Planning in PDDL2.2 Domains with LPG-*TD*

Alfonso Gerevini Alessandro Saetti Ivan Serina Paolo Toninelli
Dipartimento di Elettronica per l'Automazione
Università degli Studi di Brescia
Via Branze 38, 25123 Brescia, Italy
{gerevini,saetti,serina}@ing.unibs.it

Introduction

LPG-*TD* is an extension of the LPG planner (Gerevini, Saetti, & Serina 2003; 2004) that can handle most of the features of PDDL2.2 (Edelkamp & Hoffmann 2003), the standard planning language of the 4th International Planning Competition (IPC-4).[1] In particular, LPG-*TD* is an incremental fully-automated planner generating plans for problems in domains involving:

- STRIPS actions;
- durative actions;
- actions and goals involving numerical expressions;
- operators with universally quantified effects;
- operators with existentially quantified preconditions;
- operators with disjunctive preconditions;
- operators with implicative preconditions;
- timed initial literals (deterministic unconditional exogenous events);
- predicates derived by domain axioms;
- maximization or minimization of complex plan metrics.

Like the previous version of LPG, the new version is based on a stochastic local search in the space of particular "action graphs" derived from the planning problem specification. In LPG-*TD*, this graph representation has been extended to deal with the new features of PDDL2.2, as well to improve the management of durative actions and of numerical expressions (already supported by PDDL2.1 (Fox & Long 2003)).

In the following, we briefly describe the main novelties of LPG-*TD*, which include some new techniques for planning problems involving *timed initial literals* and *derived predicates*, and some general improvements of all phases of the planner (pre-processing, search and post-processing).

[1]The "*TD*" extension in the name of the planner is an abbreviation of "Timed initial literals and Derived predicates", the two main new features of PDDL2.2.

Handling Timed Initial Literals

Timed initial literals represent facts (predicates instantiated with constants) that become true or false at certain time points, independently of the actions in the plan. They correspond to particular exogenous events known by the planner (Edelkamp & Hoffmann 2003). A fact can become true or false several times through different timed initial literals, defining a set of disjoint *temporal windows* where the fact holds. For example, the first problem of the `Satellite` domain in IPC-4 has two timed initial literals

```
(at 139.00 (visible antenna0 satellite0)),
(at 219.04 (not (visible antenna0 satellite0)))
```

defining a single temporal window for the fact

```
(visible antenna0 satellite0).
```

According to PDDL2.2, the fact involved by a timed initial literal can appear in the preconditions of an action, while it can never appear in its effects. We call such preconditions *timed preconditions*, and we represent them as particular nodes of the action graph. If a plan action a has a timed precondition p of type "overall" involving a fact f, p is satisfied when the interval identified by the start time and the end time of a is contained into at least one temporal window associated with f. Similar conditions can be defined for the other possible types of preconditions in a durative action.

Essentially, an unsatisfied timed precondition involving a fact f in a is treated by either (i) removing a from the plan under construction, or making some changes to the plan that make the execution of a compatible with a temporal window associated with f, i.e., by (ii) appropriately postponing the start time of a, or (iii) removing one or more actions that permit to decrease the start time of a.

In the new version of LPG, the graph-based plan representation, the pre-processing phase (reachability analysis and computation of the "mutex relations"), and the search techniques have been extended to perform such plan modifications when dealing with unsatisfied timed preconditions.

Handling Derived Predicates

Derived predicates are predicates that can not be achieved directly by the domain actions. A derived predicate $P(\overline{x})$ is true at a certain time t during the execution of a plan iff it can be derived from the facts that are true at time t through a set of rules specified in the domain formalization. Each of these rules is of the form

$$\textit{if } \phi(\overline{x}) \textit{ then } P(\overline{x}),$$

where $\overline{x}$ is a tuple of variables, and $\phi(\overline{x})$ a logical formula (a precise syntactic and semantic definition of domain rule is given in (Edelkamp & Hoffmann 2003)).

A typical example of derived predicate in the `Blocksworld` domain is *above*, which can be derived by using the following rule:

$$\textit{if } \big(on(x,y) \vee \exists z\, above(x,z) \wedge above(z,y)\big)$$
$$\textit{then } (above(x,y).$$

In PDDL2.2, a derived predicate can be a precondition of an action or a goal of the planning problem, which we call *derived precondition* (we treat problem goals as preconditions of a special final action). A derived precondition of an action a is satisfied if it is implied by the domain rules and the facts that are true when a is executed.

Essentially, an unsatisfied derived precondition d in a is treated by either (i) removing a from the current plan, or (ii) adding one or more actions that modify the set of the facts that are true when the action can be executed in the plan, so that d becomes true by applying of one or more domain rules. For example, consider a simple `Blocksworld` problem where the initial state is

`(on-table a), (on-table b), (on c b)`

and the goal is `(above a b)`. When the domain rule of the previous example is available, it is easy to see that the goal can be achieved by just adding to the (initially empty) plan the action `stack(a,c)` making `(on a c)` true.

In the new version of LPG, the graph-based plan representation, the pre-processing phase (reachability analysis and computation of the mutex relations), and the search techniques have been extended to take possible domain rules into account.

Further Extensions

In addition to the treatment of timed initial literals and derived predicated, the new version of our planner includes several revisions and extensions with respect to the version that took part in the previous competition. Such changes concern the pre-processing phase, the search phase, and post-processing phase of the planner. In the following, we give a list of them.

Pre-processing

- The algorithm for computing mutex relations has been revised to make it faster than the original algorithm described in (Gerevini, Saetti, & Serina 2003).
- Some actions are automatically identified as "useless actions", and they can be pruned away at parsing time or they can be neglected during search.
- The computation of the reachability information for numerical domains has been improved to derive more accurate information that are exploited by the heuristic function evaluating the search neighborhood.

Search

- We have developed new heuristics for evaluating the search neighborhood specialized for the different variants of a planning domain supported by PDDL2.2.
- The basic local search strategy (Walkplan) has been extended with a "tabu list" helping to escape from local minima.

Post-processing

- We have developed a technique for increasing the degree of parallelism in the plans generated by LPG for domains with durative actions and numerical expressions. This is done by an algorithm that, starting from the set of the actions forming the plan and their ordering constraints identified by the planner, tries to reduce the plan makespan.

Finally, at the time of writing, the development of LPG-*TD* is still in progress. In particular, we are experimenting a pre-processing technique for the automatic ordering of the problem goals, and we are developing an extension of the representation for handling actions involving conditional effects.

Acknowledgments

We would like to thank all previous members of the LPG team and particularly Alberto Bettini, Marco Lazzaroni, Sergio Spinoni.

References

Edelkamp, S, Hoffmann, J. 2003. PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition Technical Report N. 194, Albert Ludwigs Universität Institüt für Informatik, Freiburg, Germany.

Fox, M., and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. JAIR 20:61–124.

Gerevini, A., Saetti, A., and Serina, I. Planning through Stochastic Local Search and Temporal Action Graphs in LPG. 2003. JAIR 20:239–290.

Gerevini, A., Saetti, A., and Serina, I. 2004. An Empirical Analysis of Some Heuristic Features for Local Search in LPG . In *Proceedings of ICAPS-04*.