

Landmarks for Numeric Planning Problems

Enrico Scala*, Patrik Haslum*, Daniele Magazzeni**, Sylvie Thiébaux*

*The Australian National University, Canberra (AUS) - firstname.lastname@anu.edu.au

**King's College London, London, WC2R 2LS - firstname.lastname@kcl.ac.uk

Abstract

The paper generalises the notion of landmarks for reasoning about planning problems involving propositional and numeric variables. Intuitively, numeric landmarks are regions in the metric space defined by the problem whose crossing is necessary for its resolution. The paper proposes a relaxation-based method for their automated extraction directly from the problem structure, and shows how to exploit them to infer what we call disjunctive and additive hybrid action landmarks. The justification of such a disjunctive representation results from the intertwined propositional and numeric structure of the problem. The paper exercises their use in two novel admissible LP-Based numeric heuristics, and reports experiments on cost-optimal numeric planning problems. Results show the heuristics are more informed and effective than previous work for problems involving a higher number of (sub)goals.

Introduction

Numeric planning [Fox and Long, 2003] is the extension of classical/STRIPS planning where also numeric variables can participate into the definition of (sub)goals and actions of the problem. Despite being undecidable in the general case [Helmert, 2002], efficient reasoning in this setting is a key ingredient for planning in real world contexts, where propositional and numeric reasoning has to be done in an combined way [Della Penna *et al.*, 2009; Scala *et al.*, 2016b; Fox and Long, 2006; Löhr *et al.*, 2012; Fox *et al.*, 2011].

In propositional planning, landmarks [Hoffmann *et al.*, 2004], meaning atoms/actions necessary to achieve the goal, have been a key mechanism to devise informed heuristics used by both optimal and satisficing state-of-the-art planners [Keyder *et al.*, 2010; Helmert and Domshlak, 2009; Pommerening *et al.*, 2014; Haslum *et al.*, 2012]. In addition landmarks elicit some of the problem's structure, which may be used in other ways, for example to decompose the problem. Finding the complete set of landmarks is as hard as solving it [Hoffmann *et al.*, 2004]; thus, most works approximate the landmark set using some kind of problem relaxation, and the hope is that such a relaxation still contains useful structural information. Recently, new relaxations for numeric

planning have been devised [Scala *et al.*, 2016a], which, under some conditions, improve on previous interval-based relaxation heuristics [Hoffmann, 2003; Scala *et al.*, 2016b].

This paper studies landmarks for numeric planning through the lens of the subgoaling relaxation [Scala *et al.*, 2016a]. For this relaxation we derive a hybrid version of the AND/OR graph introduced by Keyder *et al.* [2010], capturing achievers and dependencies of both numeric and propositional subgoals; this gives us a method for inferring both types of landmarks. Numeric landmarks are metric conditions that are *necessarily* achieved in every plan solving the problem. Extending previous cost-partitioning techniques [Karpas and Domshlak, 2009], we formulate a linear program that estimates not only which actions are necessary, but also can quantify the number of times that actions need to be executed in any plan solving the problem. The result is a landmark-based admissible heuristics for cost-optimal numeric planning, which we show to be more effective than the admissible numeric heuristic proposed by Scala *et al.* [2016a] for certain kinds of problems. Other (mainly inadmissible) heuristics for numeric planning have been studied in the past [Hoffmann, 2003; Coles and Coles, 2011; Eyerich *et al.*, 2012]; yet, to the best of our knowledge, no other work has developed them from a landmarks perspective. This paper attempts to close this gap.

An Illustrative Example

Consider an N-dimensional, metric Travelling Salesman Problem: An agent, initially at $\langle x_0, \dots, x_{n-1} \rangle$, must visit each of a set of regions, w_0, w_1, \dots, w_{m-1} , in any order, minimising its total movement. Each region w_j is defined by a conjunction of simple inequalities $\bigwedge_{0 \leq i < n} (x_i + k_{i,j} \{ \leq, \geq \} 0)$ that bounds it in each dimension. The position of the agent is represented by n numeric state variables. A goal proposition, g_i , for each region is achievable by an action $visit_i$ whose precondition is that the agent is in the region. Because the agent's movement is encoded numerically, the propositional landmarks of this problem are only the goals g_0, \dots, g_{m-1} , and the corresponding heuristic estimate is weak. The hybrid h_{hbd}^{max} admissible heuristic [Scala *et al.*, 2016a], on the other hand, does capture dependencies between numeric and propositional reachability, but since it assumes subgoal independence it will see only the single most costly goal. The intuition here is that even though the order of visits is not fixed,

any plan solving the problem has to *intersect* the goal regions at some point. These are necessary metric conditions; in other words, landmarks. This paper shows how such numeric landmarks can be found and used in a much more informed admissible estimate.

Fundamentals and Previous Work

This paper focuses on numeric planning problems, which extend propositional planning with numeric state variables. Numeric variables can appear in conditions $\xi \triangleright k$ where ξ is an arithmetic expression, $\triangleright \in \{\leq, <, =, >, \geq\}$ and k is a constant. In the fragment we consider, action preconditions and goals are conjunctions of numeric conditions and/or propositions, ξ is linear, and actions can assign boolean variables and/or increase/decrease the value of numeric variables by a constant factor. Conflicting assignments are not allowed. The semantics of numeric planning can be defined in terms of a state transition system following PDDL 2.1 level 2 [Fox and Long, 2003]. We adopt the set-theoretic representation [Ghalab *et al.*, 2004].

Given a set of boolean (F) and numeric variables (X), a numeric planning problem consists in finding an action sequence $\langle a_0, \dots, a_{n-1} \rangle$ which, starting from an initial assignment to all variables (the initial state s_0), brings the system to a state (s_n) satisfying given goal conditions G (i.e., $s_n \models G$). Each action has to be applicable in the state resulting from the plan prefix executed before it. Each action a_i has a cost $c(a_i) \in \mathbb{R}^{\geq 0}$ and the cost of a plan $\pi = \langle a_0, \dots, a_{n-1} \rangle$ is $\sum_{0 \leq i < |\pi^*|} c(a_i)$; the objective is to minimize cost.

Landmarks

Propositional landmarks [Hoffmann *et al.*, 2004] are propositions that need to be true at some point in any plan solving the problem, whereas action landmarks are the actions whose execution is necessary to achieve the goal: any plan solving the problem has to contain them.

A sound and complete algorithm to find landmarks is not tractable [Hoffmann *et al.*, 2004]; therefore previous methods extract landmarks using a relaxed version of the problem. Intuitively, tighter relaxations bring more landmarks, but are expensive to compute. Looser relaxations are cheaper but bring limited information, resulting in fewer landmarks [Keyder *et al.*, 2010]. This generates an obvious trade-off highlighting the importance of choosing the right relaxation.

Possible Achievers for Simple Numeric Conditions

Simple numeric conditions are linear numeric inequalities affected by only constant increase/decrease actions.¹ For this reason, they match the numeric planning fragment we are interested in. As shown by Scala *et al.* [2016a], possible achievers of simple numeric conditions can be obtained using a parametric regression operation:

¹Note that effects can be considered constant if their state dependent nature is ignored. Although possible and computationally feasible, this can make relaxation-based heuristics highly inadmissible in general [Scala *et al.*, 2016a].

Definition 1. Let $c = (\sum_{x \in X} w_{x,c}x) + w_{n,c} \triangleright 0$, with $\triangleright \in \{\leq, <, >, \geq\}$, be a simple numeric condition. The m -times regression $c^{r(a,m)}$ of c through action a is:

$$c^{r(a,m)} \equiv \sum_{x \in X} w_{x,c}(k_{x,a}m + x) + w_{n,c} \triangleright 0 \quad (1)$$

where $k_{x,a}$ is the constant additive effect of a on x .

We say that a is a possible achiever of c in a state s if there exists an m such that $s \models c^{r(a,m)}$. All the other elements of (1) are coefficients extracted from the action model [Scala *et al.*, 2016a]. In the simple numeric condition case, possible achievers can be detected statically by looking at the relational operator (\triangleright), the action coefficients and the targeted condition. This results from the regression being linear in time, and hence, having a constant derivative. Yet, the number m of repetitions needed to achieve the condition is a state-dependent information.

In the numeric h^1 's family of heuristics [Scala *et al.*, 2016a], possible achievers over-approximate reachable numeric conditions, and have been used to restrict regression towards the actual relevant actions. This induces a computable relaxation of the problem. These possible achievers represent necessary but not sufficient conditions to achieve a numeric condition; a possible achiever ignores negative effects on other conditions that might be necessary to achieve - though it still captures interesting internal conflicting effects for multi-variables inequalities. Interestingly, as for the propositional case, possible achievers are also a necessary (though not sufficient) condition for an action to be considered a landmark for the problem. Next section defines numeric landmarks; then we show how to compute and use them.

We use the term numeric achiever (NA) for an action that is a possible achiever of a numeric condition and propositional achiever (PA) when it sets to true the value of a boolean variable. An action can be both an NA of a (numeric) condition, and a PA of another (propositional) condition.

Numeric Landmarks and Hybrid Action Landmarks

A numeric landmark resembles a propositional landmark, yet it extends its interpretation to the numeric case.

Definition 2 (Numeric Landmark). A numeric condition $c : \xi \triangleright k$ is said to be a numeric landmark if any plan solving the problem induces a trajectory of states S such that $\exists s \in S$ such that $s \models c$.

A numeric landmark can be geometrically understood as defining a region in the metric space that the plan trajectory has to hit at some step of the execution. Goals and initial state atoms are landmarks by definition.

An action that must be present in any plan is called an action landmark. We call these hybrid (action) landmarks, since they may be justified by both propositional or numeric landmarks. A landmark is called causal if it belongs to the precondition of some hybrid action landmark or to the goal set.

Numeric planning is more expressive than propositional planning. For this reason, a complete numeric and hybrid action landmark discovery procedure is not tractable. To overcome this problem, we approximate the landmarks set using

the landmarks of the underlying subgoal relaxation given by h_{hbd}^{max} , and defined by Scala et al. [2016a]. The intuition is that landmarks can be seen as *necessary* subgoals. Let s and G be our initial state and goal condition, respectively, a condition c (either numerical or propositional) is a landmark if at least one of the following holds: i) c is in the initial state, ii) c is a goal iii) the removal of any action preconditioned on c makes $h_{hbd}^{max}(s, G) = \infty$.

To compute landmarks we propose a hybridized AND/OR Landmark formulation [Keyder et al., 2010]; then we show how the extracted landmarks can be used to infer what we call disjunctive and additive hybrid action landmarks.

AND/OR Graph for numeric planning

An AND/OR Graph is a directed graph whose vertices are partitioned into AND nodes (V_{and}) and OR nodes (V_{or}), with edges only between AND and OR nodes or vice-versa.

In planning, AND/OR graphs have been used to compactly represent relaxed state spaces [Keyder et al., 2010]: V_{and} contains a node for each action (including a dummy initial action having empty preconditions and the initial state as its effect), and V_{or} contains a node for each proposition. There is an edge (a, b) if either a represents an action achieving proposition b , or if a is a proposition appearing in the precondition of action b . The AND/OR graph constructed this way nicely captures the semantics of the delete-free relaxation.

The extension to numeric planning is straightforward. V_{or} is extended with a node for each numeric condition that appears in problem; V_{and} is unchanged. Given an action a and a numeric condition b , there is an edge (a, b) if a is a possible achiever (NA) of b , as defined above. There is an edge (b, a) if the precondition of a contains b . In this case, edges can represent not only the single execution of an action, to achieve a propositional condition, but also the repeated execution necessary to achieve a numeric condition, as captured by the closed form of Eq. 1. The resulting encoding has only a single copy of each relevant action, also for the numeric part. This interpretation allows us to focus on the qualitative implications that the numeric structure has on reachability, so that, similarly to the propositional case, we can avoid the explicit layering of relaxed states in a traditional planning graph-like formulation [Hoffmann, 2003; Gerevini et al., 2008; Coles et al., 2010; Coles and Coles, 2011]. Most importantly, it enables the (almost) off-the-shelf use of Keyder et al.’s landmark discovery procedure for the intertwined inference of propositional and numeric landmarks.

Landmarks Extraction

Landmarks for the relaxed problem are those OR nodes of the associated AND/OR graph that lie on the intersection of all possible *justifications*, or paths to the goal nodes. This can be tackled using the following recurrence relation [Keyder et al., 2010]²:

$$LM(v) = \begin{cases} \{v\} \cup \bigcap_{\langle v', v \rangle \in E} LM(v') & \text{if } v \in V_{OR} \\ \bigcup_{\langle v', v \rangle \in E} LM(v') & \text{if } v \in V_{AND} \end{cases} \quad (2)$$

The set of propositional and numeric landmarks is given by: $LM(G) = \bigcup_{v \in G} LM(v)$. The intent of the intersection in the first line of Eq. 2, is to infer the strongest implicant of the landmarks of each of achievers (PAs or NAs) of a condition, i.e., a set of conditions that are implied by each of the sets of conditions $LM(v')$. In the propositional case this operation corresponds to intersecting these sets. Taking the intersection, treating numeric conditions as syntactic atoms, is valid also in the hybrid AND/OR graph. However, a stronger implicant can be found by examining the metric structure of the conditions. We consider numeric conditions pairwise: comparing two numeric conditions c_1 and c_2 , if one implies the other, that is, if $s \models c_1 \Rightarrow s \models c_2$ holds for every state s , then the weaker (implied) condition, c_2 , is clearly the strongest implicant of the two.

This dominance relation can be checked syntactically. If the coefficient vectors of the two conditions are linearly dependent (that is, one can be obtained by scaling the other), then c_1 implies c_2 if and only if after scaling the right-hand side of c_1 is greater than that of c_2 in the direction of the inequality. For example, consider the two sets of conditions $\{(y > 5)\}$ and $\{(2y > 9)\}$. Since they are syntactically different, set intersection yields the empty set. However, since $y > 5$ is equivalent to $2y > 10$ it implies $2y > 9$, so the implicant is the weaker condition ($2y > 9$).

In the general case, the strongest implicant of two sets of linear inequalities that is also a set of linear inequalities is the convex hull of the regions in the metric space defined by the disjuncts [Balas, 1998]. This characterization could potentially allow us to infer more landmarks, though at the cost of increased computational complexity. Our current implementation is therefore limited considering single conditions pairwise, as explained above. We leave the study of the more complex forms of reasoning, and its computational complexity implications, as a future work.

The union of numeric condition sets can also be refined analogously, leaving in this case only the stricter numeric condition. While this does not affect the *information* extracted, it may remove redundant numeric conditions.

The combined propositional and numeric landmark discovery enjoys three important properties studied by Keyder et al. [2010] in the propositional case: the system of equations $LM(\cdot)$ admits only one maximal solution (with respect to set inclusion); the resulting landmarks set contains all the causal landmarks w.r.t. the subgoal relaxation; all the landmarks are sound. The first two follow from the relationship between actions and numeric conditions being captured *symbolically* in the AND/OR graph construction. Thus, Keyder’s et al. Theorem 1 still applies. Soundness follows from the fact that edges have relaxed any negative effect between conditions, thus all the landmarks found using the procedure are also landmarks for the non-relaxed problem. Of course, there may be landmarks in the problem that are not landmarks of

²Note here we use a slightly different formulation where the explicit initial state is substituted by a dummy action. The recurrence reaches a base case when the initial dummy action is selected.

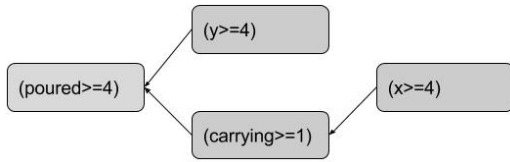


Figure 1: An excerpt of the landmark’s DAG generated for GARDENING domain. In order to pour the plant ($poured \geq 4$) the agent has to carry some water ($carrying \geq 1$), and at the same time reach a particular position in the map ($x \geq 4$ and $y \geq 4$). These are required for the plant to be poured, and to reach the tap to load the necessary amount of water.

the relaxation, and thus not identified by the procedure. This is way more complicated in the numeric case, since the number of conditions that can be expressed on the problem is infinite. Note here that we not only find all the causal landmarks for the relaxation, but because the intersection is sensitive to the metric structure of the problem, we can approximate some *disjunctive* landmark as well.

The existence of a unique maximal solution guarantees computability of the recurrence relation via a polynomial label-correcting algorithm. Yet the computation, although polynomial in the number of conditions, can be expensive, despite the use of efficient data-structures and looking only at those actions that are relaxed reachable. We will study its impact experimentally.

Orderings

As by-product of this novel AND/OR graph interpretation and Eq. 2, *orderings* among landmarks can also be extracted. A natural order \prec_n between two landmarks c_0, c_1 , ($c_0 \prec_n c_1$) establishes that any valid plan needs to satisfy c_0 first, and then c_1 . This can be inferred looking at whether $c_0 \in LM(c_1)$. Other classic forms of orderings seem however more difficult to characterize in the numeric setting and probably require a more profound re-interpretation. At the moment we can also support greedy necessary ordering \prec_{gn} (as defined by Keyder et al. [2010]), under the assumption that, given $c_0 \prec_{gn} c_1$, c_0 is a propositional landmark.

The validity of such ordering relationships is preserved by construction, since, although these conditions have a different semantics, the kind of reasoning performed at this level is, again, completely qualitative. Figure 1 shows an example of numeric landmarks and natural orderings inferred in the GARDENING domain [Francès and Geffner, 2015].

Disjunctive/Additive Hybrid Landmarks

Quantitative information can also be captured from the AND/OR graph. It is in fact sufficient to store for each possible way of achieving a condition, information regarding the number of times actions need to be repeated in order to achieve such a condition (i.e., using Eq. 1). This quantitative information can be used to obtain an admissible estimate by minimising over the number of repetitions needed for all the NAs/PAs. To account for this optimisation problem systematically we propose a generalised form of cost partitioning

by Karpas and Domshlak [2009] that considers numeric and propositional conditions altogether. The formulation contains a linear constraint for each numeric and propositional landmark found and minimizes the cost.

In particular: let O be the set of reachable actions (according to the relaxation), we denote with $O(c)$ the subset of O containing the NAs for a numeric condition c , and $O(p)$ the set of PAs for a propositional condition p . Given $a \in O(c)$ with $\lambda(a, c)$ we define the *contribution* of an action a to achieve a condition c , and with $\gamma(a)$ its cost. With $t(c)$ we define the target value for a condition c . In the propositional case the contribution of each action as well as the target value is 1. For the numeric case the contribution and the target value can be computed in closed-form on a per-state basis by manipulating Eq. 1. For the \geq and $>$ case³ $\lambda(a, c) = \sum_{x \in X} w_{x,c} k_{x,a}$ ($k_{x,a}$ is the result of the additive effects of a onto c and $w_{x,c}$ represents the weight of the action on the numeric condition); $t(c) = -(\sum_{x \in X} w_{x,c} x + w_{n,c})$ (this captures what is missing to satisfy condition c). Note that both $\lambda(a, c)$ and $t(c)$ are to be evaluated on the state.

For each action we introduce a counting variable $Y(a)$. Each $O(p/c)$ is de facto a *disjunctive/additive hybrid landmark*. In the propositional case this disjunction implies the existence of at least one of these actions in any plan solving the problem. In the numeric case the equation is a weighted summation of variables that needs to satisfy a given bound. Each plan has to contain actions such that the resulting constraint is satisfied. This is effectively an *additive hybrid landmark*. Note that an additive hybrid landmark captures actions repeatable an arbitrary number of times, so directly exploiting numeric structural properties of the problem. These hybrid landmarks can be encoded in an integer linear problem, from which we take its continuous relaxation⁴:

$$\begin{aligned}
 & \text{minimize } \sum_{a \in O} \gamma(a) Y(a) \\
 & \text{subject to } \sum_{a \in O(c)} \lambda(a, c) Y(a) \geq t(c), c \in LM(G) \\
 & Y(a) \in \mathbb{R}^{\geq 0}, \forall a \in O
 \end{aligned} \tag{3}$$

The result of this optimisation problem is our landmark-based heuristic: h_{hbd}^{lma} , where the subscript stands for *HyBriD*.

Theorem 1. *If $LM(G)$ contains only sound landmarks, the optimal cost induced by Eq. 3 gives us an admissible estimate, i.e., $h_{hbd}^{lma}(s) \leq h^*(s)$*

Proof Sketch. Easy by contradiction; assume the cost produced by the linear program was overestimating the real cost; this implies at least one constraint requiring some *unnecessary* action. However, this contradicts either the formulation of NAs/PAs or the soundness of the landmarks, or both. \square

³ \leq and $<$ can be reduced to \geq and $>$, respectively, by a simple mathematical manipulation.

⁴The continuous relaxation (LP) of a MIP has polynomial worst-case complexity, while its integral counter-part is NP-hard. The LP cost lower bounds the integer solution cost.

The resulting heuristic is both polynomial (inferring the landmark plus solving the Linear Program), and because admissible it also is safe-pruning: i.e., it gives infinity values only for actually unsolvable problems. Its computational cost trivially derives from Keyder et al.’ procedure [2010].

Getting More Landmarks

Numeric planning makes explicit structural and metric properties of the search space. Such information is crucial not only to infer repeated action applications within the heuristic, but also to constrain the solutions space by adopting redundant constraints. Redundant constraints have been used in scheduling [Getoor *et al.*, 1997], but also in planning [Scala *et al.*, 2016a] to tighten the relaxation. It is known that tighter relaxations lead to more landmarks [Keyder *et al.*, 2014], therefore our research question is: can we obtain more landmarks using redundant constraints? How useful is that?

Consider two numeric goals $c_1 : x \geq 10$ and $c_2 : y \geq 10$, an initial state $\langle x = 0, y = 0 \rangle$ and three unit cost actions $a : \langle \emptyset, \{x += 5, y -= 5\} \rangle$, $b : \langle \emptyset, \{x += 1\} \rangle$ $c : \langle \emptyset, \{y += 1\} \rangle$. This formulation generates two additive hybrid landmarks: $5Y(a) + Y(b) \geq 10$ and $Y(c) \geq 10$. The estimated cost is 12, given by executing c 10 times, and a twice.

For the purpose of studying the impact of redundant landmarks we adopt the same technique presented in Scala et al. [2016a]. For each set of conditions S (action precondition and goal), we augment S with a number of redundant constraints obtained looking at some of the *necessary* condition for the satisfaction of each pair of numeric conditions in S . Given $c_0 : \xi \triangleright k$ and $c_1 : \xi' \triangleright' k'$ the redundant numeric condition added is $red(c_0, c_1) : \xi + \xi' \text{ weaker}(\triangleright, \triangleright') k + k'$. The weaker operator intuitively selects the less strict comparator.

In our previous example the generated redundant constraint is $x + y \geq 20$ and requires the additive hybrid landmark $Y(b) + Y(c) \geq 20$: the overall estimate here is 20.

Redundant constraints approximate conjunctive landmarks in a very interesting way whilst preserving admissibility. This however comes at a price in our landmark inference and our LP heuristic, since they both have polynomial worst case complexity in the *number* of conditions. We study their benefit experimentally; the resulting heuristic is h_{hbd}^{lma+} .

Theorem 2. $h_{hbd}^{lma}(s)$ with redundant constraints is potentially more informative, whilst still admissible

$$h_{hbd}^{lma}(s) \leq h_{hbd}^{lma+}(s) \leq h^*(s)$$

Experiments

Our experimental evaluation compares the landmark heuristics with the hybrid \hat{h}_{hbd}^{rmax} heuristic [Scala *et al.*, 2016a], which is, to our knowledge, the only existing admissible heuristic for cost-optimal numeric planning.

We apply the heuristics in a standard forward state space planner using A^* , so as to minimise any noise in the evaluation. The cost-partitioning problem is solved with CPLEX 12.63. Ties are broken preferring higher g-values. All heuristics are computed on a per-state basis.⁵

⁵The implementation is part of the ENHSP planning sys-

The evaluation is done over four numeric, non-temporal domains that have featured in the International Planning Competition⁶, the SAILING and FARMLAND domains from our earlier paper [Scala *et al.*, 2016a], and the COUNTERS and GARDENING domains by Frances & Geffner [2015].

COUNTERS is a very simple numeric domain with N numeric variables (X_1, X_2, \dots, X_N) that have to be set to a specific configuration $(X_1 < X_2 \wedge X_2 < X_3 \wedge \dots \wedge X_{N-1} < X_N)$ through unit increase and decrease actions. Instances scale with N (from 2 to 40), and are split in three groups, with the initial values all 0, ordered inversely to the goal, or random. GARDENING models an agent that has to water a number N of plants located in a $M \times M$ grid, can carry a limited amount of water, and can be refill at a specific location. Moving, watering and refilling actions have only numeric effects. Instances scale in the number of plants (from 1 to 3) and grid size (4 to 10).

SAILING models N boats navigating in an unbounded 2-D space which must reach and rescue M persons scattered in the sea. To rescue a person, a boat must be inside a region defined by a set of inequalities. Instances scale with N and M (both from 1 to 5). FARMLAND models the problem of allocating manpower to farms whilst preserving a hard constraint on a metric measuring benefit. Each farm contributes differently to the overall metric. Instances scale with the number of farms (from 2 to 6), and workers at disposal (from 100 to 1000).

Table 1 shows a summary of the results. We make the following general observations: In almost all domains, h_{hbd}^{lma} and h_{hbd}^{lma+} require fewer node expansions than \hat{h}_{hbd}^{rmax} (we discuss one exception in detail in the next subsection). Where the reduction is substantial, this also translates into improved runtime or coverage. In the domains where using redundant constraints generates more landmarks, h_{hbd}^{lma+} also expands fewer nodes than h_{hbd}^{lma} . Among the IPC domains, ZENO TRAVEL is only one in which an action precondition (or goal) contains more than one numeric condition, and therefore the only one in which a redundant constraint is generated. However, in this domain it does not lead to more landmarks being found, and there is a runtime overhead.

We also tried disabling dominance testing in the intersection of numeric conditions, as described in the section on landmark extraction. This configuration is faster (both landmark heuristics solve one more instance in DEPOT) and finds slightly fewer landmarks only in ZENO TRAVEL.

In-Depth: SAILING. The SAILING domain highlights a significant difference between the landmark-based heuristics and \hat{h}_{hbd}^{rmax} . As we scale up the number of goals (persons to be rescued) with a single boat, the landmark heuristics, which add up actions needed to reach the precondition area of each rescue action, remains informative, while \hat{h}_{hbd}^{rmax} , which considers only the single most expensive goal requires much more search. However, when the number of boats becomes greater than one, each goal has multiple achievers (boats) with dis-

tem, open sourced at <https://bitbucket.org/enricode/the-enhsp-planner>. Experiments have run on a Xeon E5-2660 V3 with 8G of RAM.

⁶See <http://www.icaps-conference.org/index.php/Main/Competitions> for information on these domains.

	Coverage			Cpu-Time			Expanded Nodes			Average LMs	
	\hat{h}_{hbd}^{rmax}	h_{hbd}^{lma}	h_{hbd}^{lma+}	\hat{h}_{hbd}^{rmax}	h_{hbd}^{lma}	h_{hbd}^{lma+}	\hat{h}_{hbd}^{rmax}	h_{hbd}^{lma}	h_{hbd}^{lma+}	h_{hbd}^{lma}	h_{hbd}^{lma+}
COUNTERS(11)	2	2	3	<1	<1	<1	10.5	10.0	6.0	19.18	276.5
COUNTERS-INV(11)	2	2	2	<1	1.1	<1	160.5	133.5	8.5	19.18	270
COUNTERS-RND(33)	6	8	8	<1	<1	<1	5.5	5.5	5.5	9.72	140.4
GARDENING(63)	63	63	63	23.9	10.6	16.4	34442.3	19845.7	17398.7	7.71	22.06
SAILING (1 boat, 5 tot)	5	5	5	5.5	6.6	5.6	12571.6	3454.0	1307.0	9	12
SAILING (2-5 boats, 20 tot)	10	0	0	NA	NA	NA	NA	NA	NA	3	3
SAILING (1 boat 1-20 persons, 20 tot)	14	20	20	105.3	29.1	7.2	638449.8	24802.6	2032.0	31.5	42
FARMLAND(30)	30	30	30	6.2	6.6	6.4	4641.1	4327.2	1883.6	2.67	9.67
ZENO TRAVEL(23)	7	7	7	226.3	117.8	173.8	191702.4	36446.7	36446.7	18.78	18.78
ROVER(20)	4	4	4	6.1	4.1	3.9	3079.0	296.0	296.0	12.75	12.75
SATELLITE(20)	2	3	3	110.7	4.1	4.2	387929.5	522.5	522.5	16.6	16.6
DEPOT(20)	3	3	3	5.4	8.5	9.2	1956.5	746.5	746.5	26.36	26.36

Table 1: Results of the three heuristics across three domains. The timeout is 1800 secs. The run-time is in seconds. We report the average number of expanded nodes only among instances solved by all systems. In parenthesis is the total number of instances. Bold font is used to emphasize the best results

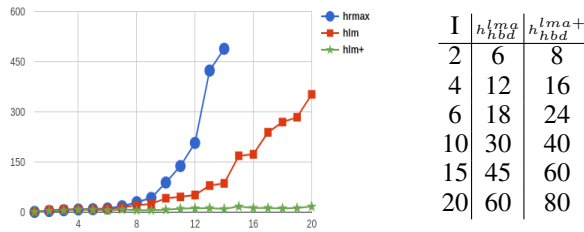


Figure 2: In depth analysis on the SAILING domain. X-axes scales with number of goals (persons to be rescued). Y-Axes reports Cpu-Time in seconds. The table shows the number of extracted landmarks without and with redundant constraints.

#C	OPL	Cpu-Time			Expanded Nodes			$h(s_0)$		
		\hat{h}_{hbd}^{rmax}	h_{hbd}^{lma}	h_{hbd}^{lma+}	\hat{h}_{hbd}^{rmax}	h_{hbd}^{lma}	h_{hbd}^{lma+}	\hat{h}_{hbd}^{rmax}	h_{hbd}^{lma}	h_{hbd}^{lma+}
2	1	0.2	0.2	0.2	2	2	2	1	1	1
3	3	0.2	0.2	0.2	4	4	4	2	2	2
4	6	0.5	0.6	0.7	19	18	10	2	3	4
5	10	1	1	1.1	231	192	109	2	4	6
6	15	3.8	4.0	3.4	3752	2482	1086	2	5	8
7	21	45.6	20.3	13.4	71980	39615	16216	2	6	10
8	28	TO	TO	511.2	NA	NA	265163	2	7	12
9	36	TO	TO	TO	NA	NA	NA	2	8	14

Table 2: COUNTERS: focus on small instances. Timeout 1800 secs. Last column: estimate in the initial state. OPL stands for optimal plan length, #C is the number of counters

joint preconditions, so that the landmark heuristics reduce to simply counting unsatisfied goals.

To show this, Table 1 separates results for subsets SAILING instances that scale in different ways. We also created additional instances with a single boat and more goals (up to 20). Figure 2 shows planner run-times with the three heuristics, as well the number of landmarks found, on this instance set only. With redundant constraints, the run-time increases nearly linearly with the number of goals.

In-Depth: COUNTERS. The COUNTERS instances used by Frances & Geffner [2015] scale up very rapidly, having 2, 4, 8, 12, 20, and 24 counters. This set is intended for evaluating satisficing planners, and does not reveal meaningful results for optimal planning (as can be seen in Table 1 only a

few problems are solved with any heuristic). We created additional instances, increasing problem size (number of counters) by steps of one from 2 up to 9, which is the smallest that no configuration solves. Table 2 shows results on this set. The landmark heuristic values without redundant constraints equate to h^+ . Redundant constraints tighten the underlying relaxation enabling more landmarks to be found, and therefore the heuristic value increases faster with problem size, though not as fast as the optimal plan length, which grows quadratically.

Conclusion

Landmarks have proven very useful in propositional planning, and recently the notion has also been applied to temporal propositional [Karpas *et al.*, 2015] and HTN [Elkawkagy *et al.*, 2012] planning. However, over a decade since first introduced in classical planning [Hoffmann *et al.*, 2004], this paper is the first to explain how landmarks and landmark extraction mechanisms can be developed for planning with both propositional and numeric variables.

We have applied our proposal to cost-optimal numeric planning, yet the presented mechanisms can be beneficial in other forms of intertwined numeric and propositional reasoning as well. Landmarks allow automatic identification of problem decompositions, as well as inducing (in)admissible heuristics. We believe this extension allows similar, perhaps even more powerful mechanisms. Numeric planning in fact makes explicit structural, metric information about the problem that is difficult to capture in a purely propositional representation. As future work we plan to improve landmark extraction with stronger reasoning about the disjunction of sets of numeric conditions, and extend the characterisation to support autonomous processes and temporal landmarks together.

Acknowledgements

This work is supported by ARC project DP140104219, “Robust AI Planning for Hybrid Systems”. The Australian Research Council through the ICT Centre of Excellence Program. We thank Da An for help in the implementation, and the anonymous reviewers for their useful comments.

References

- [Balas, 1998] Egon Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89(1-3):3–44, 1998.
- [Coles and Coles, 2011] Amanda Jane Coles and Andrew Coles. LPRPG-P: relaxed plan heuristics for planning with preferences. In *Proceedings of ICAPS*, 2011.
- [Coles et al., 2010] A. J. Coles, A. I. Coles, M. Fox, and D. Long. Forward-chaining partial-order planning. In *Proceedings of ICAPS*, 2010.
- [Della Penna et al., 2009] Giuseppe Della Penna, Daniele Magazzeni, Fabio Mercorio, and Benedetto Intrigila. Upmuphi: A tool for universal planning on PDDL+ problems. In *Proceedings of ICAPS*, 2009.
- [Elkawkagy et al., 2012] Mohamed Elkawkagy, Pascal Bercher, Bernd Schattenberg, and Susanne Biundo. Improving hierarchical planning performance by the use of landmarks. In *Proceedings of AAI*, 2012.
- [Eyerich et al., 2012] Patrick Eyerich, Robert Mattmüller, and Gabriele Röger. Using the context-enhanced additive heuristic for temporal and numeric planning. In *Towards Service Robots for Everyday Environments - Recent Advances in Designing Service Robots for Complex Tasks in Everyday Environments*, pages 49–64. 2012.
- [Fox and Long, 2003] Maria Fox and Derek Long. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res. (JAIR)*, 20:61–124, 2003.
- [Fox and Long, 2006] Maria Fox and Derek Long. Modelling mixed discrete-continuous domains for planning. *J. Artif. Intell. Res. (JAIR)*, 27:235–297, 2006.
- [Fox et al., 2011] Maria Fox, Derek Long, and Daniele Magazzeni. Automatic construction of efficient multiple battery usage policies. In *Proceedings of ICAPS*, 2011.
- [Francès and Geffner, 2015] Guillem Francès and Hector Geffner. Modeling and computation in planning: Better heuristics from more expressive languages. In *Proceedings of ICAPS*, 2015.
- [Gerevini et al., 2008] Alfonso Gerevini, Ivan Saetti, and Alessandro Serina. An approach to efficient planning with numerical fluents and multi-criteria plan quality. *Artificial Intelligence*, 172(8-9):899–944, 2008.
- [Getoor et al., 1997] Lise Getoor, Greger Ottosson, Markus P. J. Fromherz, and Björn Carlson. Effective redundant constraints for online scheduling. In *Proceedings of IAAI*, 1997.
- [Ghallab et al., 2004] Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Automated planning - theory and practice*. Elsevier, 2004.
- [Haslum et al., 2012] Patrik Haslum, John K. Slaney, and Sylvie Thiébaux. Minimal landmarks for optimal delete-free planning. In *Proceedings of ICAPS*, 2012.
- [Helmert and Domshlak, 2009] Malte Helmert and Carmel Domshlak. Landmarks, critical paths and abstractions: What’s the difference anyway? In *Proceedings of ICAPS*, 2009.
- [Helmert, 2002] Malte Helmert. Decidability and undecidability results for planning with numerical state variables. In *On the Role of Ground Actions in Refinement Planning*, pages 44–53, 2002.
- [Hoffmann et al., 2004] Jörg Hoffmann, Julie Porteous, and Laura Sebastia. Ordered landmarks in planning. *J. Artif. Intell. Res. (JAIR)*, 22:215–278, 2004.
- [Hoffmann, 2003] Jörg Hoffmann. The metric-ff planning system: Translating “ignoring delete lists” to numeric state variables. *J. Artif. Intell. Res. (JAIR)*, 20:291–341, 2003.
- [Karpas and Domshlak, 2009] Erez Karpas and Carmel Domshlak. Cost-optimal planning with landmarks. In *Proceedings of IJCAI*, 2009.
- [Karpas et al., 2015] Erez Karpas, David Wang, Brian C Williams, Patrik Haslum, et al. Temporal landmarks: What must happen, and when. In *Proceedings of ICAPS*, 2015.
- [Keyder et al., 2010] Emil Keyder, Silvia Richter, and Malte Helmert. Sound and complete landmarks for and/or graphs. In *Proceedings of ECAI*, 2010.
- [Keyder et al., 2014] Emil Ragip Keyder, Jörg Hoffmann, and Patrik Haslum. Improving delete relaxation heuristics through explicitly represented conjunctions. *J. Artif. Intell. Res. (JAIR)*, 50:487–533, 2014.
- [Löhr et al., 2012] Johannes Löhr, Patrick Eyerich, Thomas Keller, and Bernhard Nebel. A planning based framework for controlling hybrid systems. In *Proceedings of ICAPS*, 2012.
- [Pommerening et al., 2014] Florian Pommerening, Gabriele Röger, Malte Helmert, and Blai Bonet. Lp-based heuristics for cost-optimal planning. In *Proceedings of ICAPS*, 2014.
- [Scala et al., 2016a] Enrico Scala, Patrik Haslum, and Sylvie Thiébaux. Heuristics for numeric planning via subgoaling. In *Proceedings of IJCAI*, 2016.
- [Scala et al., 2016b] Enrico Scala, Patrik Haslum, Sylvie Thiébaux, and Miquel Ramírez. Interval-based relaxation for general numeric planning. In *Proceedings of ECAI*, 2016.