Adapting LPGP to Plan with Deadlines

Stephen Cresswell and **Alexandra Coddington** [1]

Abstract.
This paper describes two approaches that enable the AI Planner LPGP to reason about domains with exogenous events and goals with duration: the first investigates how such domains may be encoded using the planning domain definition language PDDL2.1 level 3, while the second involves directly modifying LPGP. Both approaches have been tested in a number of domains and conclusions are drawn about the relative merits of the two approaches.

1 INTRODUCTION

The focus of this work is to examine ways of reasoning about goals (states brought about by a planning agent executing a plan) or exogeneous events (events that occur independently of the planning agent's activities) which may either have fixed deadlines, may occur prior to or after some deadline, or may be made true within some interval of time. As a basis for this work we make use of syntax provided by PDDL2.2 (an extension of PDDL2.1 [3]) which enables timed literals to be specified, as these allow exogenous events that are both deterministic and unconditional to be expressed (such as the opening and closing times of shops). We aim to show how mechanisms that are required to solve problems that include timed literals may also be used to reason about goals with deadlines. In order to demonstrate our approach we use LPGP [4], a temporal planner inspired by Graphplan [1] which is designed to reason with PDDL2.1 level 3 durative actions. In contrast with earlier Graphplan-inspired temporal planners [5] in which the structure of the planning graph is exploited to represent the flow of time and durations are attached to actions using strong constraints to prevent any illegal interactions, LPGP inverts the way in which time is attached to states and actions—duration is attached to fact layers while the *start* and *end* points of durative actions are instantaneous. The graph layers, instead of modelling the flow of time, represent only the logical temporal structure of the plan by capturing the points at which events occur in the plan. LPGP reasons about durative actions by solving constraints derived from the durations of actions as a linear programming problem—durative actions introduce the constraint that the total duration of the fact layers in the graph between the *start* and *end* points of actions must equal the duration of the action. If the linear programming problem is unsolvable, the plan is invalid and search must backtrack.

In the following sections we briefly describe two approaches to encoding domains and problems including timed literals and goals with deadlines using pure PDDL2.1. We then describe how modifications made to LPGP allow it to reason with timed literals and goals with deadlines more directly. Finally we describe a number of example problem domains and present some results (for more details about this work see [2]).

[1] University of Strathclyde, Glasgow, United Kingdom. email:{Stephen.Cresswell, Alex.Coddington}@cis.strath.ac.uk

2 ENCODING DEADLINES AND EVENTS

It is possible to encode domains and problems involving deadlines and exogenous events in pure PDDL2.1, as any deadline or exogenous event can be encoded as a condition or effect of a dummy durative action. The domain must be contrived firstly so that such dummy durative actions are forced to be included in the plan and secondly so that they can only occur at fixed timepoints corresponding to their deadline. We decided to experiment with this approach as it has the advantage that no modifications need to be made to LPGP which has been implemented to plan with PDDL2.1 level 3 specified domains and problems. Two different approaches were used to encode the domains as illustrated in Figure 1 where vertical bars are used to indicate the instantaneous *start* and *end* points of durative actions.

Figure 1 (i), known as the *wrapper* encoding, is based on the approach used by the IPC4 committee[2] in order to create pure PDDL2.1 encodings of PDDL2.2 specified problems. The PDDL2.1 encoding

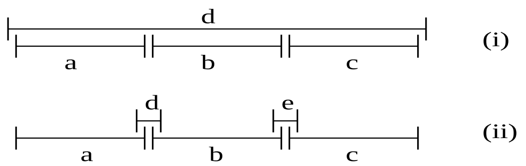

Figure 1. Encodings to force a plan to include the fi xed sequence a, b, c.

must ensure that the durative actions a, b and c occur in the required order, and that the enclosing interval d, represented as a durative *wrapper* action, is used to prevent gaps occurring between them. a, b and c are dummy durative actions that may either be used to represent a sequence of exogenous events or goals with instantaneous deadlines, or are durative actions that must be forced to occur before some single deadline (encoded as the dummy durative action d).

In Figure 1 (ii), known as the *Halsey clip* encoding, the interval d clips together a and b while the interval e clips together b and c. The duration of the clip interval imposes a maximum gap between the two actions. This representation has the advantage that it can be used even when action durations are not fixed in advance. It is suitable primarily for encoding sequences of exogenous events or goals with instantaneous deadlines. However, if only a single goal is to be satisfied by some deadline, clip intervals such as d and e will not be necessary. Details of the algorithms used for both encodings as well as an example domain can be found in [2].

[2] IPC4—the 4th International Planning Competition committee, see `http://ipc.icaps-conference.org`

3 MODIFYING LPGP

Although it is possible to model exogenous events and goals with deadlines purely by using specially contrived PDDL2.1 domain models, this approach has many disadvantages. In particular, in order to anchor an action in place, it must be fixed within a sequence of intervals, and that sequence must be arranged so that it fits within an enclosing interval. This presents two problems: (a) it causes an unnecessary proliferation of dummy durative actions whose only purpose is to keep other actions in position; (b) the planner is free to consider inserting gaps within these action sequences and although this cannot lead to a successful plan, a lot of search may be required before the planner discovers that such plans are invalid. For these reasons we decided to modify LPGP to enable it to handle exogenous events and goals with deadlines directly.

In order to represent deadlines, fact layers associated with deadlines are constrained in time relative to the initial state by directly imposing constraints in the linear programming model. This is achieved by enabling LPGP to recognise deadlines as a special type of action. The duration of such *deadline* actions is interpreted by LPGP as a minimum time interval between the initial state and current state. When including a *deadline* action, constraints posted to the linear programming solver require the sum of the durations of all fact layers between the initial and current state to be equal to the time period associated with the deadline. Figure 2 shows a partially completed plan graph in which the *deadline* action *F-dead* occurs. This action is given distinguished treatment—when *F-dead* is included in the plan (e.g. at action layer $k + 3$), a constraint is posted to the linear programming solver to ensure that the sum of the durations of *all* fact layers (e.g. $d_1 + ... + d_k + ... + d_{k+3}$) between the initial state and the action layer $k + 3$ equals the deadline associated with *F-dead*. This constraint fixes the time of action layer $k + 3$ precisely.

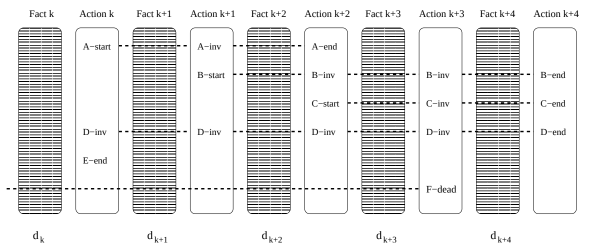

Figure 2. A partially completed graph search where search has reached back to layer k, with the addition of a deadline action (F-dead).

If we have a simple deadline or exogenous event that requires a goal or proposition to be achieved or to occur at a certain time, it is appropriate to specify that the overall duration of the fact layers occurring between the initial state and the state containing the goal/event is equal to ($=$) the specified duration (the period of time that passes between the initial state and the time the goal or exogenous event must take place). However, in some domains we may wish to express constraints containing both $=$ and $\leq$ (and possibly $\geq$). This requires a simple modification as such constraints can easily be dealt with by the linear programming solver.

4 RESULTS AND CONCLUSIONS

In order to determine whether LPGP can reason about exogenous events and goals with deadlines, a set of experiments was performed using several test domains as follows.

DriverLogShift domain is a version of the DriverLog domain from the IPC3 competition, with the modification that the drivers are only on duty for certain periods of time and off duty otherwise. The problem used here was very simple, with 3 timed exogenous events and a very simple goal.

DriverLogDeadlines has a single deadline restricting the makespan of the whole plan.

DriverLogMultiDeadlines has instantaneous deadlines at fixed times during the plan. The problem involved moving a single package between various locations by different deadlines.

ZenoTravel domain is the version extended by the IPC4 committee in order to illustrate the use of timed initial literals to encode the opening and closing of fuel stations. We retained the sequence of 6 events in the original problem, but simplified the goal condition.

Results are presented in Table 1. Although encodings of deadlines and exogenous events are possible in pure PDDL2.1, LPGP was unable to produce results using either the *wrapper* or the *Halsey clip* encodings while VHPOP was able to handle both encodings.

Table 1. Timings in seconds for various test problems. All timings taken on a 400MHz Pentium III with 256Mb RAM.

Planner Representation	LPGP Customised	VHPOP Clips	VHPOP Wrapper
ZenoTravel, timed literals, simple	29.59	13.14	-
DriverLogShift	17.61	7.13	3.68
DriverLogDeadlines	352.30	n/a	147.98
DriverLogMultiDeadlines	0.63	1.71	10.29

For LPGP, the customised modification is very much more efficient than the generic encoding as it only requires one instantaneous action per timed literal or deadline. Using the *wrapper* encoding, LPGP translates each durative action into *start*, *invariant*, and *end* actions, so for n events, we end up adding $3(n + 1)$ instantaneous actions, some of which require exact synchronisation. For the *Halsey clip* encoding this is even worse requiring $3(2n - 1)$ actions. Table 1 shows that despite the modifications made to LPGP, it is still outperformed by VHPOP. We believe that this is probably due to underlying inefficiencies in the Graphplan style search and will be conducting experiments to determine whether or not this theory holds. Because both planners could only cope with small problems it is difficult to draw any firm conclusions concerning their respective merits.

In this paper we have demonstrated two approaches that enable LPGP to reason about exogenous events and goals with deadlines and shown how minor modifications made to LPGP have significantly improved its performance. It is our intention to investigate whether using local search strategies might improve the performance of LPGP. In addition we are also investigating the use of LTL to directly express control rules which may guide the search process for LPGP. In general, our conclusions are that planning with durative actions, exogenous events and goals with deadlines is difficult and an area ripe for extensive further research.

REFERENCES

[1] A. Blum and M. Furst, 'Fast planning through plan-graph analysis', in *Proc. of the 14th Int. Joint Conf. on AI (IJCAI95)*, (1995).

[2] S. N. Cresswell and A. M. Coddington, 'Planning with timed literals and deadlines', in *Proceedings of the 21st Workshop of the UK Planning and Scheduling SIG*, pp. 22–35, (2003). ISSN 1368-5708.

[3] M. Fox and D. Long, 'PDDL+', Technical report, Department of Computer Science, University of Durham, Durham, DH1 3LE, UK, (2001).

[4] D. Long and Fox. M, 'Exploiting a graphplan framework in temporal planning', in *Proceedings of the 13th Int. Conference on Automated Planning and Scheduling ICAPS 2003*, pp. 52–61. AAAI Press, (2003).

[5] D. E. Smith and D. S. Weld, 'Temporal planning with mutual exclusion reasoning', in *Proceedings of IJCAI-99, Stockholm*, pp. 325–337, (1999).