Proceedings of the Eighth International Symposium on Combinatorial Search (SoCS-2015)

Towards a Reformulation Based Approach for Efficient Numeric Planning: Numeric Outer Entanglements

Lukáš Chrpa
PARK Research group
School of Computing & Engineering
University of Huddersfield

Enrico Scala
Research School of Computer Science
Australian National University

Mauro Vallati
PARK Research group
School of Computing & Engineering
University of Huddersfield

Abstract

Restricting the search space has shown to be an effective approach for improving the performance of automated planning systems. A planner-independent technique for pruning the search space is *domain and problem reformulation*. Recently, *Outer Entanglements*, which are relations between planning operators and initial or goal predicates, have been introduced as a reformulation technique for eliminating potential undesirable instances of planning operators, and thus restricting the search space. Reformulation techniques, however, have been mainly applied in classical planning, although many real-world planning applications require to deal with numerical information.

In this paper, we investigate the usefulness of reformulation approaches in planning with numerical fluents. In particular, we propose and extension of the notion of outer entanglements for handling numeric fluents. An empirical evaluation, which involves 150 instances from 5 domains, shows promising results.

Introduction

While propositional planning turned out to be a reasonable abstraction for several contexts, when planning has to be applied to realistic systems such as robotic systems, spatial and resource constraints become crucial to generate plans which are actually feasible.

For handling numerical information, the PDDL language has been extended (Fox and Long 2003). Unfortunately, the numeric extension makes the planning task much more complex than the propositional case, and if not restricted, numerical planning can be even undecidable (Helmert 2002). Given the importance of numerical reasoning for real-world planning applications, and the complexity of the topic, it is worthy investigating techniques that allow to improve the performance of planners in such scenarios.

In the propositional case, learning of problem structural knowledge exploited by reformulation techniques has shown to speed up plan generation, often considerably. Reformulation can be done, for instance, by considering macro-operators (Newton et al. 2007; Botea et al. 2005) or splitting complex operators (Areces et al. 2014). Interestingly, some of them has also shown good results when exploited in numerical planning. For instance, macro-operators have also appeared in the context of online numeric planning (Scala 2014). Besides macro-operators – that represent "shortcuts" in the search space – pruning techniques (e.g. stubborn sets (Wehrle and Helmert 2014)) have been also proposed for propositional planning. They aim at restricting the search space, thus boosting the planning process. *Outer entanglements*, relations between planning operators and initial or goal predicates, have been recently introduced as a reformulation technique for removing potential undesirable instances of planning operators. Despite not preserving optimality outer entanglements can considerably reduce the number of grounded operator instances, often with remarkable impact on planners' performance (Chrpa and Barták 2009; Chrpa and McCluskey 2012).

In this paper, we extend the notion of outer entanglements to numerical fluents. In particular, we define *numeric outer entanglements* as relations between planning operators and initial assignments of numerical fluents, or goal numeric expressions. We empirically evaluate their impact on the planning process by using 5 numeric domains and 3 numeric planners. Results show the potentiality of the approach in terms of coverage.

Background

This section reports some terminology used throughout the paper. For further details on the semantics, the interested reader is referred to (Fox and Long 2003).

Informally, a numeric planning problem is the task of finding a sequence of actions that applied starting from an initial state, leads to a state in which the goal is satisfied. Formally, a numeric planning task can be described by the tuple $\Pi = \langle D, I, G, Objs \rangle$ where *D* represents the domain encoding the universe *F* and *X* of propositions and numeric fluents and a set of operators *O*. *I*, *G* and *Objs* represent a problem in *D*. I is the initial state, *G* is a set of goals and *Objs* a set of objects upon which *F*, *X* and the *O* can be instantiated. A state s is the pair $\langle F(s), X(s) \rangle$, where *F(s)* specifies which atoms hold in s, while *X(s)* represents what value is assigned to each numeric variable in s. An operator o is the tuple $\langle par(o), pre(o), eff(o) \rangle$, which respectively represent the parameter, the precondition and effect sets of the operator. Precondition involves both propositional statements and numeric conditions in form of inequalities `({<,<=,==,>=,>} exp exp')`. Effects of the operator include classical add and delete list ($add(o)$, $del(o)$), but also a set of numeric effects of the form `(op x exp)` where x is the fluent affected by the operation and $op \in increase, decrease, assign$. exp and exp' are arithmetical expressions over variables from X. An instance of operator o is obtained grounding its parameters with a set of objects which have to be consistent with the types defined in par(o).

A solution plan π for Π is a sequence of actions whose execution leads the state I to a state $s_{|\pi|}$ satisfying G and $sol(\Pi)$ is a set of all solution plans of Π. Moreover, a solution plan is such that $I \models pre(a_0)$, $I[a_0] \models pre(a_1)$ ... $I[a_0, a_1, ...a_{n-2}] \models pre(a_{n-1})$ where the operation $s[a]$ defines the state resulting from the application of a in s.

Propositional Outer Entanglements

Outer Entanglements are relations between planning operators and initial or goal predicates, and have been introduced as a technique for eliminating potential unnecessary instances of these operators (Chrpa and McCluskey 2012).

Informally, entanglements by init capture situations where only instances of a given operator that require the same instances of a certain predicate as initial ones are needed to solve the problem. Similarly, entanglements by goal capture situations where only instances of a given operator that achieve the same instances of a certain predicate as goal ones are needed to solve the problem.

For example, in the BlocksWorld domain, the operator unstack is entangled by init with the predicate on, since unstacking blocks only from their initial positions is sufficient. Similarly, the operator stack is entangled by goal with the predicate on, since stacking blocks only to their goal positions is sufficient.

Definition 1 (Propositional Outer Entanglement) *Given a Planning Task $\Pi = \langle D, I, G, Objs\rangle$, we say that operator o is **entangled by I (resp. G)** with predicate p in Π if and only if $p \in pre(o)$ (resp. $p \in add(o)$) and $\exists\pi \in sol(\Pi)$ such that $\forall a \in \pi$ where $a \in grounding(o)$, and $\forall p_{gnd}$, where $p_{gnd} \in grounding(p)$, it holds: $p_{gnd} \in pre(a) \Rightarrow p_{gnd} \in I$ (resp. $p_{gnd} \in add(a) \Rightarrow p_{gnd} \in G$).*

Outer entanglements have been used as a reformulation technique, as they can be directly encoded into a problem and domain model. The way outer entanglements are encoded is inspired by one of their properties: given a static predicate p_s, an operator *o* is entangled by init with p_s if and only if $p_s \in pre(o)$ (Chrpa and Barták 2009). Introducing supplementary static predicates, "clones" of the predicates involved in the outer entanglement, and putting them into preconditions of operators in the outer entanglement relation (both init and goal) will *filter* instances of these operators which do not follow the entanglement conditions. For a formal description of the reformulation process, the interested reader is referred to (Chrpa and McCluskey 2012).

Numeric Entanglements

For the numeric case the idea is to relate the applicability of an action to a particular configuration of states in X. Numeric conditions are often used to represent consumable and continuous resources (e.g., the fuel required to perform a fly action). In this context, numeric entanglements can be used to understand if there exists a relation between an operator and the initial resource at disposal.

Therefore, analogously to the propositional case, a numeric entanglement is defined as follows:

Definition 2 (Numeric Outer Entanglement by Init) *Given a Planning Task $\Pi = \langle D, I, G, Objs\rangle$, we say that an operator o is **numerically entangled by I** with c if $\exists\pi \in sol(\Pi)$ such that $\forall a \in \pi$ where $a \in grounding(o)$ then $c_{gnd} \in pre(a) \Rightarrow I \models c_{gnd}$, that is the evaluation of the condition c in I is satisfied in an arithmetical way.*

It is worth noting that this definition only deals with the by-init case. Extending the notion of outer entanglements for considering relation between operators and numeric condition in the goal is not straightforward. The problem relies on the fact that a numeric condition does not inherit the same state based interpretation of the propositional case. In the classical case in fact it is possible to reason on action effects and goal statements homogeneously since both represent partial world states. The propositional outer entanglement condition by goal is hence quite symmetrical to the one defined for the by-init case. On the contrary, action numeric effects are functions of the state in which the action is applied. The resulting execution is not independent from the particular context.

However, it is possible to find a restricted definition of numeric entanglements by goal. The idea is to consider only those relations between numeric effects and numeric goals in which it is possible to show that the execution of the action always satisfies the numeric condition. Formally:

Definition 3 (Numeric Outer Entanglement by Goal) *Given a planning task $\Pi = \langle D, I, G, Objs\rangle$, we say that an operator is **numerically entangled by G** with c if $\exists\pi \in sol(\Pi)$ such that $\forall$ $a \in \pi$ where $a \in grounding(o)$ it holds that $eff^{gnd}_{num} \in eff_{num}(a) \Rightarrow s^*[eff^{gnd}_{num}] \models c_{gnd}$ where $s^* \in 2^P \times R^{|X|}$ and $c_{gnd} \in G$.*

State s^* represents any possible state within the space of states generated by the planning problem, $s^*[eff^{ground}_{num}]$ is the state resulting from the application of a numeric effect of *a* on s^*. Numeric effects and numeric conditions can be quite complex as they may involve complicated expression. A sufficient condition for finding this kind of relation is to restrict the attention to simple forms of numeric conditions and operators. In particular, numeric conditions containing only one numeric fluent (e.g., `( > (power r1) 5)`) *assign* effects which assign a value to the numeric fluent object of the condition (e.g, `(= (power r1 10) ))`[1].

Reformulation

Definition 2 states that, if there exists an operator entangled by init with some condition *c*, then there exists a solution for the problem that does not include any instance of that operator whose entangled precondition is not satisfied. In order

[1]Focusing on a simplified form of numeric effects turns out to be fruitful also in other contexts, such as (Coles et al. 2013)

to prune those instances, for each entangled condition we reformulate the problem by inserting a new set of numeric fluents having the same parameters of the fluents involved in the entangled condition. As for the propositional case, these fluents have to encode the initial state of the system and have to be static (no actions can modify their value). Then, we can exploit these fluents for building a new set of static preconditions. As a matter of fact, it holds that if an action is numerically conditioned by a comparison c in which all the involved fluents are static, then the action is numerically entangled with the initial status with c.

For instance, if one has the entangled condition `(> (fuel r1) 5)`, it is necessary to encode the new fluent `(fuel1 r1)`, set the initial value such that `fuel1=fuel`, and add a new condition in the operator, having `fuel1` as parameter `(> (fuel1 r1) 5)`. Then the only possible groundings for this operator will be the ones satisfying the numeric entanglement condition by init.

Reformulating problems for numeric entanglements by goal is achieved in a similar way. In particular our reformulation focuses on entanglements by goal where the numeric condition in the goal is of the form `({<,<=,==,>=,>} f k)` where f is an element from X, k is a constant real number and the operator's effect is $(= f'z')$. Here f' is the abstracted version of f and z' is another real number. To preserve only operator's entangled with the condition, we add a new precondition to the action which is satisfied only when the grounding of f' corresponds to f. This reformulation assumes that there are no other comparisons in the goal involving the same numeric fluent. Generalising the reformulation for a larger set of entanglements is object of our future work.

Learning Entanglements

Discovering valid propositional entanglements has been shown to be infeasible from a practical point of view since the problem of deciding if an operator is entangled for a given propositional condition is as hard as the planning problem itself (Chrpa, McCluskey, and Osborne 2012). Although formal results do not exist for the numeric case, the arising decision problem for numeric entanglements is likely to be at least as hard as for the propositional case.

To alleviate this problem this paper proposes an approximate approach; i.e., given a set of easy instances for a given domain, and a set of plans representing valid solutions for those instances, we generalise the entanglement detected as valid entanglement for that domain. Although such an approach is generally incomplete, lessons learnt from the propositional case indicate that making a problem unsolvable rarely occurs (Chrpa and Barták 2009). Algorithm 2 details the learning strategy implemented.

The procedure iterates over a set of plans, and for each of them keeps track of the number of times a given condition is the entanglement (initial) condition for a corresponding given problem. Both propositional and numeric entanglements can be detected whenever an instance of an operator is inspected. For the propositional case it suffices to see if the initial state "contains" the current proposition. For the numeric case it is necessary to evaluate the numeric constraint by using the initial values for the numeric fluents. If the inequality is satisfied then the counter is increased.

The approach extends the learning approach pursued by (Chrpa and McCluskey 2012) where a *flaw-ratio* is used to allow a small percentage of "errors" in the training plans, since they might, for instance, contain useless instances of an operator that might cause a given condition to be not considered as an entanglement condition. If a flaw-ratio is used, detected entanglements have to be cross-validated, so the training problems are reformulated and passed to the planner. If the planner fails in any of the reformulated problems, the flaw-ratio is reduced and the learning procedure is repeated.

In the algorithm the function *abs* abstracts a given action instance to the corresponding operator. The procedure focuses on entanglements by init. The extension to entanglements by goal is straightforward.

Algorithm 2: Learning Propositional and Numeric Entanglements by Init

Input: D: Domain
Ps: Problems set
Plans: Plans set
f: flaw-ratio
Output: $prop\_ent(D) \cup num\_ent(D)$

set #sat and #occ to 0 for each operator's condition
forall the $P \in Ps$ **do**
    plan = pick_solution($\langle D, Ps\rangle, Plans$);
    forall the *action* $a \in$ *plan* **do**
        forall the $c \in Pre(a)$ **do**
            if $c \in F(D)$ **then**
                if $c \in init(P)$ **then**
                    #sat(c,abs(a)) += 1
            else if *c is a num condition* **then**
                if $init(P) \models c$ **then**
                    #sat(c,abs(a)) += 1
            #occ(c,abs(a)) += 1;
ents=$\{(c, o) : \exists o \in O(D), c \in pre(o), \frac{\#sat(c,o)}{\#occ(c,o)} \geq (1 - f)\}$
return *ents*

Experimental Analysis

In this section we present the results of our preliminary experimental study examining the usefulness of numerical entanglements. We have considered problem instances from 5 well-known planning domains: Depots, DriverLog, Rovers, Satellite and ZenoTravel. These domains were selected because they have a numeric formulation, and random generators are available. For each domain, we generated 30 benchmark problems as testing instances. The 5 easiest problems used in planning competitions are used for training purposes. A runtime cutoff of 900 CPU seconds was used. As planners, we considered LPG (Gerevini, Saetti, and Serina 2003), FF (Hoffmann 2003) and Colin (Coles et al. 2012). All the experiments were run on a machine with 3.0 Ghz CPU and

FF

Domain	ΔIPC Score			Δ Solved		
	Num	**Prop**	**All**	**Num**	**Prop**	**All**
Depots	+0.2	+3.8	+7.2	0	0	0
Driverlog	+4.1	+2.7	+3.6	+3	+2	+3
Rovers	+1.4	+0.7	+0.6	0	0	−1
Satellite	–	–	–	–	–	–
Zeno	+0.1	+8.5	+9.4	0	0	0
Total	+5.8	+15.7	+20.4	+3	+2	+2

LPG

Domain	ΔIPC Score			Δ Solved		
	Num	**Prop**	**All**	**Num**	**Prop**	**All**
Depots	−0.9	−3.0	+3.7	0	0	0
Driverlog	−0.3	−1.5	+2.0	0	0	0
Rovers	0.0	−9.0	−8.4	0	−8	−8
Satellite	−2.2	−1.1	−2.2	0	0	0
Zeno	+0.2	+0.9	+2.4	0	0	0
Total	−3.2	−13.7	−2.5	0	−8	−8

Table 1: Delta of performance, with regards to original formulation, of IPC score and solved problems achieved by FF (upper) and LPG (lower) when exploiting numerical entanglements (Num), propositional entanglements (Prop) and both (All). Entanglements are extracted using training plans generated by the corresponding planner.

4GB of RAM. In this experimental analysis, the IPC score as defined in the Agile track of IPC 2014 is used. For a planner $\mathcal{C}$ and a problem p, $Score(\mathcal{C}, p)$ is 0 if p is unsolved, and $1/(1+\log_{10}(T_p(\mathcal{C})/T_p^*))$, where $T_p(\mathcal{C})$ is the CPU-time needed by planner $\mathcal{C}$ to solve problem p and T_p^* is the CPU-time needed by the best considered planner, otherwise. The IPC score on a set of problems is given by the sum of the scores achieved on each considered problem.

Table 1 shows the results of LPG and FF on the considered benchmark instances. Results are shown in terms of delta of performance between the original problem formulation, and formulations exploiting numerical only, propositional only, or both entanglements. Entanglements have been extracted by considering plans generated, on the 5 training problems, by the corresponding planner: i.e., LPG is exploiting entanglements learned by considering training plans generated by itself. Entanglements have different impact on the two planners. FF is able to achieve considerably better performance while exploiting them. It should be noted that in Satellite, no entanglements have been extracted by analysing FF training plans. On the other hand, when considering LPG, they seem to have a detrimental effect on most of the domains. Interestingly, we can observe that there is some sort of synergy between numerical and propositional entanglements: using both usually further improve performance. This is not true in Rovers, where numerical entanglements used without propositional ones allow both LPG and FF to achieve better performances. Entanglements constrain the search space by pruning some instances of operators. Whereas it is often beneficial, sometimes it might cause dead-ends occurrence in the search space. If a planner is stuck in a dead-end it must backtrack to recover from it, and thus spend much more time in the planning process. This seems to be an issue especially for local search based planners such as LPG. Similar peculiarities of entanglements have already been discussed in the literature (Chrpa and McCluskey 2012).

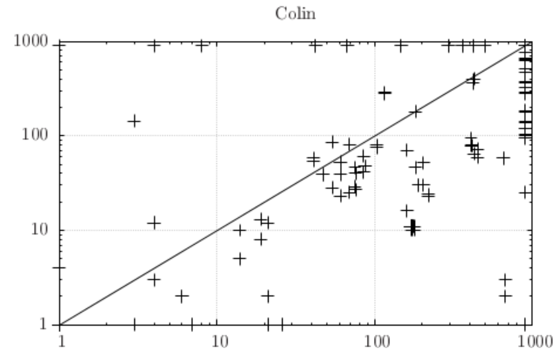

Figure 1: CPU time of Colin exploiting original problem formulation (x-axis) wrt problems reformulated by including all the entanglements extracted by either FF or LPG (y-axis). Crosses at 900 seconds indicate unsolved problems.

In order to understand the impact of entanglements on different planners, we run Colin on problems reformulated with entanglements extracted by considering training plans generated by either FF or LPG. Figure 1 shows an overall comparison in terms of CPU time between Colin using the original problem formulation, and both propositional and numerical entanglements. Remarkably, Colin's performance is usually improved when exploiting either FF or LPG entanglements. This possibly indicates that extracted entanglements have good quality, even though they are not identical.

Summary and Future Work

Although dealing with numerical information is of critical importance for many real-world planning applications, many approaches have been proposed for improving the performance of classical planning systems. Within the class of reformulation techniques, outer entanglements have been used for planner-independent pruning of the search space, often with significant impact on the planning performance.

In this paper, we extended the notion of outer entanglements, by considering numerical fluents. In particular, we defined *numeric outer entanglements* as relations between operators and initial numerical fluents assignments. An empirical analysis, which involved 150 benchmarks, showed that numeric outer entanglements are promising for improving planning performance.

As a future work we plan to evaluate the impact of numeric entanglements on plan quality, and extend them to cover more complex numeric expressions as well as inferring relations between operators, that can be used to build macro operators.

Acknowledgements. This work was supported in part by ARC project DP140104219, Robust AI Planning for Hybrid Systems and by the UK EPSRC Autonomous and Intelligent Systems Programme (grant no. EP/J011991/1). We would like to acknowledge the use of the University of Huddersfield Queensgate Grid.

References

Areces, C.; Bustos, F.; Dominguez, M.; and Hoffmann, J. 2014. Optimizing planning domains by automatic action schema splitting. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, (ICAPS)*.

Botea, A.; Enzenberger, M.; Müller, M.; and Schaeffer, J. 2005. Macro-FF: improving AI planning with automatically learned macro-operators. *Journal of Artificial Intelligence Research (JAIR)* 24:581–621.

Chrpa, L., and Barták, R. 2009. Reformulating planning problems by eliminating unpromising actions. In *Proceedings of SARA*, 50–57.

Chrpa, L., and McCluskey, T. L. 2012. On exploiting structures of classical planning problems: Generalizing entanglements. In *Proceedings of ECAI*, 240–245.

Chrpa, L.; McCluskey, T. L.; and Osborne, H. 2012. Reformulating planning problems: A theoretical point of view. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference, Marco Island, Florida. May 23-25, 2012*.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2012. Colin: Planning with continuous linear numeric change. *Journal of Artificial Intelligence Research*.

Coles, A. J.; Coles, A.; Fox, M.; and Long, D. 2013. A hybrid LP-RPG heuristic for modelling numeric resource flows in planning. *J. Artif. Intell. Res. (JAIR)* 46:343–412.

Fox, M., and Long, D. 2003. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res. (JAIR)* 20:61–124.

Gerevini, A.; Saetti, A.; and Serina, I. 2003. Planning through stochastic local search and temporal action graphs in lpg. *J. Artif. Intell. Res.(JAIR)* 20:239–290.

Helmert, M. 2002. Decidability and undecidability results for planning with numerical state variables. In *On the Role of Ground Actions in Refinement Planning.*, 44–53.

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

Newton, M. A. H.; Levine, J.; Fox, M.; and Long, D. 2007. Learning macro-actions for arbitrary planners and domains. In *Proceedings of the International Conference on Automated Planning and Scheduling, ICAPS*, 256–263.

Scala, E. 2014. Plan repair for resource constrained tasks via numeric macro actions. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2014, Portsmouth, New Hampshire, USA, June 21-26, 2014*.

Wehrle, M., and Helmert, M. 2014. Efficient stubborn sets: Generalized algorithms and selection strategies. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2014, Portsmouth, New Hampshire, USA, June 21-26, 2014*.