable X, $w(X)$, is $|Ctx(X) \cap V_U|$, where $V_U = V \setminus V_K$ and V_K is the set of variables that are always known.*

The width of a problem is then:

Definition 9 *The width $w(P)$ of a conformant or contingent problem P, whether deterministic or not, is $\max_X w(X)$, with X ranging over the variables that appear in a goal or action precondition in P.*

The key theorem can then be expressed as:

Theorem 10 *Belief tracking in P is exponential in $w(P)$.*

The proof for this theorem follows from the results below where an algorithm that achieves this complexity bound is presented. The significance of the theorem is that *belief tracking over planning domains with width bounded by a constant becomes polynomial in the number of problem variables.* We'll see examples of this below.

This complexity bound is similar to the ones obtained for *deterministic* conformant and contingent problems (Palacios and Geffner 2009; Albore, Palacios, and Geffner 2009). The main difference is that the new account applies to *non-deterministic* problems as well. The new account is simpler and more general, but as we will see, it is also slightly less tight on some deterministic problems.

Examples

We consider the width of some benchmark domains, starting with DET-Ring (Cimatti, Roveri, and Bertoli 2004). In this domain, there is a ring of n rooms and an agent that can move forward or backward along this ring. Each room has a window which can be opened, closed, or locked when closed. Initially, the status of the windows is not known and the agent does not know his initial location. The goal is to have all windows locked. A plan for this *deterministic conformant* problem is to repeat n times the actions $(close, lock, fwd)$, skipping the last fwd action. Alternatively, the action fwd can be replaced by the action bwd throughout the plan. The state variables for the problem encode the agent location $Loc \in \{1, \ldots, n\}$, and the status of each window, $W(i) \in \{open, closed, locked\}$, $i = 1, \ldots, n$. The location variable Loc is (causally) relevant to each window variable $W(i)$, but no window variable $W(i)$ is relevant to Loc or $W(k)$ for $k \neq i$. No variable is always known and the largest contexts are for the window variables $W(i)$ that include two variables, $W(i)$ itself and Loc. As a result the width of the domain is 2, which is independent of the number of state variables that grows with the number of rooms n.

NON-DET-Ring is a variation of the domain where any agent move, fwd or bwd, has a non-deterministic effect on the status of all windows that are not locked, capturing the possibility of external events that can open or close unlocked windows. This non-determinism has no effect on the relevance graph over the variables. As a result, the change has no effect on the contexts or domain width that remains bounded and equal to 2 for any number of rooms n.

The last version of the domain considered by Cimatti, Roveri, and Bertoli is NON-DET-Ring-Key, where a key is required to lock the windows. The initial position of the key is not known, yet if the agent tries to collect the key from a room and the key is there, the agent will hold the key. A conformant plan for this problem is to repeat the actions $pick$ and fwd, n times, skipping the last fwd action, following then the plan for DET-Ring. In NON-DET-Ring-Key, there is an additional state variable, $LocK \in \{1, \ldots, n, H\}$, that represents the key location that includes the possibility of being held. The agent location Loc is relevant to $LocK$ which is relevant to each window variable $W(i)$. As a result, both the size of the contexts $Ctx(W(i))$ and the problem width increase by 1. So, the problem width remains bounded and independent of n, with value 3.[1]

In the presence of partial observability, the analysis is similar but it is necessary to consider the relevance relationships that arise due to the presence of observable variables. For example, one can express that the agent can always observe whether is holding the key or not, by having an observation variable $Y \in \{H, nH\}$ with model formulas $W_a(Y = H)$ and $W_a(Y = nH)$ given by $LocK = H$ and $LocK \neq H$ respectively for all actions a. The only new relevance relationship among state variables that arises from this observable variable is between Loc and $LocK$, as both are causally relevant to Y. Before, Loc was relevant to $LocK$ but not the other way around. Yet this does not affect the domain width that remains 3 for any n.

Factored Belief Tracking

We focus now on a belief tracking algorithm that achieves the complexity bound established in Theorem 10 and hence is exponential in the problem width. The key is to exploit the relevance relations encoded in the variable contexts for decomposing beliefs. In particular, if no variable is relevant to any other variable, the problem width is 1 and beliefs over each variable can be maintained separately.

For a problem $P = \langle V, I, A, G, V', W \rangle$ and a state-variable $X \in V$, we will denote by P_X the planning problem P *projected* over the variables in the context of X, $Ctx(X)$. The projected problem $P_X = \langle V_X, I_X, A_X, G_X, V'_X, W_X \rangle$ is such that V_X is given by the state variables in $Ctx(X)$

[1] The problem can also be encoded by making 'holding key' a *precondition* rather than a *condition* for locking the windows. In such an encoding, the variable $LocK$ is no longer relevant to the window variables $W(i)$ according to the definitions, as then $LocK = H$ must be known with certainty, and hence uncertainty about the windows variables $W(i)$ is not affected by uncertainty about $LocK$. The result is that in such an encoding, the domain width reduces to 2.

only, I_X and G_X are the initial and goal formulas I and G logically projected over the variables in V_X,[2] A_X is A but with the preconditions and conditional effects projected over the variables in V_X,[3] V'_X is V', and W_X are the formulas $W_a(Y = y)$ in W projected over the variables in V_X. Notice that variables Y that are both state and observation variables in P but are not relevant to X, will belong to V'_X but not to V_X, meaning that they will be just observation variables in the projected problem P_X. Moreover, the formulas for such variables Y in W_X will become $W_a(Y = y) = true$ for all $y \in D_Y$, meaning that in the problem P_X, the observations $Y = y$ will be possible for any y, regardless of the state and last action done. Such observations will thus be completely irrelevant in P_X and will have no effect on the resulting beliefs.

We state two important results for the projected problems P_X. With no loss of generality, we assume that I contains just literals.[4]

Proposition 11 *If an execution $a_0, o_0, a_1, o_1, \ldots$ is possible in P, then the execution $a_0, o_0, a_1, o_1, \ldots$ is possible in the projected problem P_X.*

Proposition 12 *A literal $X = x$ or $X \neq x$ is known to be true in the belief state b that results from a possible execution in P iff the same literal is known to be true in the belief state b_X that results from the same execution in the projected problem P_X.*

The last result is critical because belief tracking in the projected problem P_X is exponential in the width of X:

Proposition 13 *Flat belief tracking in the projected problem P_X for a variable X in P is exponential in $w(X)$.*

This is direct from Theorem 2 as the width of X corresponds to the number of state variables in the projected problem P_X that are not always known. It then follows from this and Proposition 12 that:

Theorem 14 Flat belief tracking *over each of the* projected problems *P_X, for X being a variable in an action precondition or goal of P, provides a* factored algorithm *for belief tracking over P that is exponential in the width $w(P)$ of P.*

In other words, in order to determine whether a goal or action precondition $X = x$ is true after an execution $a_0, o_0, a_1, o_1, \ldots$, we just need to consider this execution over the projected problem P_X and test the literal $X = x$ in the resulting belief. Since there is a linear number of preconditions and goals variables, and belief tracking in each of the projected problems P_X is exponential in $w(X)$, the result in the theorem follows.

As an illustration, in order to test the goals $W(i) = locked$ in the domain NON-DET-Ring-Key, $i = 1, \ldots, n$, we just need to do flat belief tracking using Equations 1 and 2 over each of the projected problems $P_{W(i)}$, each featuring the 3 state variables in the context of $W(i)$; namely, the variable $W(i)$, the agent location Loc, and the key location $LocK$. All the other variables in the problem P can be ignored for determining the status of the $W(i)$ variables, and indeed, they are not part of the projected problem $P_{W(i)}$.

It is important to notice that the vector of beliefs b_X that result from an execution $a_0, o_0, a_1, o_1, \ldots$ over each of the projected problems P_X, allows us to determine the status of the X variables, and hence the status of preconditions and goals. Yet this *factored form of belief tracking does not provide a factored representation of the global belief* b that would result from the same execution over the original problem P. That is, $X = x$ is true or false in b_X iff $X = x$ is true or false in b respectively, yet the belief b itself cannot be obtained from the composition of the factored beliefs b_X. More precisely, the factorization is complete for determining the truth of goals and preconditions but not for determining the truth of arbitrary formulas. As an illustration, if P contains three boolean state variables X_1, X_2, and X_3, all them initially true, and an action a is applied that has the conditional effect $X_1 \rightarrow X_2, X_3 | \neg X_2, \neg X_3$, making thus both X_2 and X_3 true or both false, then if X_2 and X_3 are each relevant to a different goal, the formula $X_2 \equiv X_3$ will follow from the resulting global belief b but not from a composition of the factored beliefs b_2 and b_3 that does not track the coupling between X_2 and X_3. The theory implies that tracking this relation is not necessary for tracking the truth of precondition and goal variables as long as X_2 and X_3 are not both relevant to one such variable Z. The same situation arises if the non-deterministic effect is changed to a deterministic effect $X_1 \rightarrow \neg X_2, \neg X_3$, and X_1 is initially unknown.

Experiments

We have experimentally tested the factored belief tracking algorithm over a few planning domains. We used the algorithm in the context of simple planning procedures and heuristics. The vector b of local beliefs b_X that result a given execution $a_0, o_0, a_1, o_1, \ldots$ and initial belief is used to determine several elements of the resulting planners; namely, termination (i.e., whether b is a goal belief), the actions that apply in b, the observations o that may follow from a in b, the representation of the resulting belief b_a^o, and the heuristic $h(b)$ that approximates the cost to the goal. The experiments were conducted on a Xeon 'Woodcrest' machine at 2.33 GHz with 8 Gb of RAM.

The first two domains are DET-Ring-Key and NON-DET-Ring-Key, explained above, from (Cimatti, Roveri, and Bertoli 2004). Both domains are conformant and their width is 3. The number of unknown variables is $n+2$ where n is the number of rooms. The number of problem states is $n^2 \times 3^n$ and the number of belief states exponential in this number. The planner KACMBP by Cimatti, Roveri, and Bertoli

[2] The logical projection over a formula F over a subset S of its variables, is the formula F' defined over the variables in S only, such that the valuations that satisfy F' are exactly those that can be extended into valuations that satisfy F (Darwiche and Marquis 2002).

[3] The projection of a conditional effect $C \rightarrow E^1 | \cdots | E^1$ is the conditional effect $C_X \rightarrow E_X^1 | \cdots | E_X^n$ where the body C and the effects E^i are replaced by their logical projections C_X and E_X^i.

[4] Arbitrary formulas in DNF can always be captured in the initial situation by means of literals and dummy non-deterministic actions that are forced to apply just once at the beginning of any plan.

uses an OBDD-based belief representation and cardinality heuristics, and can solve problems with up to $n = 20$ rooms, producing plans with 206 steps in slightly more than 1000 seconds. Other conformant planners such as T0 cannot be used as the problem is non-deterministic. Tables 1a and 1b show the scalability of the belief tracking algorithm in the context of a greedy best-first search with a heuristic $h(b)$, similar to the one in (Albore, Ramirez, and Geffner 2011), given by $\sum_{i=1,n} h(b_i)$, where b_i is the belief factor in the projected problem for the goal variable $W(i)$ representing the status of the i-th window, and $h(b_i)$ is the fraction of states in b_i where the goal $W(i) = locked$ is false. As displayed in the Tables, the resulting planner scales up to instances with up to 80 rooms, producing plans with 616 steps in 187.37 seconds.

We considered also a contingent version of the problem where the agent can detect whether the key is in the room. In this case, we combined the belief tracking algorithm with an *on-line planner* that uses a version of the AO* algorithm (Nilsson 1980), run for a fixed number of iterations (10 iterations), as a lookahead mechanism for selecting the action to do next (Bonet and Geffner 2012), similar to the use of A* in (Koenig and Sun 2009). The heuristic used in this lookahead search was the same as above, and ties among the best actions were broken randomly. The results are shown in Table 1c, where averages resulting from the on-line planner applied to different sampled hidden states are shown. Simulations that do not reach the goal in 500 steps are cut, contributing 500 steps to the average.

The last domain is a simple version of the Wumpus problem where an agent has to pick up the gold in a grid of size 10×10, inhabited by a hidden Wumpus, Pits, and other objects to be avoided, each one emitting a distinct signal that can be detected from nearby cells. The location of the gold is also unknown. The gold emits a distinct signal that can be detected only from the same cell. Table 2 shows the results for running the belief tracking algorithm in the context of the above on-line planner and a heuristic given by the distance to the nearest unvisited cell. The width of the problem is 1, while the number of unknown variables is $n + 1$, where n is the number of objects to avoid. The state space is in the order of 100^{n+1}, and the belief space is exponential in this number. As it can be seen from the table, up to $n = 6$ objects can be solved.

grid dim.	#objects	avg. steps	avg. time
10×10	0	57.4 ± 46	43.6 ± 37
10×10	1	137.6 ± 204	113.7 ± 167
10×10	2	145.8 ± 200	195.7 ± 259
10×10	3	191.2 ± 177	538.0 ± 438
10×10	4	114.0 ± 57	953.6 ± 506
10×10	5	48.0 ± 34	1552.6 ± 1001
10×10	6	129.6 ± 105	8714.7 ± 4716

Table 2: Results for the Wumpus domain on a 10×10 grid with a variable number of objects plus the gold. A problem with n objects has $100\binom{100}{n+1}$ states and $O(2^{100^{n+1}})$ belief states. Each data point is the average over 5 random instances. Problems are solved with factored belief tracking and an on-line search algorithm that relies on a bounded AO* lookahead search (Bonet and Geffner 2012). Standard deviations are high because the difficulty of the instances for a fixed number of objects varies widely.

Discussion

The formulation in the paper is closely related to recent translation-based approaches to conformant and contingent planning that compile beliefs away (Palacios and Geffner 2009; Albore, Palacios, and Geffner 2009). These translations, however, assume that the problems are *deterministic*, and they can all be understood as mapping the uncertain initial situation I into a DNF formula $\varphi_L = \vee_i t_i$ for each precondition and goal literal L, where each 'tag' or assumption t_i is a conjunction of literals that are uncertain in the initial situation and relevant to L. The width of L is related to the size $|t_i|$ of the assumptions required for φ_L to be valid and to 'cover' L. The assumptions t_i are valid if one of the them is true in I (i.e., $I \models \varphi_L$), and they cover L if for each t_i there is a state s such that 1) s satisfies $I \cup \{t_i\}$ and 2) $I \cup \{t_i\}$ implies all literals L' true in s that are relevant to L. Our account yields similar widths on most deterministic benchmarks, and is both simpler and more general because it handles non-determinism. Yet it is also less tight on some deterministic problems. As an illustration, if $I = \{x_1 \vee \cdots \vee x_n\}$ and the actions are a_i, each with conditional effect $x_i \rightarrow G$, $i = 1, \ldots, n$, the conformant problem with goal G has width 1 in Palacios and Geffner's account, but width n in our account. The relevance account based on literals is indeed finer than the one based on variables but it is also more difficult to generalize to non-deterministic settings. This difference does not seem to have practical effects over most benchmarks where disjunctions in the initial situation are exclusive ($oneof$'s).

Another importance difference with these approaches is that complete translations are *always* exponential in the problem width not just worst case exponential. On the other hand, our complexity bound is worst case; i.e., if the variables in the contexts are highly correlated, the actual complexity of factored belief tracking can be much lower.

The notion of width appears also in Bayesian Networks where inference is exponential in the width of the network (Pearl 1988). Three differences that can be pointed out in relation to our notion of width are that 1) we exploit the knowledge that certain variables are not observable, 2) we can determine and use the knowledge that certain variables will be always known, and 3) we make use of the distinction between action conditions and preconditions in planning. As an example, a problem where an agent has to go through n doors whose status, open or closed, can only be observed when the agent is near the door, will have width no smaller than n when modeled as a Dynamic Bayesian Network, as all the door variables affect the agent location variable. In our setting, however, the problem has width 1 because the status of a door need to be known to the agent before it can open, close or walk the door.

n	steps	expanded	time
10	68	355	0.01
20	138	705	0.21
30	208	1055	1.14
40	277	1400	3.73
50	345	1740	9.65
60	415	2090	21.43
70	476	2395	39.57
80	545	2740	71.75

(a) DET-Ring-Key

n	steps	expanded	time
10	118	770	0.05
20	198	1220	0.91
30	278	1670	4.67
40	488	3210	16.39
50	438	2570	37.40
60	468	2660	56.35
70	543	3080	108.35
80	616	3480	187.37

(b) NON-DET-Ring-Key

n	avg. steps	avg. time
6	35.6 ± 2.3	1.23 ± 0.1
8	68.8 ± 13.0	6.03 ± 1.4
10	95.6 ± 27.0	16.38 ± 4.1
12	111.2 ± 16.9	33.69 ± 4.5
14	135.0 ± 16.7	64.56 ± 6.8
16	168.6 ± 25.0	114.53 ± 20.6
18	184.6 ± 14.9	185.81 ± 24.4
20	238.0 ± 33.2	386.90 ± 110.2

(c) CONT-Ring-Key

Table 1: Results for conformant and contingent Ring problems. A problem with n rooms has $n^2 3^n$ states and $O(2^{n^2 3^n})$ belief states. Panels (a) and (b) are the conformant deterministic (DET-Ring-Key) and non-deterministic (NON-DET-Ring-Key) problems respectively, while panel (c) is a contingent problem (CONT-Ring-Key). Conformant problems are solved with a (greedy) best-first search that uses the evaluation function $f(b) = h(b)$ for the heuristic h described in the text. The contingent problem is solved with factored belief tracking and an on-line search algorithm that uses a version of AO* in a bounded lookahead search (Bonet and Geffner 2012). Each data point in panel (c) is the average (and sampled standard deviation) over 5 random instances.

Conclusions

We have introduced a new notion of width for planning with incomplete information and showed that belief tracking is exponential in the problem width even in the presence of non-deterministic actions and non-deterministic sensing. We have also introduced a factored belief tracking algorithm that achieves this bound, tested it over some domains, and analyzed the width of several problems. In the future, we want to apply these results to POMDPs where uncertainty is not modeled through sets but through probability distributions. In addition, we want to use the formulation to introduce belief approximations by partitioning the projected subproblems P_X further and by using tractable forms of inference for enforcing a degree of consistency among the different partitions.

Acknowledgments. H. Geffner is partially supported by grants TIN2009-10232, MICINN, Spain, and EC-7PM-SpaceBook.

References

Albore, A.; Palacios, H.; and Geffner, H. 2009. A translation-based approach to contingent planning. In *Proc. IJCAI-09*, 1623–1628.

Albore, A.; Ramirez, M.; and Geffner, H. 2011. Effective heuristics and belief tracking for planning with incomplete information. In *Proc. ICAPS*.

Amir, E., and Russell, S. 2003. Logical filtering. In *Proc. IJCAI*, 75–82.

Bertoli, P.; Cimatti, A.; Roveri, M.; and Traverso, P. 2001. Planning in nondeterministic domains under partial observability via symbolic model checking. In *Proc. IJCAI-01*.

Bonet, B., and Geffner, H. 2000. Planning with incomplete information as heuristic search in belief space. In *Proc. of AIPS-2000*, 52–61. AAAI Press.

Bonet, B., and Geffner, H. 2012. Action selection for MDPs: Anytime AO* vs. UCT. In *Proc. AAAI-2012*.

Brafman, R., and Hoffmann, J. 2004. Conformant planning via heuristic forward search: A new approach. In *Proc. ICAPS-04*.

Bryce, D.; Kambhampati, S.; and Smith, D. E. 2006. Planning graph heuristics for belief space search. *Journal of AI Research* 26:35–99.

Cimatti, A.; Roveri, M.; and Bertoli, P. 2004. Conformant planning via symbolic model checking and heuristic search. *Artificial Intelligence* 159:127–206.

Darwiche, A., and Marquis, P. 2002. A knowledge compilation map. *J. of AI Research* 17:229–264.

Doucet, A.; De Freitas, N.; Murphy, K.; and Russell, S. 2000. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proc. UAI*, 176–183.

Goldman, R. P., and Boddy, M. S. 1996. Expressive planning and explicit knowledge. In *Proc. AIPS-1996*.

Koenig, S., and Sun, X. 2009. Comparing real-time and incremental heuristic search for real-time situated agents. *Autonomous Agents and Multi-Agent Systems* 18(3):313–341.

Nilsson, N. 1980. *Principles of Artificial Intelligence*. Tioga.

Palacios, H., and Geffner, H. 2009. Compiling Uncertainty Away in Conformant Planning Problems with Bounded Width. *Journal of Artificial Intelligence Research* 35:623–675.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

Shani, G., and Brafman, R. 2011. Replanning in domains with partial information and sensing actions. In *Proc. IJCAI-2011*, 2021–2026.

Smith, D., and Weld, D. 1998. Conformant graphplan. In *Proc. AAAI*, 889–896.