ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1712

Temporal Planning with Constants in Context

Josef Bajada[1] and **Maria Fox** and **Derek Long**

1 INTRODUCTION

Required concurrency [3] can cause actions to interfere with running continuous effects. This interference can modify the rate of change, including the polarity, of a continuous effect. In this work, we propose a mechanism to support discrete interference of rates of change caused by instantaneous actions, the start and end endpoints of other durative actions, and numeric timed initial fluents [9]. Current temporal planners have very limited support for such numeric dynamics. COLIN [2] reduces a temporal numeric planning problem to a linear program (LP), but operates on an implicit assumption that the rate of change of a durative action's continuous effect is constant throughout its execution. In this work we propose some enhancements to the algorithms used in COLIN [2], in order to support discrete interference of continuous effects, and a new planner, DICE, was developed to implement them.

2 CONSTANTS IN CONTEXT

A *context* refers to the interval between two adjacent discrete *happenings* in a temporal plan, where a happening refers to the time-stamp of a discrete state transition [5]. We only assume that the rate of change is constant throughout a temporal context. This enables a single durative action to have piecewise linear continuous effects.

Definition 2.1. A temporal *context*, $C = [t_i, t_{i+1}]$, in a plan's happening sequence, T, is an interval enclosed by two adjacent discrete happenings, t_i and t_{i+1}, where $0 \leq t_i < t_{i+1}$, and there is no intermediate *happening*, t, that is $\{t | t_i < t < t_{i+1}\} \cap T = \emptyset$.

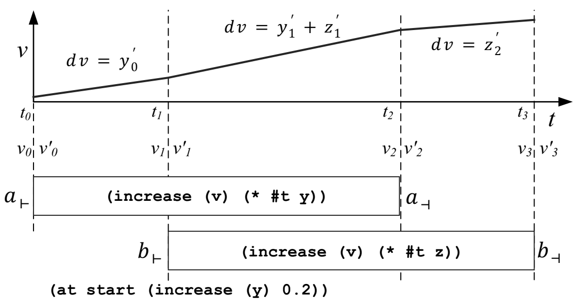

Figure 1. Linear Continuous Effects with Constants in Context.

An LP is used to verify the feasibility and validity of a temporal plan. Its variables consist of the happenings that correspond to the discrete steps of the plan, together with the values of the non-time-dependent [1] numeric fluents before and after each happening. The constraints of the LP consist of temporal constraints between the steps of the plan, mainly the ordering constraints and duration conditions of durative actions, together with numeric pre-conditions and effects of each action in the plan. The rate of change of each continuous effect on a variable, v, is computed dynamically within each temporal context, from the list of durative actions running concurrently, as illustrated in Figure 1.

The rate of change of v in temporal context i is computed from the continuous effects running throughout context i, denoted ceffs_i, on v. This is shown in Equation 1, where ceffs_i is represented as a multiset, since the same continuous effect could take place $n > 0$ times concurrently. Each continuous effect, $expr$, on v, is evaluated in the context of the discrete state, s_i, that initiated context i.

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = \sum_{\langle v, expr \rangle \to n \in \mathit{ceffs}_i} n \cdot expr(s_i) \tag{1}$$

Since continuous effects depend on action durations, whether a sequence of actions achieves a numeric goal or not could depend on the chosen schedule for those actions. We refer to such numeric goals as *schedule dependent*. When during forward search the planner evaluates a state against such goals, G, a dummy action, a_G, with precondition $pre(a_G) = G$, and effects $\mathit{eff}(a_G) = \langle \emptyset, \emptyset, \emptyset \rangle$, is appended to the current plan. If the LP for the resultant plan is solvable the schedule dependent goals are achievable with the plan.

3 A NUMERIC ENHANCED TRPG

The Temporal Relaxed Planning Graph (TRPG) [2] adapts the Metric-FF delete relaxation heuristic [6], and associates a time-stamp to each action layer, which represents the earliest time at which the actions in that layer can be applied [2]. However, the TRPG does not take into account actions whose effects indirectly enable numeric goals to be achieved. A *numeric enhanced* TRPG (TRPGne) is proposed, which takes into account richer numeric causality. It propagates effects on variables that are used in effect expressions of other variables, and identifies *implicit intermediate goals* by inferring new numeric preconditions on actions that achieve numeric goals. These preconditions are then used during relaxed plan extraction.

4 EHC WITH ASCENT BACKTRACKING

Enforced Hill-Climbing (EHC), a popular heuristic search algorithm used in classical [8], numeric [6], and temporal planners [2], suffers from an inherent weakness that hill-climbing decisions could lead to a dead end [7]. This issue becomes more evident in planning problems with temporal and numeric bounded constraints, where the heuristic can be too optimistic and leads to a constraint violation.

[1] King's College London, United Kingdom. email: josef.bajada@kcl.ac.uk

Table 1. Experimental results for three domains, performed on an Intel® Core™ i7-3770 CPU @ 3.40Ghz, allocated a maximum of 3GB RAM.

	Plantery Rover				Intelligent Pump Control				Demand-Side Electricity Management			
	DICE		UPMurphi (1.0)		DICE		UPMurphi (20.0)		DICE		UPMurphi (10.0)	
#	Time (s)	States	Time (s)	States	Time (s)	States	Time (s)	States	Time (s)	States	Time (s)	States
1	8.208	622	22.84	805,145	0.348	40	2.04	51,137	0.899	76	78.14	3,740,370
2	8.243	429	153.04	5,264,013	0.605	55	135.68	3,244,517	1.099	97	1,437.38	61,313,989
3	15.695	442	•	•	0.683	70	•	•	1.782	125	•	•
4	17.589	417	•	•	1.023	86	•	•	1.548	138	•	•
5	43.653	431	•	•	1.404	103	•	•	1.503	204	•	•
6	45.455	474	•	•	1.197	81	•	•	3.582	247	•	•
7	72.977	518	•	•	1.744	86	•	•	6.932	310	•	•
8	152.903	666	•	•	2.522	97	•	•	11.623	292	•	•
9	117.159	647	•	•	3.7	138	•	•	14.805	321	•	•
10	160.035	755	•	•	5.897	132	•	•	12.971	298	•	•

When EHC fails, most planners resort to an exhaustive search such as Weighted A*. However, this means that a wrong hill-climb late in the search could cause EHC to fail even if it is close to a solution.

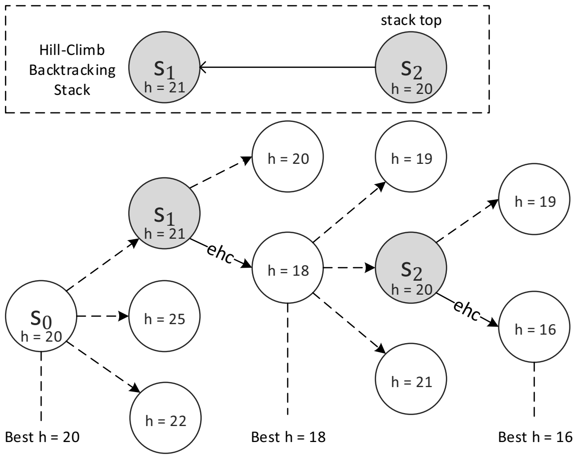

Figure 2. EHC with Ascent Backtracking.

We propose enhancing EHC with an *ascent backtracking* mechanism. A Hill-Climb Backtracking Stack of states from which an enforced hill-climb was performed is stored in memory, as illustrated in Figure 2. This introduces negligible memory overheads. When EHC fails, the state at the top of the stack is popped, to reverse the latest hill-climb. At this point all the *helpful actions* are reconsidered, without performing any enforced hill-climbing if one of them has a better heuristic value than the one encountered so far. If this also fails, the next state is popped off the stack and the process is repeated, until a solution is found or the stack is empty.

5 EVALUATION

The algorithms described above were implemented in a new planner, referred to as DICE (Discrete Interference of Continuous Effects). It was developed in Scala, which runs on the Java™ Virtual Machine. The performance of DICE was compared to that of the PDDL-based hybrid planner, UPMurphi 3.1 [4], which uses a time discretization approach to support complex non-linear functions. Tests were performed on 3 domains that feature *constants in context*. The problem instances for each domain increase in complexity incrementally. The time-step used in UPMurphi is indicated in parentheses in Table 1, which shows the results from both planners. Missing timings indicate that the planner ran out of memory.

6 CONCLUSION

In this work we have proposed a mechanism through which planning with rich numeric characteristics under required concurrency can be performed. The algorithms used by COLIN [2] were enhanced in order to support durative actions whose continuous effects change their gradient at specific time points due to interference from other discrete actions. These algorithms were implemented in a new planner called DICE. We also proposed a new heuristic, the TRPGne, which performs a better analysis of numeric causality. EHC was also complemented with ascent backtracking, which recovers from dead-ends by reversing the latest hill-climb decisions, and adds negligible memory overheads. DICE was evaluated on domains featuring these rich numeric characteristics and compared with UPMurphi [4], the only known PDDL-based planner that supports these characteristics.

ACKNOWLEDGEMENTS

This research was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) as part of the project entitled *The Autonomic Power System* (Grant Ref: EP/I031650/1).

REFERENCES

[1] J. Bajada, M. Fox, and D. Long, 'Temporal Planning with Semantic Attachment of Non-Linear Monotonic Continuous Behaviours', in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI-15)*, pp. 1523–1529, (2015).

[2] A. Coles, A. Coles, M. Fox, and D. Long, 'COLIN: Planning with Continuous Linear Numeric Change', *Journal of Artificial Intelligence Research*, **44**, 1–96, (2012).

[3] W. Cushing, S. Kambhampati, Mausam, and D. S. Weld, 'When is Temporal Planning Really Temporal?', in *20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, (2007).

[4] G. Della Penna, D. Magazzeni, F. Mercorio, and B. Intrigila, 'UPMurphi: A Tool for Universal Planning on PDDL+ Problems.', in *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS 2009)*, pp. 106–113. AAAI, (2009).

[5] M. Fox and D. Long, 'PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains', *Journal of Artificial Intelligence Research*, **20**, 61–124, (2003).

[6] J. Hoffmann, 'The Metric-FF Planning System: Translating "Ignoring Delete Lists" to Numeric State Variables', *Journal of Artificial Intelligence Research*, **20**, 291–341, (2003).

[7] J. Hoffmann, 'Where "Ignoring Delete Lists" Works: Local Search Topology in Planning Benchmarks', *Journal of Artificial Intelligence Research*, 685–758, (2005).

[8] J. Hoffmann and B. Nebel, 'The FF Planning System: Fast Plan Generation Through Heuristic Search', *Journal of Artificial Intelligence Research*, **14**, 253–302, (2001).

[9] C. Piacentini, V. Alimisis, M. Fox, and D. Long, 'An Extension of Metric Temporal Planning with Application to AC Voltage Control', *Artificial Intelligence*, **229**, 210–245, (2015).