From Conformant into Classical Planning: Efficient Translations That May be Complete Too

Héctor Palacios
Departamento de Tecnología
Universitat Pompeu Fabra
08003 Barcelona, SPAIN
hector.palacios@upf.edu

Advisor: Héctor Geffner
Departamento de Tecnología
ICREA & Universitat Pompeu Fabra
08003 Barcelona, SPAIN
hector.geffner@upf.edu

Abstract

Focusing on the computation of conformant plans whose verification can be done efficiently, Palacios and Geffner have recently proposed a polynomial scheme for mapping conformant problems P with deterministic actions into classical problems $K(P)$. The scheme is sound as the classical plans are all conformant, but is incomplete as the converse relation does not always hold. In this paper, we build on this work, and consider an alternative, more powerful translation based on the introduction of tagged literals KL/t where L is a literal in P and t is a *set* of literals in P unknown in the initial situation. The translation ensures that a plan makes KL/t true only when the plan makes L certain in P given the assumption that *t is initially true*. We show that a conformant planner based on this translation solves some interesting domains that cannot be solved apparently by other planners.[1]

Introduction

Conformant planning is the problem of finding a sequence of actions for achieving a goal in the presence of uncertainty in the actions or initial state (Goldman & Boddy 1996). While few practical problems are purely conformant, the ability to find conformant plans appears to be a necessity in contingent planning where conformant situations are an special case (null observability being an special case of partial observability) and where relaxations into conformant planning appear to provide useful heuristics (Brafman & Hoffmann 2004).

In this paper, we build on the work of (Palacios & Geffner 2006) and consider an alternative translation scheme $K_i(P)$ that overcomes some of the theoretical and practical limitations of that approach. In particular, we define a parameter $w(P)$ that we call the *conformant width* of P. The conformant width of a problem P measures the maximum number of explicit and implicit disjunctions in the initial situation that are relevant to a precondition or goal. The translation $K_i(P)$ is complete for problems with width i, being exponential only in i. We will see that almost all conformant benchmarks have actually width equal to 1. We then take advantage of this for developing and testing a conformant planner based on this translation.

[1]Full version accepted for ICAPS-2007

General Translation Scheme $K_{T,M}(P)$

Following (Palacios & Geffner 2006), a conformant planning problem P is a tuple $P = \langle F, O, I, G \rangle$ where F stands for the fluent symbols in the problem, O stands for a set of actions a, I is a set of clauses over F defining the initial situation, and G is a set of literals over F defining the goal. In addition, every action a has a precondition given by a set of fluent literals, and a set of deterministic conditional effects $C \rightarrow L$ where C is a set of fluent literals and L is a literal.

We consider a family of translations that can all be understood as arising from a common pattern that we refer as $K_{T,M}(P)$, where T and M are a set of *tags* and a set of *merges* respectively. A tag $t \in T$ is a set of literals L from P whose truth value is not known in the initial situation I. The tagged literals KL/t, where L is a literal in P and $t \in T$ is a tag, capture the conditional 'it is known that if t is true initially, then L is true', which we would write in logic as $K(t_0 \supset L)$. The key issue is that tags can be *sets* of literals and they all refer to conditions in the *initial situation* only. Roughly $\neg KL/t$ means that the conditional $K(t_0 \supset L)$ is not true, while $K\neg L/t$ means that the conditional $K(t_0 \supset \neg L)$ is true.

Each merge $m_{R,L} \in M$ is a pair $m_{R,L} = \langle R, L \rangle$ where where L is a literal in P and $R \subseteq T$ is a collection of tags in T. A collection of tags R is *valid* when one of the tags $t \in R$ must be true in I; i.e., when

$$I \models \bigvee_{t \in R} t \, .$$

We say that a merge $m = \langle R, L \rangle$ is *valid* when its collections of tags R is *valid*. We assume that all merges are valid in this sense. Merges $m_{R,L}$ will map into 'merge actions' $m_{R,L}$ with effects

$$m_{R,L} : \bigwedge_{t \in R} KL/t \rightarrow KL \, .$$

For example, a valid merge $m = \langle R, L \rangle$, with R as the set of literals of a clause $C \in I$, and one tag corresponding to each literal $l \in C$.

We assume that T always includes a tag t that stands for the empty collection of literals, that we call the *empty tag*. If t is empty, we denote KL/t simply as KL. Similarly, for a set (conjunction) C of literals $L_1, L_2, \ldots$; KC/t

stands for $KL_1/t, KL_2/t, \ldots$, while $\neg K\neg C/t$ stands for $\neg K\neg L_1/t, \neg K\neg L_2/t, \ldots$.

Wrapping up, the general translation $K_{T,M}(P)$ is:

Definition 1 ($K_{T,M}(P)$) *Let $P = \langle F, O, I, G\rangle$ be a conformant problem, then $K_{T,M}(P) = \langle F', I', O', G'\rangle$ is defined as:*

- $F' = \{KL/t, K\neg L/t \mid L \in F \text{ and } t \in T\}$
- $I' = \{KL/t \mid \text{if } L \in I \text{ or } L \in t\}$
- $G' = \{KL \mid L \in G\}$
- $O' = \{a : KC/t \rightarrow KL/t,\ a : \neg K\neg C/t \rightarrow \neg K\neg L/t \mid a : C \rightarrow L \text{ in } P\} \cup \{m_{R,L} : [\bigwedge_{t\in R} KL/t] \rightarrow KL | m_{R,L} \in M\}$

where KL is a precondition of action a in $K_{T,M}(P)$ if L is a precondition of a in P.

The translation scheme $K_{T,M}(P)$ reduces to the core translation $K_0(P)$ (Palacios & Geffner 2006), which is equivalent to 0-approximation (Baral & Son 1997). On the other hand, for suitable choices of T and M, we will see that the new translation scheme is *complete*, and under certain conditions, both *complete and polynomial*. At the same time the scheme is simpler than $K(P)$ (Palacios & Geffner 2006).

Theorem 2 (Soundness $K_{T,M}(P)$**)** *If π is a plan that solves the classical planning problem $K_{T,M}(P)$, then the action sequence π' that results from π by dropping the merge actions is a plan that solves the conformant planning problem P.*

Example 1 Consider the problem of moving an object from an unknown origin to a destination. For instance, let $I = at_1 \vee at_2$ and $G = at_3$. We can use actions: $pick(l)$ that picks up the object if the object is at l *and* the hand is empty, while if the hand is not empty, $pick(l)$ just releases the object at l, and $drop(l)$ that drops the object in a location if the object is being held. Let us assume that these are conditional effects, and that there are no preconditions.

Let us consider now the translation $K_{T,M}(P)$ with $T = \{at_1, at_2\}$, and the single merge $m_{T,L} \in M$ with $L = at_3$ that is valid as $at_1 \vee at_2$ is true in I. We can show now that the plan π'_2

$$\{pick(l_1), drop(l_3), pick(l_2), drop(l_3), m_{T,L}\}$$

for $L = at_3$ solves the classical problem $K_{T,M}(P)$ and hence, from Theorem 2, that the plan π_2 obtained from π'_2 by dropping the merge action, is a valid conformant plan for P. We can see how some of the literals in $K_{T,M}(P)$ evolve as the actions in π'_2 are done:

0:$Kat_1/at_1, Kat_2/at_2$	true in I'
1:$Khold/at_1, Kat_2/at_2$	true after $pick(l_1)$
2:$Kat_3/at_1, Kat_2/at_2$	true after $drop(l_3)$
3:$Kat_3/at_1, Khold/at_2$	true after $pick(l_2)$
4:$Kat_3/at_1, Kat_3/at_2$	true after $drop(l_3)$
5:Kat_3	true after merge $m_{T,L}$

where the merge $m_{T,L}$ is the action with the conditional effect

$$Kat_3/at_1 \wedge Kat_3/at_2 \rightarrow Kat_3$$

whose condition is true before Step 5 producing Kat_3

The Translation $K_{S0}(P)$

A *complete* instance of the translation scheme $K_{T,M}(P)$ can be obtained in a simple manner by setting

- T to the union of the empty tag and the set of possible initial states s_0 (understood as the maximal consistent set of literals in I), and
- M to the merges $m_{T,L}$ with T as above and each literal L in P.

We will denote this instance, where the tags t range over the possible initial states s_0, as $K_{S0}(P)$, which is not only sound but complete: for every conformant plan π for P, there is a plan π' for $K_{S0}(P)$, obtained by extending π with merge actions.

Width: Exploiting Relevance and Structure

The translation $K_{S0}(P)$ introduces a number of literals KL/t that is exponential in the worst case : one for each possible initial context t. If for each literal L we could then remove all literals L' that are irrelevant to L from the tags t in the translation $K_{S0}(P)$, we could end up with a bounded number of tagged literals KL/t while retaining completeness.

Definition 3 (Conformant Relevance) *The relevance relation $L \longrightarrow L'$ in P, read L is relevant to L', is defined inductively as*

1. $L \longrightarrow L$
2. $L \longrightarrow L'$ if $L \longrightarrow L''$ and $L'' \longrightarrow L'$
3. $L \longrightarrow L'$ if $a : C \rightarrow L'$ in P with $L \in C$
4. $L \longrightarrow L'$ if $L \longrightarrow \neg L''$ and $L'' \longrightarrow \neg L'$
5. $L \longrightarrow L'$ if both $\neg L$ and L' in a clause in I.

The first two clauses defining relevance stands for reflexivity and transitivity, the third captures conditions relevant to the effect, and the last captures deductive relevance in the initial situation. The fourth clause, which is the least obvious, captures conditions under which L preempts conditional effects that may delete L'. This definition is equivalent to the one in (Son & Tu 2006). For more details, see the full paper.

Let C_I the set of non-unary clauses in I along with the tautologies $L \vee \neg L$ for complementary literals L and $\neg L$ that do not appear as unary clauses in I.[2] We will denote by $C_I(L)$ the set of clauses in C_I relevant to the literal L in P.

Let us also say that a clause $c \in C_I$ *subsumes* another clause $c' \in C_I$, written $c \preceq c'$, if for every literal $L \in c$ and for some literal $L' \in c'$, $I \models L \supset L'$, and let us keep in $C^*_I(L)$, a minimal set of clauses from $C_I(L)$, such that all clauses in $C_I(L)$ are subsumed by a clause in $C^*_I(L)$. We can then define the *conformant width* parameter $w(P)$ of a problem P in terms of the number of 'irredundant' clauses in $C_I(L)$ over all preconditions and goal literals L:

Definition 4 (Conformant Width) *Let the* width of a literal *L in P, written as $w(L)$, be $w(L) = |C^*_I(L)|$, and let the* width of the conformant problem P, $w(P)$, *be the max width of any* precondition or goal literal L.

[2] In order to have polynomial but complete translations we need to assume that I is in *prime implicate (PI) form*. More details in the full paper

The Translation $K_i(P)$

The translations $K_i(P)$, parametrized with the non-negative integer i, are complete for problems with width no greater than i and have a complexity that is exponential only in i.

For example, assuming $i = 1$ we get $K_i(P)$ is $K_{T,M}$ where

- T is the union of the empty tag and the set of literals L (i.e., singletons) in some clause in $C_I(L)$ for some precondition or goal literal L,
- M is the set of merges $m_{R,L}$ where L is a precondition or goal literal L and R is the set of literals in a clause in $C_I(L)$.

The translation $K_i(P)$ applies to problems P of any width, remaining in all cases exponential only in i but polynomial in both the number of fluents and actions in P.

Theorem 5 (Soundness and Completeness of $K_i(P)$) *For conformant problems P with width bounded by i, the translation $K_i(P)$ is sound, complete, and exponential only in i.*

The Planner T_0

The conformant planner T_0 is an optimized and slightly extended version of the $K_i(P)$ translation for $i = 1$ combined with the FF classical planner v2.3 (Hoffmann & Nebel 2001). T_0 won the conformant track of the IPC-5 (Bonet & Givan 2006). The $K_1(P)$ translation is provably complete for problems with width 1, but may also solve problems with higher widths as well (later on we discuss such examples). The optimization in the $K_1(P)$ translation comes from a simple observation: all the schemes considered above are *uniform* in the sense that the same set of tags T is used over all the literals in P. Yet, whenever a tagged literal KL/t has a tag t that includes literals L' that are not relevant to L, such literals can be removed from t so that literals KL/t are encoded by means of tagged literals KL/t' where t' is the relevant part of t. This simplification decreases the size of the resulting encoding considerably without affecting the semantics of the translation. In particular, in $K_1(P)$, the tags t have a size no greater than 1 so that all literals KL/t where t contains a literal that is not relevant to L are represented effectively by the same literal KL.

Experimental Results

Table 1 shows the plan times and lengths obtained by T_0 and Conformant FF (Brafman & Hoffmann 2004) on several standard domains, taken from the Conformant-FF distribution and from the recent competition (Bonet & Givan 2006). In all these domains, T_0 scales up very well. The results of T_0 over a family of grid problems in (Palacios & Geffner 2006) are not presented. We found solutions for them with plans from one hundred to thousands of actions, but Conformant FF could not solve them within the time limits.

Summary

While few practical problems are purely conformant, the ability to find conformant plans fast appears to be a necessity in contingent planning where conformant situations are a special case. We have built on the work proposed by Palacios and Geffner, to introduce a novel and general translation scheme that maps conformant problems into classical problems; these can then be solved efficiently by an off-the-shelf-planner. The translation scheme depends on two parameters: a set of tags, referring to local contexts in the initial situations, and a set of merges that stand for exhaustive sets of tags. We have seen how different translations can be obtained from suitable choices of tags and merges, have introduced a measure of complexity in conformant planning called *conformant width*, and have introduced a translation scheme $K_i(P)$ that involves only tags of size i that is complete for problems of width $\leq i$. We observed that that most conformant benchmarks have width 1, have developed a conformant planner based on the $K_i(P)$ translation that uses the FF classical planner, and have shown that this planner exhibits good performance over the existing domains and some challenging new domains.

problem	T_0	len	CFF	len
Bomb-100-60	5,6	140	9,38	140
Sqr-8-ctr	0,07	26	70,63	50
Sqr-12-ctr	0,1	32	$> 2h$	
Sqr-64-ctr	10,68	188	$> 2h$	
Log-3-10-10	3,42	109	4,67	108
Log-4-10-10	6,52	125	4,36	121
Ring-4	0,09	13	1,37	26
Ring-5	0,1	17	27,35	45
UTS-K10	1,09	58	16,53	58
UTS-L10	0,33	88	1,64	59
Comm-24	0,7	418	37,52	359
Comm-25	0,84	453	56,13	389

Table 1: Plan times in seconds and lengths over standard domains. Run on a Linux machine running at 2.33 Ghz with 8Gb of RAM, with a cutoff of 2h or 1.8G of memory.

References

Baral, C., and Son, T. C. 1997. Approximate reasoning about actions in presence of sensing and incomplete information. In *Proc. ILPS 1997*, 387–401.

Bonet, B., and Givan, B. 2006. Results of the conformant track of the 5th int. planning competition. IPC-5.

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

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

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

Palacios, H., and Geffner, H. 2006. Compiling uncertainty away: Solving conformant planning problems using a classical planner (sometimes). In *Proc. AAAI-06*.

Son, T. C., and Tu, P. H. 2006. On the completeness of approximation based reasoning and planning in action theories with incomplete information. In *Proc. KR-06*, 481–491.