Proceedings of the Ninth International
Symposium on Combinatorial Search (SoCS 2016)

Numeric Planning via Search Space Abstraction (Extended Abstract)

León Illanes and **Sheila A. McIlraith**
Department of Computer Science
University of Toronto, Toronto, Canada
{lillanes, sheila}@cs.toronto.edu

Introduction

Many practical planning problems require reasoning with numeric values. These so-called *numeric planning problems* can induce an infinite search space and are challenging for traditional planners. Much of the work to date in numeric planning has adapted successful techniques from classical planning, often re-interpreting known heuristics in this context (e.g., (Hoffmann 2003; Coles et al. 2008)). Recent research suggests reformulation and abstraction techniques as other interesting approaches to consider (Chrpa, Scala, and Vallati 2015; Aldinger, Mattmüller, and Göbelbecker 2015).

In this work we describe an interval abstraction approach in which we produce a classical planning problem that roughly represents the numeric problem with some loss in precision. This loss implies that a single action can have multiple possible outcomes, which reveals some similarities between planning with abstractions and a form of non-deterministic planning. We leverage this insight, taking advantage of existing techniques used to deal with non-determinism (Muise, McIlraith, and Beck 2012) in order to build plans for the numeric problem out of plans for the abstract classical problem.

Numeric Planning

We define a **numeric planning problem** by introducing a set, N, of numeric fluents – functional fluents that range over the set of real numbers, $\mathbb{R}$, and extending the definition for classical planning into a tuple $P = \langle F, N, O, I, G \rangle$. Here, F is a set of propositional fluents, I defines the initial state, G defines the goal condition, and O is a set of operators that have numeric and propositional preconditions and effects. For the purposes of this paper we consider a restricted class of numeric planning problems where numeric conditions are restricted to inequalities of the form $n \geq c$ and numeric effects are assignments of the form $n \leftarrow n+c$, where $n \in N$ and $c \in \mathbb{R}$, but our approach generalizes in a straightforward manner.

Abstraction for Numeric Planning

The basic mechanism behind our abstraction is to define propositional fluents that represent specific intervals of the real numbers, aggregating together states in which numeric fluents fall within the same intervals. We select intervals to preserve distinctions in values that are relevant to plan generation such as the identification of preconditions and goal conditions, while removing unnecessary precision.

As an example, consider an automated vehicle that can move between two locations and uses 10 units of fuel to do so. To know whether or not the `move` action can be performed, the system only needs to know whether it currently has more or less than 10 units of fuel. As such, we can define two relevant intervals for fuel values – one contains all values below 10, and the other contains 10 and all values above it. In this way, when two or more otherwise identical states represent different fuel levels but fall within the same interval we group them together into a single abstract state.

Such an abstraction results in some loss of precision. If we aggregate a state in which the vehicle has exactly 10 units of fuel with another in which it has 100 and we perform the `move` action, it is uncertain if we will reach a state in which we have sufficient fuel to perform the action again. However, we can model this uncertainty as a form of non-determinism in the outcome of actions, and the task of plan generation in this abstracted space as an instance of fully observable non-deterministic (FOND) planning, exploiting algorithmic advances in FOND planning as a component of our efficient numeric planning algorithm, ASTER (AbSTract, Execute, Repair).

Our approach suggests a simple mechanism for generating an initial set of intervals for each numeric fluent by considering all numeric comparisons in preconditions or goal conditions. We use the constants mentioned in the comparisons as the endpoints for the intervals. Operator effects with multiple possible outcomes are modelled as non-deterministic effects.

Search Algorithm

Our algorithm is inspired by the FOND planner PRP (Muise, McIlraith, and Beck 2012). PRP works by finding a weak plan, extending it into a partial policy through regression or pre-image computation, and iteratively repairing this policy to handle outstanding non-deterministic outcomes.

Since our domain is not truly non-deterministic, and all non-deterministic effects are resolved based on the underlying numeric state, we avoid attempting to repair all outstanding outcomes. Instead we try to execute the partial policy using the concrete (i.e., original numeric) domain description, repairing only when we reach a state that cannot be handled by the current policy. The repair process is done through breadth-first search in the concrete space, searching from the reached state to any state already handled by the partial policy.

Experimental Evaluation

We built our implementation on top of PRP, extending it as described above. Our planner directly reads a numeric planning problem, automatically abstracts it, finds a partial policy and attempts to apply it, repairing when necessary.

We compare our approach, ASTER, to Metric-FF (Hoffmann 2003) and LPRPG (Coles et al. 2008). We use the Settlers domain from the 3rd International Planning Competition (IPC) (Long and Fox 2003), an interesting domain that exhibits complex interaction between numeric fluents.

In our experiments we consider the same problem instances used for the IPC. However, since our approach does not handle optimization, we allow the planners to ignore the numeric optimization requirements. We ran all experiments on a Linux machine with a 2.2GHz Intel Xeon E5 CPU, limiting the running time to a maximum of 30 minutes. Resulting times and plan lengths are summarized in Table 1.

For these problems, our approach is more effective than either Metric-FF or LPRPG. Indeed, it solves 3 more problems than the other two algorithms combined, and finds plans, often one or more orders of magnitude faster than the other planners, on 6 of the remaining problems.

At the same time, there is a small but somewhat consistent degradation in the quality of the plans found. Nonetheless, whenever our algorithm terminates, it does so in under 5 minutes of time. This suggests that our algorithm could be used as a first attempt to find a solution in a very short amount of time before attempting to find a better plan with some other method. The suboptimal solution discovered by APR2 can be used as a fallback whenever the optimal algorithm doesn't terminate in time.

Extensions and Future Work

We are interested in extending the approach to use abstractions in which propositional fluents represent more expressive numeric conditions than those described in this work. Similarly, we want to consider numeric effects that involve more elaborate numeric transformations. Understanding the possible outcomes of these transformations over the numeric variables requires some form of reasoning about the theory over which the conditions and effects are defined. Techniques for dealing with transformations over abstracted numeric variables have been extensively studied within the field of Abstract Interpretation for Static Analysis of Software (Cousot and Cousot 1977). We believe that it is possible to adapt some of the techniques and more expressive abstractions used in that field to suit our context.

Acknowledgements: The authors gratefully acknowledge funding from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Pblm	Time (s)			Plan length		
	M-FF	LPRPG	ASTER	M-FF	LPRPG	ASTER
ipc-1	3.98	**0.29**	17.59	**52**	59	64
ipc-2	**0.01**	0.10	15.75	**25**	26	26
ipc-3	24.15	38.25	**17.14**	**101**	113	105
ipc-4	93.41	**0.41**	20.42	76	**70**	80
ipc-5	0.28	**0.23**	21.86	76	76	**72**
ipc-6	**10.45**	43.90	20.94	**77**	117	89
ipc-7	T/O	T/O	**33.69**	—	—	**184**
ipc-9	T/O	T/O	T/O	—	—	—
ipc-10	T/O	607.52	**63.74**	—	245	**167**
ipc-11	1798.61	892.11	**42.66**	**195**	262	200
ipc-12	539.36	T/O	**38.63**	**147**	—	162
ipc-13	1105.52	T/O	**70.98**	**230**	—	241
ipc-14	T/O	T/O	T/O	—	—	—
ipc-15	1114.18	T/O	**84.39**	**235**	—	240
ipc-16	T/O	T/O	**129.95**	—	—	**340**
ipc-17	T/O	T/O	T/O	—	—	—
ipc-18	T/O	T/O	T/O	—	—	—
ipc-19	T/O	T/O	T/O	—	—	—
ipc-20	T/O	T/O	**280.10**	—	—	**389**
solved	10	8	**14**			

Table 1: Results on the Settlers domain, where ASTER finds more solutions than Metric-FF and LPRPG combined. ASTER often finds a solution more than an order of magnitude faster than the other planners. T/O is time out.

References

Aldinger, J.; Mattmüller, R.; and Göbelbecker, M. 2015. Complexity of interval relaxed numeric planning. In *KI 2015: Advances in Artificial Intelligence*. Springer. 19–31.

Chrpa, L.; Scala, E.; and Vallati, M. 2015. Towards a reformulation based approach for efficient numeric planning: Numeric outer entanglements. In *Proc. of the 8th Symposium on Combinatorial Search (SOCS)*.

Coles, A.; Fox, M.; Long, D.; and Smith, A. 2008. A Hybrid Relaxed Planning Graph-LP Heuristic for Numeric Planning Domains. In *Proc. of the 18th Intl. Conf. on Automated Planning and Sched. (ICAPS)*, 52–59.

Cousot, P., and Cousot, R. 1977. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, 238–252. ACM.

Hoffmann, J. 2003. The Metric-FF planning system: Translating "ignoring delete lists" to numeric state variables. *Journal of Artificial Intelligence Research* 20:291–341.

Long, D., and Fox, M. 2003. The 3rd international planning competition: Results and analysis. *Journal of Artificial Intelligence Research* 20:1–59.

Muise, C.; McIlraith, S. A.; and Beck, J. C. 2012. Improved Non-deterministic Planning by Exploiting State Relevance. In *Proc. of the 22nd Intl. Conf. on Automated Planning and Sched. (ICAPS)*.