Dissertation Abstract: Numeric Planning

Johannes Aldinger
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Georges-Köhler-Allee 52
79110 Freiburg, Germany
aldinger@informatik.uni-freiburg.de

Extended Abstract[1]

Planning is the art to automatically find solutions to problems where a model of the world is described in terms of variables and actions. A solution to such a planning problem is a sequence of actions (called *plan*) transforming the initial situation to a state that satisfies a goal description. In *classical planning*, the world is described by Boolean variables and the field is well studied with eminent advances of the state of the art in the last decades. However, classical planning is not expressive enough, e.g. for many interesting real-world applications that rely on numeric quantities. Therefore, we contemplate on numeric planning in this dissertation.

Background

For planning, the domain description language PDDL is the standard to model planning problems. Fox and Long (2003) extended this language in order to model more expressive planning problems. In PDDL2.1, the expressiveness of the planning language is classified into layers. Classical planning problems can be expressed in layer 1. Layer 2 allows for numeric, rational valued, variables and can therefore express physical properties (such as the velocity of a vehicle) as well as resources (such as the fuel level of a vehicle). In layer 3, actions can have a duration in order to model planning problems requiring the concurrent execution of actions. Changes to the world happen at specific instants (start of action, end of action), an assumption that is lifted in layer 4, where continuous change of variables (e.g. the concurrent filling and draining of a tub) can be modeled as well. Finally, layer 5 allows for exogenous events: the world is dynamic and events can happen without the influence of the planning agent. This dissertation aims at shedding light on numeric planning expressible with PDDL2.1, layer 2, also known as numeric planning with instantaneous actions.

A numeric planning task $\Pi = \langle \mathcal{V}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$ is a 4-tuple where $\mathcal{V}$ is a set of numeric variables v with domain $\mathbb{Q}^\infty := \mathbb{Q} \cup \{-\infty, \infty\}$. $\mathcal{O}$ is a set of operators, $\mathcal{I}$ the initial state and $\mathcal{G}$ the goal condition. A *numeric expression* $\mathsf{e}_1 \circ \mathsf{e}_2$ is an arithmetic expression with operators $\circ \in \{+, -, \times, \div\}$ and expressions e_1 and e_2 recursively defined over variables $\mathcal{V}$ and constants from $\mathbb{Q}$. A *numeric constraint* $(\mathsf{e}_1 \bowtie \mathsf{e}_2)$ compares numeric expressions e_1 and e_2 with $\bowtie \in \{\leq, <, =, \neq\}$. A *condition* is a conjunction of propositions and numeric constraints. A *numeric effect* is a triple $(\mathsf{v} \circ\!\!= \mathsf{e})$ where $\mathsf{v} \in \mathcal{V}$, $\circ\!\!= \in \{:=, +=, -=, \times=, \div=\}$ and e is a numeric expression. Operators $o \in \mathcal{O}$ are of the form $\langle \mathrm{pre} \rightarrow \mathrm{eff} \rangle$ and consist of a condition pre and a set of effects $\mathrm{eff} = \{\mathrm{eff}_1, \ldots, \mathrm{eff}_n\}$ containing at most one numeric effect for each numeric variable and at most one truth assignment for each propositional variable.

The semantics of a numeric planning task is straightforward. For constants $c \in \mathbb{Q}$, $s(c) = c$. Numeric expressions $(\mathsf{e}_1 \circ \mathsf{e}_2)$ for $\circ \in \{+, -, \times, \div\}$ are recursively evaluated in state s: $s(\mathsf{e}_1 \circ \mathsf{e}_2) = s(\mathsf{e}_1) \circ s(\mathsf{e}_2)$. a numeric constraints $(\mathsf{e}_1 \bowtie \mathsf{e}_2)$, with expressions e_1, e_2 and $\bowtie \in \{\leq, <, =, \neq\}$, $s \models (\mathsf{e}_1 \bowtie \mathsf{e}_2)$ is satisfied iff $s(\mathsf{e}_1) \bowtie s(\mathsf{e}_2)$.

Related Work

Extending classical delete relaxation heuristics to numeric problems has been done before, albeit only for a subset of numeric tasks, where numeric variables can only be manipulated in a restricted way. The Metric-FF planning system (Hoffmann 2003) tries to convert the planning task into a *linear numeric* task, which ensures that variables can "grow" in only one direction. When high values of a variable are beneficial to fulfill the preconditions, *decrease* effects are considered harmful. Another approach to solve *linear* numeric planning problems is to encode numeric variables in a linear program and solve constraints with an LP-solver. Coles et al. (2008) analyze the planning problem for consumers and producers of resources to obtain a heuristic that ensures that resources are not more often consumed than produced or initially available. The RANTANPLAN planner (Bofill, Arxer, and Villaret 2015) uses linear programs in the context of planning as satisfiability modulo theories. Instead, we are interested in approaching numeric planning supporting *all* arithmetic base operations.

Contributions

The objective of this dissertation is to provide planning system for numeric planning with instantaneous actions. At the core stands the Fast Downward planning system (Helmert 2006) for classical planning. Our intent is to extended Fast

[1] Parts of this extended abstract were adopted from previous work (Aldinger, Mattmüller, and Göbelbecker 2015)

Downward with full numeric functionality without impairing its performance on classical domains. Numeric Fast Downward (NFD), the numeric extension of Fast Downward has to alter almost all components of the Fast Downward planning systems. Fast Downward transforms the PDDL input into more convenient and effective data structures. During the translation phase, the task is grounded translated into a multi-valued SAS^+ representation. During a knowledge compilation step, domain transition graph and causal graph are determined. Finally, different search algorithms can be used together with a multitude of heuristics in order to solve the planning task. Extending Fast Downward requires major modifications in all steps, and NFD has to deal with interactions between numeric variables, interactions between numeric and multi-valued variables and new challenges coming from the numeric abilities (e.g. an operator can now have infinitely many different outcomes, depending on the previous state). Some of the extensions for NFD could be adopted from Temporal Fast Downward (TFD) (Eyerich, Mattmüller, and Röger 2009), a temporal planner based on an earlier version of Fast Downward, e.g. the handling of numeric expressions by recursively introducing auxiliary variables for each expression. The evaluation of these auxiliary variables is then handled by numeric axioms. Other extensions had to be developed from scratch, either because the Fast Downward evolved from the time when TFD branched from it, or because some features were never implemented for TFD (e.g. the detection of unreachable world states early on can simplify internal data structures).

In order to obtain a baseline heuristic for numeric planning, we implement numeric extensions of the relaxation heuristics from classical planning $h_{\max}$, $h_{\text{add}}$and h_{FF}. In order to do so, we found that the theoretical base for relaxed numeric planning was not set and had to be established first. Numeric planning is undecidable (Helmert 2006) in general, while classical planning is PSPACE-complete (Bylander 1994). Nevertheless, plans exist for many numeric problems and for many numeric problems we can prove unsolvability, so we seek guidance for these problems. This guidance is obtained by heuristics and in order to be tractable we want the estimate to be computable in polynomial time. The idea of a relaxation heuristic is that every fact that is achieved once during planning remains achieved. The problem is simplified as the set of achieved facts grows monotonously. We studied different extensions to relaxation for numeric planning (Aldinger, Mattmüller, and Göbelbecker 2015) and found intervals to be suitable. The idea of an interval relaxation is to store the lower and the upper bound of achievable values in an interval, ensuring that the reachable values can only grow at each step. The methods to deal with intervals have been studied in the field of interval arithmetic. Nevertheless, a major obstacle in numeric planning has to be overcome: the repeated application of numeric operators. While relaxed operators are idempotent in classical planning, the same operator can alter the state of the world arbitrarily often (e.g. $\langle \emptyset \rightarrow \mathsf{x} \mathrel{+}= 1\rangle$ can increase $x_0 = [0, 0]$ to $x_i = [0, i]$ after i steps). We analyzed conditions under which this repeated application of operators can be captured in polynomial time, and how interval relaxed plans can be derived by explicating the number of repetitions. For *acyclic* numeric planning tasks, i.e. tasks where variables do not depend directly or indirectly on themselves, we proved that the interval relaxation in P. For cyclic tasks, we can introduce cycle breaker actions that artificially set the reachable values of a variable to $(-\infty, \infty)$. While this impairs the quality of the heuristic estimate, it ensures that the heuristic can be computed in polynomial time.

On the practical side we use numeric planning in the context of earth observation satellites application of numeric planning to earth observation satellites (Aldinger and Löhr 2013). An Earth observation satellite equipped with heavy optical sensors has to slew towards regions of interest while orbiting Earth. The number of observation sites exceeds the capability of the satellite and attitude dynamic constraints have to be satisfied.

Open Research Ideas

In the near future we are interested in addressing the open question whether the restriction to acyclic numeric planning tasks can be weakened. We are also interested in tackling another problem inherent to relaxation heuristics: the cyclic resource transfer problem (Coles et al. 2008). Numeric variables are frequently used to model resources. If an operator can transfer resources from one location to another, this is modeled by reducing the quantity at the source location while increasing it at the target. In the relaxed problem, the quantity of the resource is not decreased at the source location, and as such a resource can be “produced” by moving it around. This deteriorates heuristic estimates in many (relevant) numeric planning problems. Coles et al. (2008) use linear programming to ensure that no more resources are consumed than produced. Linear programs are also used in the numeric planning system RANTANPLAN by Bofill, Arxer, and Villaret (2015). We believe that linear programming can be fruitful for numeric planning in many ways and opens many promising research directions for future work.

References

Aldinger, J., and Löhr, J. 2013. Planning for Agile Earth Observation Satellites. In *Proceedings of the ICAPS-2013 Workshop on Planning in Continuous Domains (PCD 2013)*, 9–17.

Aldinger, J.; Mattmüller, R.; and Göbelbecker, M. 2015. Complexity of Interval Relaxed Numeric Planning. In *Proceedings of the 38th German Conference on Artificial Intelligence (KI 2015)*.

Bofill, M.; Arxer, J. E.; and Villaret, M. 2015. The RANTANPLAN Planner: System Description. In *Proceedings of the ICAPS-15 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems (COPLAPS 2015)*, 1–10.

Bylander, T. 1994. The Computational Complexity of Propositional STRIPS Planning. *Artificial Intelligence 69* 165–204.

Coles, A.; Fox, M.; Long, D.; and Smith, A. 2008. A Hybrid Relaxed Planning Graph-LP Heuristic for Numeric Planning

Domains. In *Proceedings of the 20th International Conference on Automated Planning and Search (ICAPS 2008)*, 52–59.

Eyerich, P.; Mattmüller, R.; and Röger, G. 2009. Using the Context-enhanced Additive Heuristic for Temporal and Numeric Planning. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS 2009)*, 130–137.

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

Helmert, M. 2006. The Fast Downward Planning System. *Journal of Artificial Intelligence Research 26 (JAIR)* 191–246.

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