Easy and Hard Conformant Planning

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

Even under polynomial restrictions on plan length, conformant planning remains a very hard computational problem as plan verification itself can take exponential time. This heavy price cannot be avoided in general although in many cases conformant plans are verifiable efficiently by means of simple forms of disjunctive inference. We report an efficient but incomplete planner capable of solving non-trivial problems quickly. In this work, we show that this is possible by mapping conformant into classical problems that are then solved by an off-the-shelf classical planner. The formulation is sound as the classical plans obtained are all conformant, but it is incomplete as the inverse relation does not always hold. Atoms L/X_i that represent conditional beliefs 'if X_i then L' are introduced in the classical encoding and combined with suitable actions when certain invariants are verified. Empirical results over a wide variety of problems illustrate the power of the approach. We propose extensions to this formulation.

Introduction

Conformant planning is a form of planning where a goal is to be achieved when the initial situation is not fully known and actions may have non-deterministic effects (Goldman & Boddy 1996). Conformant planning is computationally harder than classical planning, as even under polynomial restrictions on plan length, plan verification remains hard (Turner 2002). This additional complexity cannot be avoided in general. This difference in complexity explains why it is still very easy to come up with simple and small conformant problems that no general domain-independent planner can solve, while the same is no longer true for classical planners. The main motivation of this work is to narrow this gap by developing an approach that targets 'simple' conformant problems effectively. The approach will not be complete but it will provide solutions to *some non-trivial conformant planning problems* by translating them intro *classical planning problems*(Palacios & Geffner 2006). New problems are fed into a classical planner. The translation is sound as the classical plans are all conformant, but it is incomplete as the converse does not always hold. The translation scheme accommodates 'reasoning by cases' by means of an 'split-protect-and-merge' strategy; namely, atoms L/X_i that represent conditional beliefs 'if X_i then L' are introduced in the classical encoding that are then combined by suitable actions when certain invariants in the plan are verified.

While several effective but incomplete formulations of conformant planning have been formulated before, like 0-approximation (Baral & Son 1997), none, as far as we know, can represent these types of plans, while those planners that can *represent* them (Cimatti, Roveri, & Bertoli 2004; Brafman & Hoffmann 2004), are not able to *compute* them except for very small problems.

Conformant Planning

For a conformant planning problem, if the number m of possible initial states $s_0 \in Init$ is bounded and actions are deterministic, the conformant planning problem P with a fixed horizon n can be mapped in the SAT problem over the formula (Palacios & Geffner 2005)

$$\bigwedge_{s_0 \in Init} T^{s0}(P, n) \tag{1}$$

where if $T(P, n)$ is the propositional theory that encodes the problem P with horizon n. $T^{s0}(P, n)$ is $T(P, n)$ with two modifications: first, fluent literals L_0 (L at time 0) are replaced by true/false iff L is true/false in the (complete) state s_0, and second, fluent literals L_i, $i > 0$, are replaced by 'fresh' literals L_i^{s0}, one for each $s_0 \in Init$.

Eq. 1 can be thought as expressing m 'classical planning problems', one for each possible initial state $s_0 \in Init$, that are *coupled* in the sense that they all share the same set of actions; namely, the action variables are the only variables shared across the different subtheories $T^{s0}(P, n)$ for $s_0 \in Init$.

For bounded m, the resulting class of conformant planning problems with a fixed horizon can be mapped polynomially into SAT, generalizing the SAT encoding of classical planning problems which corresponds to $m = 1$ (Kautz & Selman 1996). Also, for a sufficiently large horizon, this formulation is *complete*. In other words, for this interesting class of problems, the formulation of Eq 1 takes advantage of the reduced complexity without restricting the inferences at all. However, expressivity and complexity, however, are not the only problems; *efficiency* or control is the other. A planner using Eq. 1 naively will not scale.

We have already proposed two approaches to optimal classical conformant planning based on logical formulations (Palacios *et al.* 2005; Palacios & Geffner 2005). Both of them translate the problem into CNF, and obtain a plan by doing logical operations and search. The logical approach has been very importan on optimal classical planning (Kautz & Selman 1996), where they map it into SAT. In vplan (Palacios *et al.* 2005) we presented a complete optimal planner that rejects plan candidate by checking through model counting that it does not work for some initial state. In cf2sat (Palacios & Geffner 2005) (for *conformant2sat*) we construct a new propositional formula by doing logical operations as forgetting (Lin & Reiter 1994) and conditioning. The models of these new formula are all the possible plan. We feed that formula into a SAT solver to obtain a plan. Logical operations in both planners became feasible by compiling the propositional theory into d–DNNF (Darwiche 2002), a formal norm akin to OBDD. We obtained good results on some very complex domains but failed to scale in more simple problems.

One way to trade off completeness for efficiency in conformant planning results from approximating belief states (Bonet & Geffner 2000). For example, the 0-approximation introduced in (Baral & Son 1997) represents belief states bel by means of two sets: the set of literals that are true in bel, and the set of literals that are false in bel. Variables which do not appear in either set are unknown.

Conformant planning under the 0-approximation is thus no more complex, theoretically, than classical planning. The problem however is that the 0-approximation is strongly incomplete, as it does not capture any non-trivial form of disjunctive inference. For example, given a disjunction $p \vee q$ and an action a that maps either p or q into r, the semantics will not validate a as a conformant plan for r. Indeed, disjunctions that are not tautologies are thrown away.

Translation

The translation scheme maps a conformant planning problems P into a classical planning problems $K(P)$. We describe the contents of $K(P)$ in two parts, starting with the basic core $K_0(P)$. We assume that P is given by tuples of the form $\langle F, O, I, G\rangle$ 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 conditional effects $C \rightarrow L$ where C is a set of fluent literals and L is a literal. We assume that actions are all *deterministic* and hence that all uncertainty lies in the initial situation. We will usually refer to the conditional effects $C \rightarrow L$ of an action a as the *rules* associated with a, and sometimes write them as $a : C \rightarrow L$. Also, we use the expression $C \wedge X \rightarrow L$ to refer to rules with literal X in their bodies. In both cases, C may be empty. Last, when L is a literal, we take $\neg L$ to denote the complement of L.

Definition 1 (Core Translation) [1] *The core translation maps the conformant problem P into the classical problem $K_0(P) = \langle F', O', I', G'\rangle$ where*

- $F' = \{KL, K\neg L \mid L \in F\}$
- $I' = \{KL, \neg K\neg L \mid L \in I\} \cup \{\neg KL', \neg K\neg L' \mid L' \notin I\}$
- $G' = \{KL \mid L \in G\}$
- $O' = O$ *but with conditional effect $a : C \rightarrow L$ replaced by $a : KC \rightarrow KL$ and $a : \neg K\neg C \rightarrow \neg K\neg L$.*

For any literal L in P, KL denotes its 'epistemic' counterpart in $K_0(P)$ whose meaning is that L is known. We write KC for $C = L_1 \wedge L_2 \ldots$ as an abbreviation for $KL_1 \wedge KL_2 \ldots$, and $\neg K\neg C$ for $\neg K\neg L_1 \wedge \neg K\neg L_2 \ldots$.

The intuition behind the translation is simple: first, complementary literals L and $\neg L$ whose status is not known in the initial situation in P are 'negated', by mapping them into the literals $\neg KL$ and $\neg K\neg L$ that are jointly consistent. This mapping removes all uncertainty from $K_0(P)$. In addition, to ensure soundness, each conditional effect $a : C \rightarrow L$ in P maps, not only into the 'supporting' rule $a : KC \rightarrow KL$ but also into the 'cancellation' rule $a : \neg K\neg C \rightarrow \neg K\neg L$ that guarantees that literal $K\neg L$ is deleted (prevented to persist) when action a is applied except when C is known to be false.

We extend the translation further so that the disjunctions in P are taken into account in a form that is similar to the Disjunction Elimination inference rule used in Logic

$$\text{If } X_1 \vee \cdots \vee X_n,\ X_1 \supset L, \ldots, \text{and } X_n \supset L \text{ then } L \quad (2)$$

For this, we will create new atoms in $K(P)$, written L/X_i, that aim to capture the conditional beliefs $X_i \supset L$. Then, the resulting classical encoding will be such that once these atoms are 'achieved' for each $i = 1, \ldots, n$, and when they are suitably 'protected', the literal L will be made 'achievable' by an extra 'dummy' action with conditional effect similar to (2). In principle, any rule $a : C \wedge X_i \rightarrow L$ in P with X_i uncertain can be used to produce a rule $a : KC \rightarrow L/X_i$ in $K(P)$, meaning that if KC is known and a is applied, then if X_i was true, L will become true.

Rule 2 (Split) *For each rule $a : C \wedge X_i \rightarrow L$ in P where X_i is a literal that appears in a disjunction $X : X_1 \vee \cdots \vee X_n$, then add to $K(P)$ the atoms L/X_j, $j = 1, \ldots, n$, all initialized to **false**, and the rules $a : KC \rightarrow L/X_i$.* [2]

The combinations of the conditional beliefs represented by the atoms L/X_i is achieved by means of extra actions added to the classical encoding $K(P)$ that generalize (2) slightly, allowing some of the cases X_i to be disproved:[3]

Rule 3 (Merge) *For each disjunction $X : X_1 \vee \cdots \vee X_n$ and atom L in P such that L/X_i is an atom in $K(P)$, add to $K(P)$ a new action $a_{X,L}$ with conditional effect*

$$(L/X_1 \vee K\neg X_1) \wedge \cdots \wedge (L/X_n \vee K\neg X_n) \wedge FLAG_{X,L} \rightarrow L$$

*where $FLAG_{X,L}$ is a fluent initialized to **true.** If $L = X_i$ for some $i \in [1, n]$, remove the conjunct $(L/X_i \vee K\neg X_i)$ from the rule body.*

[1] We will present a simplified subset of the transformation rules due to lack of space. In particular, we will assume that every action only has one rule and no preconditions. The general translation appears in (Palacios & Geffner 2006).

[2] If we want L/X_i to mean exactly that '*right after the action a*, if X_i is true, then L is true', some additional care is needed about the other rules of the action. Details in the full paper.

[3] When using the classical plans obtained from $K(P)$ as conformant plans in P, such 'dummy' actions must be removed.

	cf2cs(ff)		CFF	
Problem	Time	Length	Time	Length
Bomb-100-1	0,84	199	96,2	199
Bomb-100-60	9.64	140	23,53	140
Cube-7-Ctr	0,02	24	38,2	39
Cube-9-Ctr	0,05	33	—-	—-
Cube-11-Ctr	0,09	42	—-	—-
Sqr-8-Ctr	0,03	22	140,5	50
Sqr-12-Ctr	0,04	32	—-	—-
Sqr-64-Ctr	9,66	188	—-	—-
Grid-4-4	0,06	25	0,11	25
Grid-4-5	0,05	30	0,14	30
Safe-50	0,05	50	134,4	50
Safe-70	0,08	70	561,8	70
Safe-100	0,28	100	—-	—-

Table 1: Plan times and lengths obtained by a classical planner (FF) over $K(P)$ translation (cf2cs(ff)) in relation to Conformant FF for various conformant problems P. Times in seconds. The symbol '—-' means cutoff exceeded (30 mins or 800Mb)

A key distinction from Logic is that the disjunction $X_1 \vee \cdots \vee X_n$ and the conditional beliefs 'if X_i then L' represented by the atoms L/X_i need all be **preserved** until they are combined together to yield L. This is the purpose of the boolean $FLAG_{X,L}$ that is initially set to true, but which is deleted when an action is taken in a context where it is not possible to prove that 1) L is preserved (if true), 2) the disjunction $X \vee L$ is preserved, and 3) the conditional beliefs represented by the atoms L/X_i achieved are preserved. This is accomplished by extending $K(P)$ with the rules that delete $FLAG_{X,L}$ when it is necessary.

These rules more detailed and other rules can be read in (Palacios & Geffner 2006). They yield expressivity without sacrificing efficiency, as they manage to *accommodate non-trivial forms of disjunctive inference in a classical theory without having to carry disjunctions explicitly in the belief state:* some disjunctions are represented by atoms like L/X_i, and others are maintained as *invariants* enforced by the resulting encoding.

Theorem 2 (Soundness $K(P)$**)** *Any plan that achieves the literal KL in $K(P)$ is a plan that achieves L in the conformant problem P.*

Experimental Results

We have implemented the translation program cf2cs that takes a conformant planning problem P as input and outputs a classical problem $K(P)$. Table 1 shows the plan times and lengths obtained by Conformant FF (Brafman & Hoffmann 2004) vs. cf2cs(ff) (FF planner fed with the problem generated by cf2cs). Translations only require a few seconds. Among the existing benchmarks, not included in the table, there are three domains, Sorting-Nets, (Incomplete) Blocks, and Ring, which cannot be handled by our translation scheme.

Discussion & Future Work

In vplan (Palacios *et al.* 2005) we presented a complete optimal planner that reject plans candidate that does not work for some initial state. In cf2sat (Palacios & Geffner 2005) we proposed to generate a propositional formula that encodes all the possible conformant planners, and called a SAT solver over it. In both cases we require an exponential process step of compiling into d–DNNF.

We have introduced a translation scheme that enables a wide class of conformant planning problems to be solved by an off-the-shelf classical planner. The translation accounts for a limited form of 'reasoning by cases' by means of an 'split-protect-and-merge' strategy; namely, atoms L/X_i that represent conditional beliefs 'if X_i then L' are introduced, and when certain invariants are verified, they are combined. This translation is incomplete because it is equivalent to a transformation like cf2sat, but considering only simple disjunctions of fluents instead of every initial state.

We want to explore allowing combinations of disjunctions by introducing atoms L/X_iY_j. For rules $a : C \wedge L \rightarrow M$, we can add $a : KC \wedge L/X_i \rightarrow M/X_i$, but in many cases it can lead to an exponential number of added atoms. However, we hope that some domains such as the Rings can be solved by a combination of these new rules, even when the new transformation will not be complete. We want to detect whether a problem is suitable for doing those additional transformations by using causal graphs. It will allow us, for instance, to see how many labels we need to consider for a variable.

The results presented here suggest to look for new propositional theories similar to Eq. 1. We can split on atoms that really need to be considered for solving the problem. We also can combine cf2cs, for easy problems, with cf2sat, for more complex problems, and obtain an hybrid planner that scales in a broader set of benchmarks. Moreover, as the plans obtained by cf2cs do not appear to be suboptimal, we want to identify when optimality holds and guarantee that. We also want to look for similar rules that allow transformations of other kinds of non-probabilistic uncertain planning, such as contingent planning.

Related Work

We did not compare the performance of cf2cs with many of the planners available because our goal is to map *some* conformant problems into classical planning. We compare with CFF (Brafman & Hoffmann 2004) as a way to show that our results are encouraging with respect to the state of the art. Most of them try to give a suboptimal solution to any conformant problem (Brafman & Hoffmann 2004; Cimatti, Roveri, & Bertoli 2004; Ferraris & Giunchiglia 2000). FragPlan (Kurien, Nayak, & Smith 2002) try to solve the general problem, but it can be used in more realistic environments where a partially conformant plan are needed.

Acknowledgements

We thank Blai Bonet for the PDDL parser and Joerg Hoffmann for providing FF and CFF.

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

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

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

Darwiche, A. 2002. On the tractable counting of theory models and its applications to belief revision and truth maintenance. *J. of Applied Non-Classical Logics*.

Ferraris, P., and Giunchiglia, E. 2000. Planning as satisfiability in nondeterministic domains. In *Proceedings AAAI-2000*, 748–753.

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

Kautz, H., and Selman, B. 1996. Pushing the envelope: Planning, propositional logic, and stochastic search. In *Proceedings of AAAI-96*, 1194–1201. AAAI Press / MIT Press.

Kurien, J.; Nayak, P.; and Smith, D. 2002. Fragment-based conformant planning. In *Proc. 13th Int. Conf. on Automated Planning and Scheduling (ICAPS-2002)*.

Lin, F., and Reiter, R. 1994. Forget it! In *Working Notes, AAAI Fall Symposium on Relevance*, 154–159. American Association for Artificial Intelligence.

Palacios, H., and Geffner, H. 2005. Mapping conformant planning to sat through compilation and projection. In *1st Workshop on QCSP - CP-2005*.

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

Palacios, H.; Bonet, B.; Darwiche, A.; and Geffner, H. 2005. Pruning conformant plans by counting models on compiled d-DNNF representations. In *Proc. of the 15th Int. Conf. on Planning and Scheduling (ICAPS-05)*. AAAI Press.

Turner, H. 2002. Polynomial-length planning spans the polynomial hierarchy. In *JELIA '02: Proc. of the European Conference on Logics in AI*, 111–124. Springer-Verlag.