LM-Cut and Operator Counting Heuristics for Optimal Numeric Planning with Simple Conditions

Ryo Kuroiwa, Alexander Shleyfman, Chiara Piacentini, Margarita P. Castro, J. Christopher Beck
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Canada, ON M5S 3G8
rkuroiwa@mie.utoronto.ca, shleyfman.alexander@gmail.com, {chiarap, mpcastro, jcb}@mie.utoronto.ca

Abstract

We consider optimal numeric planning with numeric conditions consisting of linear expressions of numeric state variables and actions that increase or decrease numeric state variables by constant quantities. We build on previous research to introduce a new variant of the numeric $h^{\max}$ heuristic based on the delete-relaxed version of such planning tasks. Although our $h^{\max}$ heuristic is inadmissible, it yields a numeric version of the classical LM-cut heuristic which is admissible. Further, we prove that our LM-cut heuristic neither dominates nor is dominated by the existing numeric heuristic $h^{\max}_{\text{hbd}}$. Admissibility also holds when integrating the numeric cuts into the operator-counting (OC) heuristic producing an admissible numeric version of the OC heuristic. Through experiments, we demonstrate that both these heuristics compete favorably with the state-of-the-art heuristics: in particular, while sometimes expanding more nodes than other heuristics, numeric OC solves 19 more problem instances than the next closest heuristic.

Introduction

In this paper, we consider numeric planning with instantaneous actions. The presence of numeric state variables introduces a further degree of complexity over classical planning, making plan existence undecidable in general (Helmert 2002). Nevertheless, since the introduction of numeric variables in PDDL2.1 (Fox and Long 2003), several methods have been proposed to solve satisficing and optimal variants of numeric planning problems (Hoffmann 2003b; Shin and Davis 2005; Gerevini, Saetti, and Serina 2008; Eyerich, Mattmüller, and Röger 2009; Coles et al. 2013; Scala et al. 2016; Illanes and McIlraith 2017; Li et al. 2018; Scala, Haslum, and Thiébaux 2016; Scala et al. 2017; Aldinger and Nebel 2017; Piacentini et al. 2018a,b). We consider here optimal planning with so-called *simple* numeric conditions (where numeric variables can be increased or decreased by constant quantities and pre-conditions are inequalities involving linear expressions) and focus on the delete-relaxed version of such problems to develop heuristics.

First, we introduce a variation of the numeric $h^{\max}$ heuristic based on sub-goaling relaxations (Scala, Haslum, and Thiébaux 2016). Next, building upon our version of the relaxed $h^{\max}$, we extend the well-known classical LM-cut heuristic (Helmert and Domshlak 2009) to account for numeric effects and include it in the operator-counting framework (Pommerening et al. 2015). We prove that while our version of the relaxed $h^{\max}$ is inadmissible, our LM-cut and, therefore, operator counting heuristics are admissible and show the theoretical relationship among these heuristics.

The empirical evaluation indicates a good trade-off between informativeness and computational time, in particular, these heuristics favorably compete with several classes of state-of-the-art heuristics and overall achieve higher coverage. The experimental results over classical planning tasks with resources show the potential of explicitly modelling resources as numeric state variables, instead of using their propositional representation.

Preliminaries

Numeric Planning

We consider a fragment of numeric planning restricted to the STRIPS formalism (Fikes and Nilsson 1971) with the addition of numeric state variables. We focus our attention on a restricted class of numeric planning defined by Hoffmann (2003a), called the *restricted numeric task* (RT). Formally, RT is defined as a 5-tuple $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$, where $\mathcal{F}_p$ is a finite sets of facts and $\mathcal{N}$ is a set of numeric variables. Numeric variables $v \in \mathcal{N}$ have rational values in $\mathbb{Q}$. A state s is a tuple $\langle s_p, s_n \rangle$, where $s_p \subseteq \mathcal{F}_p$ is a set of facts and s_n is a full assignment over the variables in $\mathcal{N}$; $s[v]$ indicates the value of the variable $v \in \mathcal{N}$.[1]

Conditions can be either propositional $\psi \in \mathcal{F}_p$ or numeric, defined as $\psi : v \trianglerighteq w$, with $\trianglerighteq \in \{\geq, >\}$, $v \in \mathcal{N}$ and $w \in \mathbb{Q}$. The set of all numeric conditions is denoted by $\mathcal{F}_n$. A propositional condition $\psi \in \mathcal{F}_p$ is satisfied by the state s if $\psi \in s_p$. A numeric condition $\psi : v \trianglerighteq w \in \mathcal{F}_n$ is satisfied by s if $s[v] \trianglerighteq w$.

An action $a \in \mathcal{A}$ is a triplet $\langle \mathsf{pre}(a), \mathsf{eff}(a), \mathsf{cost}(a) \rangle$, where $\mathsf{pre}(a)$ are the preconditions, $\mathsf{eff}(a)$ the effects, and $\mathsf{cost}(a) \in \mathbb{R}^{0+}$ is the cost. Preconditions are defined as $\mathsf{pre}_p(a) \cup \mathsf{pre}_n(a)$, with propositional and numeric conditions, respectively. Effects are a triplet $\mathsf{eff}(a) =$

[1] For simplicity, we sometimes write $s = s_p \cup s_n$ as a minor abuse of notation.

$\langle \mathsf{add}(a), \mathsf{del}(a), \mathsf{num}(a)\rangle$, where $\mathsf{add}(a), \mathsf{del}(a) \subseteq \mathcal{F}_p$ are added and deleted facts, and $\mathsf{num}(a)$ is the set of additive numeric effects of the from $(v \mathrel{+}= k)$, where $v \in \mathcal{N}$ and $k \in \mathbb{Q}$. We assume that actions have at most one numeric effect on each numeric variable. We say that action a is applicable in state s if $s \models \mathsf{pre}_p(a) \cup \mathsf{pre}_n(a)$. The result of applying a in s is denoted by $s[\![a]\!] = \langle s'_p, s'_n\rangle$, with $s'_p = s_p \setminus \mathsf{del}(a) \cup \mathsf{add}(a)$, while for each $v \in \mathcal{N}$, $s'_n[v] = s_n[v]+k$ if $(v \mathrel{+}= k) \in \mathsf{num}(a)$, and $s'_n[v] = s_n[v]$ otherwise. We replace $\psi : v > w$ with $\psi' : v \geq w+\varepsilon$, where $\varepsilon > 0$ is a sufficiently small constant, assuming only $\geq$ conditions.

The goal conditions $G = G_p \cup G_n$ denote propositional and numeric conditions, respectively. The set of numeric conditions $\bar{\mathcal{F}}_n = \{\psi \in \mathcal{F}_n \mid \psi \in G_n \vee \exists a \in \mathcal{A} : \psi \in \mathsf{pre}_n(a)\}$ is called *active numeric facts*. Note that $\bar{\mathcal{F}}_n$ is finite. When a state s satisfies a condition ψ or a set of conditions Ψ, we write $s \models \psi$ and $s \models \Psi$, respectively. The goal condition G is a subset of $\mathcal{F}_p \cup \mathcal{F}_n$. We say that s_* is a *goal state* if $s_* \models G$.

An *s-plan* is an action sequence π that can be applied successively in s and results in a goal state $s_* \models G$. A plan for Π is an s_I-plan. The *cost of an s-plan* π is the sum of all its action costs and an *optimal s-plan* has minimal cost among all possible s-plans. The optimal cost of an s-plan is denoted by $h^*(s)$. The cost of Π is $h^*(s_I)$.

A *disjunctive fact landmark* $L_\mathcal{F} \subseteq \mathcal{F}_p \cup \mathcal{F}_n$ is a set of facts such that in the execution of any plan π, there is a state s such that $s \models \psi$ for some $\psi \in L_\mathcal{F}$. A *disjunctive action landmark* $L \subseteq \mathcal{A}$ is a set of actions such that set-wise it holds $L \cap \pi \neq \emptyset$, for every plan π for Π.

If a Π^{RT} does not have any numeric state variable ($\mathcal{N} = \emptyset$), we have a *classical* (STRIPS) *planning task* Π. A numeric variable $v \in \mathcal{N}$ is called a *resource* variable if it has a non-negative domain $[0, R_v]$, where $R_v \in \mathbb{Q}^{0+}$ is the maximum capacity, and v is affected only by actions When a numeric state variable $v \in \mathcal{N}$ has non-negative domain $[0, R_v]$, where $R_v \in \mathbb{Q}^{0+}$ is the maximum capacity and is affected only by actions which are applicable in s if $s[\![a]\!][v] \in [0, R_v]$, it is called *resource* variable. If all the numeric variables are resources and the effects are the type $v \mathrel{+}= k^a_v$ with $k^a_v < 0$ for all $a \in \mathcal{A}$, $v \in \mathcal{N}$, we have a resource-constrained planning task (RCP) (Nakhost, Hoffmann, and Müller 2012), while if there exists at least one action with $k^a_v > 0$, we have a planning task with resources (RP) (Wilhelm, Steinmetz, and Hoffmann 2018). RT generalizes RP by allowing numeric variables to have domain $\mathbb{Q}$ and numeric conditions of the form: $\psi : v \geq w^\psi_0$, with $w^\psi_0 \in \mathbb{Q}$.

Scala, Haslum, and Thiébaux (2016) introduce planning with simple numeric conditions (SC), an extension of RT where numeric preconditions are $\psi : \sum_{v \in \mathcal{N}} v \cdot w^\psi_v \geq w^\psi_0$, with $w^\psi_v, w^\psi_0 \in \mathbb{Q}$. Tasks with SC only are called SC tasks (SCT). They can be reduced to RT by introducing a new numeric variable for each SC and modifying numeric effects so that they change the variable by the net effects on the SC, as shown in the supplementary material (Kuroiwa et al. 2021).

Delete-free RT. We say that an RT is delete-free if for each action $a \in \mathcal{A}$ it holds that $\mathsf{del}(a) = \emptyset$ and all numeric effects are of the form $(v \mathrel{+}= k^a_v)$ with $k^a_v > 0$. The *support function* $\mathsf{supp} : \mathcal{F}_p \cup \mathcal{F}_n \rightarrow 2^\mathcal{A}$ for such tasks is defined to be $\mathsf{supp}(\psi) = \{a \in \mathcal{A} \mid \psi \in \mathsf{add}(a)\}$ if $\psi \in \mathcal{F}_p$ and $\mathsf{supp}(v \geq w_0) = \{a \in \mathcal{A} \mid v \mathrel{+}= k^a_v \in \mathsf{num}(a)\}$ if $v \geq w_0 \in \mathcal{F}_n$. The optimal cost of the delete-free RT is denoted by h^+.

The delete-relaxed version of an RT Π^{RT}, Π^+, is a delete-free RT, where the effects $\mathsf{del}(a)$ and $\forall (v \mathrel{+}= k^a_v) \in \mathsf{num}(a)$ if $k^a_v \leq 0$ are ignored for all $a \in \mathcal{A}$. The optimal cost of Π^+, h^+, is an admissible estimate of the optimal cost of Π^{RT}, since all plans for Π^{RT} are also plans for Π^+. In what follows, we consider heuristics for delete-relaxed versions of RTs.

LM-Cut in Classical Planning

Helmert and Domshlak (2009) introduce the LM-cut heuristic for classical planning and show that it produces excellent estimates of h^+ for many tasks. The heuristic is based on disjunctive action landmarks, computed as a cut in a *labelled weighted digraph* called justification graph.

A *labelled weighted digraph* is formally defined by a triplet $\mathcal{G} = \langle N, E, \mathsf{W}\rangle$, where N are the vertices of the graph, $E \subseteq N \times N \times A$ are labelled edges of the graph, where A denotes the label set, and $\mathsf{W} : E \rightarrow \mathbb{R}^{0+}$ is the weight function on edges. An interleaving sequence of vertices and labels $(n_0, a_0, n_1, \ldots, a_m, n_{m+1})$ (that can be viewed as a sequence of edges) is called a *path* if for each $i \in \{0, \ldots, m\}$ it holds that $(n_i, n_{i+i}; a_m) \in E$. When the labels are evident from the context we denote such path by $\mathsf{path}(n_0, n_{m+1})$ and say that n_{m+1} is *reachable* from n_0 if such a path exists. Given two disjoint sets $N_1, N_2 \subseteq N$, we define a *directed cut* to be $(N_1, N_2) = \{(n_1, n_2; a) \in E \mid n_1 \in N_1, n_2 \in N_2\}$. We also assume that $N_1 \cup N_2 = N$. The weights of the path π and the cut L are denoted respectively as

$$\mathsf{W}(\pi) = \sum_{e \in \pi} \mathsf{W}(e) \quad \text{and} \quad \mathsf{W}(L) = \min_{e \in L} \mathsf{W}(e).$$

Lastly, for a vertex $n_0 \in N$, we define the set of edges *incident to* n_0 as $\mathsf{in}(n_0) = \{(n, n_0; a) \in E \mid n \in N, a \in A\}$ and the *in-border* of $E' \subseteq E$ as $\partial^{\mathsf{in}}(E') = \{n_0 \in N \mid \mathsf{in}(n_0) \cap E' \neq \emptyset\}$.

The computation of the LM-cut heuristic is performed in rounds, leveraging the concept of cost partitioning (see Definition 1) to guarantee the admissibility of the heuristic. Starting with $h^{\text{LM-cut}}(s) = 0$, each round proceeds as follows:[2]

1. Compute the $h^{\max}$ values (Bonet and Geffner 2001) of all relevant facts. If $\max_{g \in G} h^{\max}(g) = 0$ return the computed heuristic value. If there is $g \in G$ such that $h^{\max}(g) = \infty$ return ∞.
2. Use the $h^{\max}$ values to construct a Justification Graph (JG) and using this graph compute a disjunctive action landmark L with nonzero cost c. Update $h^{\text{LM-cut}}(s) \mathrel{+}= c$.
3. Reduce the costs of all actions in L by c.
4. Go to Step 1, using the updated action costs.

[2]A formal definition of the computation can be found in the literature (Helmert and Domshlak 2009; Bonet and Helmert 2010).

LM-cut in RT Planning

We can use the exact same schema as described in the previous section to compute an admissible heuristic for RT planning. To prove that the heuristic remains admissible when considering numeric state variables, we need the following building blocks: (1) an assurance that the cost-partitioning concept is valid in RT, (2) an $h^{\max}$ heuristic that handles numeric conditions, and (3) a JG with numeric preconditions and effects. These building blocks and the appropriate proofs are discussed below.

Cost-Partitioning in RT Planning

In their works, Katz and Domshlak (2008) and Yang et al. (2008) independently proposed an approach to additively combine individual admissible heuristic estimates.

Definition 1. *Given a planning task Π, its **cost-partitioning** is a family of planning tasks $\{\Pi_i\}_{i=1}^n$ where each task differs from Π only by its cost function cost_i and it holds that $\forall a \in \mathcal{A} : \sum_{i=1}^n \mathsf{cost}_i(a) \leq \mathsf{cost}(a)$.*

Proposition 1. *Given a planning task Π, cost partitioning $\{\Pi_i\}_{i=1}^n$, and an admissible heuristic function h_i for each Π_i, the heuristic function $h(s) = \sum_{i=1}^n h_i(s)$ is admissible.*

Here we give a simplified version of the claim proposed by Katz and Domshlak. The proof of this, less general, claim is given in the supplementary material (Kuroiwa et al. 2021). Note that the proof does not rely on any formalism properties and depends only on the transition system. Therefore, Prop. 1 holds for numeric planning.

h^{max} in Numeric Planning

While the computation of $h^{\max}$ in classical planning (Bonet and Geffner 2001) can be done in polynomial time, with numeric state variables, the problem of calculating $h^{\max}$ is NP-hard even if we are restricted to RTs.

Let $h^{\max}(s) = \hat{h}(s, G)$ be defined as a maximal fixed-point of the following recursive equations: $\hat{h}(s, F) = \max_{\psi \in F} \hat{h}(s, \psi)$ for $F \subseteq \mathcal{F}_p \cup \mathcal{F}_n$ and

$$\hat{h}(s, \psi) = \begin{cases} 0 & \text{if } s \models \psi \\ \min\limits_{a \in \mathsf{supp}(\psi)} \hat{h}(s, \mathsf{pre}_\psi(a)) + \mathsf{cost}(a) & \text{otherwise.} \end{cases}$$

Here, $\mathsf{pre}_\psi(a)$ is $\mathsf{pre}(a)$ if $\psi \in \mathcal{F}_p$, or $\mathsf{pre}(a) \cup \{v \geq w - k_v^a\}$ if $\psi : v \geq w \in \mathcal{F}_n$. Note that $h^{\max}$ can be computed in a finite number of steps (Scala, Haslum, and Thiébaux 2016).

Proposition 2. *Given an* RT *and a state s, the problem of computing $h^{\max}$ is NP-hard (Kuroiwa et al. 2021).*

As in the classical case, the RT heuristic can be interpreted using both interval (Aldinger and Nebel 2017) and subgoaling relaxations (Scala, Haslum, and Thiébaux 2016), but because of its intractability, a further relaxation is needed. Scala et al. (2016) modified $h^{\max}$ by introducing a function $\mathsf{m}_a(s, \psi)$ that, intuitively, accounts for the number of times action a must be applied in state s to reach fluent ψ.

Definition 2. *Given a delete-free* RT, *a state s, a fact $\psi \in \mathcal{F}$, and action $a \in \mathsf{supp}(\psi)$, We define m_a as*

$$\mathsf{m}_a(s, \psi) = \begin{cases} 0 & \text{if } s \models \psi \\ \frac{w - s[v]}{k^a} & \text{if } s \not\models \psi \wedge \psi : v \geq w \in \mathcal{F}_n \\ 1 & \text{otherwise,} \end{cases}$$

where $k^a \in \mathbb{Q}^{0+}$ is the numeric effect of action a on variable v, i.e., $(v \mathrel{+}= k^a) \in \mathsf{num}(a)$. If $a \notin \mathsf{supp}(\psi)$, we say $\mathsf{m}_a(s, \psi) = \infty$.

This definition allows us to restrict our fluents to the set of active numeric facts, $\bar{\mathcal{F}}_n$. Note that in contrast to $\mathcal{F}_n$, $\bar{\mathcal{F}}_n$ is a finite set. This notation allows us to define another version of $h^{\max}$ that we denote by $h^{\max}_{\mathsf{cri}}$.[3]

Definition 3. *Given an* RT *and a state s, we define heuristic function $h^{\max}_{cri}(s) := h^{\max}_{cri}(s, G)$ as follows. As always, for a set of facts $F \subseteq \mathcal{F}_p \cup \mathcal{F}_n$:*

$$h^{\max}_{cri}(s, F) = \max_{\psi \in F} h^{\max}_{cri}(s, \psi).$$

For a fact $\psi \in \mathcal{F}_p \cup \mathcal{F}_n$, $h^{\max}_{cri}(s, \psi) = 0$ if $s \models \psi$, or otherwise

$$h^{\max}_{cri}(s, \psi) = \min_{a \in \mathsf{supp}(\psi)} h^{\max}_{cri}(s, \mathsf{pre}(a)) + \mathsf{m}_a(s, \psi)\mathsf{cost}(a).$$

Unfortunately, as we show in Ex. 1, $h^{\max}_{\mathsf{cri}}$ is not an admissible heuristic. In contrast, in their work, Scala et al. (2016) propose another level of relaxation, decoupling preconditions and numeric effects of actions, that assures the admissibility of their $h^{\max}$ version (i.e., $h^{\max}_{\mathsf{hbd}}$).

Definition 4. *Given an* RT *and a state s, the heuristic function $h^{\max}_{hbd}$ is defined exactly as $h^{\max}_{cri}$, except when $\psi \in \mathcal{F}_n$ is a numeric fact, in this case*

$$h^{\max}_{hbd}(s, \psi) = \min_{a \in \mathsf{supp}(\psi)} h^{\max}_{hbd}(s, \mathsf{pre}(a)) + \min_{a \in \mathsf{supp}(\psi)} \mathsf{m}_a(s, \psi) \cdot \mathsf{cost}(a).$$

Note that while these definitions address all subsets of $\mathcal{F}_p \cup \mathcal{F}_n$, in practice we need to compute $h^{\max}_x$ only for the fluents in the finite set $\mathcal{F}_p \cup \bar{\mathcal{F}}_n$, which allows us to compute both $h^{\max}_{\mathsf{cri}}$ and $h^{\max}_{\mathsf{hbd}}$ in polynomial time in the size of the RT. It is also important to note that for any state s in a given RT, $h^{\max}_{\mathsf{hbd}}(s, \psi') \leq h^{\max}_{\mathsf{cri}}(s, \psi')$ because $h^{\max}_{\mathsf{hbd}}$ decouples preconditions and effects of actions that have numeric fluents, while $h^{\max}_{\mathsf{cri}}$ does not.

The Justification Graph and LM-cut

To follow the scheme of Helmert and Domshlak (2009) for the numeric version of LM-cut, we need to construct a JG using our $h^{\max}$ estimates.

Helmert and Domshlak (2009) defined a JG to have a singleton goal set and all operators to have exactly one precondition, one add effect, and no delete effects. This defines a directed weighted graph $\mathcal{G}$ whose vertices correspond to the facts of the planning task and labeled weighted edges correspond to actions. In the classical case, the weight of the shortest path from a vertex representing some fact in s to another fact ψ, corresponds to the $h^{\max}(s, \psi)$ value, hence the graph is said to *justify* $h^{\max}$.

To meet the requirement of having exactly one precondition, the $h^{\max}$-values need to be preserved. Intuitively speaking, the precondition choice function defined by Bonet

[3]The notation "cri" is chosen since this version of $h^{\max}$ admits the critical path value in our version of the justification graph.

and Helmert (2010) maps each action to one of its preconditions, which in classical planning is done via maximizing $h^{\max}$. Thus, for $h^{\max}_{\text{cri}}$, we can write as follows.

Definition 5. *Given an* RT*, a* **precondition choice function** $\mathsf{pcf}_{cri} : \mathcal{S} \times \mathcal{A} \to \mathcal{F}_p \cup \bar{\mathcal{F}}_n$ *is the function that satisfies the condition*

$$\mathsf{pcf}_{cri}(s, a) \in \operatorname*{argmax}_{\psi \in \mathsf{pre}(a)} h^{\max}_{cri}(s, \psi).$$

For actions with no precondition, we add an artificial fact denoted by $\emptyset$. If $\mathsf{pre}(a) = \emptyset$, we write $\mathsf{pcf}_{cri}(s, a) = \emptyset$.

In contrast to $h^{\max}_{\text{cri}}$, $h^{\max}_{\text{hbd}}$ decouples numeric effects of an action from its preconditions. To make $\mathsf{pcf}_{\text{hbd}}$ justify $h^{\max}_{\text{hbd}}$, $\mathsf{pcf}_{\text{hbd}}$ is defined to map each action/effect pair $\langle a, \psi\rangle$, not each action, to one of the preconditions of some $\hat{a}$ in $\mathsf{supp}(\psi)$. Thus, it is possible that $\mathsf{pcf}(s, a, \psi) \neq \mathsf{pcf}(s, a, \psi')$ for $\psi \neq \psi'$ and $\mathsf{pcf}(s, a, \psi) \notin \mathsf{pre}(a)$. The result is a precondition choice function defined as follows:

$$\mathsf{pcf}_{\text{hbd}}(s, a, \psi) \in \operatorname*{argmax}_{\psi' \in \mathsf{pre}(\hat{a})} h^{\max}_{\text{hbd}}(s, \psi'),$$

where $\hat{a} = a$ if $\psi \in \mathcal{F}_p$, or otherwise

$$\hat{a} \in \operatorname*{argmin}_{a' \in \mathsf{supp}(\psi)} h^{\max}_{\text{hbd}}(s, \mathsf{pre}(a')).$$

Our definitions of JG for $h^{\max}_{\text{cri}}$ and $h^{\max}_{\text{hbd}}$ differ only on the precondition chosen by the pcf version. We use the generalised term pcf as a short notation for both function $\mathsf{pcf}_{\text{cri}}$ and $\mathsf{pcf}_{\text{hbd}}$, where $\mathsf{pcf}_{\text{cri}}$ is extended with a mute variable ψ.

Definition 6. *Given an* RT *and a state s, the* **justification graph** *is a labelled weighted digraph $\mathcal{G} = \langle N, E, \mathsf{W}\rangle$ with*

1. *a set of vertices $N = \{n_\psi \mid \psi \in \mathcal{F}_p \cup \bar{\mathcal{F}}_n \cup \{\emptyset\}\}$;*
2. *a set of labeled edges:*

$$E = \hat{E} \cup \{(n_\psi, n_{\psi'}; a) \mid a \in \mathsf{supp}(\psi'), \psi = \mathsf{pcf}(s, a, \psi')\};$$

where for simplicity we include the following zero-cost edges

$$\hat{E} = \{(n_\emptyset, n_\psi; a_0) \mid s \models \psi\};$$

3. *and a weight function $\mathsf{W} : E \to \mathbb{R}^{0+}$ defined as*

$$(n_\psi, n_{\psi'}; a) \mapsto \mathsf{m}_a(s, \psi') \cdot \mathsf{cost}(a).$$

The function $\mathsf{lbl} : E \to \mathcal{A}$ is defined as $(n_\psi, n_{\psi'}; a) \mapsto a$.

The construction of the JG is at most quadratic in the size of the RT problem and, when no numeric conditions are present, the construction is identical to that of the JG used in classical planning formalisms such as STRIPS or FDR (Bäckström and Nebel 1995; Helmert 2006).

Admissibility

We now turn to the admissibility status of the four heuristics discussed above, as summarized in Table 1.

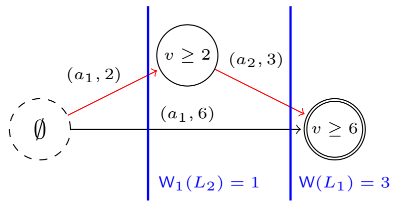

Figure 1: A JG for the RT in Ex. 1. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

$h^{\max}$ Heuristics. While Scala et al. (2016) proved the admissibility of $h^{\max}_{\text{hbd}}$, we show in Ex. 1 that $h^{\max}_{\text{cri}}$ is inadmissible.

Example 1. *Let $\langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G\rangle$ be an* RT *with $\mathcal{F}_p = \emptyset$ and $\mathcal{N} = \{v\}$. Let $s_I = \{v = 0\}$, $G = \{v \geq 6\}$, and $\mathcal{A} = \{a_1, a_2\}$, where*

action	pre	eff	cost
a_1	$\emptyset$	$v \mathrel{+}= 1$	1
a_2	$v \geq 2$	$v \mathrel{+}= 2$	1

Note, that in this case $\bar{\mathcal{F}}_n = \{v \geq 2, v \geq 6\}$. The JG of the first and second iterations can be seen in Figure 1, where the critical path is indicated in red and the landmarks of numeric LM-cut are denoted by blue lines. Thus, we have:

$$h^{\max}_{hbd}(s_I) = 3 < h^{\text{LM-cut}}_{cri}(s_I) = h^*(s_I) = 4 < h^{\max}_{cri}(s_I) = 5.$$

The example also shows that $h^{\max}_{\text{hbd}}$ does not dominate $h^{\text{LM-cut}}_{\text{cri}}$ even in the case where all variables are numeric. This complements the fact that $h^{\max}$ is dominated by $h^{\text{LM-cut}}$ in classical planning (Helmert and Domshlak 2009).

$h^{\text{LM-cut}}$ Heuristics. The JGs based on $\mathsf{pcf}_{\text{cri}}$ and $\mathsf{pcf}_{\text{hbd}}$ admit the property of justifying $h^{\max}_{\text{cri}}$ and $h^{\max}_{\text{hbd}}$, correspondingly. However, it is important to note the following.

Proposition 3. *The LM-cut heuristic based on the pcf_{hbd} JG, $h^{\text{LM-cut}}_{hbd}$, is inadmissible.*

The intricate example of inadmissibility of $h^{\text{LM-cut}}_{\text{hbd}}$ is shown in the supplementary material (Kuroiwa et al. 2021).

We now turn to the proof of admissibility of $h^{\text{LM-cut}}_{\text{cri}}$. Since $h^{\text{LM-cut}}_{\text{hbd}}$ is inadmissible, we construct the JG using the $\mathsf{pcf}_{\text{cri}}$ function. To simplify the notation, in what follows, we drop the subscript and write $h^{\text{LM-cut}}$ instead of $h^{\text{LM-cut}}_{\text{cri}}$.

Without compromising the admissibility, we restrict the vertices of $\mathcal{G}$ to the following: for each $n_\psi \in N$ it holds that there are facts $\psi' \in s$ and $\psi'' \in G$ such that there

h^y_x	max	LM-cut
hbd	✓(Scala et al. 2016)	✗ (Prop. 3)
cri	✗ (Ex. 1)	✓(Thm. 1)

Table 1: Admissibility chart: ✓– admissible, ✗– inadmissible. The h^y_x is the heuristic name where $y \in \{\max, \text{LM-cut}\}$ and $x \in \{\text{cri}, \text{hbd}\}$.

exist a $\mathsf{path}(n_{\psi'}, n_\psi)$ and a $\mathsf{path}(n_\psi, n_{\psi''})$. Note that both checks are polynomial, and assure that $h^{\max}_{\mathsf{cri}}(s, \psi) < \infty$. To establish a goal vertex for our computation we choose

$$g \in \operatorname*{argmax}_{\psi \in G} h^{\max}_{\mathsf{cri}}(s, \psi).$$

Definition 7. *Given an* RT*, a state* s*, and the JG* $\mathcal{G} = \langle N, E, \mathsf{W}\rangle$*, the* **goal zone** *of the graph* $\mathcal{G}$ *is the set of vertices that can reach the goal fact* g *at zero cost and is defined as:*

$$N^g = \{n_\psi \in N \mid \exists \mathsf{path}(n_\psi, n_g) : \mathsf{W}(\mathsf{path}(n_\psi, n_g)) = 0\}.$$

The **before-goal zone** *is a set of vertices that can be reached from the vertex* $n_\emptyset$ *without passing through* N^g*:*

$$N^0 = \{n_\psi \in N \mid \exists \mathsf{path}(n_\emptyset, n_\psi) : \mathsf{path}(n_\emptyset, n_\psi) \cap N^g = \emptyset\}.$$

Lastly, the **beyond-goal zone** *is* $N^b = (N \setminus N^g) \setminus N^0$.

The RT problem is unsolvable if there are no directed paths from $n_\emptyset$ to n_g. Whereas, if $n_\emptyset \in N^g$ there is a path from $n_\emptyset$ to n_g such that $\mathsf{W}(\mathsf{path}(n_\emptyset, n_g)) = 0$, which in turn means that heuristic value for s is 0.

Scala et al. (2017) defined landmarks for numeric planning. Following Helmert and Domshlak (2009), we show below how to extract such landmarks from JGs.

Lemma 1. *Assume an* RT *of a non-zero optimal cost. Let* $\mathcal{G}$ *be the JG corresponding to* Π^{RT}*, and let* L *be a directed cut in* $\mathcal{G}$ *that separates* $n_\emptyset$ *from* n_g*, such that* $\mathsf{W}(L) > 0$*. Then,*

1. $\partial^{\mathsf{in}}(L)$ *is a disjunctive fact landmark.*
2. $\mathsf{lbl}(L)$ *is a disjunctive action landmark.*

The proof of this claim is presented in the supplementary material (Kuroiwa et al. 2021). We also note that this property does not necessarily hold for a JG that is constructed based on $\mathsf{pcf}_{\mathsf{hbd}}$.

Now, we are ready to state and prove our main theoretical claim that states that numeric LM-cut is admissible.

Theorem 1. *Let* $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s, G\rangle$ *be a solvable* RT *with a non-zero optimal cost. Let* $\mathcal{G} = \langle N, E, \mathsf{W}\rangle$ *be the JG corresponding to* Π^{RT}*, and let* N^0, N^b *and* N^g *be before-, beyond- and goal zones, as defined above. Let* $L = (N^0, N^g \cup N^b)$ *be a directed cut in* $\mathcal{G}$*. We define the weight of the cut to be the weight of the minimal edge in the cut, i.e.*

$$\mathsf{W}(L) = \min_{e \in L} \mathsf{W}(e).$$

$h_1(s) = \mathsf{W}(L)$ *is admissible for* Π^{RT}_1*, where* Π^{RT}_1 *is a copy of* Π^{RT}*, with the augmented cost function* cost_1

$$\mathsf{cost}_1(a) = \begin{cases} \frac{\mathsf{W}(L)}{\mathsf{m}^L_a} & \text{if } a \in \mathsf{lbl}(L) \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathsf{m}^L_a = \min_{(n_\psi, n_{\psi'}; a) \in L} \mathsf{m}_a(s, \psi')$*, i.e., the minimum over all* m*'s of the same action edges in the cut.*

Proof. First, we show that $\mathsf{cost}_1(a) \leq \mathsf{cost}(a)$ for all $a \in \mathcal{A}$. The claim is clear for $a \notin \mathsf{lbl}(L)$, otherwise there is $(n_\psi, n_{\psi'}; a) \in L$ such that $\mathsf{W}(n_\psi, n_{\psi'}; a) = \mathsf{m}^L_a \cdot \mathsf{cost}(a)$, thus

$$\mathsf{cost}_1(a) = \frac{\mathsf{W}(L)}{\mathsf{m}^L_a} \leq \frac{\mathsf{m}^L_a \cdot \mathsf{cost}(a)}{\mathsf{m}^L_a} = \mathsf{cost}(a).$$

To finish the claim, all is left to show that the weight of L is an admissible estimate for the solution of Π^{RT}_1. Assume in contradiction that it is not, i.e., there is a plan π such that

$$\mathsf{cost}_1(\pi) < \mathsf{W}(L).$$

By Lem. 1 point 1, $\partial^{\mathsf{in}}(L)$ is a disjunctive fact landmark. Thus, there is at least one fluent in $\partial^{\mathsf{in}}(L)$ that is achieved by the plan π.[4] We denote by ψ_0 the first fluent in $\partial^{\mathsf{in}}(L)$ that is achieved by π. Note that $\mathsf{lbl}(\mathsf{in}(n_\psi)) = \mathsf{supp}(\psi)$; we write $L_\psi = L \cap \mathsf{in}(n_\psi)$. By construction of the JG, we have that

$$\min_{a \in \mathsf{lbl}(L_{\psi_0})} \mathsf{m}_a(s, \psi_0)\mathsf{cost}_1(a)$$

constitutes a lower bound on achieving the fluent ψ_0. Intuitively, $\mathsf{m}_a(s, \psi_0)$ is the (not necessarily integer) number of times that action a should be applied from s to achieve ψ_0. Thus, there is an action $\hat{a}_0 \in \mathsf{lbl}(L_{\psi_0})$ such that

$$\mathsf{m}_{\hat{a}_0}(s, \psi_0)\mathsf{cost}_1(\hat{a}_0) \leq \mathsf{cost}_1(\pi) < \mathsf{W}(L).$$

The fact that $\mathsf{m}^L_{\hat{a}_0} \leq \mathsf{m}_{\hat{a}_0}(s, \psi_0)$ allows us to conclude with the following contradiction

$$\mathsf{W}(L) \leq \mathsf{m}_{\hat{a}_0}(s, \psi_0)\frac{\mathsf{W}(L)}{\mathsf{m}^L_{\hat{a}_0}} \leq \mathsf{m}_{\hat{a}_0}(s, \psi_0)\mathsf{cost}_1(\hat{a}_0). \quad \square$$

Using Prop. 1 and Thm. 1, we finalize our claim.

Corollary 1 (Admissibility)**.** *The LM-cut heuristic for* RT *is admissible, and can be computed in polynomial time.*

Proof. Admissibility follows directly from Prop. 1 and Thm. 1. The computation of $h^{\max}_{\mathsf{cri}}(s, g)$ and the construction of the JG are both polynomial in RT and the cuts L are produced in polynomial time in the size of JG. Thus, if we show that the number of such cuts does not exceed $|\mathcal{A}|$ we can prove our claim. We show that for each L there is at least one action $a \in \mathsf{lbl}(L)$ where the cost is reduced to zero.

Let $(\psi, \psi'; a) \in L$ be the edge where $\mathsf{W}(L)$ achieves its minimum ($\mathsf{W}(L) > 0$). By definition of W we have that

$$\mathsf{W}(L) = \mathsf{W}(\psi, \psi'; a) = \mathsf{m}_a(s, \psi')\mathsf{cost}(a) = \mathsf{m}^L_a\mathsf{cost}(a).$$

Thus, the updated cost of $a \in \mathcal{A}$, $\mathsf{cost}_1(a)$, is

$$\mathsf{cost}(a) - \frac{\mathsf{W}(L)}{\mathsf{m}^L_a} = \mathsf{cost}(a) - \frac{\mathsf{m}^L_a\mathsf{cost}(a)}{\mathsf{m}^L_a} = 0. \quad \square$$

Finally, we show that $h^{\text{LM-cut}}$ does not dominate $h^{\max}_{\mathsf{hbd}}$.

Example 2. *Let* $\langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G\rangle$ *be an* RT *with* $\mathcal{F}_p = \{p, g\}$ *and* $\mathcal{N} = \{v\}$*. Let, the rest of elements in the tuple be* $s_I = \{v = 0\}$*,* $G = \{g\}$*, and* $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$*, where*

[4]This is a minor abuse of notation: fluent ψ corresponds to the node $n_\psi \in \partial^{\mathsf{in}}(L)$.

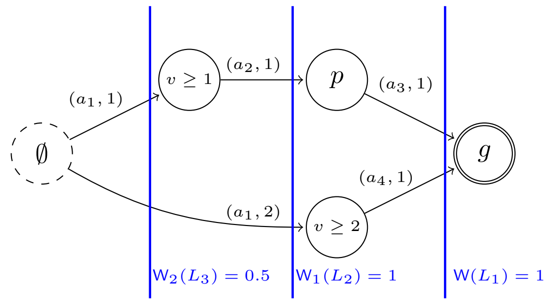

Figure 2: A JG for the RT in Ex. 2. The functions W, W$_1$, and W$_2$ denote the consequent cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

action	pre	eff	cost
a_1	$\emptyset$	$v \mathrel{+}= 1$	1
a_2	$v \geq 1$	p	1
a_3	p	g	1
a_4	$v \geq 2$	g	1

In this case we have $\bar{\mathcal{F}}_n = \{v \geq 1, v \geq 2\}$. The JG of all iterations can be seen in Figure 2, where the landmarks of numeric LM-cut are denoted by blue lines. Thus, we have

$$h^{\text{LM-cut}}(s_I) = 2.5 < h^{\max}_{hbd}(s_I) = h^*(s_I) = 3.$$

Hence, we have that $h^{\text{LM-cut}}$ does not dominate $h^{\max}_{hbd}$.

Using Ex. 1 and Ex. 2 we conclude the section with the following proposition.

Proposition 4. *$h^{\text{LM-cut}}$ and $h^{\max}_{hbd}$ are incomparable.*

Operator Counting

The operator-counting (OC) framework combines linear/integer programming (LP/IP) based heuristics using the optimal cost for the following problem as a heuristic value (Pommerening et al. 2014):

$$\begin{aligned} &\text{minimize} && \textstyle\sum_{a \in \mathcal{A}} \mathsf{cost}(a) X_a && (1)\\ &\text{subject to} && X_a \geq 0, \forall a \in \mathcal{A} && (2)\\ &&& C && (3) \end{aligned}$$

where X_a for all $a \in \mathcal{A}$ is a decision variable representing the number of occurrence of action a in a plan, and C is a set of OC constraints: linear inequalities over X_a such that for every plan π, $\forall a \in \mathcal{A}, X_a = \mathsf{count}(\pi, a)$ is a feasible solution where $\mathsf{count}(\pi, a)$ is the number of occurrences of action a in π. Since the optimal cost of the LP/IP problem is a lower bound of the cost of every plan, OC heuristics are admissible. Adding OC constraints does not remove feasible solutions for any plan and tightens the bound. Therefore, different types of OC constraints can be used together to improve the heuristic informativeness.

While the OC framework was originally proposed for classical planning, a recent work has applied it to numeric planning (Piacentini et al. 2018b). They introduced the state equation constraints (SEQ) (Bonet 2013) and the delete relaxation constraints (Imai and Fukunaga 2015) into numeric planning tasks with simple conditions (SCT).

LM-Cut for Operator-Counting Constraints

Given a disjunctive action landmark L, the landmark constraint is as follows (Bonet and Helmert 2010):

$$\sum_{a \in \mathsf{lbl}(L)} X_a \geq 1. \tag{4}$$

The above inequality is a valid OC constraint. In classical planning, the landmarks extracted by $h^{\text{LM-cut}}$ can be used to generate the landmark constraints (Pommerening et al. 2014). Here, we generalize this approach.

Theorem 2. *Given an RT, let L be a cut set obtained by the LM-cut heuristic. Let $\mathsf{count}(\pi, a)$ be the number of times action a appears in a plan π. The following holds for any π*

$$\sum_{a \in \mathsf{lbl}(L)} \frac{\mathsf{count}(\pi, a)}{\mathsf{m}^L_a} \geq 1. \tag{5}$$

Proof. Let π be a plan for the RT, and let $L = (N^0, N^g \cup N^b)$ be the cut in the JG. Recall that $\partial^{\mathsf{in}}(L)$ is a disjunctive fact landmark. Thus, there is at least one fluent in $\partial^{\mathsf{in}}(L)$ that is achieved by the plan π. Let ψ_0 be the first fluent in $N^g \cup N^b$ that is achieved by π, and let a_0 be the action in π that achieves ψ_0, i.e., $a_0 \in \mathsf{lbl}(L_{\psi_0}) \cap \pi$, where $\mathsf{lbl}(L_{\psi_0}) = \mathsf{supp}(\psi_0) \cap \mathsf{lbl}(L)$.

Next, note that if $m_{a_0}(s, \psi_0) \leq 1$, it holds that

$$1 \leq \frac{\mathsf{count}(\pi, a_0)}{\mathsf{m}_{a_0}(s, \psi_0)} \leq \frac{\mathsf{count}(\pi, a_0)}{\mathsf{m}^L_{a_0}} \leq \sum_{a \in \mathsf{lbl}(L)} \frac{\mathsf{count}(\pi, a)}{\mathsf{m}^L_a}.$$

Thus, assume that $\mathsf{m}_{a_0}(s, \psi_0) > 1$ and $\psi_0 \in \bar{\mathcal{F}}_n$ is a numeric fluent. Assume that ψ_0 is of the form $v \geq w_0$. For each $a \in \mathsf{supp}(\psi_0)$ it holds that $(v \mathrel{+}= k^a) \in \mathsf{num}(a)$. Since π achieves ψ_0 we can write

$$w_0 \leq s[v] + \sum_{a \in \mathsf{lbl}(L_{\psi_0})} k^a \cdot \mathsf{count}(\pi, a).$$

Now, subtract from both sides of the inequality $s[v]$ and subsequently divide by $w_0 - s[v] > 0$:

$$\begin{aligned} 1 \leq &\sum_{a \in \mathsf{lbl}(L_{\psi_0})} \frac{k^a}{w_0 - s[v]} \cdot \mathsf{count}(\pi, a) = \\ &\sum_{a \in \mathsf{lbl}(L_{\psi_0})} \frac{\mathsf{count}(\pi, a)}{\mathsf{m}_a(s, \psi_0)} \leq \sum_{a \in \mathsf{lbl}(L)} \frac{\mathsf{count}(\pi, a)}{\mathsf{m}^L_a}. \quad \square \end{aligned}$$

From Thm. 2, we derive the following OC constraint:

$$\sum_{a \in \mathsf{lbl}(L)} \frac{X_a}{\mathsf{m}^L_a} \geq 1, \tag{6}$$

where L is a cut obtained by the LM-cut heuristic.

Experimental Evaluation

In all the experiments, we evaluate the heuristics by using them in A^* search imposing a 30-minute time-limit and 4 GB memory limit on an Intel(R) Xeon(R) CPU E5-2620 @

2.00GHz processor. We implemented the heuristics in Numeric Fast Downward (NFD) (Aldinger and Nebel 2017)[5] using C++11 with GCC 7.5.0 on Ubuntu 18.04 and using CPLEX 12.10 as a mathematical programming solver.

Domains with simple numeric conditions are taken from the literature (Scala et al. 2016, 2017, 2020). We exclude ZENOTRAVEL because some conditions are not simple numeric conditions (Piacentini et al. 2018b). From COUNTERS, we exclude three instances that are in SMALLCOUNTERS. For SAILING, in addition to the original instances (Scala et al. 2016), we include the instances with a single boat (Scala et al. 2017), removing duplicates. Since multiple configurations solve all instances in FARMLAND, GARDENING, and SAILING, we also added satisficing versions of these domains (FARMLAND-SAT, GARDENING-SAT, and SAILING-SAT) excluding instances appearing in the optimal versions. A task is translated into an RT when computing the numeric LM-cut. We implemented the numeric LM-cut and LP/IP-based heuristics. For the numeric heuristics, we add the redundant constraints to the goal conditions and preconditions of actions in the same way as Scala et al. (2016).

Comparison with Propositional LM-Cut

To evaluate the benefit of considering numeric conditions in LM-cut, we compare the numeric and propositional versions. In the numeric domains, the propositional LM-cut ignores numeric conditions, i.e., all numeric conditions are achieved with zero cost when computing heuristic values.

We omit the domains COUNTERS, FARMLAND-SAT, and CHILDSNACK because both versions of LM-cut solve no instances. The results in the top of Table 2 show that the numeric version of $h^{\text{LM-cut}}$ outperforms the propositional.

In addition, we consider a set of domains containing resource variables from the optimal IPC track, translated into RP tasks (Wilhelm, Steinmetz, and Hoffmann 2018). If there are multiple versions for the same domain, we use the latest one. Search is performed on the original space, while the numeric heuristic is computed using the transformed task. The results of this comparison are shown in the bottom of Table 2. The translation into numeric version increases the coverage on three domains while reducing the number of the expanded states and search time on seven domains. However, the propositional version solves more instances than the numeric version on four domains.

Comparison with Numeric Heuristics

We compare the numeric LM-cut heuristic ($h^{\text{LM-cut}}$) with the following numeric heuristics: the interval relaxation based max heuristic h^{irmax} (Aldinger and Nebel 2017), the numeric max heuristic $\hat{h}^{\text{rmax}}_{\text{hbd+}}$ (Scala et al. 2020), the numeric landmark heuristic $h^{\text{lm+}}_{\text{hbd}}$ (Scala et al. 2017), and the generalized sub-goaling heuristic $h^{\text{gen}}_{\text{hbd}}$ (Scala et al. 2020). For $\hat{h}^{\text{rmax}}_{\text{hbd+}}$, $h^{\text{lm+}}_{\text{hbd}}$, and $h^{\text{gen}}_{\text{hbd}}$, in addition to our implementations in NFD, we evaluate the original implementations in ENHSP-19[6] using OpenJDK 11.0.9.1.

[5]https://github.com/Kurororoo/numeric-fast-downward

[6]https://sites.google.com/view/enhsp/

	propositional			numeric		
	c.	t.	e.	c.	t.	e.
SMALLCOUNTERS (8)	6	8.75	160932	**7**	**0.46**	**5738**
COUNTERS-INV (11)	**2**	0.04	1734	**2**	**0.00**	**34**
COUNTERS-RND (33)	6	0.01	476	**9**	**0.00**	**6**
FARMLAND (30)	11	21.39	219468	**30**	**0.05**	**456**
GARDENING (63)	**63**	7.78	198429	**63**	**1.57**	**18622**
GARDENING-SAT (51)	10	93.12	2308994	**12**	**15.39**	**163362**
SAILING (40)	9	92.17	2283679	**40**	**0.19**	**1619**
SAILING-SAT (40)	3	64.76	3085470	**14**	**0.98**	**23241**
DEPOTS (20)	**7**	**95.36**	**49827**	**7**	97.37	49873
ROVERS (20)	**4**	**6.91**	**125692**	**4**	7.44	130760
SATELLITE (20)	**2**	36.70	78620	**2**	**23.34**	**47351**
TOTAL (364)	123	-	-	**190**	-	-
ELEVAT-11 (20)	**18**	159.35	54559	**18**	**136.18**	**31038**
FREEC (80)	15	174.41	82292	**18**	**61.23**	**376**
MPRIME (35)	**22**	48.16	5091	**22**	**16.44**	**449**
MYSTRY (30)	**17**	68.89	2763	16	**16.12**	**46**
NOMYST-11 (20)	14	28.73	4868	**20**	**12.72**	**488**
OPENST-14 (20)	0	-	-	**3**	-	-
PARCPR-11 (20)	**13**	**11.83**	23101	**13**	21.50	**22969**
PATHWAY (30)	**5**	**14.03**	14621	**5**	38.60	**14609**
PIPEST (50)	**17**	**88.66**	**75427**	16	173.95	76620
PIPESNOT (50)	**12**	**76.52**	**64296**	6	466.06	64425
ROVERS (40)	**7**	**2.82**	**15767**	**7**	8.55	16258
TPP (30)	**6**	**0.97**	4935	**6**	6.80	**4403**
TRANSP-14 (20)	**6**	104.73	85366	**6**	**85.94**	**82886**
WOODWOR-11 (20)	**12**	170.20	78288	**12**	**123.26**	**19620**
ZENOT (20)	**13**	**39.53**	**8764**	12	122.38	**8764**
TOTAL (505)	177	-	-	**180**	-	-

Table 2: Coverage ('c.'), average time ('t.'), and # of states expanded ('e.') by the propositional and numeric LM-cut heuristics. The time and # of states are averaged over instances solved by both versions. The numeric domains are in the upper and the classical domains are in the lower half.

We show the experimental results in the left-hand side of Table 3. On FARMLAND-SAT, since instances are large, NFD runs out of memory when translating PDDL files to SAS+ files and does not solve any instance. Our LM-cut heuristic improves coverage by 17 tasks compared to the next best heuristic and attains the smallest run-time on 8 of the 11 domains, while expanding more states than at least one competitor in all domains except DEPOTS.

Comparison in the OC Framework

In classical planning, the combination of the LM-cut constraints and SEQ outperforms the individual components (Pommerening et al. 2014). To examine whether this is also the case with numeric planning, we evaluate the following OC heuristics using LP: $h^{\text{LM-cut}}_{\text{LP}}$ which uses the LM-cut constraints (Eq. (6)), $h^{\text{SEQ}}_{\text{LP}}$ which uses the numeric planning version of SEQ, and $h^{\text{LM-cut, SEQ}}_{\text{LP}}$ which uses both. In addition, we compare these heuristics with h^{c}_{IP}, an OC heuristic that uses IP with the delete-relaxation constraints and SEQ (Piacentini et al. 2018b). The results of this comparison are shown in the right-hand side of Table 3.

While $h^{\text{LM-cut}}_{\text{LP}}$ and $h^{\text{SEQ}}_{\text{LP}}$ are complementary on most of the domains, the coverage of $h^{\text{LM-cut, SEQ}}_{\text{LP}}$ is equal to the maxi-

	h^{irmax}	$\hat{h}^{\text{rmax}}_{\text{hbd+}}$	$\hat{h}^{\text{rmax}}_{\text{hbd+}}$	$h^{\text{lm+}}_{\text{hbd}}$	$h^{\text{lm+}}_{\text{hbd}}$	$h^{\text{gen}}_{\text{hbd}}$	$h^{\text{gen}}_{\text{hbd}}$	$h^{\text{LM-cut}}$	h^{c}_{IP}	h^{c}_{LP}	$h^{\text{SEQ}}_{\text{LP}}$	$h^{\text{LM-cut}}_{\text{LP}}$	$h^{\text{LM-cut, SEQ}}_{\text{LP}}$
implementation	N	E	N	E	N	E	N	N	N	N	N	N	N
domain	Coverage												
SmallCounters (8)	6	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**8**	**8**	**8**	7	**8**
Counters (8)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	4	5	**8**	0	**8**
Counters-Inv (11)	2	2	2	2	2	9	**11**	2	6	7	**11**	2	**11**
Counters-Rnd (33)	6	6	7	10	9	32	**33**	9	21	23	**33**	9	**33**
Farmland (30)	11	**30**	**30**	**30**	**30**	20	**30**	**30**	**30**	**30**	**30**	**30**	**30**
Farmland-SAT (20)	0	**7**	0	6	0	0	0	0	**0**	**0**	**0**	**0**	**0**
Gardening (63)	**63**	**63**	**63**	**63**	**63**	53	**63**	**63**	**63**	**63**	**63**	**63**	**63**
Gardening-SAT (51)	**12**	**12**	**12**	**12**	**12**	1	11	**12**	14	14	12	12	**15**
Sailing (40)	8	22	25	20	20	6	9	**40**	**40**	20	9	**40**	**40**
Sailing-SAT (40)	3	7	9	3	6	1	5	**14**	**24**	3	3	14	12
Depots (20)	5	4	5	4	3	1	1	**7**	1	2	6	**7**	**7**
Rovers (20)	**4**	3	**4**	3	**4**	1	2	**4**	2	**4**	**4**	**4**	**4**
Satellite (20)	1	1	1	1	1	0	1	**2**	1	1	1	**2**	**2**
Total (364)	121	164	165	161	157	131	173	**190**	214	180	188	190	**233**
domain	Time (s)												
SmallCounters (8)	10.43	1.76	0.87	2.75	2.00	25.55	2.98	**0.46**	0.92	0.16	**0.02**	37.96	0.03
Counters-Inv (11)	0.06	0.18	0.01	0.23	0.01	0.34	0.01	**0.00**	0.23	0.03	**0.01**	0.02	0.01
Counters-Rnd (33)	0.02	0.08	0.00	0.21	0.01	0.25	0.01	**0.00**	1.59	0.34	**0.04**	196.64	0.05
Farmland (30)	15.32	0.32	**0.04**	0.71	0.22	6.71	0.37	0.05	43.01	6.98	**1.30**	89.20	1.50
Gardening (63)	1.23	1.53	0.42	3.59	2.29	205.52	12.78	**0.29**	12.26	3.73	1.78	5.69	**0.31**
Gardening-SAT (51)	0.91	2.73	0.49	4.86	2.48	234.60	14.27	**0.31**	51.05	17.72	13.04	95.55	**1.05**
Sailing (40)	4.64	0.73	0.07	1.08	0.08	659.57	22.47	**0.04**	5.03	3.17	430.31	**0.18**	0.21
Sailing-SAT (40)	28.73	1.89	0.68	14.04	**0.23**	205.05	16.31	0.61	66.29	8.69	478.74	**2.09**	2.47
Depots (20)	0.01	0.23	0.01	0.68	0.15	82.07	12.65	**0.01**	35.83	0.76	0.06	**0.03**	0.04
Rovers (20)	1.10	1.14	0.99	1.99	7.72	382.17	323.81	**0.52**	233.33	47.33	6.31	**4.98**	7.10
Satellite (20)	-	-	-	-	-	-	-	-	8.37	13.75	10.09	0.26	**0.17**
domain	# States expanded												
SmallCounters (8)	118909	12665	12668	2904	2906	**2011**	2012	5738	**13**	**13**	**13**	93963	**13**
Counters-Inv (11)	1390	160	160	**8**	**8**	**8**	**8**	34	**8**	**8**	**8**	26	**8**
Counters-Rnd (33)	316	6	**5**	**5**	6	6	**5**	6	**15**	**15**	**15**	359847	**15**
Farmland (30)	217074	463	456	463	**421**	462	449	456	**441**	**441**	**441**	16854	**441**
Gardening (63)	16477	5724	5762	**3849**	4010	8628	5810	4042	**139**	140	6367	18249	235
Gardening-SAT (51)	8499	3444	3461	1839	**1802**	4958	3148	2254	420	**412**	27856	307511	576
Sailing (40)	130300	1130	1130	630	**62**	61980	31815	632	**90**	306	2283679	150	150
Sailing-SAT (40)	967917	15052	15052	15052	**175**	15052	15052	15052	**8544**	9032	3085470	8845	8845
Depots (20)	55	55	55	98	134	147	150	**29**	**17**	59	134	29	29
Rovers (20)	12716	3842	12680	370	12712	**348**	12629	16003	**7683**	**7683**	15982	18360	17540
Satellite (20)	-	-	-	-	-	-	-	-	**113**	1080	51366	274	**113**

Table 3: Coverage, average time in seconds, and # of states expanded by different numeric heuristics on simple-condition domains. 'N' and 'E' mean the implementations in NFD and ENHSP-19, respectively. The time and # of states are averaged over instances solved by all of h^{irmax}, $\hat{h}^{\text{rmax}}_{\text{hbd+}}$, $h^{\text{lm+}}_{\text{hbd}}$, $h^{\text{gen}}_{\text{hbd}}$, and $h^{\text{LM-cut}}$ in the left-hand side, and by all of the OC heuristics in the right-hand side. The coverage is highlighted in bold if it is the highest among its heuristics family (right- and left-hand sides of the table).

mum of $h^{\text{LM-cut}}_{\text{LP}}$ and $h^{\text{SEQ}}_{\text{LP}}$. Furthermore, on three domains, $h^{\text{LM-cut, SEQ}}_{\text{LP}}$ expands fewer states and finds solutions faster than both of the components. In terms of the total coverage, $h^{\text{LM-cut, SEQ}}_{\text{LP}}$ dominates all of the evaluated heuristics including h^{c}_{IP}. While h^{c}_{IP} expands fewer states than $h^{\text{LM-cut, SEQ}}_{\text{LP}}$, the latter is faster to compute. h^{c}_{LP}, the LP version of h^{c}_{IP}, is also slower than $h^{\text{LM-cut, SEQ}}_{\text{LP}}$, indicating that the delete-relaxation constraints are informative, but slow to compute.

Conclusion

This paper presents an extension of the LM-cut heuristic to numeric planning problems with simple numeric conditions. In order to obtain an admissible estimate, we introduce a new variant of $h^{\max}$ and theoretically analyze the effect of different relaxations both on $h^{\max}$ and $h^{\text{LM-cut}}$. Although our admissible version of $h^{\text{LM-cut}}$ does not show any theoretical dominance over the existing numeric heuristic $h^{\max}_{hbd}$, empirically it better approximates the plan cost, improving on the coverage. The strong performance is even more evident when combining LM-cut with SEQ constraints using the operator-counting framework, as it achieves state-of-the-art performance in most numeric domains. The explicit use of numeric state variables also produces more accurate LM-cut heuristic estimations in most cases.

References

Aldinger, J.; and Nebel, B. 2017. Interval Based Relaxation Heuristics for Numeric Planning with Action Costs. In *Proc. SOCS*, 155–156.

Bäckström, C.; and Nebel, B. 1995. Complexity Results for SAS$^+$ Planning. *Computational Intelligence* 11(4): 625–655.

Bonet, B. 2013. An Admissible Heuristic for SAS+ Planning Obtained from the State Equation. In *Proc. IJCAI*, 2268–2274.

Bonet, B.; and Geffner, H. 2001. Planning as Heuristic Search. *AIJ* 129(1): 5–33.

Bonet, B.; and Helmert, M. 2010. Strengthening Landmark Heuristics via Hitting Sets. In *Proc. ECAI*, 329–334.

Coles, A. J.; Coles, A.; Fox, M.; and Long, D. 2013. A Hybrid LP-RPG Heuristic for Modelling Numeric Resource Flows in Planning. *JAIR* 46: 343–412.

Eyerich, P.; Mattmüller, R.; and Röger, G. 2009. Using the Context-Enhanced Additive Heuristic for Temporal and Numeric Planning. In *Proc. ICAPS*, 130–137.

Fikes, R. E.; and Nilsson, N. J. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *AIJ* 2: 189–208.

Fox, M.; and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *JAIR* 20: 61–124.

Gerevini, A.; Saetti, A.; and Serina, I. 2008. An approach to efficient planning with numerical fluents and multi-criteria plan quality. *AIJ* 172(8-9): 899–944.

Helmert, M. 2002. Decidability and Undecidability Results for Planning with Numerical State Variables. In *Proc. AIPS*, 303–312.

Helmert, M. 2006. The Fast Downward Planning System. *JAIR* 26: 191–246.

Helmert, M.; and Domshlak, C. 2009. Landmarks, Critical Paths and Abstractions: What's the Difference Anyway? In *Proc. ICAPS*, 162–169.

Hoffmann, J. 2003a. The Metric-FF Planning System: Translating "Ignoring Delete Lists" to Numeric State Variables. *JAIR* 20: 291–341.

Hoffmann, J. 2003b. *Utilizing Problem Structure in Planning, A Local Search Approach*, volume 2854 of *LNCS*. Springer.

Illanes, L.; and McIlraith, S. A. 2017. Numeric Planning via Abstraction and Policy Guided Search. In *Proc. IJCAI*, 4338–4345.

Imai, T.; and Fukunaga, A. 2015. On a Practical, Integer-Linear Programming Model for Delete-Free Tasks and its Use as a Heuristic for Cost-Optimal Planning. *JAIR* 54: 631–677.

Katz, M.; and Domshlak, C. 2008. Optimal Additive Composition of Abstraction-based Admissible Heuristics. In *Proc. ICAPS*, 174–181.

Kuroiwa, R.; Shleyfman, A.; Piacentini, C.; Castro, M. P.; and Beck, J. C. 2021. Supplement for LM-cut and Operator Counting Heuristics for Numeric Planning with Simple Conditions: Counter Examples and Additional Proofs. https://tidel.mie.utoronto.ca/pubs/Kuroiwa_ICAPS2021_supplement.pdf.

Li, D.; Scala, E.; Haslum, P.; and Bogomolov, S. 2018. Effect-Abstraction Based Relaxation for Linear Numeric Planning. In *Proc. IJCAI*, 4787–4793.

Nakhost, H.; Hoffmann, J.; and Müller, M. 2012. Resource-Constrained Planning: A Monte Carlo Random Walk Approach. In *Proc. ICAPS*, 181–189.

Piacentini, C.; Castro, M. P.; Ciré, A. A.; and Beck, J. C. 2018a. Compiling Optimal Numeric Planning to Mixed Integer Linear Programming. In *Proc. ICAPS*, 383–387.

Piacentini, C.; Castro, M. P.; Ciré, A. A.; and Beck, J. C. 2018b. Linear and Integer Programming-Based Heuristics for Cost-Optimal Numeric Planning. In *Proc. AAAI*, 6254–6261.

Pommerening, F.; Helmert, M.; Röger, G.; and Seipp, J. 2015. From Non-Negative to General Operator Cost Partitioning. In *Proc. AAAI*, 3335–3341.

Pommerening, F.; Röger, G.; Helmert, M.; and Bonet, B. 2014. LP-Based Heuristics for Cost-Optimal Planning. In *Proc. ICAPS*, 226–234.

Scala, E.; Haslum, P.; Magazzeni, D.; and Thiébaux, S. 2017. Landmarks for Numeric Planning Problems. In *Proc. IJCAI*, 4384–4390.

Scala, E.; Haslum, P.; and Thiébaux, S. 2016. Heuristics for Numeric Planning via Subgoaling. In *Proc. IJCAI*, 3228–3234.

Scala, E.; Haslum, P.; Thiébaux, S.; and Ramírez, M. 2020. Subgoaling Techniques for Satisficing and Optimal Numeric Planning. *JAIR* 68: 691–752.

Scala, E.; Ramírez, M.; Haslum, P.; and Thiébaux, S. 2016. Numeric Planning with Disjunctive Global Constraints via SMT. In *Proc. ICAPS*, 276–284.

Shin, J.; and Davis, E. 2005. Processes and continuous change in a SAT-based planner. *AIJ* 166(1-2): 194–253.

Wilhelm, A.; Steinmetz, M.; and Hoffmann, J. 2018. On Stubborn Sets and Planning with Resources. In *Proc. ICAPS*, 288–297.

Yang, F.; Culberson, J.; Holte, R.; Zahavi, U.; and Felner, A. 2008. A General Theory of Additive State Space Abstractions. *JAIR* 32: 631–662.