Planning with Temporally Extended Preferences by Heuristic Search

Jorge Baier and **Jeremy Hussell** and **Fahiem Bacchus** and **Sheila McIlraith**
Department of Computer Science
University of Toronto
Toronto, Canada
[jabaier|hussell|fbacchus|sheila]@cs.toronto.edu

Abstract

In this paper we describe a planner that extends the TLPLAN system to enable planning with temporally extended preferences specified in PDDL3, a variant of PDDL that includes descriptions of temporal plan preferences. We do so by compiling preferences into nondeterministic finite state automata whose accepting conditions denote achievement of the preference described by the automaton. Automata are represented in the planning problem through additional predicates and actions. With this compilation in hand, we are able to use domain-independent heuristics to guide TLPLAN towards plans that realize the preferences. We are entering our planner in the *qualitative preferences* track of IPC5, the 2006 International Planning Competition. As such, the planner description provided in this paper is preliminary pending final adjustments in the coming weeks.

Introduction

Standard goals in planning allow us to distinguish between plans that satisfy the goal and those that do not, however, they fail to discriminate between the quality of different successful plans. Preferences, on the other hand, express information about how "good" a plan is thus allowing us to distinguish between desirable successful plans and less desirable successful plans.

PDDL3 (Gerevini & Long 2005) is an extension of previous planning languages that includes facilities for expressing preferences. It was designed in conjunction with the 2006 International Planning Competition. One of the key features of PDDL3 is that it supports temporally extended preference statements, i.e., statements that express preferences over sequences of events. In particular, in the *qualitative preferences* category of the planning competition preferences can be expressed with temporal formulae that are a subset of LTL (linear temporal logic). A plan satisfies a preference whenever the sequence of states generated by the plan's execution satisfies the LTL formula representing the preference.

PDDL3 allows each planning instance to specify a problem-specific metric used to compute the value of a plan. For any given plan, over the course of its execution various preferences will be violated or satisfied with some preference perhaps being violated multiple times. The plan value metric can depend on the preferences that are violated and the number of times that they are violated. The aim in solving the planning instance is to generate a plan that has the best metric value, and to do this the planner must be able to "monitor" the preferences to determine when and how many times different preferences are being violated. Furthermore, the planner must be able to use this information to guide its search so that it can find best-value plans.

We have crafted a preference planner that uses various techniques to find best-value plans. Our planner is based on the TLPLAN system (Bacchus & Kabanza 1998), extending TLPLAN so that fully automated heuristic-guided search for a best-value plan can be performed. We use two techniques to obtain heuristic guidance. First, we translate temporally extended preference formulae into nondeterministic finite state automata that are then encoded as a new set of predicates and action effects. When added to the existing predicates and actions, we thus obtain a new planning domain containing only standard ADL-operators. Second, once we have recovered a standard planning domain we can use a modified relaxed plan heuristic to guide search. In what follows, we describe our translation process and the heuristic search techniques we use to guide planning. We conclude with a brief discussion of related work.

Translation of LTL to Finite State Automata

TLPLAN already has the ability to evaluate LTL formulae during planning. It was originally designed to use such formulae to express search control knowledge. Thus one could simply express the temporally extended preference formulae in TLPLAN directly and have TLPLAN evaluate these formulae as it generates plans. The difficulty, however, is that this approach is by itself not able to provide any heuristic guidance. That is, there is no obvious way to use the partially evaluated LTL formulae maintained by TLPLAN to guide the planner towards satisfying these formulae (i.e., to satisfy the preferences expressed in LTL).

Instead our approach is to use the techniques presented in (Baier & McIlraith 2006) to convert the temporal formulae into nondeterministic finite state automata. Intuitively the states of the automata "monitor" progress towards satisfying the original temporal formula. In particular, as the world is updated by actions added to the plan, the state of the automata is also updated dependent on changes made to the world. If the automata enters an accepting state then the sequence of worlds traversed by the partial plan has satisfied the original temporal preference formula.

There are various issues involved in building efficient automata from an arbitrary temporal formula, and more details are provided in (Baier & McIlraith 2006). However, once the automaton is built, we can integrate it with the planning domain by creating an augmented planning domain. In the augmented domain there is a predicate specifying the current set of states that the automata could be in (it is a non-deterministic automata so there are a set of current states). Moreover, for each automata, we have a single predicate (the *accepting predicate*) that is true iff the automata has reached an accepting condition, denoting satisfaction of the preference. In addition, we define a post-action update sequence of ADL operators, which take into account the changes just made to the world and the current state of the automata in order to compute the new set of possible automata states. This post-action update is performed immediately after any action of the domain is performed. TLPLAN is then asked to generate a plan using the new augmented domain.

To deal with multiple preference statements, we apply this method to each of the preferences in turn. This generates multiple automata, and we combine all of their updates into a single ADL action (actually to simplify the translation we use a pair of ADL actions that are always executed in sequence).

A number of refinements must be made however to deal with some of the special features of PDDL3. First, in PDDL3 a preference can be scoped by a universal quantifier. Such preferences act as parameterized preference statements, representing a set of individual preference statement one for each object that is a legal binding of the universal variable. To avoid the explosion of automata that would occur if we were to generate an distinct automata for each binding, we translate such preferences into "parameterized" automata. In particular, instead of having a predicate describing the current set of states the automata could be in, we have a predicate with extra arguments which specifies what state the automata could be in for different objects. Similarly, the automata update actions generated by our translator are modified so that they can handle the update for all of the objects through universally quantified conditional effects.

Second, PDDL3 allows preference statements in action preconditions. These preferences refer to conditions that must ideally hold true immediately before performing an action. These conditions are not temporal, i.e., they refer only to the state in which the action is performed. Therefore, we do not model these preferences using automata but rather as conditional effects of the action. If the preference formula does not hold and the action is performed, then, as an effect of the action, a counter is incremented. This counter, representing the number of times the precondition preference is violated, is used to compute the metric function, described below.

Third, PDDL3 specifies its metric using an "is-violated" function. The is-violated function takes as an argument the name of a preference type, and returns the number of times preferences of this type were violated. Individual preferences are either satisfied or violated by the current plan. However, many different individual preferences can be grouped into a single type. For example, when a preference is scoped by a universal quantifier, all of the individual preference statements generated by different bindings of the quantifier yield a preference of the same type. Thus the is-violated function must be able to count the number of these preferences that are violated. Similarly, action precondition preferences can be violated multiple times, once each time the action is executed under conditions that violated the precondition preference. The automata we construct utilizes TLPLAN's ability to manipulate functions to keep track of these numbers.

Finally, PDDL3 allows specification of hard temporal constraints, which can also be viewed as being hard temporally extended goals. We also translate these constraints into automata. The accepting predicate of these automata are then treated as additional final-state goals. Moreover, we use TLPLAN's ability to incrementally check temporal constraints to prune from the search space those plans that already have violated the constraint.

Heuristic Search

The new augmented planning domain no longer has temporally extended preferences. Instead, the domain is much like a standard planning domain. Thus, we can compute relaxed plans and use those relaxed plans to compute heuristics.

In particular, we have augmented TLPLAN to allow it to compute relaxed state sequences: sequences of states that can be generated from the current state when ignoring the delete effects of actions. Notice that since the automata predicates are part of the new domain, the relaxed state sequences include predicates describing the "relaxed state" of the automata. Thus in the relaxed sequence of states not only can we compute various goal distance functions, but we can also compute various functions that depend on automata states. That is, we can compute information about the distance to satisfying various preferences. Since each preference is given a different weight in valuing a plan we can even weight the "distance to satisfying a preference" differently depending on the value of the preference.

Specifically, our heuristic function is a combination of the following functions, which are evaluated over partial plans. (We continue to work on these functions.)

Goal distance A function that is a measure of how hard it is to reach the goal. It is computed using the relaxed plan graph (similar to the one used by the FF planner (Hoffmann & Nebel 2001)). It computes a heuristic distance to the goal facts using a variant of the heuristic proposed by (Zhu & Givan 2005). The exact value of the k exponent in this heuristic is still being finalized.

Preference distance A measure of how hard it is to reach the preference goals, i.e., how hard it is to reach the accepting states of the various preference automata. Again, we use Zhu & Givan's heuristic to compute this distance.

Optimistic metric A lower bound[1] for the metric function

[1]Without loss of generality, we assume that we are minimizing the metric function.

of any plan that completes the partial plan, i.e., the best metric value that the partial plan could possibly achieve if completed to satisfy the goal. We compute this number assuming that no precondition preferences will be violated in the future, and assuming that all temporal formulae that are not currently violated by the partial plan will be true in the completed plan. To determine whether a temporal formula is not violated by the partial plan, we simply verify that its automaton is currently in a state from which there is a path to an accepting state. Finally, we assume that the goal will be satisfied at the end of the plan.

Discounted metric A weighting of the metric function evaluated in the relaxed states. Let $M(s_0)$ be the metric value of a state s_0, and $s_1, \ldots, s_n$ be the relaxed states reachable from state s until a fixed point is found. The discounted metric for s and discount factor r, $D(s, r)$, is computed as:

$$D(s,r) = M(s_0) + \sum_{i=0}^{n-1} (M(s_{i+1}) - M(s_i))r^i.$$

The factor of r we are finally going to use is not yet decided.

The final heuristic function is obtained by a combination of the functions defined above.

Our planner is able to return plans with incrementally improving metric value. It does best-first search using the heuristic described above. At all times, it keeps the metric value of the best plan found so far. Additionally, the planner prunes from the search space all those plans whose optimistic metric is worse than the best metric found so far. This is done by dynamically adding a new TLPLAN hard constraint into the planning domain.

Discussion

The technique we use to plan with temporally extended preferences presents a novel combination of techniques for planning with temporally extended goals, and for planning with preferences.

A key enabler of our planner is the translation of LTL preference formulae into automata, exploiting work described in (Baier & McIlraith 2006). There are several papers that address related issues. First is work that compiles temporally extended goals into classical planning problems such as that of Rintanen (Rintanen 2000), and Cresswell and Coddington (Cresswell & Coddington 2004). Second is work that exploits automata representations of temporally extended goals (TEGs) in order to plan with TEGs, such as Kabanza and Thiébaux's work on TLPLAN (Kabanza & Thiébaux 2005) and work by Pistore and colleagues (Lago, Pistore, & Traverso 2002). A more thorough discussion of this work can be found in (Baier & McIlraith 2006).

There is also a variety of previous work on planning with preferences. In (Bienvenu, Fritz, & McIlraith 2006) the authors develop a planner for planning with temporally extended preferences. Their planner performs best first-search based on the optimistic and pessimistic evaluation of partial plans relative to preference formulae. Preference formulae are evaluated relative to partial plans and the formulae progressed, in the spirit of TLPLAN, to determine aspects of the formulae that remain to be satisfied. Also noteworthy is the work of Son and Pontelli (Son & Pontelli 2004) who have constructed a planner for planning with temporally extended goals using answer-set programming (ASP). Their work holds promise however ASP's inability to deal efficiently with numbers has hampered their progress. Brafman and Chernyavsky (Brafman & Chernyavsky 2005) recently addressed the problem of planning with preferences by specifying qualitative preferences over possible goal states using TCP-nets. Their approach to planning is to compile the problem into an equivalent CSP problem, imposing variable instantiation constraints on the CSP solver, according to the TCP-net. This is a promising method for planning, though at the time of publication of their paper, their planner did not deal with temporal preferences.

References

Bacchus, F., and Kabanza, F. 1998. Planning for temporally extended goals. *Ann. of Math Art. Int.* 22(1-2):5–27.

Baier, J. A., and McIlraith, S. 2006. Planning with first-order temporally extended goals. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*. To appear.

Bienvenu, M.; Fritz, C.; and McIlraith, S. 2006. Planning with qualitative temporal preferences. In *Proceedings of the Tenth International Conference on Knowledge Representation and Reasoning (to appear)*.

Brafman, R., and Chernyavsky, Y. 2005. Planning with goal preferences and constraints. In *Proceedings of The International Conference on Automated Plann ing and Scheduling*.

Cresswell, S., and Coddington, A. 2004. Compilation of LTL goal formulas into PDDL. In *ECAI-04*, 985–986.

Gerevini, A., and Long, D. 2005. Plan constraints and preferences for pddl3. Technical Report 2005-08-07, Department of Electronics for Automation, University of Brescia, Brescia, Italy.

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

Kabanza, F., and Thiébaux, S. 2005. Search control in planning for temporally extended goals. In *Proc. ICAPS-05*.

Lago, U. D.; Pistore, M.; and Traverso, P. 2002. Planning with a language for extended goals. In *Proc. AAAI/IAAI*, 447–454.

Rintanen, J. 2000. Incorporation of temporal logic control into plan operators. In *Proc. ECAI-00*, 526–530.

Son, T., and Pontelli, E. 2004. Planning with preferences using logic programming. In Lifschitz, V., and Niemela, I., eds., *Proceedings of the 7th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-2004)*, number 2923 in Lecture Notes in Computer Science. Springer. 247–260.

Zhu, L., and Givan, R. 2005. Simultaneous heuristic search for conjunctive subgoals. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-2005)*, 1235–1241.