Journal of Artificial Intelligence Research 75 (2022) 1477-1548 Submitted 06/2022; published 11/2022

The LM-Cut Heuristic Family for Optimal Numeric Planning with Simple Conditions

Ryo Kuroiwa RYO.KUROIWA@MAIL.UTORONTO.CA
Department of Mechanical and Industrial Engineering
University of Toronto
Canada

Alexander Shleyfman SHLEYFMAN.ALEXANDER@GMAIL.COM
The Department of Computer Science
Bar-Ilan University
Israel

Chiara Piacentini CHIARA@AUGMENTA.AI
Augmenta Inc
Canada

Margarita P. Castro MARGARITA.CASTRO@ING.PUC.CL
Department of Industrial Engineering and Systems
Pontificia Universidad Católica de Chile
Chile

J. Christopher Beck JCB@MIE.UTORONTO.CA
Department of Mechanical and Industrial Engineering
University of Toronto
Canada

Abstract

The LM-cut heuristic, both alone and as part of the operator counting framework, represents one of the most successful heuristics for classical planning. In this paper, we generalize LM-cut and its use in operator counting to optimal numeric planning with *simple conditions* and *simple numeric effects*, i.e., linear expressions over numeric state variables and actions that increase or decrease such variables by constant quantities. We introduce a variant of $h^{\max}_{\text{hbd}}$ (a previously proposed numeric $h^{\max}$ heuristic) based on the delete-relaxed version of such planning tasks and show that, although inadmissible by itself, our variant yields a numeric version of the classical LM-cut heuristic which is admissible. We classify the three existing families of heuristics for this class of numeric planning tasks and introduce the LM-cut family, proving dominance or incomparability between all pairs of existing max and LM-cut heuristics for numeric planning with simple conditions. Our extensive empirical evaluation shows that the new LM-cut heuristic, both on its own and as part of the operator counting framework, is the state-of-the-art for this class of numeric planning problem.

1. Introduction

The presence of numeric state variables in planning problems introduces an additional degree of complexity over classical planning, making plan existence undecidable in the general case (Helmert, 2002). Nevertheless, since the introduction of numeric variables in PDDL2.1 (Fox & Long, 2003), in addition to approaches to satisficing numeric planning (Hoffmann, 2003b; Shin & Davis, 2005; Gerevini, Saetti, & Serina, 2008; Eyerich, Mattmüller, & Röger, 2009; Coles, Coles, Fox, & Long, 2013; Scala, Ramírez, Haslum, & Thiébaux, 2016; Scala, Haslum, Thiebaux, & Ramirez, 2016b; Illanes & McIlraith, 2017; Li, Scala, Haslum, & Bogomolov, 2018), a number of exact techniques have appeared in the literature for restricted cases of numeric planning. One approach is model-based, compiling a numeric planning task to another problem such as mixed integer linear programming (Piacentini, Castro, Ciré, & Beck, 2018b) and optimization modulo theories (Leofante, Giunchiglia, Ábrahám, & Tacchella, 2020). Another approach is heuristic search, the main focus of this paper.

In the past decade, a number of heuristics have been developed for optimal numeric planning with *simple conditions*, that is, where numeric variables can be increased or decreased by constant quantities and where preconditions are inequalities involving linear expressions (Scala, Haslum, & Thiébaux, 2016a; Scala, Haslum, Magazzeni, & Thiébaux, 2017; Piacentini, Castro, Ciré, & Beck, 2018a; Piacentini et al., 2018b; Kuroiwa, Shleyfman, Piacentini, Castro, & Beck, 2021). All but one of these heuristics are delete-relaxation heuristics that are either an approximation of numeric $h^{\max}$ or (as we show in Section 7) a set of operator counting (OC) constraints that are used to estimate/compute h^* via linear/integer programming (LP/IP), respectively (Pommerening, Röger, Helmert, & Bonet, 2014; Piacentini et al., 2018b). Thus, the already existing heuristics for optimal numeric planning with simple conditions can be divided into three families:

- $h^{\max}$ Relaxations: h^{irmax} (Aldinger & Nebel, 2017) and $h^{\max}_{\mathrm{hbd}}$ (Scala et al., 2016), with the latter extended to $\hat{h}^{\mathrm{rmax}}_{\mathrm{hbd}+}$ (Scala et al., 2020);
- Operator-counting: $h^{\mathrm{lm}+}_{\mathrm{hbd}}$ (Scala et al., 2017), h^{c}_{IP}, h^{c}_{LP} (Piacentini et al., 2018b);
- Generalised Subgoaling: $h^{\mathrm{gen}}_{\mathrm{hbd}}$ (Scala et al., 2020).

Figure 1 shows the relationship of all existing admissible heuristics for numeric planning with simple conditions and the main contribution of this paper: the $h^{\text{LM-cut}}$ heuristic family. We show through theoretical and experimental analysis that our heuristic is the state-of-the-art for this class of planning problems. This paper is an extension of our conference paper on the LM-cut heuristic for numeric planning (Kuroiwa et al., 2021), with the following additional contributions:

1. We propose novel variants of numeric LM-cut heuristics, $h^{\text{LM-cut}}_{\mathrm{ir}}$, $h^{\text{LM-cut}}_{\mathrm{ir,m}}$, $h^{\text{LM-cut}}_{\mathrm{ir,m+}}$, $h^{\text{LM-cut}}_{\mathrm{cri,+}}$, and $\overline{h}^{\text{LM-cut}}_{\mathrm{cri}}$, with theoretical and experimental comparisons.
2. We provide a thorough empirical evaluation of new operator-counting heuristics combining existing constraints with the LM-cut constraints.
3. We examine the complexity of delete-free restricted numeric planning.

The rest of the paper is as follows. Section 2 defines the problem and the necessary background. Sections 3 and 4 evaluate the computational complexity of the heuristics h^+ and $h^{\max}$ for numeric planning with simple conditions. We show that, although computing

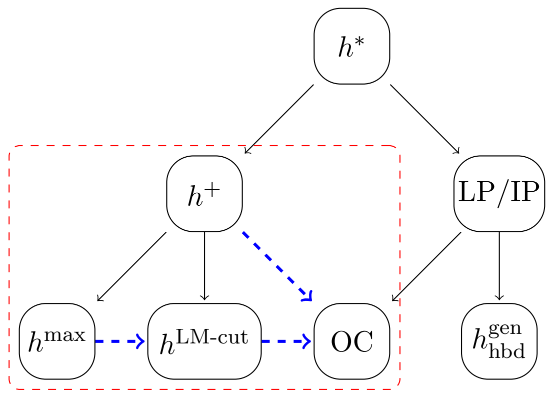


Figure 1: The schematic representation of heuristics for restricted numeric planning tasks. Each box corresponds to a family of heuristics, solid arrows represent relaxations, and dashed arrows represent information propagation. For example, the $h^{\text{LM-cut}}$ heuristic family is a relaxation of h^+, i.e., delete-relaxation, it uses information obtained by $h^{\max}$ heuristics, and can be used to provide strengthening information to the OC heuristics. The red dashed box represents the focus of this work.

these heuristics is NP-hard, checking whether there is a plan of bounded length in delete-free planning tasks still lies in NP.

In Section 5, we show that the well-known classical LM-cut heuristic (Helmert & Domshlak, 2009) based on $h^{\max}$ can be modified to account for numeric conditions and simple numeric effects. We identify the criterion of $h^{\max}$ relaxations that determines the admissibility of the resulting LM-cut heuristic variant. By doing so, we introduce an inadmissible variation of numeric $h^{\max}$ based on subgoaling relaxations (Scala et al., 2016a) that results in admissible LM-cut heuristics and grants us state-of-the-art performance. We also theoretically compare all admissible LM-cut versions and $h^{\max}$ relaxation variants presented in the literature; for each pair, we show either dominance or incomparability relations.

Section 6 presents a detailed empirical comparison of numeric and classical LM-cut variants in both classical and numeric settings, showing strong performance of our novel heuristics in both settings. In Section 7, we use the LM-cut heuristics to produce constraints for the operator-counting framework (Pommerening, Helmert, Röger, & Seipp, 2015). We compare our methods with all other operator-counting constraints in the literature. The empirical evaluation of the resulting admissible heuristics indicates a good trade-off between informativeness and computational time. These heuristics favorably compete with the state-of-the-art heuristics and overall achieve higher coverage. Lastly, Section 8 presents experimental evaluations for all available heuristics and Section 9 concludes our study.

2. Preliminaries

We consider a fragment of numeric planning restricted to the Strips formalism (Fikes & Nilsson, 1971) with the addition of numeric state variables. We first present a subclass of numeric planning defined by Hoffmann (2003a), called the *restricted numeric planning task*

(RT). Formally, an 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 $v \in \mathcal{N}$ in state s. For simplicity, we sometimes write $s = s_p \cup s_n$ as a minor abuse of notation.

Conditions can be either propositional $\psi \in \mathcal{F}_p$ or numeric. A numeric condition is defined as $\psi : v \trianglerighteq w$, with $\trianglerighteq \in \{\geq, >\}$, $v \in \mathcal{N}$ and $w \in \mathbb{Q}$, and 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 $s[v] \trianglerighteq w$. When a state s satisfies a condition ψ, or a set of conditions Ψ, we write $s \models \psi$ and $s \models \Psi$, respectively. If s does not satisfy ψ, we write $s \not\models \psi$. In what follows, we replace $\psi : v > w$ with $\psi' : v \geq w + \varepsilon$, where $\varepsilon > 0$ is a sufficiently small constant, assuming only $\geq$ conditions.

An action $a \in \mathcal{A}$ is a triplet $\langle \mathsf{pre}(a), \mathsf{eff}(a), \mathsf{cost}(a) \rangle$, where $\mathsf{pre}(a)$ is the set of 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 $\mathsf{eff}(a) = \langle \mathsf{add}(a), \mathsf{del}(a), \mathsf{num}(a) \rangle$, where $\mathsf{add}(a), \mathsf{del}(a) \subseteq \mathcal{F}_p$ are added and deleted facts, and $\mathsf{num}(a)$ is the set of numeric effects, i.e., an assignment of a numeric variable $v \mathrel{+}= k$ where $k \in \mathbb{Q}$ is a constant quantity. We assume that each action has at most one numeric effect on each numeric variable. We say that an action a is applicable in the state s if $s \models \mathsf{pre}_p(a) \cup \mathsf{pre}_n(a)$. The result of applying action a to state s is denoted by $s[\![a]\!] = \langle s'_p, s'_n \rangle$, where $s'_p = (s_p \setminus \mathsf{del}(a)) \cup \mathsf{add}(a)$, and for each variable $v \in \mathcal{N}$, $s'_n[v] = s_n[v] + k$ if $(v \mathrel{+}= k) \in \mathsf{num}(a)$, and $s'_n[v] = s_n[v]$ otherwise.

The set of goal conditions $G = G_p \cup G_n$ denotes propositional and numeric conditions, respectively. We say that s_* is a *goal state* if $s_* \models G$. 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*, that is, the set of all numeric conditions that are either goal conditions and preconditions for some action. Note that $\bar{\mathcal{F}}_n$ is finite since the number of the goal conditions and the preconditions of the actions is finite.

An *s-plan* is an action sequence π that can be applied successively to state s and results in a goal state $s_* \models G$. The cost of an s-plan π is the sum of all its action costs and an *optimal s-plan* has the minimal cost among all possible s-plans. The optimal cost of an s-plan is denoted by $h^*(s)$. Therefore, a plan for planing task Π is an s_I-plan and the minimum 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 $L \cap \pi \neq \emptyset$, for every plan π for Π.

If a Π^{RT} does not have any numeric state variables ($\mathcal{N} = \emptyset$), we have a *classical* (Strips) *planning task* Π. A numeric variable $v \in \mathcal{N}$ is called a *resource* variable if it has non-negative domain $[0, R_v]$, where $R_v \in \mathbb{Q}^{0+}$ is the maximum capacity, and it is only affected by actions $a \in \mathcal{A}$ with effects of the type $v \mathrel{+}= k_v^a \in \mathbb{Q}$. If all the numeric variables are resources and $k_v^a \in \mathbb{Q}^{0-}$ for all $a \in \mathcal{A}$, $v \in \mathcal{N}$, we have a resource-constrained planning task (RCP) (Nakhost, Hoffmann, & Müller, 2012), while if there exists at least one action with $k_v^a \in \mathbb{Q}^{+}$, we have a planning task with resources (RP) (Wilhelm, Steinmetz, & Hoffmann, 2018). RT generalizes RP by allowing numeric variables to have domain $\mathbb{Q}$ and numeric conditions of the form: $\psi : v \geq w_0^{\psi}$, with $w_0^{\psi} \in \mathbb{Q}$.

2.1 Planning with Simple Conditions

Scala et al. (2016a) introduce numeric planning with simple conditions (sc), an extension of rt where numeric preconditions can be written as $\psi : \sum_{v \in \mathcal{N}} v \cdot w_v^\psi \geq w_0^\psi$, with $w_v^\psi, w_0^\psi \in \mathbb{Q}$. Tasks with only sc 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.

Given an sct $\Pi^{\text{num}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$, a transformed task Π^{RT} is defined as the 5-tuple $\langle \mathcal{F}_p, \mathcal{N}^{\text{RT}}, \mathcal{A}^{\text{RT}}, s_I^{\text{RT}}, G^{\text{RT}} \rangle$. For every numeric expression mentioned in every numeric condition $\psi \in \bar{\mathcal{F}}_n$, we add an additional numeric variable $v^\psi \in \mathcal{N}^{\text{RT}}$, with $s_I[v^\psi] = \sum_{v \in \mathcal{N}} w_v^\psi s_I[v]$. Each numeric condition is replaced by $v^\psi \geq w_0^\psi$ and, for every action $a \in \mathcal{A}$, a numeric effect on every variable v^ψ must be added, with the form $v^\psi \mathrel{+}= \sum_{v \in \mathcal{N}} w_v^\psi k_v^a$, where $v \mathrel{+}= k_v^a \in \mathbb{Q}$ are the original numeric effects of the action. This translation is polynomial in the number of active numeric conditions of the planning task.

2.2 Delete-Free rt and Delete-Relaxed rt

In his seminal work on numeric planning, Hoffmann (2003a) defines a delete-free (or monotonic) rt as follows: for each action $a \in \mathcal{A}$ it holds that $\mathsf{del}(a) = \emptyset$ and all numeric effects are of the form $(v \mathrel{+}= k_v^a) \in \mathsf{num}(a)$ with $k_v^a > 0$. The *support function* $\mathsf{supp} : \mathcal{F}_p \cup \mathcal{F}_n \to 2^{\mathcal{A}}$ 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 $\mathsf{supp}(v \geq w_0) = \{a \in \mathcal{A} \mid v \mathrel{+}= k_v^a \in \mathsf{num}(a)\}$ if $v \geq w_0 \in \mathcal{F}_n$.

An rt can be relaxed to a delete-free rt by setting $\mathsf{del}(a) = \emptyset$ for each action a and removing numeric effects of the form $(v \mathrel{+}= k)$ with $k < 0$. We call the resulting delete-free task the delete-relaxed rt. The optimal cost of the delete-relaxed rt is denoted by h^+.

3. The Complexity of Delete-Free Numeric Planning

We start with an important complexity result for delete-free numeric planning. It is well-known that classical planning with actions having empty delete lists is NP-complete. Here we obtain the same result for delete-free rt. The lower bound, NP-hardness, follows from the fact that any classical Strips problem is also an rt problem with no numeric effects. However, obtaining the upper bound result is somewhat less trivial, since as shown by the following example, there can be plans with exponentially many actions.

Example 1. *Let $\Pi = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$ be an* rt *with $\mathcal{F}_p = \emptyset$ and $\mathcal{N} = \{v\}$. Let $s_I = \{v = 0\}$, $G = \{v \geq 2^k\}$, and $\mathcal{A} = \{a\}$, where $a = \langle \emptyset, \{\langle v \mathrel{+}= 1 \rangle\}, 1 \rangle$. The optimal plan of Π consists of applying action a 2^k times to s_I.*

In contrast to classical delete-free planning where each optimal plan is bounded by the number of actions in the task (i.e., each action is applied at most once along an optimal plan), in the example above the unique optimal plan consists of 2^k actions, and therefore cannot be checked in polynomial-time in a direct fashion. Thus, to prove membership in NP, we need to employ a different way to check the plan. To this end, we employ the following two observations:

1. if action a was applied at some point t along a plan, it can be applied at any point $t' > t$ afterwards,

2. let $v \mathrel{+}= k_v^a$ be an effect of action a, then if we apply action a consecutively m_a times the resulting effect on variable v is $v \mathrel{+}= \mathsf{m}_a \cdot k_v^a$.

We use these observations to construct a non-deterministic Turing machine (NTM) to solve the following decision problem.

Name. Bounded-Plan Length problem (BPL)
Instance. An RT Π and a number K.
Question. Is there a plan π for Π such that $\mathsf{cost}(\pi) \leq K$?

At each decision point the NTM should "guess" the tuple (a, m_a), where $a \in \mathcal{A}$ is the action that should be applied next, and m_a is the number of times that a should be applied. We denote the application of m_a actions a to a state s as $a^{\mathsf{m}_a}[\![s]\!]$. Then, the result of this application is $a^{\mathsf{m}_a}[\![s]\!] = \langle s'_p, s'_n \rangle$, where $s'_p = (s_p \setminus \mathsf{del}(a)) \cup \mathsf{add}(a) = s_p \cup \mathsf{add}(a)$, and $s'_n[v] = s_n[v] + \mathsf{m}_a \cdot k_v^a$ if $(v \mathrel{+}= k_v^a) \in \mathsf{num}(a)$, and $s'_n[v] = s_n[v]$ otherwise, for each $v \in \mathcal{N}$. Note that the complexity of this application is linear in $\mathcal{F}_p$ and logarithmic in m_a. The machine terminates, returning ACCEPT, if a state s is a goal state and, for a decision path π (a compact representation of a plan), the following holds:

$$\sum_{(a,\mathsf{m}_a)\in\pi} \mathsf{m}_a \cdot \mathsf{cost}(a) \leq K.$$

Otherwise, the NTM returns REJECT. Note, that we can restrict the NTM to choose an action a at most once and the upper bound on m_a is given by the input, i.e., for each action $a \in \mathcal{A}$:

$$\mathsf{m}_a \leq \frac{w_0^{\max}}{k^{\min}}, \text{ where } w_0^{\max} = \max\{w_0 \mid (v \geq w_0) \in \bar{\mathcal{F}}_n\}$$
$$\text{and } k^{\min} = \min\{k_v^a \mid \exists\, a \in \mathcal{A} : (v \mathrel{+}= k_v^a) \in \mathsf{num}(a)\}.$$

The NTM requires at most $(\log w_0^{\max} - \log k^{\min}) \cdot |\mathcal{A}|$ guesses, assuming that the m_as are guessed bit by bit. Thus, we can conclude with the following proposition.

Proposition 1. *The decision problem* BPL *for delete-free* RT*s is NP-complete.*

The NP membership may also be obtained by first transforming the RT into an integer linear program (Piacentini et al., 2018a), and then applying the theorem from Papadimitriou (1981). Note, however, that the straightforward proof presented here provides a better intuition on the nature of the BPL problem.

4. $h^{\max}$ in Numeric Planning

This section presents and compares different $h^{\max}$ variants for RT in the literature, since $h^{\max}$ is one of the major components of the LM-cut heuristic. We also introduce a variant $h^{\max}_{\text{cri}}$, which is a key component of the numeric variant of LM-cut heuristic described in the next section, even though we show that it is an inadmissible heuristic.

4.1 NP-completeness of $h^{\max}$ with Numeric State Variables

We start with the following observation: while computing $h^{\max}$ in classical planning can be done in polynomial time (Bonet & Geffner, 2001), the problem of calculating $h^{\max}$ with numeric state variables is NP-hard even if we restrict the problem to RTs. The proof of this observation follows. First, for a given state s we define $h^{\max}(s) = \hat{h}(s, G)$ to be a maximal fixed-point of the following recursive equations:

$$\hat{h}(s, F) = \max_{\psi \in F} \hat{h}(s, \psi) \quad \text{for} \quad F \subseteq \mathcal{F}_p \cup \mathcal{F}_n,$$

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

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 $\psi : v \geq w \in \mathcal{F}_n$. The fixed-point can be computed in a finite number of time steps (Scala et al., 2016a).

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

Proof. We show this result by reduction from the minimization version of the unbounded knapsack problem (mUKP), which is proved to be NP-hard (Zukerman, Jia, Neame, & Woeginger, 2001). Let $[n] := \{1, \ldots, n\}$ be a set of n elements, each element $i \in [n]$ has a value w_i and cost c_i, all positive rational (real) numbers. Let w_0 be a rational number. Then, the mUKP problem is given by the following optimization problem:

$$\min \sum_{i=1}^{n} c_i x_i \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_i x_i \geq w_0 \tag{2}$$

$$x_i \in \mathbb{N}_0, i \in [n]. \tag{3}$$

We assume w_0 is positive, as $x_i = 0$ for all $i \in [n]$ is a trivial solution for $w_0 \leq 0$. Consider $C^*(w_0)$ to be the optimal cost for an mUKP with constraint $\sum_{i=1}^{n} w_i x_i \geq w_0$. Then, the following recursive formula holds:

$$C^*(w_0) = \min_{i \in [n]} C^*(w_0 - w_i) + c_i.$$

Given an instance of the mUKP, we can build an RT with no propositional variables and with one numeric variable v. For every element i we have an action a_i such that $\mathsf{pre}(a_i) = \emptyset$, $\mathsf{num}(a_i) = \{v \mathrel{+}= w_i\}$, and $\mathsf{cost}(a_i) = c_i$. We set $s[v] = 0$ and G contains one condition $v \geq w_0$. This RT instance is constructed in $O(n)$ time. We show that the solution of the mUKP is equivalent to the solution of $h^{\max}$. By definition,

$$h^{\max}(s) = \hat{h}(s, G) = \hat{h}(s, \{v \geq w_0\}) = \hat{h}(s, v \geq w_0).$$

Because no actions have preconditions, we get that

$$\hat{h}(s, v \geq w_0) = \begin{cases} 0 & \text{if } w_0 \leq 0 \\ \min_{a_i \in \mathsf{supp}(v \geq w_0)} \hat{h}(s, \{v \geq w_0 - w_i\}) + \mathsf{cost}(a_i) & \text{otherwise.} \end{cases}$$

Since $\mathsf{num}(a_i) = \{v \mathrel{+}= w_i\}$ for each action a_i, $\mathsf{supp}(v \geq w_0) = \{a_i \mid i \in [n]\}$, and

$$\hat{h}(s, v \geq w_0) = \min_{i \in [n]} \hat{h}(s, v \geq w_0 - w_i) + c_i$$

for $w_0 > 0$. Thus, $h^{\max}(s) = \hat{h}(s, v \geq w_0)$ is equivalent to $C^*(w_0)$. $\square$

4.2 Relaxations of $h^{\max}$

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

Definition 1. *Given a delete-free* RT*, a state s, a fact $\psi \in \mathcal{F}$, and action a, We define an* ***action multiplier*** m_a *as*[1]

$$\mathsf{m}_a(s, \psi) = \begin{cases} 0, & \textit{if } s \models \psi, \\ \frac{w - s[v]}{k^a}, & \textit{if } a \in \mathsf{supp}(\psi) \wedge \psi : v \geq w \in \mathcal{F}_n, \\ 1, & \textit{if } a \in \mathsf{supp}(\psi) \wedge \psi \in \mathcal{F}_p, \\ \infty, & \textit{otherwise} \end{cases}$$

where $k^a \in \mathbb{Q}$ is the numeric effect of action a on variable v, i.e., $v \mathrel{+}= k^a \in \mathsf{eff}(a)$.

This definition allows us to restrict our facts 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. Using this action multiplier, Scala et al. (2016a) propose an additional level of relaxation, decoupling preconditions and numeric effects of actions, that assures the admissibility of their $h^{\max}$ variant, i.e., $h^{\max}_{\text{hbd}}$.

Definition 2. *Given an* RT *and a state s, the heuristic function $h^{\max}_{hbd}(s) := h^{\max}_{hbd}(s, G)$ is defined as*

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

for any set of facts $F \subseteq \mathcal{F}_p \cup \mathcal{F}_n$, and

$$h^{\max}_{hbd}(s, \psi) = \begin{cases} 0, & \textit{if } s \models \psi, \\ \min\limits_{a \in \mathsf{supp}(\psi)} h^{\max}_{hbd}(s, \mathsf{pre}(a)) + \mathsf{cost}(a), & \textit{if } \psi \in \mathcal{F}_p, \\ \min\limits_{a \in \mathsf{supp}(\psi)} h^{\max}_{hbd}(s, \mathsf{pre}(a)) + \min\limits_{a \in \mathsf{supp}(\psi)} \mathsf{m}_a(s, \psi) \cdot \mathsf{cost}(a), & \textit{if } \psi \in \mathcal{F}_n \end{cases}$$

for any fact $\psi \in \mathcal{F}_p \cup \mathcal{F}_n$.

The fixed point of the above equations may not be unique, but one can be computed in polynomial time in the number of active numeric conditions and actions (Scala et al., 2016a). In the computation, first, the values are initialized as $h^{\max}_{\text{hbd}}(s, \psi) = \infty$ for each $\psi \in \mathcal{F}_p \cup \bar{\mathcal{F}}_n$. If $h^{\max}_{\text{hbd}}(s) = \infty$ for a state s, it is a dead-end, i.e., a goal state is not reachable

1. Here, and in what follows, we assume that the first matching condition will be active.

from s. Note that compared to the heuristics proposed in this paper, $h^{\max}_{\text{hbd}}$ is applicable for more general numeric planning tasks than RT and SCT, where preconditions and goal conditions can be disjunctive (Scala et al., 2020).

Another relaxation of $h^{\max}$ is the repetition relaxation based max heuristic, h^{irmax}, proposed by Aldinger and Nebel (2017). h^{irmax} relaxes the number of executions of actions to achieve numeric conditions. Although it is originally defined for more general numeric planning tasks, we show a simplified definition of h^{irmax} for RT here. To stay consistent with our previous notation we write $h^{\max}_{\text{ir}}$ instead of h^{irmax}.

Definition 3. *Given an* RT *and a state s, the heuristic function $h^{\max}_{ir}(s) := h^{\max}_{ir}(s, G)$ is defined as*

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

for any set of facts $F \subseteq \mathcal{F}_p \cup \mathcal{F}_n$, and

$$h^{\max}_{ir}(s, \psi) = \begin{cases} 0 & \textit{if } s \models \psi, \\ \min_{a \in \mathsf{supp}(\psi)} h^{\max}_{ir}(s, \mathsf{pre}(a)) + \mathsf{cost}(a) & \textit{otherwise} \end{cases}$$

for a fact $\psi \in \mathcal{F}_p \cup \mathcal{F}_n$.

Lastly, we define a variant of $h^{\max}$ that we denote by $h^{\max}_{\text{cri}}$. The notation "cri" is chosen since this version of $h^{\max}$ admits the critical path value in our version of the justification graph, which is explained below. This heuristic is defined specifically for the numeric LM-cut heuristic, which we present in the next section.

Definition 4. *Given an* RT *and a state s, the heuristic function $h^{\max}_{cri}(s) := h^{\max}_{cri}(s, G)$ is defined as*

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

for any set of facts $F \subseteq \mathcal{F}_p \cup \mathcal{F}_n$, and

$$h^{\max}_{cri}(s, \psi) = \begin{cases} 0 & \textit{if } s \models \psi, \\ \min_{a \in \mathsf{supp}(\psi)} h^{\max}_{cri}(s, \mathsf{pre}(a)) + \mathsf{m}_a(s, \psi) \cdot \mathsf{cost}(a) & \textit{otherwise} \end{cases}$$

for a fact $\psi \in \mathcal{F}_p \cup \mathcal{F}_n$.

This heuristic can be interpreted as a combination of $h^{\max}_{\text{hbd}}$ and $h^{\text{add}}_{\text{hbd}}$, an inadmissible heuristic proposed by Scala et al. (2016a); $h^{\max}_{\text{cri}}$ uses the same formula as $h^{\max}_{\text{hbd}}$ for a set of facts and the same formula as $h^{\text{add}}_{\text{hbd}}$ for a single fact. Also, it is important to note that for any state s in a given RT, $h^{\max}_{\text{hbd}}(s, \psi') \leq h^{\max}_{\text{cri}}(s, \psi')$ because $h^{\max}_{\text{hbd}}$ decouples preconditions and effects of actions that have numeric facts, while $h^{\max}_{\text{cri}}$ does not.

Even though these heuristic definitions (i.e., $h^{\max}_{\text{hbd}}$, $h^{\max}_{\text{ir}}$, and $h^{\max}_{\text{cri}}$) address all subsets of $\mathcal{F}_p \cup \mathcal{F}_n$, in practice, we only need to compute the fixed point of $h^{\max}_x$, where x can be hbd, ir, or cri, for the facts in the finite set $\mathcal{F}_p \cup \bar{\mathcal{F}}_n$. We can therefore compute all three heuristics in polynomial time in the sizes of the sets of facts, active numeric conditions, and actions of RT.

In what follows we show that $h^{\max}_{\text{cri}}$ is in fact an *inadmissible* heuristic with Example 2. This is quite interesting considering the similarities of $h^{\max}_{\text{cri}}$ to $h^{\max}_{\text{ir}}$ and $h^{\max}_{\text{hbd}}$, which are known to be admissible (Scala et al., 2016a; Aldinger & Nebel, 2017).

Example 2. *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

We show the $h^{\max}$*-values of facts for all three heuristics in Table 1.*

a	ψ	$\mathsf{m}_a(s_I, \psi)$	ψ	$h^{\max}_{\text{hbd}}(s_I, \psi)$	$h^{\max}_{\text{ir}}(s_I, \psi)$	$h^{\max}_{\text{cri}}(s_I, \psi)$
a_1	$v \geq 2$	2	$v \geq 2$	1	1	2
a_1	$v \geq 6$	6	$v \geq 6$	3	1	5
a_2	$v \geq 2$	1				
a_2	$v \geq 6$	3				

Table 1: The action multipliers for each action and the $h^{\max}$-values for each ψ.

It is easy to check that an optimal plan for this task is $\pi = \langle a_1, a_1, a_2, a_2\rangle$ *with cost* $h^*(s_I) = 4$. *We also have that* $h^{\max}_{cri}(s_I, v \geq 2) = 2$ *and* $h^{\max}_{cri}(s_I, v \geq 6) = 5$, *which results in an inadmissible heuristic:*

$$h^{\max}_{cri}(s_I) = 5 > h^*(s_I) = 4.$$

The derivation of the heuristic values in the previous example are most easily seen using the concept of a justification graph (JG). As the JG is also key to obtain LM-cut estimates, we delay this presentation to the next section.[2]

5. LM-Cut in RT Planning

We now introduce the LM-cut heuristic for RT planning problems. We start by describing the well-known classical version and show how to extend it to its numeric counterpart. We then present several variants of LM-cut heuristics that arise from different choices for the $h^{\max}$ relaxation. We conclude this section with a theoretical comparison of these variants and show the dominance relationships among them.

5.1 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 planning tasks. We now provide a complete description of the LM-cut heuristic in classical planning, which is fundamental for our extension to numeric tasks, as shown in Section 5.2.

The LM-cut heuristic is computed as a cut in a *labelled weighted digraph* called justification graph (JG). Given the large number of concepts associated with this heuristic, we first formally define labelled weighted digraphs, cuts, and JGs. We then introduce the goal zone and before-goal zone sets of vertices in a JG. Finally, using these definitions, we present the LM-cut heuristic in classical planning.

2. Interested readers can see the JG representation of Example 2 in Appendix B, Example 11, Figure 16.

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 $\mathsf{W} : E \to \mathbb{R}^{0+}$ is the weight function on edges. Function $\mathsf{lbl} : E \to A$ is defined as $(n, n', a) \mapsto a$, that is, a function that returns the label of a given edge. For a set of edges $E' \subseteq E$, we define $\mathsf{lbl}(E') = \{\mathsf{lbl}(e) \mid e \in E'\}$ (i.e., the set of labels associated with edges in E'). An interleaved 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 $i \in [m] := \{1, \ldots, m\}$ it holds that $(n_i, n_{i+i}, a_i) \in E$. 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 path exists. Given two disjoint sets of nodes $N_1, N_2 \subseteq N$, we define a *directed cut* to be $(N_1, N_2) = \{(n_1, n_2, a) \in E \mid n_1 \in N_1, n_2 \in N_2\}$. The weight of a path π and a 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 $\partial^{\mathsf{in}}(E') = \{n_0 \in N \mid \mathsf{in}(n_0) \cap E' \neq \emptyset\}$.

Example 3. *Let $\mathcal{G} = \langle N, E, \mathsf{W} \rangle$ be a labelled weighted digraph with $N = \{p, q, r, s\}$, and E and W given by*

$e \in E$	$\mathsf{W}(e)$	$\mathsf{lbl}(e)$
(p, q, a_1)	1	a_1
(p, r, a_2)	2	a_2
(q, s, a_3)	2	a_3
(r, s, a_4)	1	a_4

Figure 2 depicts graph $\mathcal{G}$, where a cut $L = (\{p\}, \{q, r\})$ is represented by a vertical line. For this cut, $\mathsf{W}(L) = 1$, $\mathsf{lbl}(L) = \{a_1, a_2\}$, and $\partial^{\mathsf{in}}(L) = \{q, r\}$.

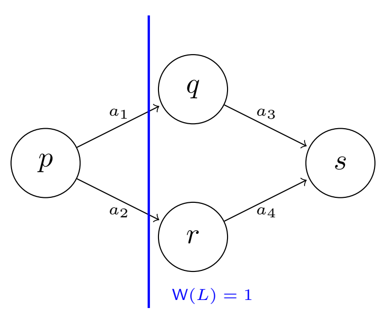


Figure 2: Example of a labelled weighted digraph.

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. These conditions define a directed weighted graph $\mathcal{G}$ whose vertices correspond to the facts of the planning task and labelled weighted edges correspond to actions. 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 one precondition in a JG, one of the preconditions is chosen for an action while justifying $h^{\max}$ values. Intuitively speaking, the precondition choice function defined by Bonet and Helmert (2010) maps each action to one of its preconditions, which is done via maximizing $h^{\max}$. Definition 5 formally defines the function that chooses the precondition of each action and Definition 6 formalizes JG for classical planning.

Definition 5. *Given a classical planning task and a state s, a* **precondition choice function** $\mathsf{pcf} : \mathcal{S} \times \mathcal{A} \to \mathcal{F}_p$ *is the function that satisfies the condition*

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

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

1. *a set of vertices* $N = \{n_\psi \mid \psi \in \mathcal{F}_p \cup \{\emptyset\}\}$*;*
2. *a set of labelled edges* $E = \hat{E} \cup \{(n_\psi, n_{\psi'}, a) \mid a \in \mathsf{supp}(\psi'), \psi = \mathsf{pcf}(s, a)\}$*; where we include the following zero-cost edges* $\hat{E} = \{(n_\emptyset, n_\psi, a_0) \mid s \models \psi\}$ *with a dummy action* a_0 *(i.e.,* $\mathsf{pre}(a_0) = \mathsf{eff}(a_0) = \emptyset$ *and* $\mathsf{cost}(a_0) = 0$*);*
3. *and a weight function* $\mathsf{W} : E \to \mathbb{R}^{0+}, (n_\psi, n_{\psi'}, a) \to \mathsf{cost}(a)$*.*

Given a JG as presented in Definition 6, we now introduce the two zones (i.e., sets of nodes) needed to define a cut over the JG.

Definition 7. *Given a classical planning task, a state s, and the JG* $\mathcal{G} = \langle N, E, \mathsf{W} \rangle$*, the goal fact g is defined as the most costly fact in G with respect to* $h^{\max}$ *and state s, that is,*

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

The **goal zone** *of the graph* $\mathcal{G}$ *is the set of vertices that can reach 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* $N^b = (N \setminus N^g) \setminus N^0$*.*

Definition 8 presents the LM-cut heuristic for classical planning. The heuristic iteratively creates a JG for the task, finds a cut, and adds its cost to the heuristic value. The procedure then updates the cost of the actions present in the cut and re-builds the JG for the updated task. The procedure continues until no more positive cost actions can be added.

Definition 8. *Given a classical planning task* $\Pi = \langle \mathcal{F}_p, \emptyset, \mathcal{A}, s_I, G \rangle$ *and a state s, the heuristic value of the LM-cut heuristic,* $h^{\text{LM-cut}}(s)$*, is computed by the following procedure.*

1. *Let* $h^{\text{LM-cut}}(s) = 0$.

2. *Initialize* $h^{\max}(s, \psi) = \infty$ *and compute a fixed-point of* $h^{\max}(s, \psi)$ *for each* $\psi \in \mathcal{F}_p$ *in* Π. *Let* g *be the goal fact in Definition 7, i.e.,* $g \in \operatorname{argmax}_{g' \in G} h^{\max}(s, g')$. *If* $h^{\max}(s, g) = 0$ *return* $h^{\text{LM-cut}}(s)$. *Otherwise, if* $h^{\max}(s, g) = \infty$ *return* ∞.

3. *Construct the JG for* Π *and* s *as in Definition 6.*

4. *Let* $L = (N^0, N^g)$ *be a cut in the JG where* N^0 *is the before-goal zone and* N^g *is the goal zone in Definition 7. Increase* $h^{\text{LM-cut}}(s)$ *by* $\mathsf{W}(L)$.

5. *Let* $\mathcal{A}^c = \{\langle \mathsf{pre}(a), \mathsf{eff}(a), \mathsf{cost}^c(a)\rangle \mid a \in \mathcal{A}\}$ *where*

$$\mathsf{cost}^c(a) = \begin{cases} \mathsf{cost}(a) - \mathsf{W}(L), & \textit{if } a \in \mathsf{lbl}(L), \\ \mathsf{cost}(a), & \textit{if } a \notin \mathsf{lbl}(L). \end{cases}$$

   Update Π *to be* $\langle \mathcal{F}_p, \emptyset, \mathcal{A}^c, s_I, G\rangle$ *and go back to Step 2.*

Example 4 illustrates all the steps of the LM-cut heuristic for a small classical planning task.

Example 4. *Let* $\Pi = \langle \mathcal{F}_p, \emptyset, \mathcal{A}, s_I, G\rangle$ *be a classical planning task with* $\mathcal{F}_p = \{p, q, r, g_1, g_2\}$, $s_I = \emptyset$, $G = \{g_1, g_2\}$, *and* $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5\}$, *where*

actions	pre	eff	cost
a_1	$\emptyset$	p	2
a_2	$\emptyset$	p, q	3
a_3	$\emptyset$	r	1
a_4	p	g_1	0
a_5	p, q	g_2	0
a_6	r	g_2	0

We show $h^{\max}$*-values of facts, action costs, and the cut in each iteration in Table 2 and the JGs in Figure 3, where nodes from which the goal proposition is not reachable are ignored. The cuts extracted in the first two iterations,* L_1 *and* L_2*, are visually represented by vertical lines in the figure. In the first JG, since* g_1 *is reachable from* p *with a zero-cost path* $\langle a_4\rangle$*, the goal zone is* $\{n_p, n_{g_1}\}$*. In the second JG, similarly, since* g_2 *is reachable from* q *or* r *with a zero-cost path* $\langle a_5\rangle$ *or* $\langle a_6\rangle$*, the goal zone is* $\{n_q, n_r, n_{g_2}\}$*. Note that*

$$\mathsf{W}(L_1) = \min\{\mathsf{cost}(a_1), \mathsf{cost}(a_2)\} = \mathsf{cost}(a_1) = 2$$

and

$$\mathsf{W}_1(L_2) = \min\{\mathsf{cost}^c_1(a_2), \mathsf{cost}^c_1(a_3)\} = 1$$

where cost^c_1 *is the updated cost function after the first iteration. We have*

$$h^{\max}(s_I) = h^{\max}(s_I, g_1) = 2 < h^{\text{LM-cut}}(s_I) = \mathsf{W}(L_1) + \mathsf{W}_1(L_2) = 3.$$

As shown in this example, $h^{\text{LM-cut}}$ *takes multiple goal propositions into consideration although* $h^{\max}$ *considers only one goal proposition. This results in the higher* h*-value of* $h^{\text{LM-cut}}$ *than* $h^{\max}$*. In fact,* $h^{\text{LM-cut}}$ *is proved to dominate* $h^{\max}$ *in classical planning (Helmert & Domshlak, 2009).*

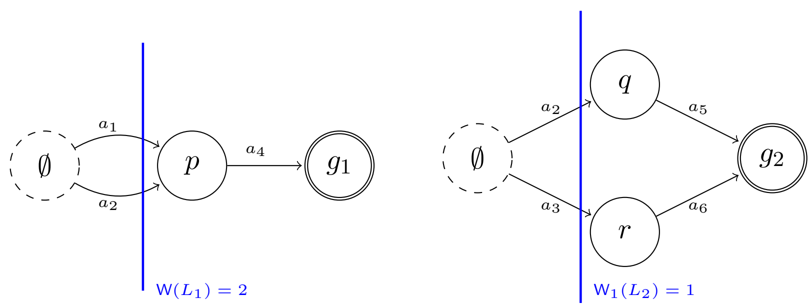
$h^{\max}(s_I, \psi)$	p	q	r	g_1	g_2
1	2	3	1	2	1
2	0	1	1	0	1
3	0	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5	a_6
1	2	3	1	0	0	0
2	0	1	1	0	0	0
3	0	0	0	0	0	0

$\mathsf{pcf}(s_I, a)$	a_1	a_2	a_3	a_4	a_5	a_6
1	$\emptyset$	$\emptyset$	$\emptyset$	p	q	r
2	$\emptyset$	$\emptyset$	$\emptyset$	q	q	r

	N^0	N^g	lbl	W
L_1	$n_\emptyset$	n_{g_1}, n_p	a_1, a_2	2
L_2	$n_\emptyset$	n_q, n_r, n_{g_2}	a_2, a_3	1

Table 2: $h^{\max}(s_I, \psi)$, $\mathsf{pcf}(s_I, a)$, $\mathsf{cost}(a)$, and the cut in each iteration.



(a) The first cut. (b) The second cut.

Figure 3: JGs for a classical planning task in Example 4. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

The admissibility of the LM-cut heuristic is ensured by the concept of cost-partitioning. In their works, Katz and Domshlak (2008) and Yang et al. (2008) independently proposed an approach to additively combine individual admissible heuristic estimates.

Definition 9. *Given a planning task Π, a* ***cost partition*** *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 $\forall a \in \mathcal{A} : \sum_{i=1}^n \mathsf{cost}_i(a) \leq \mathsf{cost}(a)$.*

The following proposition is a simplified version of the claim proposed by Katz and Domshlak (2008).

Proposition 3. *Given a planning task Π, cost partition $\{\Pi_i\}_{i=1}^n$, 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.*

Proof. Consider the case $n = 2$. For any state s, let π be an optimal s-plan with the cost $h^*(s)$ for Π. Since Π_1 and Π_2 are the same as Π except for the cost functions, π is also a valid s-plan for both Π_1 and Π_2. Let the cost of π in Π_1 and Π_2 be $h'_1(s)$ and $h'_2(s)$, respectively. Since $\forall a \in \mathcal{A} : \mathsf{cost}_1(a) + \mathsf{cost}_2(a) \leq \mathsf{cost}(a)$, it holds that $h'_1(s) + h'_2(s) \leq h^*(s)$. Let $h^*_1(s)$ and $h^*_2(s)$ be the optimal solution costs for Π_1 and Π_2. As h_1 and h_2 are admissible in Π_1 and Π_2, $h_1(s) \leq h^*_1(s) \leq h'_1(s)$ and $h_2(s) \leq h^*_2(s) \leq h'_2(s)$. Thus, $h_1(s) + h_2(s) \leq h'_1(s) + h'_2(s) \leq h^*(s)$, so $h_1(s) + h_2(s)$ is admissible in Π.

The $n > 2$ case follows by induction. $\square$

The proof of Proposition 3 does not use any property which is present in classical planning but not in numeric planning. Thus, it also holds for numeric planning.

The admissibility of the LM-cut heuristic is guaranteed by cost-partitioning. Let L_i be the cut found by the LM-cut heuristic at iteration i and W_i be the weight function after iteration i with $\mathsf{W}_0 = \mathsf{W}$. Suppose that n iterations are performed by LM-cut. Let Π_i be a classical planning task that is the same as the original task Π except that the action cost cost_i is defined as

$$\mathsf{cost}_i(a) = \begin{cases} \mathsf{W}_{i-1}(L_i) & \text{if } a \in \mathsf{lbl}(L_i) \\ 0 & \text{if } a \notin \mathsf{lbl}(L_i) \end{cases}$$

for each action a. Helmert and Domshlak (Helmert & Domshlak, 2009) showed that $\{\Pi_i\}_{i=1}^{n}$ constitutes a cost partition, and $\mathsf{W}_{i-1}(L_i)$ is admissible for Π_i. We will prove the admissibility of numeric LM-cut using the cost-partitioning in the same way.

5.2 Numeric LM-Cut

Similarly to classical LM-cut, we need an $h^{\max}$ heuristic and JGs based on the heuristic to properly define numeric LM-cut. Section 4 has introduced three tractable variants of $h^{\max}$ for numeric planning. Therefore, following the scheme of Helmert and Domshlak (2009), we now present a construction procedure for the JG in numeric planning. Since there are three $h^{\max}$ alternatives, $h^{\max}_{\mathrm{cri}}$, $h^{\max}_{\mathrm{ir}}$, and $h^{\max}_{\mathrm{hbd}}$, there can be multiple LM-cut heuristics. We first present the definition of the *numeric LM-cut heuristic family*; it is parameterized and different combinations of these parameters result in different LM-cut heuristics. The main difference between the numeric JG and its classic counterpart is that the cost of an edge $(n_\psi, n_{\psi'}, a)$ is given by $\mathsf{W_a}(s, a, \psi') \cdot \mathsf{cost}(a)$ instead of $\mathsf{cost}(a)$, where $\mathsf{W_a}$ is an action weight function, as described in Definition 10.

Definition 10. *Given an* RT *and a state s, a precondition choice function* $\mathsf{pcf} : \mathcal{S} \times \mathcal{A} \times \mathcal{F}_p \cup \bar{\mathcal{F}}_n \to \mathcal{F}_p \cup \bar{\mathcal{F}}_n$*, and an action weight function* $\mathsf{W_a} : \mathcal{S} \times \mathcal{A} \times \mathcal{F}_p \cup \bar{\mathcal{F}}_n \to \mathbb{R}^{0+}$*, the* **justification graph** *is the 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 labelled edges $E = \hat{E} \cup \{(n_\psi, n_{\psi'}, a) \mid a \in \mathsf{supp}(\psi'), \psi = \mathsf{pcf}(s, a, \psi')\}$; where we include the following zero-cost edges $\hat{E} = \{(n_\emptyset, n_\psi, a_0) \mid s \models \psi\}$ with dummy action a_0 (i.e., $\mathsf{pre}(a_0) = \mathsf{eff}(a_0) = \emptyset$ and $\mathsf{cost}(a_0) = 0$);*
3. *and a weight function $\mathsf{W} : E \to \mathbb{R}^{0+}, (n_\psi, n_{\psi'}, a) \mapsto \mathsf{W_a}(s, a, \psi') \cdot \mathsf{cost}(a)$.*

Definition 11. *Given an* RT*, a state s, a precondition choice function $\mathsf{pcf} : \mathcal{S} \times \mathcal{A} \times \mathcal{F}_p \cup \bar{\mathcal{F}}_n \to \mathcal{F}_p \cup \bar{\mathcal{F}}_n$ and the JG $\mathcal{G} = \langle N, E, \mathsf{W}\rangle$ using pcf, the goal fact g is defined as*

$$g = \mathsf{pcf}(s, a_g, g')$$

where g' is a dummy proposition and a_g is a dummy action with $\mathsf{pre}(a_g) = G$, $\mathsf{add}(a_g) = \{g'\}$, and $\mathsf{num}(a_g) = \emptyset$. The **goal zone** *of the graph $\mathcal{G}$ is the set of vertices that can reach 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* $N^b = (N \setminus N^g) \setminus N^0$*.*

We now present the definition of the numeric LM-cut heuristic family. One of the key differences with classic LM-cut is that numeric LM-cut considers the action weight function $\mathsf{W_a}$ to update the cost of an action a. This change is needed because the corresponding JG includes $\mathsf{W_a}$ in the edge costs.

Definition 12. *Given an* RT $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G\rangle$*, a state* s*, a precondition choice function* $\mathsf{pcf} : \mathcal{S} \times \mathcal{A} \times \mathcal{F}_p \cup \bar{\mathcal{F}}_n \to \mathcal{F}_p \cup \bar{\mathcal{F}}_n$*, and an action weight function* $\mathsf{W_a} : \mathcal{S} \times \mathcal{A} \times \mathcal{F}_p \cup \bar{\mathcal{F}}_n \to \mathbb{R}^{0+}$*, the heuristic value of the numeric LM-cut heuristic,* $h^{\text{LM-cut}}(s)$*, is computed by the following procedure.*

1. *Let* $h^{\text{LM-cut}}(s) = 0$*.*
2. *Construct a JG using* Π^{RT}*,* pcf*, and* $\mathsf{W_a}$ *as described in Definition 10. Let* g *be the goal fact in Definition 11. If* n_g *is unreachable from* $n_\emptyset$ *in the JG, return* ∞*. If* $\mathsf{W}(\mathsf{path}(n_\emptyset, n_g)) = 0$*, return* $h^{\text{LM-cut}}(s)$*.*
3. *Let* $L = (N^0, N^g)$ *be a cut in the JG where* N^0 *is the before-goal zone and* N^g *is the goal zone in Definition 11. Increase* $h^{\text{LM-cut}}(s)$ *by* $\mathsf{W}(L)$*.*
4. *Let* $\mathcal{A}^c = \{\langle \mathsf{pre}(a), \mathsf{eff}(a), \mathsf{cost}^c(a)\rangle \mid a \in \mathcal{A}\}$ *where*

$$\mathsf{cost}^c(a) = \begin{cases} \mathsf{cost}(a) - \frac{\mathsf{W}(L)}{\min_{(n_\psi, n_{\psi'}, a) \in L} \mathsf{W_a}(s, a, \psi')} & \text{if } a \in \mathsf{lbl}(L) \\ \mathsf{cost}(a) & \text{if } a \notin \mathsf{lbl}(L). \end{cases}$$

Update Π^{RT} *to be* $\langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}^c, s_I, G\rangle$ *and go to Step 2.*

We note that, as opposed to Definition 8, $h^{\max}$ does not appear in Definition 12 and its role is incorporated in pcf and $\mathsf{W_a}$. In classical LM-cut, $\mathsf{pcf}(s, a) \in \operatorname{argmax}_{\psi \in \mathsf{pre}(a)} h^{\max}(s, \psi)$ and $\mathsf{W_a}(s, a, n_\psi) = 1$, which results in $\mathsf{W}(\mathsf{path}(n_\emptyset, n_\psi)) = h^{\max}(s, \psi)$. Therefore, $h^{\max}(s, g)$ is used in Step 2 of Definition 8. In our numeric LM-cut, pcf and $\mathsf{W_a}$ are not necessarily determined by a single $h^{\max}$ variant. As a result, Step 2 of Definition 12 considers $\mathsf{W}(\mathsf{path}(n, n_g))$ instead of $h^{\max}(s, g)$. In addition, the weight of the cut $\mathsf{W}(L)$ is divided by $\min_{(n_\psi, n_{\psi'}, a) \in L} \mathsf{W_a}(s, a, \psi')$ in Step 4.

Now, we define different LM-cut heuristics by specifying the parameters. Except for $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$, they are defined based on $h^{\max}_{\text{cri}}$, $h^{\max}_{\text{ir}}$, and $h^{\max}_{\text{hbd}}$.

Definition 13. *Heuristic* $h_{cri}^{\text{LM-cut}}$ *is a numeric LM-cut heuristic using a precondition choice function such that*

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

and an action weight function

$$\mathsf{W_a}(s, a, \psi') = \mathsf{m}_a(s, \psi').$$

For $h_{\text{cri}}^{\text{LM-cut}}$, without compromising the admissibility, which is proven later, we restrict the set of vertices N of a JG such that for each $n_\psi \in N$ it holds that there are facts $\psi' \in s$ and $\psi'' \in G$ such that there 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 $h_{\text{cri}}^{\max}(s, \psi) < \infty$. Otherwise, the facts are irrelevant for the solution of the task. The computation of $h_{\text{cri}}^{\max}$-values and the construction of the JG, which are performed simultaneously in practice, are at most quadratic in the size of the RT problem.

Definition 14. *Heuristic* $h_{ir}^{\text{LM-cut}}$ *is a numeric LM-cut heuristic using a precondition choice function such that*

$$\mathsf{pcf}(s, a, \psi') \in \operatorname*{argmax}_{\psi \in \mathsf{pre}(a)} h_{ir}^{\max}(s, \psi)$$

and an action weight function

$$\mathsf{W_a}(s, a, \psi') = 1.$$

Definition 15. *Heuristic* $h_{hbd}^{\text{LM-cut}}$ *is a numeric LM-cut heuristic using a precondition choice function such that*

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

where $\hat{a} = a$ *if* $\psi' \in \mathcal{F}_p$ *and*

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

if $\psi' \in \mathcal{F}_n$ *and an action weight function*

$$\mathsf{W_a}(s, a, \psi') = \mathsf{m}_a(s, \psi').$$

The previous two definitions introduce the numeric LM-cut heuristics based on $h_{\text{ir}}^{\max}$ and $h_{\text{hbd}}^{\max}$, respectively. We note that action multiplier m_a is used to build both $h_{\text{cri}}^{\text{LM-cut}}$ and $h_{\text{hbd}}^{\text{LM-cut}}$, but not in $h_{\text{ir}}^{\text{LM-cut}}$ because these values are omitted in $h_{\text{ir}}^{\max}$. However, we can create another LM-cut variant, $h_{\text{ir},\mathsf{m}}^{\text{LM-cut}}$, that employs $h_{\text{ir}}^{\max}$ and includes m_a in the weight function, as introduced in the following definition.

Definition 16. *Heuristic* $h_{ir,\mathsf{m}}^{\text{LM-cut}}$ *is a numeric LM-cut heuristic using a precondition choice function such that*

$$\mathsf{pcf}(s, a, \psi') \in \operatorname*{argmax}_{\psi \in \mathsf{pre}(a)} h_{ir}^{\max}(s, \psi)$$

and an action weight function

$$\mathsf{W_a}(s, a, \psi') = \mathsf{m}_a(s, \psi').$$

To ease exposition, in what follows we write $\mathsf{pcf}(s, a)$ instead of $\mathsf{pcf}(s, a, \psi')$ when the precondition choice function depends only on a state and an action, as in $h^{\text{LM-cut}}_{\text{cri}}$, $h^{\text{LM-cut}}_{\text{ir}}$, and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$.

Example 5. *Let $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G\rangle$ be an* RT *with $\mathcal{F}_p = \{p\}$, $\mathcal{N} = \{v, u\}$, $s_I = \emptyset$, $G = \{v \geq 4, u \geq 1\}$, and $\mathcal{A} = \{a_1, a_2, a_3\}$, where*

actions	pre	eff	cost
a_1	$\emptyset$	$v \mathrel{+}= 1, p$	1
a_2	$v \geq 2$	$v \mathrel{+}= 2$	1
a_3	p	$u \mathrel{+}= 1$	1

We show the action multipliers and $h^{\max}_{hbd}$-values in Table 3. We have

$$h^{\max}_{hbd}(s_I) = h^{\max}_{hbd}(s_I, v \geq 4) = 2.$$

a	ψ	$\mathsf{m}_a(s_I, \psi)$
a_1	$v \geq 2$	2
a_1	$v \geq 4$	4
a_1	p	1
a_2	$v \geq 2$	1
a_2	$v \geq 4$	2
a_3	$u \geq 1$	1

ψ	$h^{\max}_{\text{hbd}}(s_I, \psi)$
$v \geq 2$	1
$v \geq 4$	2
p	1
$u \geq 1$	2

Table 3: The action multipliers for each action and $h^{\max}_{\text{hbd}}(s_I, \psi)$ for each ψ.

We show $h^{\max}_{cri}$-values of facts, action costs, and the cut in each iteration in Table 4 and the JGs in Figure 4, where nodes from which g is not reachable are ignored. In the figure, for each edge $(n_{\psi'}, n_\psi, a)$, we show tuple $(a, \mathsf{m}_a(s_I, \psi))$, a tuple of the label and the action multiplier. The cuts extracted in iterations, L_1, L_2, and L_3, are visually represented by vertical lines in the figure. Note that

$$\mathsf{W}(L_1) = \min\{\mathsf{m}_{a_1}(s_I, v \geq 4) \cdot \mathsf{cost}(a_1), \mathsf{m}_{a_2}(s_I, v \geq 4) \cdot \mathsf{cost}(a_2)\} = 2 \cdot \mathsf{cost}(a_1) = 2,$$

$$\mathsf{W}_1(L_2) = \mathsf{m}_{a_3}(s_I, u \geq 1) \cdot \mathsf{cost}^c_1(a_3) = \mathsf{cost}(a_3) = 1$$

where cost^c_1 is the updated cost function after the first iteration, and

$$\mathsf{W}_2(L_3) = \min\{\mathsf{m}_{a_1}(s_I, v \geq 2) \cdot \mathsf{cost}^c_2(a_1), \mathsf{m}_{a_1}(s_I, v \geq 4) \cdot \mathsf{cost}^c_2(a_1)\} = 2 \cdot \mathsf{cost}^c_2(a_1) = 1,$$

where cost^c_2 is the updated cost function after the second iteration. We have

$$h^{\text{LM-cut}}(s_I) = \mathsf{W}(L_1) + \mathsf{W}_1(L_2) + \mathsf{W}_2(L_3) = 4.$$

In this example, as in classical case, $h^{\text{LM-cut}}_{cri}$ takes multiple goal conditions into consideration while $h^{\max}_{hbd}$ only considers $v \geq 4$. This is one of the most important factors on the heuristic value difference (i.e., $h^{\max}_{hbd}(s_I) = 2$ and $h^{\text{LM-cut}}_{cri}(s_I) = 4$).

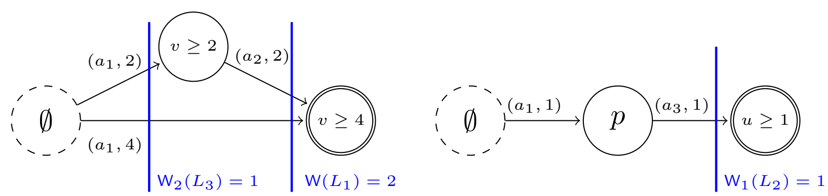
$h^{\max}_{\text{cri}}(s_I,\psi)$	$v \geq 2$	$v \geq 4$	p	$u \geq 1$
1	2	4	1	2
2	1	1	1	1.5
3	1	1	0.5	0.5
4	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3
1	1	1	1
2	0.5	0	1
3	0.5	0	0
4	0	0	0

$\mathsf{pcf}(s_I,a)$	a_1	a_2	a_3
1	$\emptyset$	$v \geq 2$	p
2	$\emptyset$	$v \geq 2$	p
3	$\emptyset$	$v \geq 2$	p

	N^0	N^g	lbl	W
L_1	$n_\emptyset, n_{v\geq2}$	$n_{v\geq4}$	a_1, a_2	2
L_2	$n_\emptyset, n_p$	$n_{u\geq1}$	a_3	1
L_3	$n_\emptyset$	$n_{v\geq2}, n_{v\geq4}$	a_1	1

Table 4: $h^{\max}_{\text{cri}}(s_I,\psi)$, $\mathsf{pcf}(s_I,a)$, $\mathsf{cost}(a)$, and the cut in each iteration.



(a) The first and third cuts.

(b) The second cut.

Figure 4: JGs for an RT in Example 5. The functions W, W_1, and W_2 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

5.3 Properties of Numeric LM-Cut

We now present several theoretical results related to the proposed numeric LM-cut heuristics. We first study the relationship between the LM-cut heuristics and their associated $h^{\max}$ variants by analyzing their JGs. We then present several theoretical results that show the admissibility (or inadmissibility) of the proposed $h^{\text{LM-cut}}$ heuristics.

To analyze the dominance relationship between $h^{\max}$ and $h^{\text{LM-cut}}$ for our numeric variants, we first need to analyze their JGs. As previously mentioned, the name *justification graph* comes from classical planning where the weight of the shortest paths from $n_\emptyset$ to n_ψ equals to the value $h^{\max}(s,\psi)$. This property is used, for example, to show that classical $h^{\text{LM-cut}}$ dominates $h^{\max}$. We would like to obtain the same property for our JG versions to show dominance between heuristics (see Section 5.5 for the heuristic comparisons). In the classical version, the fact that the JG of $h^{\text{LM-cut}}$ justifies $h^{\max}$ is almost immediate because there is not much distinction between the weight of the shortest path and the $h^{\max}$ heuristic value. In the numeric version, however, the proof is a bit more intricate, thus we need a more formal definition.

Definition 17. *Given an* RT*, a state s, and a* pcf*, let $\mathcal{G} = \langle N, E, \mathsf{W}\rangle$ be a corresponding JG. Consider $\mathsf{W}(n_\psi)$ as the weight of the shortest path from $n_\emptyset$ to n_ψ in $\mathcal{G}$.*

Let $h : S \times \mathcal{F}_p \cup \bar{\mathcal{F}}_n \to \mathbb{R}^{0+}$ to be a heuristic that, given a state s, computes a value for each fact ψ. We say that $\mathcal{G}$ justifies h if for each $n_\psi \in N$ it holds that $h(s,\psi) = \mathsf{W}(n_\psi)$.

In the following, we prove that the JGs of $h^{\text{LM-cut}}_{\text{cri}}$ and $h^{\text{LM-cut}}_{\text{ir}}$ indeed justify $h^{\max}_{\text{cri}}$ and $h^{\max}_{\text{ir}}$, respectively. We omit the proof for $h^{\text{LM-cut}}_{\text{hbd}}$ since, as we show in Appendix A, $h^{\text{LM-cut}}_{\text{hbd}}$ is inadmissible, and so it is not useful to discuss its JG.

Proposition 4. *Given an* RT*, a state s, and the justification graph of $h^{\text{LM-cut}}_{cri}$, the weight of the shortest path from $n_\emptyset$ to $n_\psi \in N$ is equal to $h^{\max}_{cri}(s,\psi)$.*

Proof. The shortest paths to other nodes can be incrementally computed in the topological order, where the first node is $n_\emptyset$. Thus, we assume that the weight $\mathsf{W}(n_\psi)$ of the shortest path from $n_\emptyset$ to n_ψ is already known for all $n_\psi \in N$. We prove by induction that $\mathsf{W}(n_\psi) = h^{\max}_{\text{cri}}(s,\psi)$ for all $n_\psi \in N$. Trivially, for $\psi = \emptyset$ it holds $\mathsf{W}(n_\emptyset) = h^{\max}_{\text{cri}}(s,\emptyset) = 0$. Then,

$$\begin{aligned}\mathsf{W}(n_\psi) &= \min_{(n_{\psi'},n_\psi,a)\in E} \mathsf{m}_a(s,\psi)\cdot \mathsf{cost}(a) + \mathsf{W}(n_{\psi'}) \\ &= \min_{(n_{\psi'},n_\psi,a)\in E} \mathsf{m}_a(s,\psi)\cdot \mathsf{cost}(a) + h^{\max}_{\text{cri}}(s,\psi').\end{aligned}$$

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

$$h^{\max}_{\text{cri}}(s,\psi') = \max_{\hat{\psi}\in\mathsf{pre}(a)} h^{\max}_{\text{cri}}(s,\hat{\psi}) = h^{\max}_{\text{cri}}(s,\mathsf{pre}(a)).$$

Because $\mathsf{supp}(\psi) = \{a \mid (n_{\psi'},n_\psi,a)\in E\}$,

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

By induction, the shortest path from $n_\emptyset$ to any node n_ψ is equal to $h^{\max}_{\text{cri}}(s,\psi)$. □

Proposition 4 shows that the JG associated with $h^{\text{LM-cut}}_{\text{cri}}$ indeed justifies $h^{\max}_{\text{cri}}$. A similar result can be obtained for the JG of $h^{\text{LM-cut}}_{\text{ir}}$, as stated in Proposition 5. The proof of this proposition is analogous to the previous one but replacing $\mathsf{m}_a(s,\psi')$ with 1.

Proposition 5. *Given an* RT*, a state s, and the justification graph of $h^{\text{LM-cut}}_{ir}$, the weight of the shortest path from $n_\emptyset$ to $n_\psi \in N$ is equal to $h^{\max}_{ir}(s,\psi)$.*

In what follows, we study the admissibility status of the six combinations of $h^{\max}$ and LM-cut heuristics discussed in the previous section. Table 5 summarizes the main results shown in this paper and the ones available in the literature for $h^{\max}_{\text{ir}}$ and $h^{\max}_{\text{hbd}}$.

The JG based of $h^{\text{LM-cut}}_{\text{cri}}$, $h^{\text{LM-cut}}_{\text{ir}}$, and $h^{\text{LM-cut}}_{\text{hbd}}$ admit the property of justifying $h^{\max}_{\text{cri}}$, $h^{\max}_{\text{ir}}$, and $h^{\max}_{\text{hbd}}$, correspondingly. It is important to note that only two out of three of the resulting LM-cut heuristics are admissible. Specifically, Proposition 6 states that $h^{\text{LM-cut}}_{\text{hbd}}$ is inadmissible. The intricate example of inadmissibility is shown in Appendix A.

h^y_x	hbd		ir		cri	
max	✓	(Scala et al., 2016a)	✓	(Aldinger & Nebel, 2017)	✗	(Sec. 3, Ex. 2)
LM-cut	✗	(App. A, Ex. 11)	✓	(Thm. 1)	✓	(Thm. 1)

Table 5: Admissibility chart: ✓– admissible, ✗– inadmissible. The h^y_x is the heuristic name where $y \in \{\max, \text{LM-cut}\}$ (rows) and $x \in \{\text{cri}, \text{hbd}, \text{ir}\}$ (columns).

Proposition 6. *The LM-cut heuristic $h_{hbd}^{\text{LM-cut}}$ is inadmissible.*

We now present the admissibility proofs for $h_{\text{cri}}^{\text{LM-cut}}$, $h_{\text{ir}}^{\text{LM-cut}}$, and $h_{\text{ir},\mathsf{m}}^{\text{LM-cut}}$. To do so, we first show the relationship between numeric landmarks (Scala et al., 2017) and the JGs associated with these LM-cut heuristics. Following Helmert and Domshlak (2009), we show in Lemma 1 how to extract such landmarks from numeric JGs. We note that this lemma holds for $h_{\text{cri}}^{\text{LM-cut}}$, $h_{\text{ir}}^{\text{LM-cut}}$, and $h_{\text{ir},\mathsf{m}}^{\text{LM-cut}}$.

Lemma 1. *Assume an* RT *of a non-zero cost and a state s. Let $\mathcal{G}$ be the JG corresponding to Π^{RT}, where $\mathsf{pcf}(s,a,\psi') \in \mathsf{pre}(a)$ holds for each $a \in \mathcal{A}$, and let L be a directed cut in $\mathcal{G}$ that separates $n_\emptyset$ from n_g, such that $\mathsf{W}(L) = \min_{e\in L} \mathsf{W}(e) > 0$. Then,*

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

Proof. Let π be a plan for Π^{RT}. Let us construct a sub-sequence π' of the plan π. Let a_g be the first action in π that achieves the atom g.

By construction $s \not\models g$, thus such an action should exist. For the action a_g we choose the first action in π that achieves $\mathsf{pcf}(s, a_g, g)$, and repeat the process until we reach a fact ψ such that $s \models \psi$. By construction, π' induces a path from $n_\emptyset$ to n_g in the JG. Thus, for every cut L that separates $n_\emptyset$ from n_g we have that at least one fact in $\partial^{\mathsf{in}}(L)$ is achieved by π', and $\pi' \cap \mathsf{lbl}(L) \neq \emptyset$. Thus, $\partial^{\mathsf{in}}(L)$ is a disjunctive fact landmark.

Note that a_0, an artificial action label, is never included in L. If $a_0 \in \mathsf{lbl}(L)$, it holds that $\exists\ (n_\emptyset, n_\psi, a_0) \in L, n_\psi \in N^g$. Since the cost of a_0 is zero, $n_\emptyset \in N^g$, and this contradicts that $L = (N^0, N^g)$. Therefore, $\mathsf{lbl}(L)$ is a disjunctive action landmark. $\square$

The proof of Lemma 1 is based on the property that $\mathsf{pcf}(s,a,\psi') \in \mathsf{pre}(a)$. The lemma cannot be extended to the JG associated with $h_{\text{hbd}}^{\text{LM-cut}}$ since $\mathsf{pcf}(s,a,\psi) \in \mathsf{pre}(a)$ does not necessarily hold.

In addition to showing how to obtain numeric landmarks from a JG, Lemma 1 allows us to state and prove the main theoretical claim of this section: the numeric LM-cut heuristics $h_{\text{cri}}^{\text{LM-cut}}$, $h_{\text{ir}}^{\text{LM-cut}}$, and $h_{\text{ir},\mathsf{m}}^{\text{LM-cut}}$ are admissible. In particular, Theorem 1 shows that the weight of a cut extracted in each iteration of numeric LM-cut is admissible for a task resulting from cost-partitioning. This is the key theoretical result to prove the admissibility.

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}, where $\mathsf{pcf}(s,a,\psi') \in \mathsf{pre}(a)$ holds for each $a \in \mathcal{A}$ and $\mathsf{W_a}(s,a,\psi') = \mathsf{m}_a(s,\psi')$. Let N^0, N^b and N^g be before-, beyond- and goal zones, as defined above. Let $L = (N^0, N^g)$ be a directed cut in $\mathcal{G}$.*

The heuristic value $h_1(s) = \mathsf{W}(L)$ is admissible for Π_1^{RT}, where Π_1^{RT} is a copy of Π^{RT}, with the augmented cost function cost_1

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

where $\mathsf{m}_a^{\min}(L) = \min_{(n_\psi, n_{\psi'}, a)\in L} \mathsf{m}_a(s,\psi')$.

Proof. First, we show that $\mathsf{cost}_1(a) \leq \mathsf{cost}(a)$ for all $a \in \mathcal{A}$ to ensure that $\mathsf{cost}^c(a) = \mathsf{cost}(a) - \mathsf{cost}_1(a) \geq 0$, i.e., action costs are non-negative in every iteration of numeric LM-cut. The cost partitioning condition $\sum_{i=1}^n \mathsf{cost}_i(a) \leq \mathsf{cost}(a)$ follows then by the telescopic argument. The claim is clear for $a \notin \mathsf{lbl}(L)$, otherwise there is $(n_\psi, n_{\psi'}, a) \in L$ such that $\mathsf{W}(n_\psi, n_{\psi'}, a) = \mathsf{m}_a^{\min}(L) \cdot \mathsf{cost}(a)$, thus

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

To finish the claim, we show that the weight of L is an admissible estimate for the solution of Π_1^{RT}. Assume in contradiction that it is not, i.e., there is a plan π such that

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

By Lemma 1 point 1, $\partial^{\mathsf{in}}(L)$ is a disjunctive fact landmark. Thus, there is at least one fact in $\partial^{\mathsf{in}}(L)$ that is achieved by the plan π.[3] We denote by ψ_0 the first fact in $\partial^{\mathsf{in}}(L)$ that is achieved by π. Note that $\mathsf{lbl}(\mathsf{in}(n_\psi)) = \mathsf{supp}(\psi)$; we write $L_\psi = L \cap \mathsf{in}(n_\psi)$. Since ψ_0 is the first fact achieved in $\partial^{\mathsf{in}}(L)$, it is achieved by an action in $\mathsf{lbl}(L)$; otherwise, ψ_0 is achieved by an action that is applicable only after achieving some other fact in $\partial^{\mathsf{in}}(L)$, which contradicts the assumption. Therefore,

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

constitutes a lower bound on achieving the fact 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

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

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

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

Note that Theorem 1 assumes that the JG utilizes the action multiplier m_a for $a \in \mathcal{A}$. This assumption holds for $h^{\text{LM-cut}}_{\text{cri}}$ and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$, but not for $h^{\text{LM-cut}}_{\text{ir}}$. Nonetheless, the proof of Theorem 1 also holds for $h^{\text{LM-cut}}_{\text{ir}}$ if we replace m_a with 1.

We formalize our admissibility results for $h^{\text{LM-cut}}_{\text{cri}}$, $h^{\text{LM-cut}}_{\text{ir}}$, and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ in the following corollaries. Note that the following proof mostly relies on the admissibility of the JG cuts (i.e., Theorem 1) and the results from the cost-partition literature (i.e., Proposition 3).

Corollary 1 (Admissibility)**.** *The LM-cut heuristics $h^{\text{LM-cut}}_{\text{cri}}$, $h^{\text{LM-cut}}_{\text{ir}}$, and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ are admissible and can be computed in polynomial time.*

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

Proof. Admissibility follows from Theorem 1 and Proposition 3. In each iteration, LM-cut increases the h-value by $\mathsf{W}(L)$, where L is the extracted cut. As shown in Theorem 1, $\mathsf{W}(L)$ is admissible for Π_1^{RT}. In Step 4 of LM-cut, the cost function is updated to be cost^c, which is defined to be $\mathsf{cost}^c(a) = \mathsf{cost}(a) - \mathsf{cost}_1(a)$ for each action a. Therefore, LM-cut incrementally performs cost-partitioning and increases the h-value by an admissible estimate for a task in the cost-partition in each iteration, so the admissibility is guaranteed by Proposition 3.

The computation of the values of $h^{\max}_{\text{cri}}(s, g)$ or $h^{\max}_{\text{ir}}(s, g)$ and the construction of the corresponding 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 $a \in \mathsf{lbl}(L)$ that goes to zero.

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

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

Thus, the updated cost of $a \in \mathcal{A}$ is

$$\mathsf{cost}_1(a) = \mathsf{cost}(a) - \frac{\mathsf{W}(L)}{\mathsf{m}_a^{\min}(L)} = \mathsf{cost}(a) - \frac{\mathsf{m}_a^{\min}(L) \cdot \mathsf{cost}(a)}{\mathsf{m}_a^{\min}(L)} = 0.$$

If we replace $\mathsf{m}_a(s, \psi')$ with 1, the claim also holds. $\square$

5.4 Tightening $h^{\text{LM-cut}}_{\text{cri}}$

In what follows, we present three different alternatives to improve the $h^{\text{LM-cut}}_{\text{cri}}$ and other possible heuristics for RT. These variants are based on rounding mechanisms and slight modifications of the action multiplier m_a in the hopes to build a heuristic that is admissible and dominates its predecessor (e.g., $h^{\text{LM-cut}}_{\text{cri}}$). While some of these results are general and can be applied to other heuristics, we mainly focus on improving $h^{\text{LM-cut}}_{\text{cri}}$.

Our first attempt utilizes a rounded-up version of the action multiplier m_a by replacing $\mathsf{m}_a(s, \psi)$ with $\lceil \mathsf{m}_a(s, \psi) \rceil$. Unfortunately, as it is shown Example 6 the resulting heuristic is inadmissible when applied to $h^{\text{LM-cut}}_{\text{cri}}$.

Example 6. *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	$\emptyset$	$v \mathrel{+}= 4$	4
a_2	$\emptyset$	$v \mathrel{+}= 2$	3

An optimal plan is $\pi = \langle a_1, a_2 \rangle$ with the cost of 7. Note that $\mathsf{m}_{a_1}(s_I, v \geq 6) = 1.5$, $\lceil \mathsf{m}_{a_1}(s_I, v \geq 6) \rceil = 2$, and $\mathsf{m}_{a_2}(s_I, v \geq 6) = \lceil \mathsf{m}_{a_2}(s_I, v \geq 6) \rceil = 3$. Thus, while for the basic version of numeric LM-cut we have

$$\begin{aligned} h^{\text{LM-cut}}_{cri}(s_I) &= \min\{\mathsf{m}_{a_1}(s_I, v \geq 6) \cdot \mathsf{cost}(a_1), \mathsf{m}_{a_2}(s_I, v \geq 6) \cdot \mathsf{cost}(a_2)\} \\ &= \min\{6, 9\} = 6 < h^*(s_I) = 7, \end{aligned}$$

for the rounded up version, the JG has two edges of the following weights (see Figure 5):

$$\lceil \mathsf{m}_{a_1}(s_I, v \geq 6)\rceil \mathsf{cost}(a_1) = 8 \text{ and } \mathsf{m}_{a_2}(s_I, v \geq 6)\mathsf{cost}(a_2) = \lceil \mathsf{m}_{a_2}(s_I, v \geq 6)\rceil \mathsf{cost}(a_2) = 9.$$

This results in

$$h^*(s_I) = 7 < 8 = \min\{\lceil \mathsf{m}_{a_1}(s_I, v \geq 6)\rceil \mathsf{cost}(a_1), \lceil \mathsf{m}_{a_2}(s_I, v \geq 6)\rceil \mathsf{cost}(a_2)\}.$$

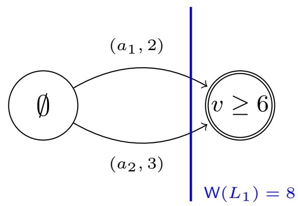


Figure 5: The JG with rounded up m_a values for an RT in Example 6.

Considering the unsuccessful attempt of rounding-up the action multipliers, we propose another variant for this multipliers, denoted by m_a^+. This variant has the property that $\mathsf{m}_a(s,\psi) \leq \mathsf{m}_a^+(s,\psi)$ and, as shown in what follows, it preserves the admissibility of the heuristic. Intuitively, this change should result in a heuristic that dominates the variant that utilize m_a. However, as we explain in what follows, this intuition holds partially at best.

We start by defining the new action multiplier m_a^+ and its corresponding $h^{\max}$ heuristic (i.e., $h^{\max}_{\text{cri},+}$). We then define the LM-cut variant based on $h^{\max}_{\text{cri},+}$, that is, $h^{\text{LM-cut}}_{\text{cri},+}$.

Definition 18. *Given an* RT *and a state s, the heuristic function $h^{\max}_{cri,+}(s) := h^{\max}_{cri,+}(s, G)$ is defined as follows. 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 $\psi \in \mathcal{F}_p \cup \mathcal{F}_n$,

$$h^{\max}_{cri,+}(s,\psi) = \begin{cases} 0 & \text{if } s \models \psi, \\ \min_{a \in \mathsf{supp}(\psi)} h^{\max}_{cri,+}(s, \mathsf{pre}(a)) + \mathsf{m}_a^+(s,\psi)\cdot \mathsf{cost}(a) & \text{otherwise} \end{cases}$$

where

$$\mathsf{m}_a^+(s,\psi) = \max\{1, \mathsf{m}_a(s,\psi)\}.$$

if $s \not\models \psi$, and $\mathsf{m}_a^+(s,\psi) = \mathsf{m}_a(s,\psi)$ otherwise.

Definition 19. *Heuristic $h^{\text{LM-cut}}_{cri,+}$ is a numeric LM-cut heuristic using a precondition choice function such that*

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

and an action weight function

$$\mathsf{W_a}(s, a, \psi') = \mathsf{m}_a^+(s, \psi').$$

In addition, given the similarities of $h^{\text{LM-cut}}_{\text{cri}}$ and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$, we define a new variant of $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ that utilize m_a^+ instead of m_a.

Definition 20. *Heuristic $h^{\text{LM-cut}}_{ir,\mathsf{m}^+}$ is a numeric LM-cut heuristic using a precondition choice function such that*

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

and an action weight function

$$\mathsf{W_a}(s, a, \psi') = \mathsf{m}_a^+(s, \psi').$$

In what follows we show that $h^{\text{LM-cut}}_{\text{cri},+}$ and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$ are indeed admissible heuristics. The admissibility proof is analogous to the proof of Theorem 1 by replacing m_a with m_a^+. Note that the admissibility is still preserved because we need to apply an action at least once to achieve a numeric condition.

Theorem 2. *The LM-cut heuristics $h^{\text{LM-cut}}_{cri,+}$ and $h^{\text{LM-cut}}_{ir,\mathsf{m}^+}$ are admissible.*

Intuitively, we would expect that $h^{\text{LM-cut}}_{\text{cri},+}$ would dominate $h^{\text{LM-cut}}_{\text{cri}}$ because $\mathsf{m}_a(s, \psi) \leq \mathsf{m}_a^+(s, \psi)$. However, as shown in Appendix B, this is not always the case. Moreover, the difference between m_a and m_a^+ is so small that in most tasks the two heuristics have the same values. Formally, for a state s where there are no actions a and facts ψ such that $\mathsf{m_a}(s, \psi) < 1$, heuristics $h^{\text{LM-cut}}_{\text{cri},+}$ and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$ are exactly the same as $h^{\text{LM-cut}}_{\text{cri}}$ and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$, respectively.

Lastly, we present another variant of $h^{\text{LM-cut}}_{\text{cri}}$ that rounds the heuristic value instead of each action multiplier m_a. This rounding strategy is valid if we assume that action costs are integer. While the assumption does not hold in general, we can generalize this improvement if the costs of the actions are rational; when computing LM-cut, we can ensure that all actions have integer costs by multiplying the action costs by some constant k and obtain the admissible estimate for the original task by dividing the computed heuristic value by k. Definition 21 presents the procedure to tighten the heuristic values.

Definition 21. *Given an* RT $\Pi^{\text{RT}} = \langle \mathcal{F}, \mathcal{N}, \mathcal{A}, s_I, G \rangle$, *state s, and heuristic h, rounded-up heuristic value $\overline{h}(s)$ is computed as follows:*

1. *Find a constant k such that $k \cdot \mathsf{cost}(a)$ is integer for all actions $a \in \mathcal{A}$.*
2. *Let $h_k(s)$ be $h(s)$ computed on $\Pi^{\text{RT}}_k = \langle \mathcal{F}, \mathcal{N}, \mathcal{A}_k, s_I, G \rangle$ where*

$$\mathcal{A}_k = \{\langle \mathsf{pre}(a), \mathsf{eff}(a), k \cdot \mathsf{cost}(a) \rangle \mid a \in \mathcal{A}\}.$$

3. *Return $\frac{\lceil h_k(s) \rceil}{k}$.*

The main advantage of the above procedure is that it guarantees that $\overline{h}$ dominates h for $h = h^{\text{LM-cut}}_{\text{cri}}$ because of the round-up step (see Theorem 4 for further details). As shown in Theorem 3, heuristic $\overline{h}$ is always admissible as long as its predecessor (i.e., h) is also admissible. The rounding procedure introduced in Definition 21 can be applied to any heuristic for RT with rational action cost, because its definition does not assume anything about the original h heuristic.

Theorem 3. *If h is admissible, then $\overline{h}$ is admissible.*

Proof. Let $h^*(s)$ and $h^*_k(s)$ be the optimal costs from s in Π^{RT} and Π^{RT}_k, respectively. By definition of Π^{RT}_k, we have $h^*_k(s) = kh^*(s)$. Since $h_k(s)$ is admissible on Π^{RT}_k, it holds that $h_k(s) \leq h^*_k(s)$. Since all actions have integer costs in Π^{RT}_k, $h^*_k(s)$ is integer, so $\lceil h_k(s) \rceil \leq h^*_k(s)$.

$$\overline{h}(s) = \frac{\lceil h_k(s) \rceil}{k} \leq \frac{h^*_k}{k} = h^*.$$

Thus, $\overline{h}$ is admissible for Π^{RT}. □

5.5 LM-Cut and Max – Theoretical Comparison

This section theoretically compares the admissible $h^{\max}$ and $h^{\text{LM-cut}}$ numeric variants presented so far. Specifically, we consider the following heuristics in our comparison:

- $h^{\max}$ relaxations: $h^{\max}_{\text{ir}}$ (Aldinger & Nebel, 2017), $h^{\max}_{\text{hbd}}$ (Scala et al., 2016);
- $h^{\text{LM-cut}}$ versions:
  - $h^{\text{LM-cut}}_{\text{ir}}$: $h^{\max}_{ir}$ based pcf and 1 as an action weight function;
  - $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$: $h^{\max}_{ir}$ based pcf and m_a as an action weight function;
  - $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$: $h^{\max}_{ir}$ based pcf and m^+_a as an action weight function;
  - $h^{\text{LM-cut}}_{\text{cri}}$: $h^{\max}_{\text{cri}}$ based pcf and m_a as an action weight function;
  - $h^{\text{LM-cut}}_{\text{cri},+}$: $h^{\max}_{\text{cri},+}$ based pcf and m^+_a as an action weight function;
  - $\overline{h}^{\text{LM-cut}}_{\text{cri}}$: $h^{\text{LM-cut}}_{\text{cri}}$ rounded up according to Definition 21.

Table 6 summarizes the pair-wise comparison among the eight heuristics and h^+. The table shows that there are few dominance relations among these heuristics. Some surprising results are, for example, that $h^{\text{LM-cut}}_{\text{cri},+}$ does not dominate $h^{\text{LM-cut}}_{\text{cri}}$ and that this heuristic is incomparable with $h^{\max}_{\text{ir}}$ and $h^{\max}_{\text{hbd}}$. In what follows, we present the dominance proofs among these heuristics: $\overline{h}^{\text{LM-cut}}_{\text{cri}}$ dominates $h^{\text{LM-cut}}_{\text{cri}}$ (Theorem 4) and heuristics $h^{\text{LM-cut}}_{\text{ir}}$ and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$ dominate $h^{\max}_{\text{ir}}$ (Theorem 5). We refer the reader to Appendix B for the examples that show that two heuristics are incomparable.

Theorem 4. *Given an* RT *task Π^{RT} and any state s, the following relation holds $\overline{h}(s) \geq h(s)$ when $h = h^{\text{LM-cut}}_{cri}$. Therefore, $\overline{h}^{\text{LM-cut}}_{cri}$ dominates $h^{\text{LM-cut}}_{cri}$.*

Proof. Let $h_k(s)$ be the h-value of s computed by $h^{\text{LM-cut}}_{\text{cri}}$ on Π^{RT}_k. We first show that

$$h_k(s) = k \cdot h^{\text{LM-cut}}_{\text{cri}}(s)$$

by examining the steps in Definition 12. In Step 1, $h_k(s)$ is initialized to be 0. In Step 2, since all action costs are scaled by the same factor, $h^{\max}_{\text{cri}}$-values are just scaled by k in Π^{RT}_k. Therefore, the precondition choice function pcf is not changed, and a JG is not changed except that the weight $\mathsf{W}_k(e)$ for an edge $e = (n_{\psi'}, n_\psi, a)$ is

$$\mathsf{W}_k(e) = \mathsf{m}_a(s, \psi) \cdot k \cdot \mathsf{cost}(a) = k \cdot \mathsf{W}(e).$$

	$h^{\max}_{\text{hbd}}$	$h^{\max}_{\text{ir}}$	$h^{\text{LM-cut}}_{\text{ir}}$	$h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$	$h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$	$h^{\text{LM-cut}}_{\text{cri}}$	$h^{\text{LM-cut}}_{\text{cri},+}$	$\overline{h}^{\text{LM-cut}}_{\text{cri}}$
h^+	$\geq$	$\geq$	$\geq$	$\geq$	$\geq$	$\geq$	$\geq$	$\geq$
$h^{\max}_{\text{hbd}}$		$\neq$	$\neq$	$\neq$	$\neq$	$\neq$	$\neq$	$\neq$
$h^{\max}_{\text{ir}}$			$\leq$	$\neq$	$\leq$	$\neq$	$\neq$	$\neq$
$h^{\text{LM-cut}}_{\text{ir}}$				$\neq$	$\neq$	$\neq$	$\neq$	$\neq$
$h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$					$\neq$	$\neq$	$\neq$	$\neq$
$h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$						$\neq$	$\neq$	$\neq$
$h^{\text{LM-cut}}_{\text{cri}}$							$\neq$	$\leq$
$h^{\text{LM-cut}}_{\text{cri},+}$								$\neq$

Table 6: Dominance relationships between the max and LM-cut heuristics for RT. '$\geq$' means the heuristic in a row dominates the heuristic in a column, '$\leq$' means the converse, and '$\neq$' means that the two heuristics are incomparable.

The before-goal zone and the goal zone are the same as those in Π^{RT} since $\mathsf{W}_k(e) = 0$ iff $\mathsf{W}(e) = 0$. In Step 3, the same cut is extracted by LM-cut, and $h_k(s)$ is increased by $k \cdot \mathsf{W}(w)$. In Step 4, the weight function ensures that the updated cost function, denoted cost^c_k, satisfies

$$\mathsf{cost}^c_k(a) = k \cdot \mathsf{cost}^c(a)$$

for each action $a \in \mathcal{A}$ since

$$\mathsf{cost}^c_k(a) = \mathsf{cost}_k(a) - \frac{\mathsf{W}_k(L)}{\mathsf{m}^{\min}_a(L)} = k \cdot \left(\mathsf{cost}(a) - \frac{\mathsf{W}(L)}{\mathsf{m}^{\min}_a(L)}\right)$$

if $a \in \mathsf{lbl}(L)$, and

$$\mathsf{cost}^c_k(a) = \mathsf{cost}_k(a) = k \cdot \mathsf{cost}(a)$$

otherwise. Since the same cut L is extracted and $h_k(s)$ is increased by $k \cdot \mathsf{W}(L)$ in each iteration, $h_k(s) = k \cdot h^{\text{LM-cut}}_{\text{cri}}(s)$. Thus,

$$\overline{h}^{\text{LM-cut}}_{\text{cri}}(s) = \frac{\lceil h_k(s) \rceil}{k} \geq \frac{k \cdot h^{\text{LM-cut}}_{\text{cri}}(s)}{k} = h^{\text{LM-cut}}_{\text{cri}}(s). \qquad \square$$

Theorem 5. *Given an* RT Π^{RT} *and any state* s*, the following relations hold:* $h^{\text{LM-cut}}_{ir}(s) \geq h^{\max}_{ir}(s)$ *and* $h^{\text{LM-cut}}_{ir,\mathsf{m}^+}(s) \geq h^{\max}_{ir}$*. Therefore,* $h^{\text{LM-cut}}_{ir}$ *and* $h^{\text{LM-cut}}_{ir,\mathsf{m}^+}$ *dominate* $h^{\max}_{ir}$*.*

Proof. Following Helmert and Domshlak (2009), we show that in each iteration of $h^{\text{LM-cut}}_{\text{ir}}$, the $h^{\max}_{\text{ir}}$-value of the task is reduced by at most the weight of the extracted cut. Note that this is not the case with $h^{\text{LM-cut}}_{\text{cri}}$; in Example 13 in Appendix B, in the second iteration, the critical path supporting the $h^{\text{LM-cut}}_{\text{cri}}$-value contains more than one action in cut L_2, so the $h^{\max}_{\text{cri}}$-value is reduced by $\mathsf{W}_1(L_2) + \frac{\mathsf{W}_1(L_2)}{2} = 3 > \mathsf{W}_1(L_2) = 2$.

Given an Π^{RT}, a state s with $h^{\max}_{\text{ir}}(s) > 0$, the JG of $h^{\max}_{\text{ir}}$, and a cut extracted by $h^{\text{LM-cut}}_{\text{ir}}$ L, let π be the shortest path from $n_\emptyset$ to n_g in the JG. By Proposition 5 we know that JG justifies $h^{\max}_{\text{ir}}$, i.e., $h^{\max}_{\text{ir}}(s) = \mathsf{W}(\pi)$.

Assume that π includes more than one label in $\mathsf{lbl}(L)$, and assume it is of the form

$$\pi = (n_\emptyset, \dots, n_i, a_i, n_{i+1}, \dots, n_j, a_j, n_{j+1}, \dots, n_g), \text{ where } a_i, a_j \in \mathsf{lbl}(L).$$

Since $a_i \in \mathsf{lbl}(L)$, there exists a node n_{ψ_0} in the goal-zone such that $a_i \in \mathsf{supp}(\psi_0)$. $n_{\psi_0} \in N^g$ implies that there is a zero-weight path from n_{ψ_0} to n_g, i.e., $\mathsf{W}(\mathsf{path}(n_{\psi_0}, n_g)) = 0$. Note that all edges with the same label have the same origin node, thus (n_i, a_i, n_{ψ_0}) is a valid edge in JG.

We define a new path $\pi' = (n_\emptyset, \dots, n_i, a_i, n_{\psi_0}, \dots, n_g)$ that coincides on its prefix with π, both on vertices and edges, starting from $n_\emptyset$ and up to a_i (note that more than one edge may have this label), and its suffix from n_{ψ_0} to n_g is a zero-cost path in the goal-zone.

Since in a JG constructed via $h^{\text{LM-cut}}_{\text{ir}}$ each edge $(n_\psi, n_{\psi'}, a)$ has the weight of $\mathsf{cost}(a)$, not $\mathsf{m}_a(s, \psi') \cdot \mathsf{cost}(a)$, two edges (n_i, n_{i+1}, a_i) and (n_i, n_{ψ_0}, a_i) have the same weight, hence $\mathsf{W}(\pi) \geq \mathsf{W}(\pi') + \mathsf{cost}(a_j)$. Recall that $a_j \in \mathsf{lbl}(L)$. Since $h^{\max}_{\text{ir}}(s) > 0$, the weight of L should be non-zero, hence $\mathsf{cost}(a_j) > 0$. Thus, $\mathsf{W}(\pi) > \mathsf{W}(\pi')$, which contradicts the assumption of π having the minimum weight. Therefore, π includes at most one edge in L.

In $h^{\text{LM-cut}}_{\text{ir}}$, the cost of an action a is updated to be $\mathsf{cost}(a) - \mathsf{W}(L)$ only if $a \in \mathsf{lbl}(L)$. Since π includes at most one edge in L, the weight of π is reduced by at most by $\mathsf{W}(L)$. Thus, the $h^{\max}_{\text{ir}}$-value of the next iteration is at least $h^{\max}_{\text{ir}}(s) - \mathsf{W}(L)$.

In the case of $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$, the action cost of a is reduced by $\frac{\mathsf{W}(L)}{\mathsf{m}^{\min}_a(L)}$. If $\mathsf{m}^{\min}_a(L) < 1$, the $h^{\max}_{\text{ir}}$-value is reduced by more than $\mathsf{W}(L)$. However, for $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$, it is guaranteed that $\mathsf{m}^{\min}_a(L) \geq 1$, so the above statement holds.

Finally, we show the dominance. Let m be the number of cuts extracted by $h^{\text{LM-cut}}_{\text{ir}}$, $\mathcal{G}_i$ be the JG in the i-th iteration, L_i be the cut extracted in the i-th iteration, and W_i be the weight function after the i-th iteration with $\mathsf{W}_0 = \mathsf{W}$. Let $h_i(s)$ be the $h^{\max}_{\text{ir}}$-value computed in Π^c_i. As we showed above, for $i \in [m]$, $h_i(s) \geq h_{i-1}(s) - \mathsf{W}_{i-1}(L_i)$. Therefore, we can bound the heuristics value by a telescoping sum

$$h^{\text{LM-cut}}_{\text{ir}}(s) = \sum_{i=1}^{m} \mathsf{W}_{i-1}(L_i) \geq \sum_{i=1}^{m} h_{i-1}(s) - h_i(s) = h_0(s) - h_m(s).$$

Since $h_m(s) = 0$ as LM-cut terminates after extracting the m-th cut,

$$h^{\text{LM-cut}}_{\text{ir}}(s) \geq h_0(s) = h^{\max}_{\text{ir}}(s).$$

The above discussion also holds for $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$. □

5.6 Empirical Evaluation of Numeric $h^{\text{LM-cut}}$ and $h^{\max}$ variants

As shown in the previous section, most numeric $h^{\max}$ and $h^{\text{LM-cut}}$ heuristics are theoretically incomparable. In what follows, we empirically evaluate the performance of these heuristics to investigate which alternatives work well in practice. We compare our numeric LM-cut variants with the $h^{\max}$-relaxation based heuristics: the repetition relaxation based max heuristic $h^{\max}_{\text{ir}}$ (Aldinger & Nebel, 2017), and the numeric max heuristic $\hat{h}^{\text{rmax}}_{\text{hbd+}}$ (Scala et al., 2020). We consider $\hat{h}^{\text{rmax}}_{\text{hbd+}}$ instead of $h^{\max}_{\text{hbd}}$ in this evaluation because the former is an improved version of the latter which has shown better empirical performance in the literature (Scala et al., 2020).

Our evaluation considers all the admissible LM-cut variants introduced in the previous sections, that is, $h^{\text{LM-cut}}_{\text{cri}}$, $h^{\text{LM-cut}}_{\text{cri},+}$, $h^{\text{LM-cut}}_{\text{ir}}$, $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$, and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$. We note that in all the tested domains actions have rational costs, which allow us to also consider $\overline{h}^{\text{LM-cut}}_{\text{cri}}$. Our $\overline{h}^{\text{LM-cut}}_{\text{cri}}$ implementation first finds the minimum non-negative integer i such that $10^i\mathsf{cost}(a)$ is integer for all $a \in \mathcal{A}$ and use $k = 10^i$. In addition, we define a randomized $h^{\text{LM-cut}}_{\text{cri}}$ variant, that is, $h^{\text{LM-cut}}_{\text{rnd}}$, where $\mathsf{pcf}(s, a)$ is chosen uniformly at random from the set $\{\psi \in \mathsf{pre}(a) \mid h^{\max}_{\text{cri}}(s, \psi) > 0\}$, and g is chosen uniformly at random from the set $\{g' \in G \mid h^{\max}_{\text{cri}}(s, g') > 0\}$. Note that Theorem 1 guarantees the admissibility of $h^{\text{LM-cut}}_{\text{rnd}}$. The main purposes of this randomize LM-cut version is as a baseline for JG construction and as a sanity check.

In all the experiments, we evaluate the heuristics inside an 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 & Nebel, 2017)[4] using C++11 with GCC 7.5.0 on Ubuntu 18.04.

In terms of implementation, we follow the LM-cut implementation for classical planning included in Fast Downward (Helmert, 2006). In the first iteration of this implementation, $h^{\max}$-values are computed and a JG is constructed by the generalized Dijkstra algorithm (Keyder & Geffner, 2008). A priority queue is initialized to contain all facts satisfied in the initial state with the priority of 0. At each step, a fact with the minimum priority is popped from the queue and marked as achieved, and its $h^{\max}$-value is set to be the priority. When all preconditions of action a are achieved, each fact ψ achieved by a, i.e., ψ such that $a \in \mathsf{supp}(\psi)$, is pushed to the queue with the priority of $\mathsf{cost}(a) + h^{\max}(s, \mathsf{pre}(a))$. This procedure is repeated until all goal conditions are achieved. After the first iteration of LM-cut, JGs are constructed incrementally; $h^{\max}$-values and pcf are recomputed only if they are changed due to updated action costs. For each action a included in the cut in the previous iteration, each fact ψ achieved by a is pushed to the queue with the priority of $\mathsf{cost}^c(a) + h^{\max}(s, \mathsf{pre}(a))$, where $\mathsf{cost}^c(a)$ is the updated cost of a. The pseudo-code is presented in Appendix C. Note that we incrementally construct JGs to compute a heuristic value for a single state; we do not incrementally compute heuristic values on multiple states, which is a method proposed by Pommerening and Helmert (2013).

We consider domains with simple conditions from the literature (Scala et al., 2016, 2017, 2020). We exclude ZENOTRAVEL because some conditions are not simple conditions (Piacentini et al., 2018b). From COUNTERS, we exclude three instances that are in SMALL-COUNTERS. In 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 add 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 numeric LM-cut. In COUNTERS-INV, COUNTERS-RND, FARMLAND, and FARMLAND-SAT, some numeric conditions are strict inequalities. In RT we convert a numeric condition $v > w$ to $v \geq w + \epsilon$ where ϵ is computed in a similar fashion as Defini-

4. https://github.com/Kurororoo/numeric-fast-downward

tion 21; we find the minimum non-negative integer i such that $10^i \cdot c_v$ is integer for each numeric effect $v \mathrel{+}= c_v$ and use $\epsilon = \frac{1}{10^i}$.

For the numeric heuristics, we add the redundant constraints to the goal conditions and preconditions of actions in the same fashion as Scala et al. (2016a).

Table 7 shows the experimental results across all the domains where we compared the number of solved instances (Coverage), the time score (Time score), and the number of expansions to solve an instance excluding the last f-layer (# States expanded). For each run, the time score is computed as $1 - \frac{\log(\max\{1,t\})}{\log(1800)}$, where t is the wall-clock time to solve the instance. Since the time limit is 30 minutes, this score is logarithmically decreasing according to the wall-clock time and takes value 1 if an instance is solved within 1 second. If the instance is not solved, we set the time score be 0. For each domain, the time score is averaged over all instances, and the number of expansions is averaged over instances solved by all methods. These settings are the same in all experiments in this paper. Farmland-SAT is omitted from this table because NFD runs out of memory when translating PDDL files to SAS+ files and, thus, the planner does not solve any instances. We note that this memory issue is not related to our heuristics since it occurs during to the translation part of NFD, which is performed before search and the heuristic computation.

In addition, Figure 6 depicts pairwise comparisons of expanded states between $h^{\text{LM-cut}}_{\text{cri}}$ and the other LM-cut variants. In each plot, points represent one instance where its x and y values are the numbers of expansions by $h^{\text{LM-cut}}_{\text{cri}}$ and the variant shown in y-axis. The points above the diagonal correspond to instances where the LM-cut variant expands more states than the $h^{\text{LM-cut}}_{\text{cri}}$ baseline. Similarly, Figure 7 compares initial h-values of $h^{\text{LM-cut}}_{\text{cri}}$ and the other LM-cut variants.

One of the most surprising results of this evaluation is that $h^{\text{LM-cut}}_{\text{rnd}}$ outperforms $h^{\text{LM-cut}}_{\text{ir}}$ and acts almost on par with $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$, despite the fact that $h^{\text{LM-cut}}_{\text{rnd}}$ randomly constructs justification graphs while $h^{\text{LM-cut}}_{\text{ir}}$ and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ are guided by $h^{\max}_{\text{ir}}$. This result may be a sign of low informativeness of the $h^{\max}_{\text{ir}}$ relaxations within the LM-cut framework. In contrast, $h^{\text{LM-cut}}_{\text{cri}}$ and $h^{\text{LM-cut}}_{\text{cri},+}$ show much better results than all other heuristics in terms of coverage, time, and number of expanded nodes, leading us to believe that the over-approximation made by $h^{\max}_{\text{cri}}$ is beneficial to the quality of the heuristics. We speculate that, because $h^{\max}_{\text{cri}}$ is inadmissible, it can be a more accurate approximation of h^* on average allowing both over- and under-approximations.

Figure 6 also shows that $h^{\text{LM-cut}}_{\text{cri},+}$ expands slightly fewer states than $h^{\text{LM-cut}}_{\text{cri}}$, however, the use of m^+ does not result in any significant improvements on the overall heuristic performance (i.e., coverage and time score). As shown in Figure 6, $\overline{h}^{\text{LM-cut}}_{\text{cri}}$ expands 0 states excluding the last f-layer in instances where $h^{\text{LM-cut}}_{\text{cri}}$ expands 100-10000 states. Table 7 also shows that $\overline{h}^{\text{LM-cut}}_{\text{cri}}$ is effective particularly in Sailing and Sailing-SAT, but the coverage is not improved in any of the domains. In Small-Counters $\overline{h}^{\text{LM-cut}}_{\text{cri}}$ expands more nodes than $h^{\text{LM-cut}}_{\text{cri}}$ while the former dominates the latter. This phenomenon is possible because these heuristics are not necessarily consistent.

6. Comparison of Propositional and Numeric LM-Cut

In this section, we address two questions related to numeric domains and numeric heuristics.

	$h^{\max}_{\text{ir}}$	$\hat{h}^{\text{rmax}}_{\text{hbd+}}$	$h^{\text{LM-cut}}_{\text{cri}}$	$h^{\text{LM-cut}}_{\text{cri},+}$	$\overline{h}^{\text{LM-cut}}_{\text{cri}}$	$h^{\text{LM-cut}}_{\text{rnd}}$	$h^{\text{LM-cut}}_{\text{ir}}$	$h^{\text{LM-cut}}_{\text{ir,m}}$	$h^{\text{LM-cut}}_{\text{ir,m+}}$
domain	Coverage								
SmallCounters (8)	6	**7**	**7**	**7**	**7**	**7**	6	**7**	**7**
Counters (8)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Counters-Inv (11)	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Counters-Rnd (33)	6	7	**9**	**9**	**9**	7	6	8	8
Farmland (30)	11	**30**	**30**	**30**	**30**	12	11	17	14
Gardening (63)	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**
Gardening-SAT (51)	**12**	**12**	**12**	**12**	**12**	**12**	**12**	**12**	**12**
Sailing (40)	8	25	**40**	**40**	**40**	14	10	13	13
Sailing-SAT (40)	3	9	**14**	**14**	**14**	3	3	3	3
Depots (20)	5	5	**7**	**7**	**7**	5	**7**	**7**	**7**
Rovers (20)	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**
Satellite (20)	1	1	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Total (344)	121	165	**190**	**190**	**190**	131	126	138	135
domain	Time score								
SmallCounters (8)	0.67	0.77	**0.79**	**0.79**	**0.79**	0.77	0.70	0.78	0.78
Counters (8)	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Counters-Inv (11)	**0.18**	**0.18**	**0.18**	**0.18**	**0.18**	**0.18**	**0.18**	**0.18**	**0.18**
Counters-Rnd (33)	0.18	0.19	**0.24**	**0.24**	**0.24**	0.19	0.18	0.23	0.23
Farmland (30)	0.31	0.74	0.74	**0.77**	0.74	0.36	0.33	0.57	0.42
Gardening (63)	0.90	0.94	**0.96**	**0.96**	**0.96**	0.94	0.91	0.95	0.95
Gardening-SAT (51)	0.12	0.14	**0.15**	**0.15**	**0.15**	0.14	0.12	0.14	0.14
Sailing (40)	0.10	0.42	0.90	0.91	**1.00**	0.18	0.12	0.19	0.19
Sailing-SAT (40)	0.03	0.14	0.20	0.20	**0.25**	0.04	0.04	0.05	0.05
Depots (20)	0.14	0.12	**0.22**	**0.22**	**0.22**	0.15	**0.22**	**0.22**	**0.22**
Rovers (20)	0.15	0.15	0.16	**0.17**	**0.17**	0.16	**0.17**	**0.17**	**0.17**
Satellite (20)	0.05	0.05	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**	**0.07**
domain	# States expanded								
SmallCounters (8)	113996	12662	**5699**	5703	5811	8108	51750	7743	7731
Counters (8)	-	-	-	-	-	-	-	-	-
Counters-Inv (11)	1325	158	**0**	30	**0**	96	1062	154	154
Counters-Rnd (33)	264	**0**	**0**	**0**	**0**	19	194	10	10
Farmland (30)	214884	57	57	56	**36**	25736	214802	15400	52778
Gardening (63)	91565	33046	15948	**15534**	15641	28124	78080	24006	22722
Gardening-SAT (51)	930434	486325	288808	**282926**	286263	426390	796766	361855	342849
Sailing (40)	1730733	50438	1544	1376	**0**	367630	1424337	325986	323133
Sailing-SAT (40)	3068452	155123	23238	23199	**8739**	1705729	3013637	668799	668693
Depots (20)	344868	344868	**38675**	38677	**38675**	107649	38677	38685	38677
Rovers (20)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Satellite (20)	1648	1648	**260**	**260**	**260**	262	**260**	**260**	**260**

Table 7: Coverage, time score, and # of states expanded excluding the last f-layer by the LM-cut variants. $h^{\max}_{\text{ir}}$ and $\hat{h}^{\text{rmax}}_{\text{hbd+}}$ are presented for comparison purposes.

1. In numeric domains, is it necessary to reason about numeric conditions or can they simply be ignored?

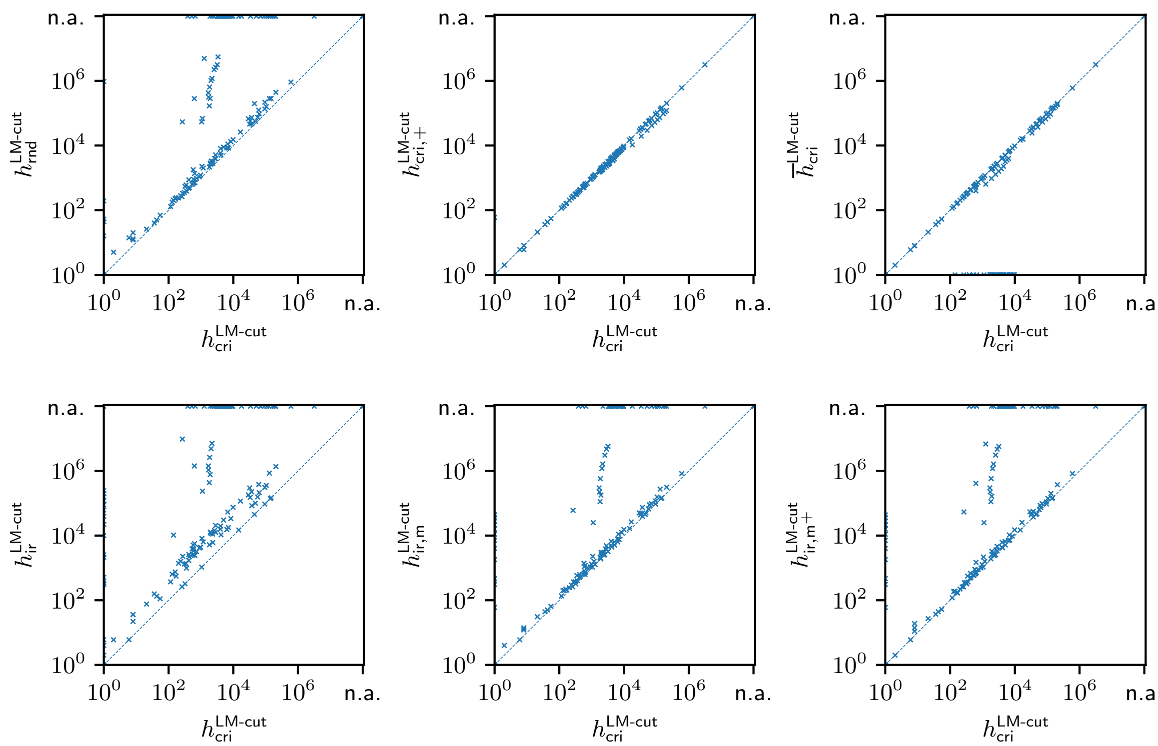


Figure 6: Comparison of the number of states expanded excluding the last f-layer by the LM-cut variants. Unsolved instances are shown at 'n.a.'. Instances solved with 0 expansions are shown at 10^0.

2. Some classical planning domains contain resource variables that can be automatically detected and represented as numeric variables (Wilhelm et al., 2018). In such domains, does a numeric reformulation of the domain with numerical reasoning (i.e., heuristics) result in better performance than the purely propositional formulation?

Note that the theoretical results in this section also apply to $\overline{h}^{\text{LM-cut}}_{\text{cri}}$, which dominates $h^{\text{LM-cut}}_{\text{cri}}$, since all action costs and heuristic values are integer in the examples.

6.1 LM-Cut: Propositional vs. Numeric Variants in Numeric Domains

Although our numeric LM-cut heuristics are designed to address numeric conditions, it is unclear if doing so is necessary. A more straightforward adaptation of the classical LM-cut to numeric planning is to ignore numeric conditions entirely by assuming that all numeric conditions are achieved with zero-cost in the computation of the JGs. When only propositional conditions are left unachieved, we can compute an admissible estimate using the propositional LM-cut heuristic.

To validate the importance of considering numeric conditions, we investigate whether numeric LM-cut provides a better estimation compared to the propositional one. In domains without propositions such as Example 2, the h-value of the propositional LM-cut is always

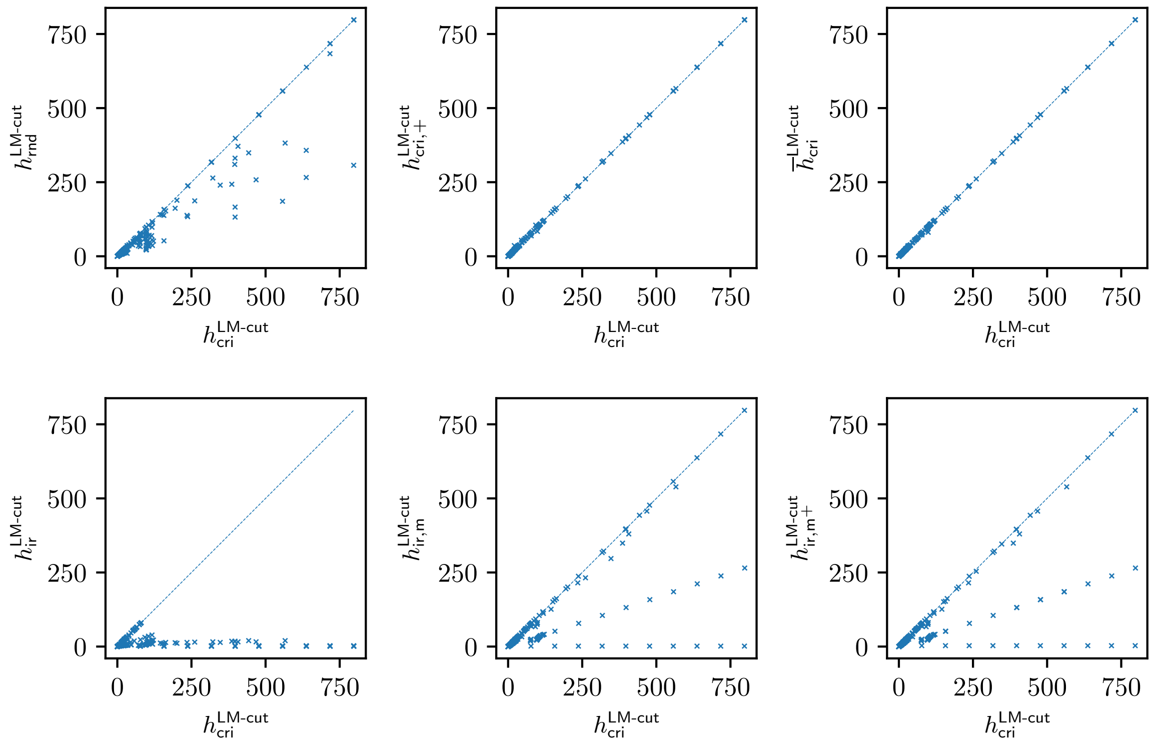


Figure 7: Comparison of the initial h-values of the LM-cut variants.

zero, so the numeric version is strictly better. However, as we show in Example 7, numeric LM-cut is not always better than the propositional, i.e., the propositional LM-cut can provide a higher h-value than its numeric counterpart.

Example 7. *Let $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G\rangle$ be an* RT *with $\mathcal{F}_p = \{p, q, r, g\}$ and $\mathcal{N} = \{v\}$. Let $s_I = \{v = 0\}$, $G = \{g\}$, and $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$, where*

action	pre	eff	cost
a_1	$\emptyset$	$v \mathrel{+}= 1, p$	1
a_2	$\emptyset$	$v \mathrel{+}= 1, q$	1
a_3	$\emptyset$	$v \mathrel{+}= 1, r$	1
a_4	$v \geq 2, p, q, r$	g	0

Table 8 shows $h^{\max}_{cri}$-values of facts and action costs in each iteration and Figure 8 depicts the corresponding JGs. In $h^{\text{LM-cut}}_{cri}$, $\mathsf{pcf}(s_I, a_4) = v \geq 2$, so a_1, a_2, and a_3 are included in the first cut, resulting in

$$h^{\text{LM-cut}}_{cri}(s_I) = 2.$$

Table 9 shows $h^{\max}$-values of facts and action costs in each iteration and Figure 9 presents the associated JGs. Since the $h^{\max}$-values of p, q, and r are the same, there can be multiple pcfs. In such a case, the tie-breaking strategy determines which fact to select for pcf. We assume that the tie-breaking strategy prefers p to q and q to r. Different tie-breaking strategies change the order of JGs but result in the same h-value for this example.

$h^{\max}_{\text{cri}}(s_I, \psi)$	$v \geq 2$	p	q	r	g
1	2	1	1	1	2
2	0	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	1	1	1	0
2	0	0	0	0

Table 8: $h^{\max}_{\text{cri}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration in Example 7.

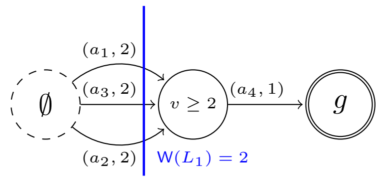


Figure 8: The JG constructed by $h^{\text{LM-cut}}_{\text{cri}}$ for an RT in Example 7.

In the propositional LM-cut, a_1, a_2, and a_3 are included in the different cuts as shown in Table 9, so we have

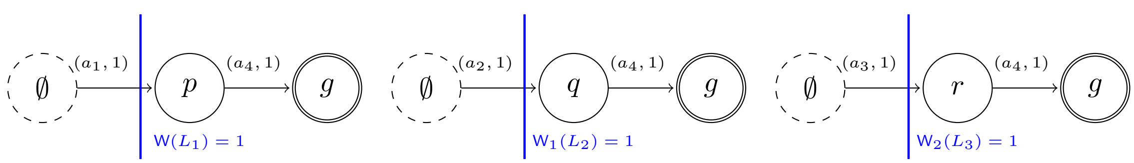
$$h^{\text{LM-cut}}(s_I) = 3 > h^{\text{LM-cut}}_{cri}(s_I) = 2.$$

Therefore, the propositional LM-cut provides a better estimate.

$h^{\max}(s_I, \psi)$	$v \geq 2$	p	q	r	g
1	0	1	1	1	1
2	0	0	1	1	1
3	0	0	0	1	1
4	0	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	1	1	1	0
2	0	1	1	0
3	0	0	1	0
4	0	0	0	0

Table 9: $h^{\max}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration.



(a) The first cut. (b) The second cut. (c) The third cut.

Figure 9: JGs constructed by propositional $h^{\text{LM-cut}}$ for an RT in Example 7. The functions W, W_1, and W_2 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

6.2 Translation of Classical Domains to Numeric Domains

Previous research has shown that some classical planning domains contain resource variables, which can be automatically detected (Wilhelm et al., 2018). Using the detected resource variables as numeric variables, we can translate the classical task into RP, which can be translated into RT. Therefore, we can obtain a numeric encoding of a task which is originally formulated with the classical planning formalism. In what follows we evaluate the benefit of encoding tasks using numeric facts instead of propositions by comparing the LM-cut heuristics in the classical and numeric versions of the same planning task.

Our comparison considers our best performing LM-cut variant, $h^{\text{LM-cut}}_{\text{cri}}$, and its classical planning counterpart $h^{\text{LM-cut}}$. In each classical panning task, we compute $h^{\text{LM-cut}}$ on the original task, while $h^{\text{LM-cut}}_{\text{cri}}$ is computed on the RTs obtained by the translation algorithm proposed by Wilhelm et al. (2018). We show that $h^{\text{LM-cut}}$ and $h^{\text{LM-cut}}_{\text{cri}}$ are incomparable in theory using two example classical planning tasks and their numeric translations obtained by the algorithm.[5]

Example 8. *Let* $\Pi = \langle \mathcal{F}_p, \mathcal{A}, s_I, G\rangle$ *be a classical planning task with* $\mathcal{F}_p = \{p_0, p_1, q_0, q_1, v_0, v_1, v_2\}$, $s_I = \{p_0, q_0, v_0\}$, *and* $G = \{p_1, q_1\}$. *Let* $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5, a_6\}$, *where*

action	pre	add	del	cost
a_1	p_0	p_1	p_0	1
a_2	p_0, v_0	v_1	v_0	1
a_3	p_0, v_1	v_2	v_1	1
a_4	p_1, v_0	v_1	v_0	1
a_5	p_1, v_1	v_2	v_1	1
a_6	q_0, v_2	q_1, v_0	q_0, v_2	1

We assume that the tie-breaking strategy prefers q_1 *to* p_1, p_1 *to* v_1, *and* v_0 *to* p_0.

We show $h^{\max}$*-values of facts and action costs in each iteration in Table 10 and JGs in Figure 10. In this figure, edge* (n_{v_2}, n_{v_0}, a_6) *is omitted. Note that* $\mathsf{cost}(a_0) = 0$, *so* L_3 *with* $\mathsf{lbl}(L_3) = \{a_1, a_2\}$ *is the last cut. We have*

$$h^{\text{LM-cut}}(s_I) = 3.$$

$h^{\max}(s_I, \psi)$	v_0	p_0	q_0	q_1	p_1	v_1	v_2
1	0	0	0	3	1	1	2
2	0	0	0	2	1	1	2
3	0	0	0	1	1	1	1
4	0	0	0	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5	a_6
1	1	1	1	1	1	1
2	1	1	1	1	1	0
3	1	1	0	1	0	0
4	0	0	0	1	0	0

Table 10: $h^{\max}(s, \psi)$ and $\mathsf{cost}(a)$ in each iteration in Figure 10.

In an FDR version of this task, if there are three variables where the first one is for p_0 *and* p_1, *the second is for* q_0 *and* q_1, *and the last is for* v_0, v_1, *and* v_2, *the last variable can be translated into a resource variable. The resulting* RT *task is* $\langle \mathcal{F}_p^{\text{RT}}, \mathcal{N}, \mathcal{A}^{\text{RT}}, s_I^{\text{RT}}, G\rangle$ *where* $\mathcal{F}_p^{\text{RT}} = \{p_0, p_1, q_0, q_1\}$, $\mathcal{N} = \{v, u\}$, $s_I^{\text{RT}} = \{p_0, q_0, v = 0, u = 0\}$. *Here, two numeric variables* v *and* u *are introduced to represent the upper and lower bounds of the resource variable.* $\mathcal{A}^{\text{RT}} = \{a_1, a_2, a_4, a_6\}$, *where*

5. Although the algorithm relies on the finite-domain representation (FDR) of classical planning tasks, we show STRIPS planning tasks since we use that formalism here. For simplicity, we only mention the characteristics of the FDR versions of our examples and do not explain the translation algorithm. While we consider the delete-relaxation, we need delete effects in this section because multiple propositions composing an FDR variable cannot hold simultaneously.

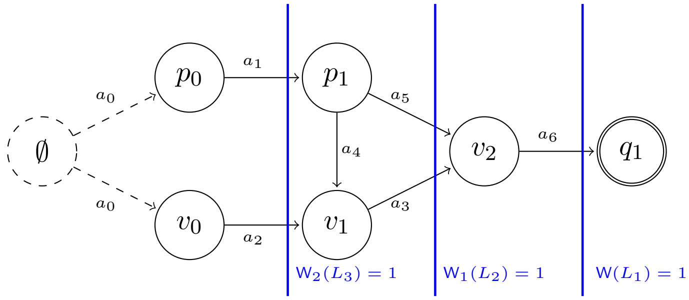


Figure 10: JGs constructed by $h^{\text{LM-cut}}$ for a classical planning task in Example 8. The functions W, W_1, and W_2 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

action	pre	add	del	num	cost
a_1	p_0	p_1	p_0	$\emptyset$	1
a_2	$p_0, u \geq -1$	$\emptyset$	$\emptyset$	$v \mathrel{+}= 1, u \mathrel{+}= -1$	1
a_4	$p_1, u \geq -1$	$\emptyset$	$\emptyset$	$v \mathrel{+}= 1, u \mathrel{+}= -1$	1
a_6	$q_0, v \geq 2$	q_1	q_0	$v \mathrel{+}= -2, u \mathrel{+}= 2$	1

We assume that the tie-breaking strategy prefers $u \geq -1$ to p_0 although the opposite results in the same heuristic value.

We show $h^{\max}_{cri}$-values of facts and action costs in each iteration in Table 11 and JGs in Figure 11. In this figure, edge $(n_{v\geq 2}, n_{u\geq -1}, a_6)$ is omitted. We have

$$h^{\text{LM-cut}}_{cri}(s_I^{\text{RT}}) = 4 > h^{\text{LM-cut}}(s_I) = 3.$$

Therefore, in this example, the h-value computed with the numeric version is more informative than that of the propositional version.

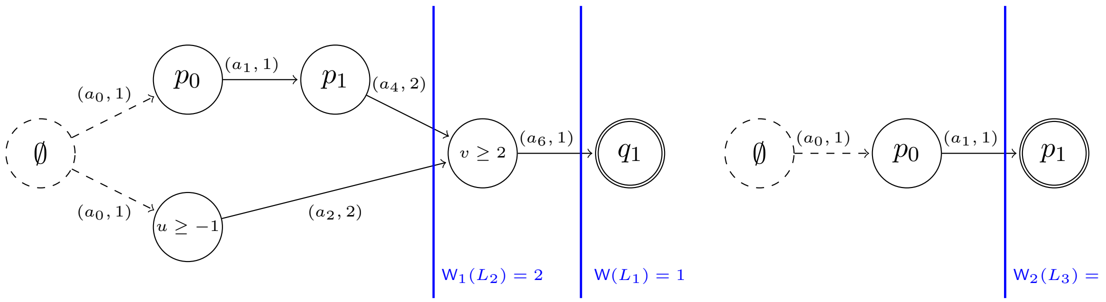
$h^{\max}_{\text{cri}}(s_I, \psi)$	$u \geq -1$	p_0	q_0	q_1	p_1	$v \geq 2$
1	0	0	0	3	1	2
2	0	0	0	2	1	2
3	0	0	0	0	1	0
4	0	0	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_4	a_6
1	1	1	1	1
2	1	1	1	0
3	1	0	0	0
4	0	0	0	0

Table 11: $h^{\max}_{\text{cri}}(s, \psi)$ and $\mathsf{cost}(a)$ in each iteration in Figure 11.

Example 9. *Let $\Pi = \langle \mathcal{F}_p, \mathcal{A}, s_I, G \rangle$ be a classical planning task. $\mathcal{F}_p = \{p_0, p_1, v_0, v_1, v_2, v_3\}$. $s_I = \{p_0, v_0\}$, and $G = \{p_1\}$. Let $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5, a_6\}$, where*

action	pre	add	del	cost
a_1	v_0	v_1	v_0	3
a_2	v_1	v_2	v_1	3
a_3	v_2	v_3	v_2	3
a_4	v_0	v_2	v_0	4
a_5	v_1	v_3	v_1	4
a_6	p_0, v_3	p_1, v_0	p_0, v_3	1



(a) The first and second cuts.

(b) The third cut.

Figure 11: JGs constructed by $h^{\text{LM-cut}}_{\text{cri}}$ for the RT translated from a classical planning task in Example 8. The functions W, W_1, and W_2 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

We show $h^{\max}$-values of facts and action costs in each iteration in Table 12 and JGs in Figure 12. In this figure, edge (n_{v_3}, n_{v_0}, a_6) is omitted. We have

$$h^{\text{LM-cut}}(s_I) = 8.$$

$h^{\max}(s_I, \psi)$	v_0	v_1	v_2	v_3	p_0	p_1
1	0	3	4	7	0	8
2	0	3	4	7	0	7
3	0	3	4	4	0	4
4	0	3	3	3	0	3
5	0	0	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5	a_6
1	3	3	3	4	4	1
2	3	3	3	4	4	0
3	3	3	0	4	1	0
4	3	2	0	3	0	0
5	0	2	0	0	0	0

Table 12: $h^{\max}(s, \psi)$ and $\mathsf{cost}(a)$ in each iteration.

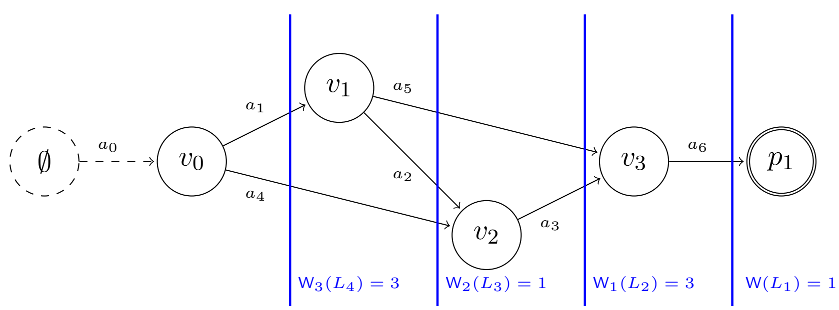


Figure 12: JGs constructed by $h^{\text{LM-cut}}$ for a classical planning task in Example 9. The functions W, W_1, W_2, and W_3 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

In an FDR version of this task, if there are two variables where the first one is for p_0 and p_1, and the second is for v_0, v_1, v_2, and v_3, the second variable can be translated into *a resource variable. The resulting* RT *task is* $\langle \mathcal{F}_p^{\mathrm{RT}}, \mathcal{N}, \mathcal{A}^{\mathrm{RT}}, s_I^{\mathrm{RT}}, G \rangle$ *where* $\mathcal{F}_p^{\mathrm{RT}} = \{p_0, p_1\}$, $\mathcal{N} = \{v, u\}$, $s_I^{\mathrm{RT}} = \{p_0, v = 0, u = 0\}$. *Here, two numeric variables* v *and* u *are introduced to represent the upper and lower bounds of the resource variable.* $\mathcal{A}^{rt} = \{a_1, a_4, a_6\}$, *where*

action	pre	add	del	num	cost
a_1	$u \geq -2$	$\emptyset$	$\emptyset$	$v \mathrel{+}= 1, u \mathrel{+}= -1$	3
a_4	$u \geq -1$	$\emptyset$	$\emptyset$	$v \mathrel{+}= 2, u \mathrel{+}= -2$	4
a_6	$p_0, v \geq 3$	p_1	p_0	$v \mathrel{+}= -3, u \mathrel{+}= 3$	1

We show $h_{cri}^{\max}$*-values of facts and action costs in each iteration in Table 13 and JGs in Figure 13. In the figure, edges* $(n_{v\geq 3}, n_{u\geq -1}, a_6)$ *and* $(n_{v\geq 3}, n_{u\geq -2}, a_6)$ *are omitted. In the numeric version, the second cut contains* $(a_4, \frac{3}{2})$, *and* $\mathsf{W}((n_{u\geq -1}, n_{v\geq 3}, a_4)) = \frac{3}{2} \cdot \mathsf{cost}(a_4) = 6$. *We have*

$$h_{cri}^{\text{LM-cut}}(s_I^{\mathrm{RT}}) = 7 \leq h^{\text{LM-cut}}(s_I) = 8.$$

Therefore, in this example, the h*-value computed with the propositional version is more informative than that of the numeric version.*

$h_{\mathrm{cri}}^{\max}(s_I, \psi)$	$u \geq -1$	$v \geq 3$	p_0	p_1
1	0	6	0	7
2	0	6	0	6
3	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_4	a_6
1	3	4	1
2	3	4	0
3	1	0	0

Table 13: $h_{\mathrm{cri}}^{\max}(s, \psi)$ and $\mathsf{cost}(a)$ in each iteration in Figure 13.

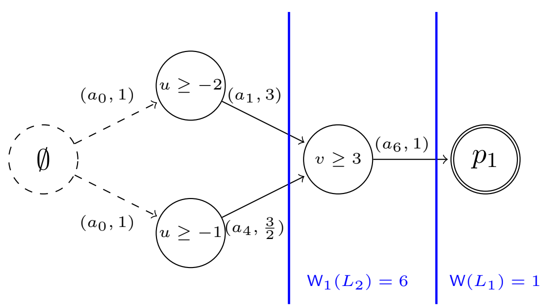


Figure 13: JGs constructed by $h_{\mathrm{cri}}^{\text{LM-cut}}$ for the translated RT from a classical planning task in Example 9. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

From Example 8 and Example 9, we can derive the following proposition.

Proposition 7. *Given a classical planning tasks with resources* Π *and a state* s, *and their counterparts translated to* RT, Π^{RT} *and* s^{RT}, *the values* $h^{\text{LM-cut}}(s)$ *and* $h_{cri}^{\text{LM-cut}}(s^{\mathrm{RT}})$ *are incomparable.*

	Propositional ($h^{\text{LM-cut}}$)			Numeric ($h^{\text{LM-cut}}_{\text{cri}}$)		
	c.	t.	e.	c.	t.	e.
SMALLCOUNTERS (8)	6	8.75	153890	**7**	**0.43**	**5699**
COUNTERS-INV (11)	**2**	0.04	1643	**2**	**0.00**	**0**
COUNTERS-RND (33)	6	0.01	409	**9**	**0.00**	**0**
FARMLAND (30)	11	21.39	217229	**30**	**0.04**	**57**
GARDENING (63)	**63**	7.78	191060	**63**	**1.50**	**15948**
GARDENING-SAT (51)	10	93.12	2207237	**12**	**15.39**	**149909**
SAILING (40)	9	92.17	2283110	**40**	**0.20**	**1617**
SAILING-SAT (40)	3	64.76	3075681	**14**	**0.98**	**23238**
DEPOTS (20)	**7**	**95.36**	**49640**	**7**	101.00	49687
ROVERS (20)	**4**	**6.91**	**0**	**4**	10.10	**0**
SATELLITE (20)	**2**	36.70	78338	**2**	**24.41**	**47278**
TOTAL (336)	123	-	-	**190**	-	-

Table 14: Coverage ('c.'), average time ('t.'), and # of states expanded ('e.') excluding the last f-layer by the propositional and numeric LM-cut heuristics in numeric domains. The time and # of states are averaged over instances solved by both versions.

6.3 LM-Cut: Propositional vs. Numeric – Experimental Evaluations

We empirically evaluate the propositional and numeric LM-cut heuristics. The computational setting is the same as it was described in Section 5.6, where we employ A^* as our search algorithm.

6.3.1 NUMERIC DOMAINS

First, we compare $h^{\text{LM-cut}}$ and $h^{\text{LM-cut}}_{\text{cri}}$ in numeric domains. Recall that the propositional LM-cut assumes that all numeric conditions are achieved with zero cost and considers only propositions in the computation of the JGs. Table 14 shows the results with COUNTERS and FARMLAND-SAT omitted because both versions of LM-cut solve no instances. In this table, as there are only two methods, we show the wall-clock time averaged over instances solved by both of the methods instead of the time score.

Table 14 shows a clear superiority of $h^{\text{LM-cut}}_{\text{cri}}$ in this domains, covering 63 more instances than $h^{\text{LM-cut}}$. Specifically, $h^{\text{LM-cut}}_{\text{cri}}$ outperforms $h^{\text{LM-cut}}$ in all three dimensions (i.e., coverage, wall time, and number of stated expanded) in most domains except for DEPOTS and ROVERS, where wall time and number of state expanded is slightly better for $h^{\text{LM-cut}}$. We note that these two domains are from the IPC and contain several propositional facts and just a few numeric facts and conditions. Therefore, these results indicate that considering numeric conditions is important particularly in domains with many numeric facts and conditions.

6.3.2 Classical Domains

Next, we evaluate the LM-cut heuristics using a set of classical planning domains containing resource variables from the classical optimal track of IPC 1998–2014. 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 on the numeric version obtained by the translation algorithm (Wilhelm et al., 2018). The results of this comparison are shown in Table 15. In addition to the time to solve an instance, we also show the search time, where the time to translate a task is excluded. Similarly to Table 14, we show solving time and search time averaged over instances solved by the two methods instead of the time score.

	Propositional ($h^{\text{LM-cut}}$)				Numeric ($h^{\text{LM-cut}}_{\text{cri}}$)			
	c.	t.	s.	e.	c.	t.	s.	e.
Elevat-11 (20)	**18**	165.26	165.25	54559	**18**	**141.62**	**141.57**	**31038**
Freec (80)	15	188.61	188.56	82292	**18**	**64.23**	**61.27**	**376**
Mprime (35)	**22**	49.51	49.46	5091	**22**	**17.23**	**10.06**	**449**
Mystry (30)	**17**	75.96	75.90	2763	16	**16.22**	**1.46**	**46**
NoMyst-11 (20)	14	29.62	29.60	4868	**20**	**12.63**	**0.39**	**488**
Openst-14 (20)	0	-	-	-	**3**	-	-	-
ParcPr-11 (20)	**13**	**11.97**	**11.97**	23101	**13**	22.30	22.28	**22969**
Pathway (30)	**5**	**14.40**	**14.40**	14621	**5**	39.10	39.09	**14609**
PipesT (50)	**17**	**92.28**	**92.27**	**75427**	16	183.64	183.16	76620
PipesNoT (50)	**12**	**79.77**	**79.75**	**64296**	6	489.09	488.80	64425
Rovers (40)	**7**	**2.92**	**2.92**	**15767**	**7**	8.89	8.88	16258
TPP (30)	**6**	**1.02**	**1.02**	4935	**6**	7.38	7.37	**4403**
Transp-14 (20)	**6**	112.76	112.74	85366	**6**	**90.38**	**90.26**	**82886**
Woodwor-11 (20)	**12**	173.65	173.63	78288	**12**	**126.35**	**126.28**	**19620**
Zenot (20)	**13**	**40.89**	**40.88**	**8764**	12	128.14	128.05	**8764**
Total (505)	177	-	-	-	**180**	-	-	-

Table 15: Coverage ('c.'), average time ('t.'), average search time excluding time to translate tasks ('s.'), and # of states expanded ('e.') excluding the last f-layer by the propositional and numeric LM-cut heuristics in classical domains with resource variables. The time and # of states are averaged over instances solved by both versions.

The translation into a numeric task increases the coverage on three domains and reduces 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. Note also that the number of states expanded is almost always smaller for the numeric version of the LM-cut. While the grounding process for the numeric version is usually marginal when compared to the total solving time, there are a few domains (e.g., Mystry and NoMyst-11) where the grounding process takes significantly more time than the search itself.

7. Operator-Counting

In classical planning, LM-cut is combined with other techniques in the operator-counting (OC) framework (Pommerening et al., 2014) to obtain strong admissible heuristics. This section generalizes LM-cut in the OC framework to numeric planning and proposes a family of novel admissible heuristics. We empirically show that our OC heuristics achieve state-of-the-art performance in RT.

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

$$\begin{aligned} \text{minimize } & \sum_{a \in \mathcal{A}} \mathsf{cost}(a) X_a \\ \text{subject to } & DX + EY \leq b, \\ & X_a \geq 0, \qquad\qquad\qquad\qquad \forall a \in \mathcal{A}, \\ & X \in \mathbb{N}_0^{|\mathcal{A}|}, Y \in \mathbb{R}^n, \end{aligned}$$

where X_a for each $a \in \mathcal{A}$ is a decision variable representing the number of occurrences of action a in a plan, and Y is a vector of auxiliary variables. The set of OC constraints is represented by $DX + EY \leq b$ and corresponds to linear inequalities over X_a such that for every plan π and $a \in \mathcal{A}$, $X_a = \mathsf{count}(\pi, a)$ satisfies $DX + EY \leq b$ where $\mathsf{count}(\pi, a)$ is the number of occurrences of action a in π.[6] Since the optimal cost of the LP/IP problem is a lower bound of the cost of every plan, the OC heuristics are admissible. Adding OC constraints does not remove feasible solutions for a task and tightens the bound (i.e., results in a stronger heuristic). Therefore, different types of OC constraints can be used together to improve the heuristic informativeness.

7.1 Operator-Counting Heuristics

The OC framework was first proposed in classical planning and recent works by Scala et al. (2017) and Piacentini et al. (2018b) applied this framework to numeric planning. Scala et al. (2017) extract landmarks from the delete-relaxed task using the AND/OR graph, while Piacentini et al. (2018b) introduced the state equation constraints (SEQ) (Bonet, 2013) and the delete-relaxation constraints (Imai & Fukunaga, 2015) into SCT.

In theory, OC heuristics do not have to be implemented in the delete-relaxed setting. However, to the best of our knowledge, all the OC heuristics for RT that account for numeric conditions, except for SEQ, under-approximate h^+. This observation may be due to the fact that all these heuristics were adapted from the delete-relaxed setting of classical planning. As shown in our experiments, considering constraints from outside the delete-relaxed setting may be the reason why the SEQ constraints are complementary to the landmarks and delete-relaxation constraints.

We show that the cuts produced by $h^{\text{LM-cut}}_{\text{cri}}$ are also delete-relaxed OC constraints (see Theorem 6 in the following subsection). Moreover, since these constraints approximate h^+ or h^* using an LP, they all can be combined in various configurations to obtain a tighter

6. Here X is a vector of decision variables X_a, $D \in \mathbb{Q}^{|\mathcal{A}| \times m}$ and $E \in \mathbb{Q}^{n \times m}$ are matrices and $b \in \mathbb{Q}^m$ is a vector.

Heuristic	OC Constraints	Solver
$h^{\text{lm+}}_{\text{hbd}}$	AND/OR landmarks	LP
h^{c}_{IP}	delete-relaxation, SEQ	IP
h^{c}_{LP}	delete-relaxation, SEQ	LP
h^{S}_{LP}	SEQ	LP
$h^{\text{S,lm}}_{\text{LP}}$	SEQ, AND/OR landmarks	LP
$h^{c,\text{lm}}_{\text{IP}}$	delete-relaxation, SEQ, AND/OR landmarks	IP
$h^{\text{LC}}_{\text{LP}}$	LM-cut (cri)	LP
$h^{\text{LC,S}}_{\text{LP}}$	LM-cut (cri), SEQ	LP
$h^{\text{LC,S}}_{+,\text{LP}}$	LM-cut (cri, +), SEQ	LP
$\overline{h}^{\text{LC,S}}_{\text{LP}}$	LM-cut (cri), SEQ	LP, rounded up
$h^{\text{LC,lm}}_{\text{LP}}$	LM-cut (cri), AND/OR landmarks	LP
$h^{\text{LC,S,lm}}_{\text{LP}}$	LM-cut (cri), SEQ, AND/OR landmarks	LP
$h^{c,\text{LC}}_{\text{IP}}$	LM-cut (cri), delete-relaxation, SEQ	IP

Table 16: Constraints and solvers used by OC heuristics. 'LM-cut (cri)' and 'LM-cut (cri, +)' indicate the OC constraints extracted by $h^{\text{LM-cut}}_{\text{cri}}$ and $h^{\text{LM-cut}}_{\text{cri},+}$ as in Theorem 7. 'LP, rounded up' used by $\overline{h}^{\text{LC,S}}_{\text{LP}}$ indicates that the heuristic value computed by LP is rounded up in the same way as $\overline{h}^{\text{LM-cut}}_{\text{cri}}$, and the admissibility is proved accordingly to Theorem 3.

approximations. Since the IP solution of the delete-relaxation constraints calculates h^+, the heuristic h^c_{IP} that combines the delete-relaxation and SEQ constraint, solving the IP, approximates the delete-relaxation from above resulting in $h^+ \leq h^c_{\text{IP}} \leq h^*$.

Table 16 shows different OC heuristic and solver configurations (i.e., LP or IP). The $h^{\text{lm+}}_{\text{hbd}}$ heuristic as well as the delete-relaxed AND/OR graph landmark extraction, denoted as 'AND/OR landmarks' in Table 16, were proposed by Scala et al. (2017). The heuristics h^c_{IP} and h^c_{LP} with the set of delete-relaxation constraints and SEQ were proposed by Piacentini et al. (2018b). All other combinations of constraints in Table 16 are novel. Note that, as in most planning problems, the informativeness of a heuristic is not the only relevant characteristic: in most cases a good heuristic combines informativeness with speed.

7.2 LM-Cut for Operator-Counting Constraints

We now present how to obtain OC constraints using the numeric LM-cut heuristic $h^{\text{LM-cut}}_{\text{cri}}$. First, let us recall the general structure of a landmark constraint in classical planning. Given a disjunctive action landmark L, the landmark constraint is as follows (Bonet & Helmert, 2010):

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

Note that this inequality is an OC constraint. In classical planning, the landmarks extracted by $h^{\text{LM-cut}}$ can be used to generate landmark constraints (Pommerening et al., 2014). We generalize this approach for numeric tasks employing the following result.

Theorem 6. *Given an* RT*, let L be a cut obtained by $h^{\text{LM-cut}}_{cri}$. Let $\mathsf{count}(\pi, a)$ be the number of times action a appears in a plan π. Then, the following relation holds for any π*

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

Proof. Let π be a plan for the RT, and let $L = (N^0, N^g)$ be the cut in the JG. Recall that $\partial^{\mathsf{in}}(L)$ is a disjunctive fact landmark by Lemma 1. Thus, there is at least one fact in $\partial^{\mathsf{in}}(L)$ that is achieved by the plan π. Let ψ_0 be the first fact in N^g 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 $\mathsf{lbl}(L_{\psi_0}) = \mathsf{supp}(\psi_0) \cap \mathsf{lbl}(L)$. Note that we can restrict the actions that achieve ψ_0 to $\mathsf{lbl}(L_{\psi_0})$, since to apply an action from $\mathsf{supp}(\psi_0) \setminus \mathsf{lbl}(L_{\psi_0})$ we need to achieve at least one fact in N^g.

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

$$1 \leq \frac{\mathsf{count}(\pi, a_0)}{\mathsf{m}_{a_0}(s, \psi_0)} \leq \frac{\mathsf{count}(\pi, a_0)}{\mathsf{m}_{a_0}^{\min}(L)} \leq \sum_{a \in \mathsf{lbl}(L)} \frac{\mathsf{count}(\pi, a)}{\mathsf{m}_a^{\min}(L)}. \tag{6}$$

Thus, assume that $\mathsf{m}_{a_0}(s, \psi_0) > 1$ and $\psi_0 \in \bar{\mathcal{F}}_n$ is a numeric fact, i.e., ψ_0 is of the form $v \geq w_0$. For each $a \in \mathsf{supp}(\psi_0)$ it holds that $v \mathrel{+}= k^a \in \mathsf{eff}(a)$. Recall that by assumption ψ_0 is the first fact in N^g that is achieved by π, thus we can write

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

To get the result stated in the theorem we need to subtract from both sides of the inequality above $s[v]$, and subsequently divide it by $w_0 - s[v]$, which is greater than zero, by the assumption that $s \not\models \psi_0$:

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

From Theorem 6, we derive the following OC constraint:

$$\sum_{a \in \mathsf{lbl}(L)} \frac{X_a}{\mathsf{m}_a^{\min}(L)} \geq 1, \tag{7}$$

where L is a cut obtained by the LM-cut heuristic. Another immediate corollary of Theorem 6 is that Constraint (7) can be used in an IP to provide a hyper-plane that can potentially speed up the work of the solver in finding h^+. However, as we see in the next subsection, this approach does not grant a performance boost.

Note that repeating the proof for Theorem 6 verbatim we can also obtain the following.

Theorem 7. *Given an* RT*, let L be a cut obtained by $h^{\text{LM-cut}}_{cri,+}$. Let $\mathsf{count}(\pi, a)$ be the number of times action a appears in a plan π. The following holds for any π*

$$\sum_{a \in \mathsf{lbl}(L)} \frac{\mathsf{count}(\pi, a)}{\mathsf{m}_a^{\min +}(L)} \geq 1 \tag{8}$$

where $\mathsf{m}_a^{\min +} = \min_{(n_\psi, n_{\psi'}, a) \in L} \mathsf{m}_a^+(s, \psi')$.

We do not prove this claim directly because the proof repeats verbatim the proof of Theorem 6, with the omission of Equation (6). As presented in the next subsection, OC heuristics that are based on the cuts of $h^{\text{LM-cut}}_{\text{cri}}$ perform slightly better than those that are based on the cuts generated by $h^{\text{LM-cut}}_{\text{cri},+}$.

7.3 Operator Counting – Experimental Evaluation

The setting we use here is the same as in Section 5.6: we plan optimally using the A^* search with a time limit of 30 minutes and memory limit of 4 GB. The search and all heuristics were implemented in NFD using C++11 with GCC 7.5.0, while the mathematical programming solver used for these heuristics is IBM CPLEX 12.10.

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:

- $h^{\text{LC}}_{\text{LP}}$ which uses the LM-cut constraints (Equation (7)),
- h^{S}_{LP} which uses the numeric planning version of SEQ, and
- $h^{\text{LC,S}}_{\text{LP}}$ which uses both.

The constraints used by the LP and IP heuristics can be found in Table 16. A task is translated into an RT when computing numeric LM-cut. 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. (2016a). The results of this comparison are shown in Table 17. Farmland-SAT is omitted in this table because no instance is solved by any configuration.

While $h^{\text{LC}}_{\text{LP}}$ and h^{S}_{LP} are complementary on most of the domains, the coverage of $h^{\text{LC,S}}_{\text{LP}}$ is equal to the maximum of $h^{\text{LC}}_{\text{LP}}$ and h^{S}_{LP}. Furthermore, on Gardening, Gardening-SAT, and Satellite, $h^{\text{LC,S}}_{\text{LP}}$ expands fewer states and finds solutions faster than both of the components.

In total, $h^{\text{LC,S}}_{\text{LP}}$ has the highest coverage among all of the evaluated heuristics including h^{c}_{IP}. Heuristic h^{c}_{IP} expands fewer states than $h^{\text{LC,S}}_{\text{LP}}$ in all domains, which is consistent with the fact that the LM-cut constraints in $h^{\text{LC,S}}_{\text{LP}}$ are an estimation of the delete-relaxation constraints in h^{c}_{IP}. In contrast, $h^{\text{LC,S}}_{\text{LP}}$ has higher time scores than h^{c}_{IP} in all domains, which suggests that the former is faster to compute than the latter. h^{c}_{LP}, the LP version of h^{c}_{IP}, is also slower than $h^{\text{LC,S}}_{\text{LP}}$, indicating that the delete-relaxation constraints are informative, but slow to compute. In the majority of the domains, the computational advantage of $h^{\text{LC,S}}_{\text{LP}}$ results in the higher coverage. However, on Sailing-SAT, h^{c}_{IP} solves 12 more instances than $h^{\text{LC,S}}_{\text{LP}}$, indicating that the informativeness of h^{c}_{IP} is more beneficial in this domain.

Using $h^{\text{LM-cut}}_{\text{cri},+}$ based constraints or the $\overline{h}^{\text{LM-cut}}_{\text{cri}}$ based rounding up and adding more constraints to h^{c}_{IP} and $h^{\text{LC,S}}_{\text{LP}}$ do not improve the coverage.

8. Overall Experimental Evaluation

This section compares the best performing heuristics from Section 5 and Section 7 (i.e., $h^{\text{LM-cut}}_{\text{cri}}$ and $h^{\text{LC,S}}_{\text{LP}}$, respectively) to the state-of-the-art heuristics in the literature. The

	$h^{\text{lm+}}_{\text{hbd}}$	h^{c}_{IP}	h^{c}_{LP}	h^{S}_{LP}	$h^{\text{S,lm}}_{\text{LP}}$	$h^{c\text{,lm}}_{\text{IP}}$	$h^{\text{LC}}_{\text{LP}}$	$h^{\text{LC,S}}_{\text{LP}}$	$h^{\text{LC,S}}_{+\text{,LP}}$	$\bar{h}^{\text{LC,S}}_{\text{LP}}$	$h^{\text{LC,lm}}_{\text{LP}}$	$h^{\text{LC,S,lm}}_{\text{LP}}$	$h^{c\text{,LC}}_{\text{IP}}$
domain	Coverage												
SmallCounters (8)	7	**8**	**8**	**8**	**8**	**8**	7	**8**	**8**	**8**	7	**8**	**8**
Counters (8)	0	4	5	**8**	**8**	3	0	**8**	**8**	**8**	0	**8**	3
Counters-Inv (11)	2	6	7	**11**	**11**	6	2	**11**	**11**	**11**	2	**11**	6
Counters-Rnd (33)	9	21	23	**33**	**33**	19	9	**33**	**33**	**33**	9	**33**	21
Farmland (30)	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**
Gardening (63)	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**
Gardening-SAT (51)	12	14	14	12	**15**	13	12	**15**	**15**	**15**	12	**15**	13
Sailing (40)	20	**40**	20	9	20	**40**	**40**	**40**	**40**	**40**	**40**	**40**	**40**
Sailing-SAT (40)	6	**24**	3	3	5	**24**	14	12	12	12	10	10	23
Depots (20)	3	1	2	6	3	2	**7**	**7**	**7**	**7**	5	5	1
Rovers (20)	**4**	2	**4**	**4**	**4**	2	**4**	**4**	**4**	**4**	**4**	**4**	2
Satellite (20)	1	1	1	1	**2**	1	**2**	**2**	**2**	**2**	**2**	**2**	1
Total (344)	157	214	180	188	202	211	190	**233**	**233**	**233**	184	229	211
domain	Time score												
SmallCounters (8)	0.75	0.94	**1.00**	**1.00**	**1.00**	0.94	0.74	**1.00**	**1.00**	**1.00**	0.74	**1.00**	0.94
Counters (8)	0.00	0.13	0.21	**0.76**	0.55	0.13	0.00	0.72	0.71	0.72	0.00	0.55	0.12
Counters-Inv (11)	0.18	0.32	0.39	**0.77**	0.61	0.31	0.18	0.73	0.73	0.73	0.18	0.61	0.31
Counters-Rnd (33)	0.23	0.34	0.41	**0.81**	0.66	0.34	0.22	0.77	0.77	0.77	0.23	0.65	0.34
Farmland (30)	0.77	0.63	0.83	**0.96**	0.94	0.61	0.73	0.95	0.95	**0.96**	0.77	0.93	0.62
Gardening (63)	0.87	0.77	0.90	0.96	0.99	0.77	0.90	**1.00**	**1.00**	**1.00**	0.86	0.99	0.76
Gardening-SAT (51)	0.09	0.14	0.18	0.18	0.25	0.13	0.12	**0.26**	0.25	**0.26**	0.09	0.25	0.13
Sailing (40)	0.49	0.62	0.34	0.07	0.49	0.61	**0.99**	**0.99**	**0.99**	0.98	0.98	0.97	0.61
Sailing-SAT (40)	0.07	0.21	0.06	0.02	0.07	0.20	**0.22**	**0.22**	**0.22**	**0.22**	**0.22**	**0.22**	0.20
Depots (20)	0.09	0.03	0.06	0.13	0.09	0.03	**0.20**	0.19	0.18	0.18	0.13	0.13	0.03
Rovers (20)	0.10	0.03	0.05	0.11	0.11	0.03	**0.12**	0.11	0.10	0.10	0.09	0.10	0.03
Satellite (20)	0.04	0.04	0.03	0.03	0.05	0.04	0.06	**0.08**	**0.08**	**0.08**	0.06	**0.08**	0.04
domain	# States expanded												
SmallCounters (8)	40366	**0**	**0**	**0**	**0**	**0**	93959	**0**	**0**	**0**	39587	**0**	**0**
Counters (8)	-	-	-	-	-	-	-	-	-	-	-	-	-
Counters-Inv (11)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Counters-Rnd (33)	116069	**0**	**0**	**0**	**0**	**0**	359419	**0**	**0**	**0**	116069	**0**	**0**
Farmland (30)	6537	**0**	**0**	**0**	**0**	**0**	16702	**0**	**0**	**0**	6537	**0**	**0**
Gardening (63)	15044	**30**	**30**	6341	**30**	**30**	15621	86	85	86	14275	**30**	**30**
Gardening-SAT (51)	321302	**151**	**151**	27506	**151**	**151**	286186	274	272	274	273409	**151**	**151**
Sailing (40)	**0**	**0**	**0**	2283110	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Sailing-SAT (40)	8652	**8357**	8820	3075681	8652	**8357**	8654	8654	8650	8654	8652	8652	**8357**
Depots (20)	50	**0**	50	50	50	**0**	6	6	6	6	6	6	**0**
Rovers (20)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Satellite (20)	26906	**98**	1076	46560	7764	**98**	264	**98**	**98**	**98**	264	**98**	**98**

Table 17: Coverage, time score, and # of states expanded excluding the last f-layer by the OC heuristics. # of states are averaged over instances solved by all methods.

experimental setting is the same as in the previous section. Domains with simple conditions are due to Scala et al. (2016a, 2017, 2020) with duplicate tasks excluded. ZenoTravel was excluded since some numeric conditions are not simple (Piacentini et al., 2018b). A task is translated into an RT when computing numeric LM-cut. 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. (2016a).

The heuristics we compare to are: 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), the generalised subgoaling heuristic $h^{\text{gen}}_{\text{hbd}}$ (Scala et al., 2020), and the optimal numeric delete-relaxation heuristic h^{c}_{IP}(Piacentini et al., 2018b). For $\hat{h}^{\mathrm{rmax}}_{\mathrm{hbd+}}$, $h^{\mathrm{lm+}}_{\mathrm{hbd}}$, and $h^{\mathrm{gen}}_{\mathrm{hbd}}$, in addition to our implementations in NFD, we evaluate the original implementations in ENHSP-19[7] using OpenJDK 11.0.9.1.

	$h^{\mathrm{gen}}_{\mathrm{hbd}}$	$h^{\mathrm{gen}}_{\mathrm{hbd}}$	$\hat{h}^{\mathrm{rmax}}_{\mathrm{hbd+}}$	$\hat{h}^{\mathrm{rmax}}_{\mathrm{hbd+}}$	$h^{\mathrm{LM\text{-}cut}}_{\mathrm{cri}}$	$h^{\mathrm{lm+}}_{\mathrm{hbd}}$	$h^{\mathrm{lm+}}_{\mathrm{hbd}}$	$h^{\mathrm{c}}_{\mathrm{IP}}$	$h^{\mathrm{LC,S}}_{\mathrm{LP}}$
implementation	E	N	E	N	N	E	N	N	N
domain	Coverage								
SmallCounters (8)	7	7	7	7	7	7	7	**8**	**8**
Counters (8)	0	0	0	0	0	0	0	4	**8**
Counters-Inv (11)	9	**11**	2	2	2	2	2	6	**11**
Counters-Rnd (33)	32	**33**	6	7	9	10	9	21	**33**
Farmland (30)	20	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**
Farmland-SAT (20)	0	0	**7**	0	0	6	0	0	0
Gardening (63)	53	**63**	**63**	**63**	**63**	**63**	**63**	**63**	**63**
Gardening-SAT (51)	1	11	12	12	12	12	12	14	**15**
Sailing (40)	6	9	22	25	**40**	20	20	**40**	**40**
Sailing-SAT (40)	1	5	7	9	14	3	6	**24**	12
Depots (20)	1	1	4	5	**7**	4	3	1	**7**
Rovers (20)	1	2	3	**4**	**4**	3	**4**	2	**4**
Satellite (20)	0	1	1	1	**2**	1	1	1	**2**
Total (364)	131	173	164	165	190	161	157	214	**233**
domain	Time score								
SmallCounters (8)	0.62	0.74	0.74	0.77	0.79	0.73	0.75	0.94	**1.00**
Counters (8)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	**0.72**
Counters-Inv (11)	0.40	**0.75**	0.18	0.18	0.18	0.18	0.18	0.32	0.73
Counters-Rnd (33)	0.45	**0.79**	0.18	0.19	0.24	0.23	0.23	0.34	0.77
Farmland (30)	0.39	0.65	0.78	0.74	0.74	0.69	0.77	0.63	**0.95**
Farmland-SAT (20)	0.00	0.00	**0.06**	0.00	0.00	0.03	0.00	0.00	0.00
Gardening (63)	0.40	0.72	0.89	0.94	0.96	0.82	0.87	0.77	**1.00**
Gardening-SAT (51)	0.01	0.04	0.12	0.14	0.15	0.09	0.09	0.14	**0.26**
Sailing (40)	0.05	0.13	0.35	0.42	0.90	0.37	0.49	0.62	**0.99**
Sailing-SAT (40)	0.01	0.03	0.11	0.14	0.20	0.05	0.07	0.21	**0.22**
Depots (20)	0.02	0.03	0.11	0.12	**0.22**	0.09	0.09	0.03	0.19
Rovers (20)	0.01	0.02	0.10	0.15	**0.16**	0.12	0.10	0.03	0.11
Satellite (20)	0.00	0.01	0.05	0.05	0.07	0.04	0.04	0.04	**0.08**
domain	# States expanded								
SmallCounters (8)	18943	18945	**0**	242923	90990	40366	40366	**0**	**0**
Counters (8)	-	-	-	-	-	-	-	-	-
Counters-Inv (11)	**0**	**0**	**0**	158	**0**	**0**	**0**	**0**	**0**
Counters-Rnd (33)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Farmland (30)	2185	1534	**0**	1753	1753	2191	657	**0**	**0**
Farmland-SAT (20)	-	-	-	-	-	-	-	-	-
Gardening (63)	8014	5456	**0**	5352	3574	3504	3522	15	40
Gardening-SAT (51)	**0**	3137	**0**	3424	1321	**0**	1019	2	6
Sailing (40)	60572	31813	**0**	1128	629	628	**0**	**0**	**0**
Sailing-SAT (40)	**0**	15050	**0**	15050	15050	**0**	**0**	**0**	**0**
Depots (20)	111	80	50	30	6	76	50	**0**	6
Rovers (20)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Satellite (20)	-	-	-	-	-	-	-	-	-

Table 18: Coverage, time score, and # of states expanded excluding the last f-layer by different numeric heuristics on SCTs. 'N' and 'E' mean the implementations in NFD and ENHSP-19, respectively. # of states are averaged over instances solved by all methods.

Table 18 shows the experimental results. Our $h^{\mathrm{LM\text{-}cut}}_{\mathrm{cri}}$ heuristic improves coverage by 17 tasks compared to the next best non-LP heuristic. The overall best performing heuristic

7. https://sites.google.com/view/enhsp/

is $h_{\text{LP}}^{\text{LC,S}}$, that is, the OC heuristic that includes the constraints produced by the LM-cut procedure and the numeric version of SEQ. This heuristic improves coverage by 19 tasks over the next best state-of-the-art heuristic h_{IP}^{c}.

9. Conclusion

We present a family of LM-cut heuristics that are extended to handle numeric planning problems with simple conditions. In order to obtain a strong and admissible estimate, we introduce an inadmissible variant of $h^{\max}$ which is used to construct a justification graph that, in turn, produces cuts for the numeric version of LM-cut. We show that the resulting heuristic, $h_{\text{cri}}^{\text{LM-cut}}$, is admissible. Moreover, we present several procedures on how to employ existing numeric versions of $h^{\max}$ to construct the justification graph and, thus, create novel numeric LM-cut variants.

We provide a thorough theoretical comparison of all versions of numeric LM-cut we developed with all numeric $h^{\max}$-based heuristics present in the literature showing if there is dominance or incompatibility relation between each pair of heuristics. Although our admissible version of LM-cut does not show any theoretical dominance over the existing heuristics, its empirical performance is much stronger, achieving a significant increase in coverage when compared with numeric $h^{\max}$ relaxations present in the literature. Moreover, we compare our version of LM-cut with the classical one and show, surprisingly, that our version obtains a better coverage not only on numeric domains, but also on classical ones that are translated to numeric domains.

We also transform the cuts produced by numeric LM-cut into operator-counting constraints. The strength of this technique is most evident when LM-cut constraints are combined with SEQ constraints within the operator-counting framework. When compared against various combinations of operator-counting constraints, this heuristic achieves state-of-the-art performance in most numeric domains.

Acknowledgments

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada. The work of Alexander Shleyfman was partially supported by the Israel Academy of Sciences and Humanities program for Israeli postdoctoral researchers. The work of Margarita Castro was supported by the National Center for Artificial Intelligence CENIA FB210017, Basal ANID.

Appendix A.

Here, we show an RT where $h^{\text{LM-cut}}_{\text{hbd}}$ returns an inadmissible heuristic value.

Example 10. *Let* $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G\rangle$ *be an* RT *with* $\mathcal{F}_p = \{p_1, p_2, p_3, p_4, p_5, g_1, g_2\}$ *and* $\mathcal{N} = \{v\}$. *Let* $s_I = \{v = 0\}$, $G = \{g_1, g_2\}$, *and* $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}\}$ *where*

action	pre	eff	cost
a_1	$\emptyset$	p_1	3
a_2	$\emptyset$	p_2	1
a_3	$\emptyset$	p_3	5
a_4	p_1	p_4	1
a_5	p_2	p_5	1
a_6	p_3	$v \mathrel{+}= 1$	0
a_7	p_4, p_5	$v \mathrel{+}= 1$	1
a_8	p_4	g_1	0
a_9	p_5	g_2	0
a_{10}	$v \geq 1$	g_1, g_2	0

The hypergraph representation of the task is shown in Figure 14. The optimal plan is $\langle a_3, a_6, a_{10}\rangle$ *with the cost of* 5.

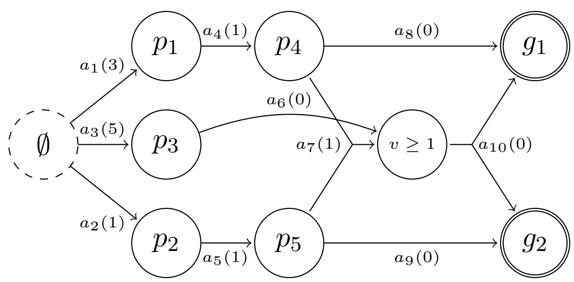


Figure 14: A hypergraph representation of Example 10. The action costs are shown in parentheses.

Consider $h^{\text{LM-cut}}_{hbd}$ *for this task using the justification graphs for* $h^{\max}_{hbd}$. *Let* L_i *be the cut extracted in the i-th iteration in the computation of* $h^{\text{LM-cut}}_{hbd}$. *For each task, we show the value of* $h^{\max}_{hbd}(s_I, \psi)$ *for each fact* ψ *and* $\mathsf{cost}(a)$ *for each action* a *is shown in Table 19.*

Since $h^{\max}_{hbd}(s_I, p_3) = 5$, $h^{\max}_{hbd}(s_I, p_4) = 4$, *and* $h^{\max}_{hbd}(s_I, p_5) = 2$ *in* Π^{RT} *and* Π^{RT}_1 *and* $h^{\max}_{hbd}(s_I, p_3) = 3$ *in* Π^{RT}_1,

$$\begin{aligned}
&h^{\max}_{hbd}(s_I, v \geq 1) \\
&= \min\{h^{\max}_{hbd}(s_I, p_3), \max\{h^{\max}_{hbd}(s_I, p_4), h^{\max}_{hbd}(s_I, p_5)\}\} + \min\{\mathsf{cost}(a_6), \mathsf{cost}(a_7)\} \\
&= h^{\max}_{hbd}(s_I, p_4) + \min\{\mathsf{cost}(a_6), \mathsf{cost}(a_7)\}
\end{aligned}$$

and

$$\mathsf{pcf}(s_I, a_6, v \geq 1) = \mathsf{pcf}(s_I, a_7, v \geq 1) = p_4.$$

$h^{\max}_{\text{hbd}}(s_I, \psi)$	p_1	p_2	p_3	p_4	p_5	$v \geq 1$	g_1	g_2
1	3	1	5	4	2	4	4	2
2	3	1	5	3	2	3	3	2
3	0	1	5	0	2	2	0	2
4	0	1	5	0	1	1	0	1
5	0	0	5	0	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
1	3	1	5	1	1	0	1	0	0	0
2	3	1	5	0	1	0	1	0	0	0
3	0	1	5	0	1	0	1	0	0	0
4	0	1	5	0	0	0	1	0	0	0
5	0	0	5	0	0	0	1	0	0	0

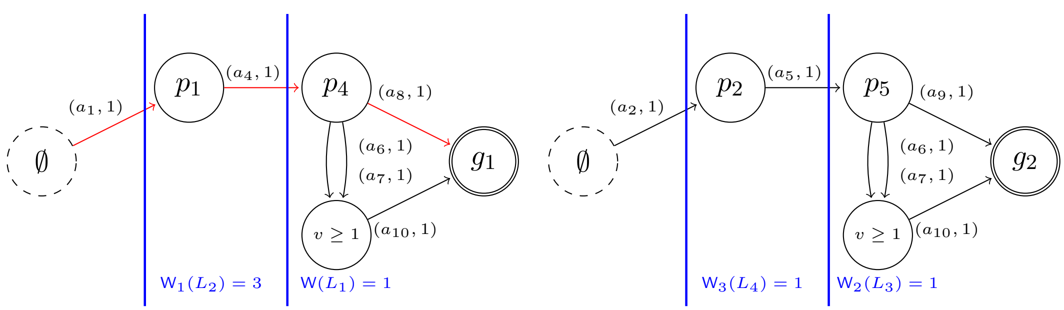
Table 19: $h^{\max}_{\text{hbd}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration.

Since $\mathsf{cost}(a_8) = 0$*,* $p_4 \in N^g$*. Likewise, in* Π^{RT}_2 *and* Π^{RT}_3*,*

$$\mathsf{pcf}(s_I, a_6, v \geq 1) = \mathsf{pcf}(s_I, a_7, v \geq 1) = p_5$$

and $p_5 \in N^g$*. The justification graphs are shown in Figure 10. While the same justification graphs are shared by* Π^{RT} *and* Π^{RT}_1*, another one is shared by* Π^{RT}_2 *and* Π^{RT}_3*. Note that* p_3 *never appears in the justification graphs because* $\mathsf{pcf}(s_I, a_6, v \geq 1) \neq p_3$*. We have*

$$h^{\max}_{hbd}(s_I) = 4 < h^*(s_I) = 5 < h^{\text{LM-cut}}_{hbd}(s_I) = 6.$$



(a) The first and second cuts. (b) The third and fourth cuts.

Figure 15: JGs for the RT in Example 10. The functions W, W_1, W_2, and W_3 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

The inadmissibility of $h^{\text{LM-cut}}_{\text{hbd}}$ follows directly from Example 10.

Next, we show that $h^{\max}_{\text{cri}}$ is inadmissible.

Example 11. *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 $\bar{\mathcal{F}}_n = \{v \geq 2, v \geq 6\}$, and the JG can be seen in Figure 16, where the critical path is indicated in red and the landmarks of numeric LM-cut are denoted by vertical lines. Thus, we have:

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

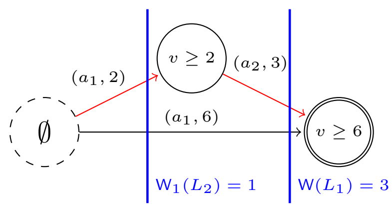


Figure 16: A JG for the RT in Example 11. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

Appendix B.

This appendix is dedicated to the proof of all incomparability relations presented in Table 6. To prove that two heuristics, h_1 and h_2, are incomparable we need to present two planning tasks Π_1 and Π_2 with two states s_1 and s_2, respectively such that $h_1(s_1) < h_2(s_1)$ and $h_2(s_2) < h_1(s_2)$. The catalog for such examples can be found in Table 20 and Table 21. In what follows, we compute and compare the h-values for the heuristics in the examples.

example	$h^{\max}_{\text{hbd}}$	$h^{\max}_{\text{ir}}$	$h^{\text{LM-cut}}_{\text{ir}}$	$h^{\text{LM-cut}}_{\text{ir,m}}$	$h^{\text{LM-cut}}_{\text{ir,m}^+}$	$h^{\text{LM-cut}}_{\text{cri}}$	$h^{\text{LM-cut}}_{\text{cri},+}$	$\overline{h}^{\text{LM-cut}}_{\text{cri}}$
Example 12	2	3	3	4	4	4	4	4
Example 13	5	3	3	4	4	4	4	4
Example 14	2	2	2	2	2	1.5	1.5	1.5
Example 15	1	2	2	1	2	1	2	1
Example 16	1	1	1	1	1	1.9	1	$[1.9, 2.9]$
Example 17	2	1	1	2	2	3	3	3
Example 18	2	2	3.9	2.9	2.9	2.9	2.9	$[2.9, 3.9]$
Example 19	2	1	1.6	2.6	2.1	2.6	2.6	$[2.6, 3.6]$

Table 20: h-values of the heuristics in examples.

Example 12. *Let $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G\rangle$ be an* RT *with $\mathcal{F}_p = \emptyset$ and $\mathcal{N} = \{v, u\}$. Let $s_I = \{v = 0, u = 0\}$, $G = \{v \geq 2\}$, and $\mathcal{A} = \{a_1, a_2\}$, where*

	$h^{\max}_{\text{ir}}$	$h^{\text{LM-cut}}_{\text{ir}}$	$h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$	$h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$	$h^{\text{LM-cut}}_{\text{cri}}$	$h^{\text{LM-cut}}_{\text{cri},+}$	$\overline{h}^{\text{LM-cut}}_{\text{cri}}$
$h^{\max}_{\text{hbd}}$	13, 12	13, 12	13, 12	13, 12	13, 12	13, 12	13, 12
$h^{\max}_{\text{ir}}$		$\leq$	15, 12	$\leq$	14, 12	14, 12	14, 12
$h^{\text{LM-cut}}_{\text{ir}}$			15, 12	18, 12	14, 12	14, 12	14, 12
$h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$				19, 15	14, 17	14, 15	14, 17
$h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$					14, 16	14, 17	14, 16
$h^{\text{LM-cut}}_{\text{cri}}$						16, 15	$\leq$
$h^{\text{LM-cut}}_{\text{cri},+}$							15, 16

Table 21: Pairs of examples with which heuristics are proved to be incompatible. In each cell, in the left example, the heuristic in a row has the higher h-value than the heuristic in a column, and vice versa in the right example. '$\leq$' means the heuristic in a column dominates the heuristic in a row.

action	pre	eff	cost
a_1	$u \geq 1$	$v \mathrel{+}= 1$	1
a_2	$\emptyset$	$v \mathrel{+}= 1, u \mathrel{+}= 1$	3

We show the h-values of the heuristics.

$$h^{\max}_{hbd}(s_I) = h^{\max}_{hbd}(s_I, v \geq 2) = \min\{2 \cdot \mathsf{cost}(a_1), 2 \cdot \mathsf{cost}(a_2)\} = 2 \cdot \mathsf{cost}(a_1) = 2.$$

Since $h^{\max}_{ir}(s_I, u \geq 1) = \mathsf{cost}(a_2) = 3$,

$$h^{\max}_{ir}(s_I) = h^{\max}_{ir}(s_I, v \geq 2) = \min\{h^{\max}_{ir}(s_I, u \geq 1) + \mathsf{cost}(a_1), \mathsf{cost}(a_2)\} = \mathsf{cost}(a_2) = 3.$$

For the LM-cut heuristics, since this task has at most one precondition for each action and only one goal condition, the JGs are the same for all iterations except for the edge weights. For $h^{\text{LM-cut}}_{ir}$*, we show* $h^{\max}_{ir}$*-values of facts and action costs in each iteration in Table 22 and the justification graph and cuts in Figure 17. We have* $h^{\text{LM-cut}}_{ir}(s_I) = 3$*.*

$h^{\max}_{\text{ir}}(s, \psi)$	$u \geq 1$	$v \geq 2$	$\mathsf{cost}(a)$	a_1	a_2
1	3	3	1	1	3
2	2	2	2	0	2
3	0	0	3	0	0

Table 22: $h^{\max}_{\text{ir}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration in Example 12.

For $h^{\text{LM-cut}}_{cri}$*, we show* $h^{\max}_{cri}$*-values of facts and action costs in each iteration in Table 23 and the justification graph and cuts in Figure 18. We have* $h^{\text{LM-cut}}_{cri}(s_I) = 4$*. Since all action costs and action multipliers are positive integer,*

$$h^{\text{LM-cut}}_{cri}(s_I) = h^{\text{LM-cut}}_{cri,+}(s_I) = h^{\text{LM-cut}}_{ir,\mathsf{m}}(s_I) = h^{\text{LM-cut}}_{ir,\mathsf{m}^+}(s_I) = 4.$$

Example 13. *Let* $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$ *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*

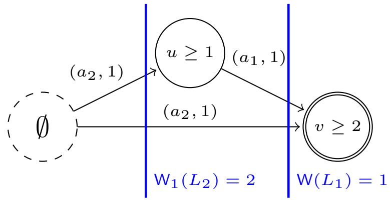


Figure 17: A JG created by $h^{\text{LM-cut}}_{\text{ir}}$ for the RT in Example 12. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

$h^{\max}_{\text{cri}}(s, \psi)$	$u \geq 1$	$v \geq 2$	$\mathsf{cost}(a)$	a_1	a_2
1	3	5	1	1	3
2	2	2	2	0	2
3	0	0	3	0	0

Table 23: $h^{\max}_{\text{cri}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration in Example 12.

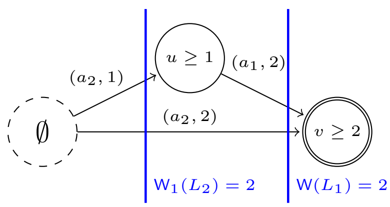


Figure 18: A JG created by $h^{\text{LM-cut}}_{\text{cri}}$ for the RT in Example 12. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

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

For the LM-cut heuristics, since each action has at most one precondition and there is only one goal condition, the justification graphs are the same except for the edge weights. We show the $h^{\max}_{ir}$-values of facts and action costs in each iteration of $h^{\text{LM-cut}}_{ir}$ in Table 24 and the justification graph and cuts in Figure 19. We have

$$h^{\max}_{ir}(s_I) = h^{\max}_{ir}(s_I, g) = 3 \text{ and } h^{\text{LM-cut}}_{ir}(s_I) = 3.$$

We show the $h^{\max}_{cri}$-values of facts and action costs in each iteration of $h^{\text{LM-cut}}_{cri}$ in Table 25 and the justification graph and cuts in Figure 20. We have

$$h^{\text{LM-cut}}_{cri}(s_I) = 4.$$

$h^{\max}_{\text{ir}}(s_I, \psi)$	$v \geq 1$	p	$v \geq 2$	g
1	2	4	2	3
2	2	4	2	2
3	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	2	2	1	1
2	2	2	0	0
3	0	0	0	0

Table 24: $h^{\max}_{\text{ir}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration in Example 13.

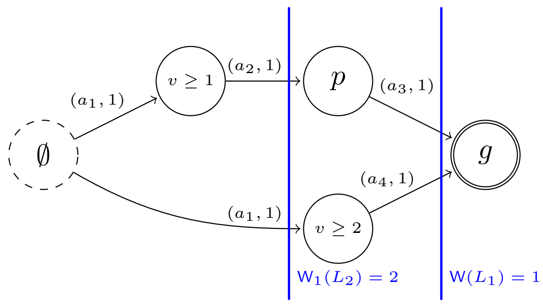


Figure 19: A JG constructed by $h^{\text{LM-cut}}_{\text{ir}}$ for the RT in Example 13. The functions W and W_1 denote the consequent cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

Since all action costs and action multipliers are positive integer,

$$\overline{h}^{\text{LM-cut}}_{cri}(s_I) = h^{\text{LM-cut}}_{cri,+}(s_I) = h^{\text{LM-cut}}_{ir,\mathsf{m}}(s_I) = h^{\text{LM-cut}}_{ir,\mathsf{m}^+}(s_I) = h^{\text{LM-cut}}_{cri}(s_I) = 4.$$

Since only one action has a numeric effect, $h^{\max}_{hbd}$ is the same as $h^{\max}_{cri}$.

$$h^{\max}_{hbd}(s_I) = h^{\max}_{cri}(s_I) = h^{\max}_{cri}(s_I, g) = 5.$$

$h^{\max}_{\text{cri}}(s_I, \psi)$	$v \geq 1$	p	$v \geq 2$	g
1	2	4	4	5
2	2	4	4	4
3	1	1	2	1
4	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	2	2	1	1
2	2	2	0	0
3	1	0	0	0
4	0	0	0	0

Table 25: $h^{\max}_{\text{cri}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration in Example 13.

Example 14. *Let $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$ be an* RT *with $\mathcal{F}_p = \{p, g\}$ and $\mathcal{N} = \{v\}$. Let $s_I = \{v = 0\}$, $G = \{v \geq 2, g\}$, and $\mathcal{A} = \{a_1, a_2\}$, where*

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

In this example, we assume that the tie-breaking strategy prefers $v \geq 2$ to g.

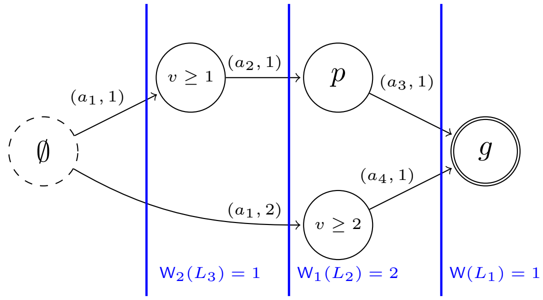


Figure 20: A JG constructed by $h^{\text{LM-cut}}_{\text{cri}}$ for the RT in Example 13. The functions W, W_1, and W_2 denote the consequent cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

ψ	$h^{\max}_{\text{hbd}}(s_I, \psi)$
p	1
$v \geq 2$	1
g	2

Table 26: $h^{\max}_{\text{hbd}}(s_I, \psi)$ for each ψ in Example 14.

We show the $h^{\max}_{hbd}$-values of facts in Table 26. We have

$$h^{\max}_{hbd}(s_I) = h^{\max}_{hbd}(s_I, g) = 2.$$

We show the $h^{\max}_{ir}$-values of facts and action costs in each iteration in Table 24 and the justification graph and cuts in Figure 21. We have

$$h^{\max}_{ir}(s_I) = h^{\text{LM-cut}}_{ir}(s_I) = 2.$$

$h^{\max}_{\text{ir}}(s_I, \psi)$	p	$v \geq 2$	g
1	1	1	2
2	1	1	1
3	0	0	0

$\mathsf{cost}(a)$	a_1	a_2
1	1	1
2	1	0
3	0	0

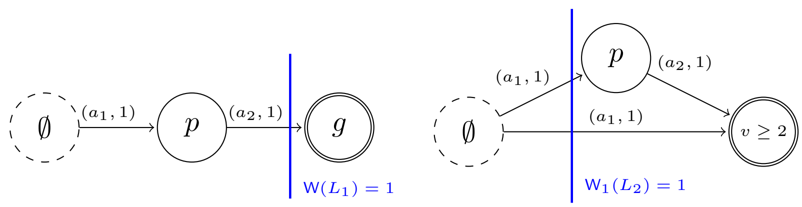
Table 27: $h^{\max}_{\text{ir}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{ir}}$ in Example 14.

For $h^{\text{LM-cut}}_{ir,\mathsf{m}}$, we show the $h^{\max}_{ir}$-values of facts and action costs in each iteration in Table 28 and the justification graph and cuts in Figure 22. We have

$$h^{\text{LM-cut}}_{ir,\mathsf{m}}(s_I) = 2.$$

Since all action multipliers are grater than or equal to 1*,*

$$h^{\text{LM-cut}}_{ir,\mathsf{m}^+}(s_I) = h^{\text{LM-cut}}_{ir,\mathsf{m}}(s_I) = 2.$$



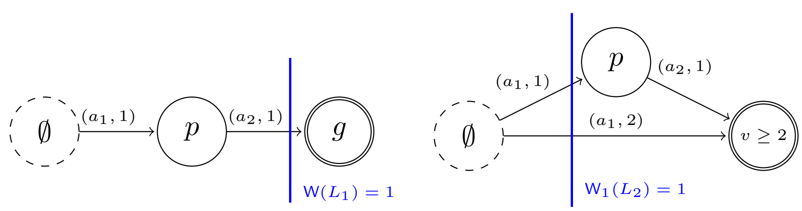
(a) The first cut. (b) The second cut.

Figure 21: JGs constructed by $h^{\max}_{\text{ir}}$ for the RT in Example 14. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

$h^{\max}_{\text{ir}}(s_I, \psi)$	p	$v \geq 2$	g
1	1	1	2
2	1	1	1
3	0	0	0

$\mathsf{cost}(a)$	a_1	a_2
1	1	1
2	1	0
3	0	0

Table 28: $h^{\max}_{\text{ir}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ in Example 14.



(a) The first cut. (b) The second cut.

Figure 22: JGs constructed by $h^{\max}_{\text{ir},\mathsf{m}}$ for the RT in Example 14. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

We show the $h^{\max}_{cri}$*-values of facts and action costs in each iteration in Table 29 and the justification graph and cuts in Figure 23. We have,*

$$h^{\text{LM-cut}}_{cri}(s_I) = 1.5.$$

Since all action costs in Π^{RT} *and action multipliers are positive integer,*

$$h^{\text{LM-cut}}_{cri,+}(s_I) = \overline{h}^{\text{LM-cut}}_{cri}(s_I) = h^{\text{LM-cut}}_{cri}(s_I) = 1.5.$$

Example 15. *Let* $\langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$ *be an* RT *with* $\mathcal{F}_p = \emptyset$ *and* $\mathcal{N} = \{v, u\}$*. Let* $s_I = \{v = 0, u = 0\}$*,* $G = \{u \geq 1\}$*, and* $\mathcal{A} = \{a_1, a_2\}$*, where*

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

$h^{\max}_{\text{cri}}(s_I, \psi)$	p	$v \geq 2$	g
1	1	2	2
2	0.5	0.5	0.5
3	0	0	0

$\mathsf{cost}(a)$	a_1	a_2
1	1	1
2	0.5	0
3	0	0

Table 29: $h^{\max}_{\text{ir}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration in Example 14.

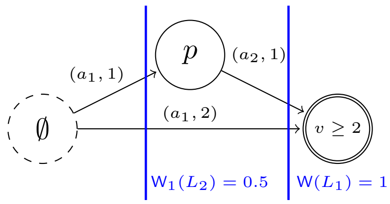


Figure 23: A JG for constructed by $h^{\text{LM-cut}}_{\text{cri}}$ for the RT in Example 14. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

We show the h-values of the heuristics.

$$h^{\max}_{hbd}(s_I) = h^{\max}_{hbd}(s_I, u \geq 1) = h^{\max}_{hbd}(s_I, v \geq 1) + 0.5 \cdot \mathsf{cost}(a_2) = 0.5 \cdot \mathsf{cost}(a_1) + 0.5 = 1.$$

For the LM-cut heuristics, since each action has at most one precondition and there is only one goal condition in this task, the justification graphs are the same except for the edge cost. Since there is only one path to the goal node in the JG, the cost is the same as the max heuristic which defines the edge weight. For $h^{\text{LM-cut}}_{ir}$,

$$h^{\text{LM-cut}}_{ir}(s_I) = h^{\max}_{ir}(s_I) = h^{\max}_{ir}(s_I, v \geq 1) + \mathsf{cost}(a_2) = \mathsf{cost}(a_1) + 1 = 2.$$

For $h^{\text{LM-cut}}_{cri}$,

$$h^{\text{LM-cut}}_{cri}(s_I) = h^{\max}_{cri}(s_I) = h^{\max}_{cri}(s_I, v \geq 1) + 0.5 \cdot \mathsf{cost}(a_2) = 0.5 \cdot \mathsf{cost}(a_1) + 0.5 = 1.$$

Since JGs are the same in $h^{\text{LM-cut}}_{ir,m}$,

$$h^{\text{LM-cut}}_{ir,\mathsf{m}}(s_I) = h^{\text{LM-cut}}_{cri}(s_I) = 1.$$

Since all action costs are integer in Π^{RT},

$$\overline{h}^{\text{LM-cut}}_{cri}(s_I) = h^{\text{LM-cut}}_{cri}(s_I) = 1.$$

For $h^{\text{LM-cut}}_{cri,+}$ and $h^{\text{LM-cut}}_{ir,\mathsf{m}^+}$, since $\mathsf{m}^+_{a_1}(s_I, v \geq 1) = 1$ and $\mathsf{m}^+_{a_2}(s_I, u \geq 1) = 1$,

$$h^{\text{LM-cut}}_{cri,+}(s_I) = h^{\text{LM-cut}}_{ir,\mathsf{m}^+}(s_I) = 1 \cdot \mathsf{cost}(a_1) + 1 \cdot \mathsf{cost}(a_2) = 2.$$

Example 16. *Let $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$ be an* RT *task with $\mathcal{F}_p = \{p, g\}$ and $\mathcal{N} = \{v\}$. Let $s_I = \{v = 0\}$, $G = \{v \geq 0.9, g\}$, and $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$, where*

action	pre	eff	cost
a_1	$\emptyset$	$v \mathrel{+}= 1$	1
a_2	$\emptyset$	p	1
a_3	$v \geq 0.9, p$	g	0
a_4	p	g	0

In this example, we assume that the tie-breaking prefers g to $v \geq 0.9$ and $v \geq 0.9$ to p.

We show the $h^{\max}_{ir}$-values and costs of actions in each iteration of $h^{\text{LM-cut}}_{ir}$ in Table 30 and the justification graphs in Figure 24. We have

$$h^{\max}_{ir}(s_I) = 1 \text{ and } h^{\text{LM-cut}}_{ir}(s_I) = 1.$$

Since all action multipliers are less than or equal to 1, $h^{\text{LM-cut}}_{cri,+}$ and $h^{\text{LM-cut}}_{ir,\mathsf{m}^+}$ are exactly the same as $h^{\text{LM-cut}}_{ir}$.

$$h^{\text{LM-cut}}_{cri,+}(s_I) = h^{\text{LM-cut}}_{ir,\mathsf{m}^+}(s_I) = h^{\text{LM-cut}}_{ir}(s_I) = 1.$$

$h^{\max}_{\text{ir}}(s_I, \psi)$	g	$v \geq 0.9$	p
1	1	1	1
2	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	1	1	0	0
2	0	0	0	0

Table 30: $h^{\max}_{\text{ir}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{ir}}$ in Example 16.

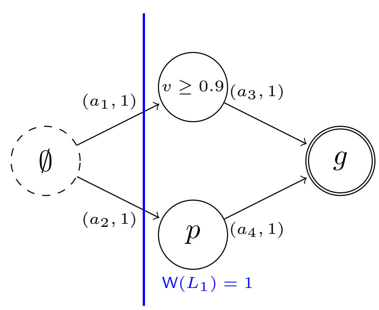


Figure 24: The JG constructed by $h^{\text{LM-cut}}_{\text{ir}}$ for the RT in Example 16.

We show the $h^{\max}_{ir}$-values and costs of actions in each iteration of $h^{\text{LM-cut}}_{ir,\mathsf{m}}$ in Table 31 and the JGs in Figure 25. We have $h^{\text{LM-cut}}_{ir,\mathsf{m}}(s_I) = 1$.

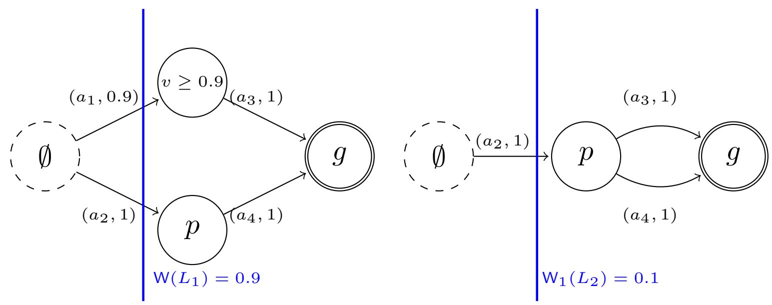
$h^{\max}_{\text{ir}}(s_I, \psi)$	g	$v \geq 0.9$	p
1	1	1	1
2	0.1	0	0.1
3	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	1	1	0	0
2	0	0.1	0	0
3	0	0	0	0

Table 31: $h^{\max}_{\text{ir}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ in Example 16.

We show the $h^{\max}_{cri}$-values and costs of actions in each iteration of $h^{\text{LM-cut}}_{cri}$ in Table 32 and the JGs in Figure 26. We have

$$h^{\text{LM-cut}}_{cri}(s_I) = 1.9.$$



(a) The first cut.

(b) The second cut.

Figure 25: JGs constructed by $h^{\text{LM-cut}}_{\text{ir,m}}$ for the RT in Example 16. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

For $\overline{h}^{\text{LM-cut}}_{cri}$,

$$\overline{h}^{\text{LM-cut}}_{cri}(s_I) = \frac{\lceil c \cdot h^{\text{LM-cut}}_{cri}(s_I) \rceil}{c} \leq h^{\text{LM-cut}}_{cri}(s_I) + 1 \leq 2.9.$$

Since $\overline{h}^{\text{LM-cut}}_{cri}(s_I) \geq h^{\text{LM-cut}}_{cri}(s_I)$,

$$1.9 \leq \overline{h}^{\text{LM-cut}}_{cri}(s_I) \leq 2.9.$$

Since only one action has a numeric effect,

$$h^{\max}_{hbd}(s_I) = h^{\max}_{cri}(s_I) = h^{\max}_{cri}(s_I, g) = 1.$$

$h^{\max}_{\text{cri}}(s_I, \psi)$	g	$v \geq 0.9$	p
1	1	0.9	1
2	0	0.9	0
3	0	0	0

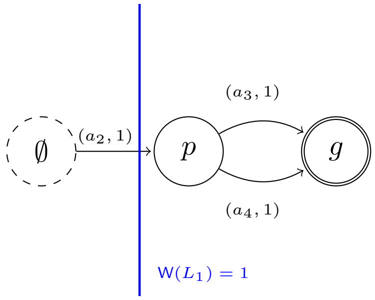
$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	1	1	1	1
2	1	0	0	0
3	0	0	0	0

Table 32: $h^{\max}_{\text{cri}}(s_I, \psi)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{cri}}$ in Example 16.

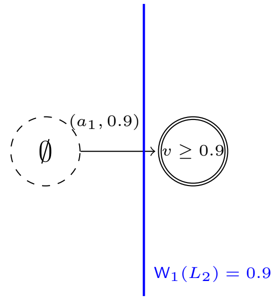
Example 17. *Let* $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$ *be an* RT *task with* $\mathcal{F}_p = \{p, g\}$ *and* $\mathcal{N} = \{v\}$. *Let* $s_I = \{v = 0\}$, $G = \{p, g\}$, *and* $\mathcal{A} = \{a_1, a_2, a_3\}$, *where*

action	pre	eff	cost
a_1	$\emptyset$	$v \mathrel{+}= 1$	1
a_2	$\emptyset$	p	1
a_3	$v \geq 2, p$	g	0
a_4	$v \geq 2$	g	0

In this example, we assume that the tie-breaking strategy prefers g *to* p *and* p *to* $v \geq 2$.



(a) The first and second cuts.



(b) The third cut.

Figure 26: JGs constructed by $h_{\text{cri}}^{\text{LM-cut}}$ for the RT in Example 16. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

$h_{\text{ir}}^{\max}(s_I, \psi)$	g	p	$v \geq 2$
1	1	1	1
2	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	1	1	0	0
2	0	0	0	0

Table 33: $h_{\text{ir}}^{\max}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h_{\text{ir}}^{\text{LM-cut}}$ in Example 17.

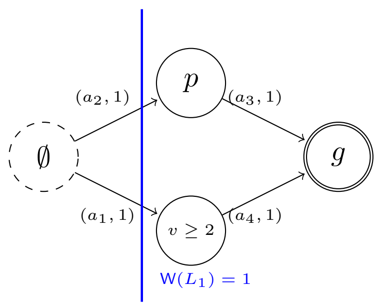


Figure 27: A JG constructed by $h_{\text{ir}}^{\text{LM-cut}}$ for the RT in Example 17.

We show the $h_{ir}^{\max}$-values of facts and action costs in each iteration of $h_{ir}^{\text{LM-cut}}$ in Table 33 and the JGs in Figure 27. We have

$$h_{ir}^{\max}(s_I) = 1 \textit{ and } h_{ir}^{\text{LM-cut}}(s_I) = 1.$$

We show the $h_{ir}^{\max}$-values of facts and action costs in each iteration of $h_{ir,\mathsf{m}}^{\text{LM-cut}}$ in Table 34 and the JGs in Figure 28. We have $h_{ir,\mathsf{m}}^{\text{LM-cut}}(s_I) = 2$.

Since all action multipliers are greater than or equal to 1,

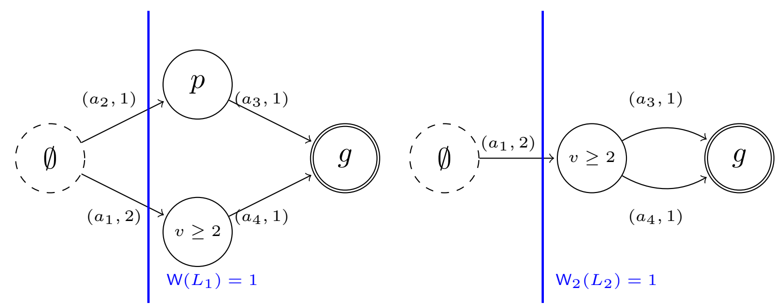
$$h_{ir,\mathsf{m}^+}^{\text{LM-cut}}(s_I) = h_{ir,\mathsf{m}}^{\text{LM-cut}}(s_I) = 2.$$

We show the $h_{cri}^{\max}$-values of facts and action costs in each iteration of $h_{cri}^{\text{LM-cut}}$ in Table 35 and the JGs in Figure 29. We have $h_{cri}^{\text{LM-cut}}(s_I) = 3$.

$h^{\max}_{\text{ir}}(s_I, \psi)$	g	p	$v \geq 2$
1	1	1	1
2	0.5	0	0.5
3	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	1	1	0	0
2	0.5	0	0	0
3	0	0	0	0

Table 34: $h^{\max}_{\text{ir}}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ in Example 17.



(a) The first cut. (b) The third cut.

Figure 28: JGs constructed by $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ for the RT in Example 17. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

Since all action costs in Π^{RT} *and action multipliers are positive integer,*

$$\overline{h}^{\text{LM-cut}}_{cri}(s_I) = h^{\text{LM-cut}}_{cri,+}(s_I) = h^{\text{LM-cut}}_{cri}(s_I) = 3.$$

Since only one action has a numeric effect,

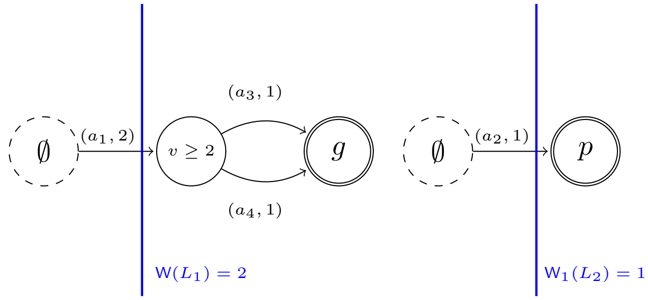
$$h^{\max}_{hbd}(s_I) = h^{\max}_{cri}(s_I) = h^{\max}_{cri}(s_I, g) = 2.$$

$h^{\max}_{\text{cri}}(s_I, \psi)$	g	p	$v \geq 2$
1	2	1	2
2	0	1	0
3	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4
1	1	1	0	0
2	0	1	0	0
3	0	0	0	0

Table 35: $h^{\max}_{\text{cri}}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{cri}}$ in Example 17.

Example 18. *Let* $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$ *be an* RT *task with* $\mathcal{F}_p = \{p, q, r, g_1, g_2\}$ *and* $\mathcal{N} = \{v\}$*. Let* $s_I = \{v = 0\}$, $G = \{v \geq 2, g_1, g_2\}$, *and* $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$, *where*



(a) The first cut. (b) The second cut.

Figure 29: JGs constructed by $h_{\text{cri}}^{\text{LM-cut}}$ for the RT in Example 17. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

action	pre	eff	cost
a_1	$\emptyset$	p	1
a_2	$\emptyset$	q	1
a_3	$\emptyset$	r	1.5
a_4	p	$v \mathrel{+}= 2$	1
a_5	q	$v \mathrel{+}= 1, g_1$	1
a_6	g_1, r	g_2	0.4
a_7	r	g_2	0.4

In this example, we assume that the tie-breaking strategy prefers g_2 to g_1, g_2 to $v \geq 2$, $v \geq 2$ to g_1, and g_1 to r.

We show the $h_{hbd}^{\max}$-values of facts in Table 36. We have

$$h_{hbd}^{\max}(s_I) = h_{hbd}^{\max}(s_I, v \geq 2) = 2.$$

ψ	$h_{\text{hbd}}^{\max}(s_I, \psi)$
p	1
q	1
g_2	1.9
$v \geq 2$	2
g_1	2
r	1.5

Table 36: $h_{\text{hbd}}^{\max}(s, \psi)$ for each ψ in Example 18.

We show the $h_{ir}^{\max}$-values of facts and action costs in each iteration of $h_{ir}^{\text{LM-cut}}$ in Table 37 and the JGs in Figure 30. We have

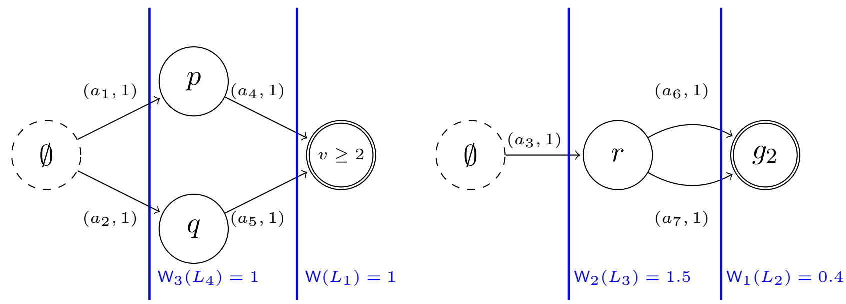
$$h_{ir}^{\max}(s_I) = h_{ir}^{\max}(s_I, v \geq 2) = 2$$

and

$$h_{ir}^{\text{LM-cut}}(s_I) = 3.9.$$

$h_{\text{ir}}^{\max}(s_I, \psi)$	p	q	g_2	$v \geq 2$	g_1	r	$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5	a_6	a_7
1	1	1	1.9	2	2	1.5	1	1	1	1.5	1	1	0.4	0.4
2	1	1	1.9	1	1	1.5	2	1	1	1.5	0	0	0.4	0.4
3	1	1	1.5	1	1	1.5	3	1	1	1.5	0	0	0	0
4	1	1	0	1	1	0	4	1	1	0	0	0	0	0
5	0	0	0	0	0	0	4	0	0	0	0	0	0	0

Table 37: $h_{\text{ir}}^{\max}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h_{\text{ir}}^{\text{LM-cut}}$ in Example 18.



(a) The first and fourth cuts. (b) The second and third cuts.

Figure 30: JGs constructed by $h_{\text{ir}}^{\text{LM-cut}}$ for the RT in Example 18. The functions W, W_1, W_2, and W_3 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

We show the $h_{ir}^{\max}$-values of facts and action costs in each iteration of $h_{ir,\mathsf{m}}^{\text{LM-cut}}$ in Table 38 and the JGs in Figure 31. We have

$$h_{ir,\mathsf{m}}^{\text{LM-cut}}(s_I) = 2.9.$$

Since all action multipliers are greater than or equal to 1,

$$h_{ir,\mathsf{m}^+}^{\text{LM-cut}}(s_I) = h_{ir,\mathsf{m}}^{\text{LM-cut}}(s_I) = 2.9.$$

We show the $h_{cri}^{\max}$-values of facts and action costs in each iteration of $h_{cri}^{\text{LM-cut}}$ in Table 39 and the JGs in Figure 32. We have

$$h_{cri}^{\text{LM-cut}}(s_I) = 2.9.$$

Since all action multipliers are greater than or equal to 1,

$$h_{cri,+}^{\text{LM-cut}}(s_I) = h_{cri}^{\text{LM-cut}}(s_I) = 2.9.$$

For $\overline{h}_{cri}^{\text{LM-cut}}$,

$$\overline{h}_{cri}^{\text{LM-cut}}(s_I) = \frac{\lceil c \cdot h_{cri}^{\text{LM-cut}}(s_I) \rceil}{c} \leq h_{cri}^{\text{LM-cut}}(s_I) + 1 \leq 3.9.$$

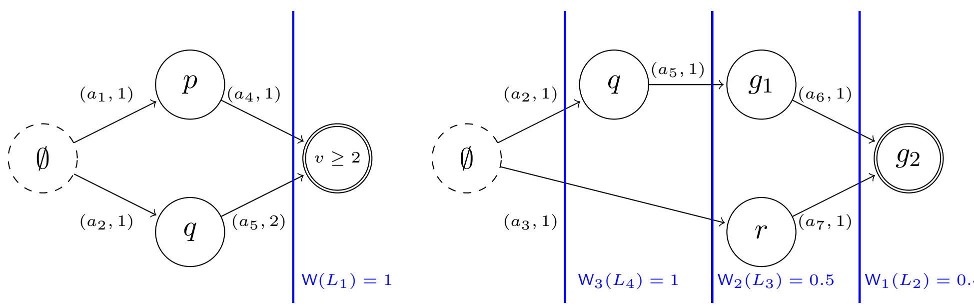
Since $\overline{h}_{cri}^{\text{LM-cut}}(s_I) \geq h_{cri}^{\text{LM-cut}}(s_I)$,

$$2.9 \leq \overline{h}_{cri}^{\text{LM-cut}}(s_I) \leq 3.9.$$

$h^{\max}_{\text{ir}}(s_I, \psi)$	p	q	g_2	$v \geq 2$	g_1	r
1	1	1	1.9	2	2	1.5
2	1	1	1.9	1	1.5	1.5
3	1	1	1.5	1	1.5	1.5
4	1	1	1	1	1	1
5	1	0	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5	a_6	a_7
1	1	1	1.5	1	1	0.4	0.4
2	1	1	1.5	0	0.5	0.4	0.4
3	1	1	1.5	0	0.5	0	0
4	1	1	1	0	0	0	0
5	1	0	0	0	0	0	0

Table 38: $h^{\max}_{\text{ir}}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ in Example 18.



(a) The first cut. (b) The second, third, and fourth cuts.

Figure 31: JGs constructed by $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ for the RT in Example 18. The functions W, W_1, W_2, and W_3 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

Example 19. *Let* $\Pi^{\text{RT}} = \langle \mathcal{F}_p, \mathcal{N}, \mathcal{A}, s_I, G \rangle$ *be an* RT *task with* $\mathcal{F}_p = \{g_1, g_2\}$ *and* $\mathcal{N} = \{v, u\}$. *Let* $s_I = \{v = 0, u = 0\}$, $G = \{v \geq 0.6, g_1, g_2\}$, *and* $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5\}$, *where*

action	pre	eff	cost
a_1	$\emptyset$	g_1	0.6
a_2	$\emptyset$	$v \mathrel{+}= 1$	1
a_3	$\emptyset$	$v \mathrel{+}= 0.3, u \mathrel{+}= 1$	1
a_4	$g_1, u \geq 2$	g_2	0
a_5	$u \geq 2$	g_2	0.1

In this example, we assume that the tie-breaking strategy prefers $v \geq 0.6$ *to* g_2, g_2 *to* g_1, *and* g_1 *to* $u \geq 2$.

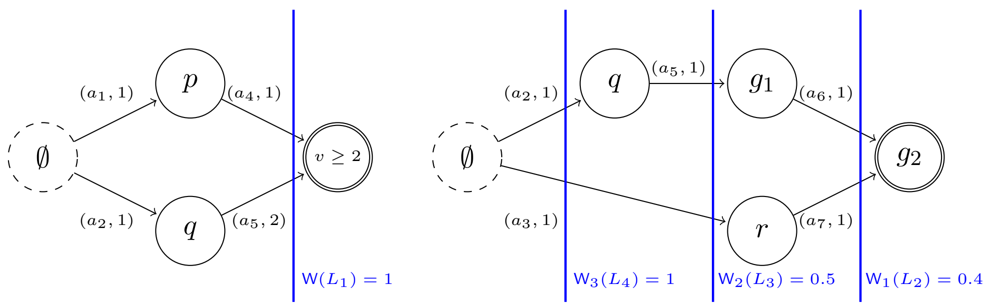
We show the $h^{\max}_{hbd}$*-values of facts in Table 40. We have*

$$h^{\max}_{hbd}(s_I) = h^{\max}_{hbd}(s_I, g_2) = 2.$$

$h^{\max}_{\text{cri}}(s_I, \psi)$	p	q	g_2	$v \geq 2$	g_1	r
1	1	1	1.9	2	2	1.5
2	1	1	1.9	1	1.5	1.5
3	1	1	1.5	1	1.5	1.5
4	1	1	1	1	1	1
5	1	0	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5	a_6	a_7
1	1	1	1.5	1	1	0.4	0.4
2	1	1	1.5	0	0.5	0.4	0.4
3	1	1	1.5	0	0.5	0	0
4	1	1	1	0	0	0	0
5	1	0	0	0	0	0	0

Table 39: $h^{\max}_{\text{cri}}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{cri}}$ in Example 18.



(a) The first cut.

(b) The second, third, and fourth cuts.

Figure 32: JGs constructed by $h^{\text{LM-cut}}_{\text{cri}}$ for the RT in Example 18. The functions W, W_1, W_2, and W_3 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

ψ	$h^{\max}_{\text{hbd}}(s_I, \psi)$
$v \geq 0.6$	0.6
g_2	2
g_1	0.6
$u \geq 2$	2

Table 40: $h^{\max}_{\text{hbd}}(s_I, \psi)$ for each ψ in Example 19.

We show the $h^{\max}_{ir}$-values of facts and action costs in each iteration of $h^{\text{LM-cut}}_{ir}$ in Table 41 and the JGs in Figure 33. We have

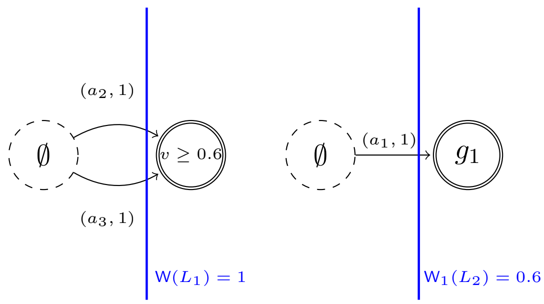
$$h^{\max}_{ir}(s_I) = h^{\max}_{ir}(s_I, v \geq 0.6) = 1$$

and

$$h^{\text{LM-cut}}_{ir}(s_I) = 1.6.$$

$h^{\max}_{\text{ir}}(s_I, \psi)$	$v \geq 0.6$	g_2	g_1	$u \geq 2$	$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5
1	1	1	0.6	1	1	0.6	1	1	0	0.1
2	0	0.1	0.6	0	2	0.6	0	0	0	0.1
3	0	0	0	0	3	0	0	0	0	0.1

Table 41: $h^{\max}_{\text{ir}}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{ir}}$ in Example 19.



(a) The first cut. (b) The second cut.

Figure 33: JGs constructed by $h^{\text{LM-cut}}_{\text{ir}}$ for the RT in Example 19. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

We show the $h^{\max}_{ir}$-values of facts and action costs in each iteration of $h^{\text{LM-cut}}_{ir,\mathsf{m}}$ in Table 42 and the JGs in Figure 34. We have

$$h^{\text{LM-cut}}_{ir,\mathsf{m}}(s_I) = 2.6.$$

$h^{\max}_{\text{ir}}(s_I, \psi)$	$v \geq 0.6$	g_2	g_1	$u \geq 2$	$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5
1	1	1	0.6	1	1	0.6	1	1	0	0.1
2	0	0.7	0.6	0.7	2	0.6	0	0.7	0	0.1
3	0	0.1	0.6	0	3	0.6	0	0	0	0.1
4	0	0	0	0	4	0	0	0	0	0.1

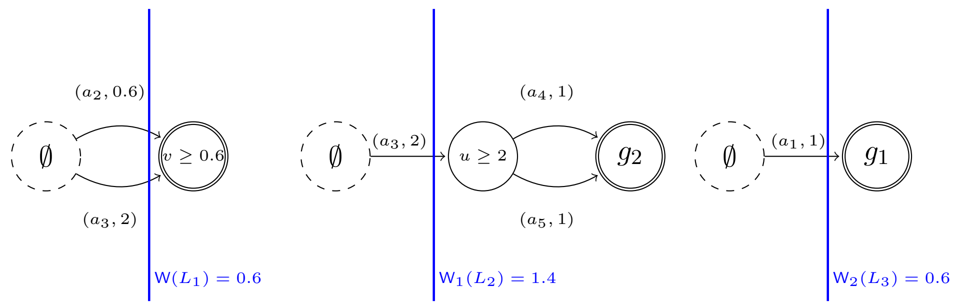
Table 42: $h^{\max}_{\text{ir}}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ in Example 19.

We show the $h^{\max}_{ir}$-values of facts and action costs in each iteration of $h^{\text{LM-cut}}_{ir,\mathsf{m}^+}$ in Table 43 and the JGs in Figure 35. We have

$$h^{\text{LM-cut}}_{ir,\mathsf{m}^+}(s_I) = 2.1.$$

We show the $h^{\max}_{cri}$-values of facts and action costs in each iteration of $h^{\text{LM-cut}}_{cri}$ in Table 44 and the JGs in Figure 36. We have

$$h^{\text{LM-cut}}_{cri}(s_I) = 2.6.$$



(a) The first cut. (b) The second cut. (c) The third cut.

Figure 34: JGs constructed by $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$ for the RT in Example 19. The functions W, W_1, and W_2 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

$h^{\max}_{\text{ir}}(s_I, \psi)$	$v \geq 0.6$	g_2	g_1	$u \geq 2$	$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5
1	1	1	0.6	1	1	0.6	1	1	0	0.1
2	0	0.6	0.6	0.5	2	0.6	0	0.5	0	0.1
3	0	0.5	0.5	0.5	3	0.5	0	0.5	0	0
4	0	0.25	0	0.25	4	0	0	0.25	0	0
5	0	0	0	0	5	0	0	0	0	0

Table 43: $h^{\max}_{\text{ir}}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$ in Example 19.

For $\overline{h}^{\text{LM-cut}}_{cri}$,

$$\overline{h}^{\text{LM-cut}}_{cri}(s_I) = \frac{\lceil c \cdot h^{\text{LM-cut}}_{cri}(s_I) \rceil}{c} \leq h^{\text{LM-cut}}_{cri}(s_I) + 1 = 3.6.$$

Since $\overline{h}^{\text{LM-cut}}_{cri}(s_I) \geq h^{\text{LM-cut}}_{cri}(s_I)$,

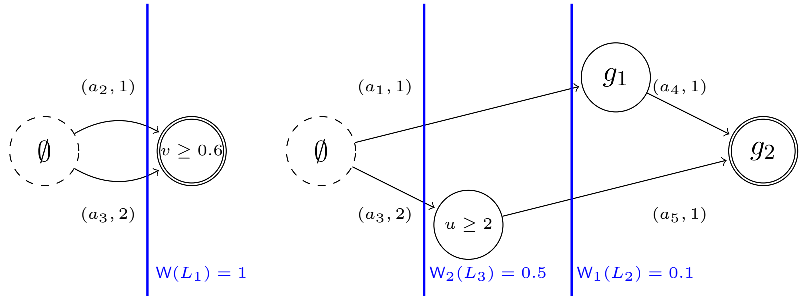
$$2.6 \leq \overline{h}^{\text{LM-cut}}_{cri}(s_I) \leq 3.6.$$

$h^{\max}_{\text{cri}}(s_I, \psi)$	$v \geq 0.6$	g_2	g_1	$u \geq 2$	$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5
1	0.6	2	0.6	2	1	0.6	1	1	0	0.1
2	0	0.1	0.6	0	2	0.6	1	0	0	0.1
3	0	0	0	0	3	0	1	0	0	0.1

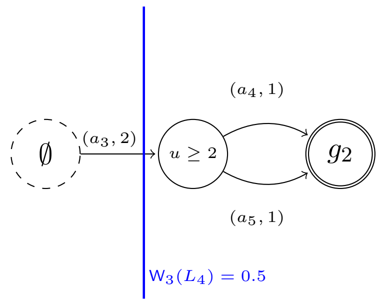
Table 44: $h^{\max}_{\text{cri}}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{cri}}$ in Example 19.

We show the $h^{\max}_{cri,+}$*-values of facts and action costs in each iteration of* $h^{\text{LM-cut}}_{cri,+}$ *in Table 45 and the JGs in Figure 37. We have*

$$h^{\text{LM-cut}}_{cri,+}(s_I) = 2.6.$$

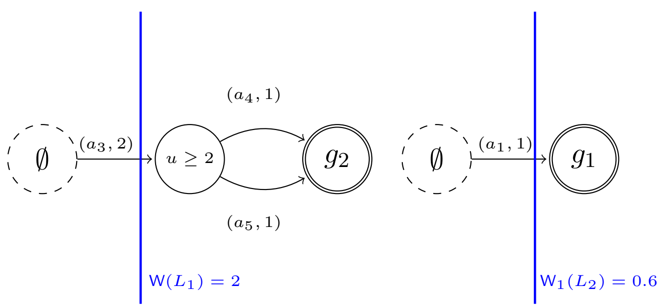


(a) The first cut. (b) The second and third cuts.



(c) The fourth cut.

Figure 35: JGs constructed by $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}^+}$ for the RT in Example 19. The functions W, W_1, W_2, and W_3 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.



(a) The first cut. (b) The second cut.

Figure 36: JGs constructed by $h^{\text{LM-cut}}_{\text{cri}}$ for the RT in Example 19. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

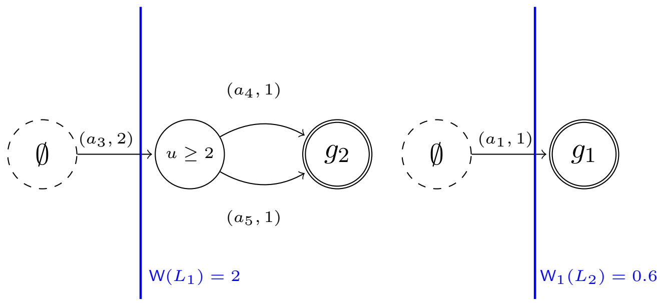
Appendix C.

In this appendix, we present the pseudo-code of the implementation of algorithms to construct JGs for numeric LM-cut. Algorithm 1 shows the procedure to construct a JG given a state s. This algorithm is based on the generalized Dijkstra's algorithm (Key-

$h^{\max}_{\text{cri},+}(s_I,\psi)$	$v \geq 0.6$	g_2	g_1	$u \geq 2$
1	1	2	0.6	2
2	0	0.1	0.6	0
3	0	0	0	0

$\mathsf{cost}(a)$	a_1	a_2	a_3	a_4	a_5
1	0.6	1	1	0	0.1
2	0.6	1	0	0	0.1
3	0	1	0	0	0.1

Table 45: $h^{\max}_{\text{cri},+}(s_I)$ and $\mathsf{cost}(a)$ in each iteration of $h^{\text{LM-cut}}_{\text{cri},+}$ in Example 19.



(a) The first cut. (b) The second cut.

Figure 37: JGs constructed by $h^{\text{LM-cut}}_{\text{cri},+}$ for the RT in Example 19. The functions W and W_1 denote the cut weights of the LM-cut procedures, where action costs are reduced in each iteration.

der & Geffner, 2008), and $f(\psi)$ is the priority of fact ψ in a priority queue Q. While a JG is a triplet $\langle N, E, \mathsf{W}\rangle$, the set of nodes $N = \mathcal{F}_p \cup \bar{\mathcal{F}}_n$ and the weight function $\mathsf{W} : (n_\psi, n_{\psi'}, a) \to \mathsf{W_a}(s, a, \psi') \cdot \mathsf{cost}(a)$ are independent of a state. The algorithm computes the set of edges E and the goal fact g. We assume that the JG is constructed using pcf that justifies some variant of $h^{\max}$, $\hat{h}$, defined in the following form:

$$\hat{h}(s, F) = \max_{\psi \in F} \hat{h}(s, \psi)$$

for any set of facts $F \subseteq \mathcal{F}_p \cup \mathcal{F}_n$, and

$$\hat{h}(s,\psi) = \begin{cases} 0 & \text{if } s \models \psi, \\ \min_{a \in \mathsf{supp}(\psi)} \hat{h}(s, \mathsf{pre}(a)) + \hat{\mathsf{m}}_a(s,\psi) \cdot \mathsf{cost}(a) & \text{otherwise} \end{cases}$$

for a fact $\psi \in \mathcal{F}_p \cup \mathcal{F}_n$. The action multiplier $\hat{\mathsf{m}}_a$ is m_a for $h^{\max}_{\text{cri}}$, m^+_a for $h^{\max}_{\text{cri},+}$, and 1 for $h^{\max}_{\text{ir}}$. Because $\mathsf{pcf}(s, a, \psi)$ is determined only by s and a for these variants, we use $\mathsf{pcf}(s, a)$ instead of $\mathsf{pcf}(s, a, \psi)$ in the pseudo-code. The value of $f(\psi)$ converges to $\hat{h}(s, \psi)$. While the algorithm is common in $h^{\text{LM-cut}}_{\text{cri}}$, $h^{\text{LM-cut}}_{\text{cri},+}$ $h^{\text{LM-cut}}_{\text{ir}}$, $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}}$, and $h^{\text{LM-cut}}_{\text{ir},\mathsf{m}+}$, we need to modify it for $h^{\text{LM-cut}}_{\text{rnd}}$. In $h^{\text{LM-cut}}_{\text{rnd}}$, while m_a is used for $\hat{\mathsf{m}}_a$, lines 14 and 20 are replaced with a random selection from $\{\psi \in \mathsf{pre}(a) \mid s \not\models \psi\}$ and $\{g \in G \mid s \not\models g\}$, respectively.

After the first iteration of the LM-cut, costs of actions included in the extracted cut are updated, and the precondition choice function may be changed. With the extracted cut L and the updated action cost cost^c, subsequent JGs are incrementally constructed by Algorithm 2 until the goal fact g can be reached with zero cost. For $h^{\text{LM-cut}}_{\text{rnd}}$, we need to modify the algorithm by replacing lines 13 and 19 with random selections.

Algorithm 1 An algorithm to construct a JG.

Require: State s.

Ensure: E is the set of edges and g is the goal proposition in the JG.

$E \leftarrow \emptyset$.
$Q \leftarrow \emptyset$.
for all $\psi \in \mathcal{F}_p \cup \bar{\mathcal{F}}_n$ **do**
  $\hat{h}(s, \psi), f(\psi) \leftarrow \infty$.
  if $s \models \psi$ **then**
    $E \leftarrow E \cup \{(n_\emptyset, n_\psi, a_0)\}$.
    $f(\psi) \leftarrow 0$.
    $Q \leftarrow Q \cup \{\psi\}$.
while $Q \neq \emptyset$ **do**
  $\psi \leftarrow \hat{\psi} \in \operatorname{argmin}_{\psi' \in Q} f(\psi)$.
  $Q \leftarrow Q \setminus \{\psi\}$.
  $\hat{h}(s, \psi) = f(\psi)$.
  for all $a \in \{a' \in \mathcal{A} : \psi \in \mathsf{pre}(a') \wedge \forall \psi' \in \mathsf{pre}(a'), \hat{h}(s, \psi') < \infty\}$ **do**
    $\mathsf{pcf}(s, a) = \psi$.
    for all $\psi' \in \mathcal{F}_p \cup \bar{\mathcal{F}}_n : a \in \mathsf{supp}(\psi')$ **do**
      $E \leftarrow E \cup \{(n_{\mathsf{pcf}(s,a)}, n_{\psi'}, a)\}$.
      if $\hat{\mathsf{m}}_a(s, a) \cdot \mathsf{cost}(a) + \hat{h}(s, \mathsf{pcf}(s, a)) < f(\psi')$ **then**
        $f(\psi') \leftarrow \hat{\mathsf{m}}_a(s, a) \cdot \mathsf{cost}(a) + \hat{h}(s, \mathsf{pcf}(s, a))$.
        $Q \leftarrow Q \cup \{\psi'\}$.
$g \leftarrow \hat{g} \in \operatorname{argmax}_{g' \in G} \hat{h}(s, g')$.
return E, g

Algorithm 2 An algorithm to incrementally construct a JG.

Require: State s and the set of edges E in the JG.
Ensure: E is the set of edges and g is the goal proposition in the updated JG.

$Q \leftarrow \emptyset$.
for all $a \in \mathsf{lbl}(L)$ **do**
  for all $\psi \in \mathcal{F}_p \cup \bar{\mathcal{F}}_n : a \in \mathsf{supp}(\psi)$ **do**
    if $\hat{\mathsf{m}}_a(s, a, \psi) \cdot \mathsf{cost}^c(a) + \hat{h}(s, \mathsf{pre}(a)) < f(\psi)$ **then**
      $f(\psi) \leftarrow \hat{\mathsf{m}}_a(s, a, \psi) \cdot \mathsf{cost}^c(a) + \hat{h}(s, \mathsf{pre}(a))$.
      $Q \leftarrow Q \cup \{\psi\}$.
while $Q \neq \emptyset$ **do**
  $\psi \leftarrow \hat{\psi} \in \operatorname{argmin}_{\psi' \in Q} f(\psi)$.
  $Q \leftarrow Q \setminus \{\psi\}$.
  $\hat{h}(s, \psi) \leftarrow f(\psi)$.
  for all $a \in \mathcal{A} : \psi \in \mathsf{pre}(a)$ **do**
    if $\hat{h}(s, \psi) < \max_{\psi' \in \mathsf{pre}(a) \setminus \{\psi\}} \hat{h}(s, \psi')$ **then**
      $\mathsf{pcf}(s, a) \leftarrow \hat{\psi} \in \operatorname{argmax}_{\psi' \in \mathsf{pre}(a) \setminus \{\psi\}} \hat{h}(s, \psi')$.
    for all $\psi' \in \mathcal{F}_p \cup \bar{\mathcal{F}}_n : a \in \mathsf{supp}(\psi')$ **do**
      $E \leftarrow (E \setminus \{(n_\psi, n_{\psi'}, a)\}) \cup \{(n_{\mathsf{pcf}(s,a)}, n_{\psi'}, a)\}$.
      if $\hat{\mathsf{m}}_a(s, a) \cdot \mathsf{cost}^c(a) + \hat{h}(s, \mathsf{pcf}(s, a)) < f(\psi')$ **then**
        $f(\psi') \leftarrow \hat{\mathsf{m}}_a(s, a) \cdot \mathsf{cost}^c(a) + \hat{h}(s, \mathsf{pcf}(s, a))$.
        $Q \leftarrow Q \cup \{\psi'\}$.
$g \leftarrow \hat{g} \in \operatorname{argmax}_{g' \in G} \hat{h}(s, g')$.
return E, g

References

Aldinger, J., & Nebel, B. (2017). Interval based relaxation heuristics for numeric planning with action costs. In *KI 2017: Advances in Artificial Intelligence*, pp. 15–28.

Bonet, B. (2013). An admissible heuristic for SAS+ planning obtained from the state equation. In *Proc. IJCAI*, pp. 2268–2274.

Bonet, B., & Geffner, H. (2001). Planning as heuristic search. *Artif. Intell.*, *129*(1), 5–33.

Bonet, B., & Helmert, M. (2010). Strengthening landmark heuristics via hitting sets. In *Proc. ECAI*, pp. 329–334.

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

Eyerich, P., Mattmüller, R., & Röger, G. (2009). Using the context-enhanced additive heuristic for temporal and numeric planning. In *Proc. ICAPS*, pp. 130–137.

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artif. Intell.*, *2*, 189–208.

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

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

Helmert, M. (2002). Decidability and undecidability results for planning with numerical state variables. In *Proc. AIPS*, pp. 303–312.

Helmert, M. (2006). The Fast Downward planning system. *J. Artif. Intell. Res.*, *26*, 191–246.

Helmert, M., & Domshlak, C. (2009). Landmarks, critical paths and abstractions: What's the difference anyway?. In *Proc. ICAPS*, pp. 162–169.

Hoffmann, J. (2003a). The metric-FF planning system: Translating "ignoring delete lists" to numeric state variables. *J. Artif. Intell. Res.*, *20*, 291–341.

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

Illanes, L., & McIlraith, S. A. (2017). Numeric planning via abstraction and policy guided search. In *Proc. IJCAI*, pp. 4338–4345.

Imai, T., & Fukunaga, A. (2015). On a practical, integer-linear programming model for delete-free tasks and its use as a heuristic for cost-optimal planning. *J. Artif. Intell. Res.*, *54*, 631–677.

Katz, M., & Domshlak, C. (2008). Optimal additive composition of abstraction-based admissible heuristics. In *Proc. ICAPS*, pp. 174–181.

Keyder, E., & Geffner, H. (2008). Heuristics for planning with action costs revisited. In *Proc. ECAI*, pp. 588–592.

Kuroiwa, R., Shleyfman, A., Piacentini, C., Castro, M. P., & Beck, J. C. (2021). LM-cut and operator counting heuristics for optimal numeric planning with simple conditions. In *Proc. ICAPS*. 210–218.

Leofante, F., Giunchiglia, E., Ábrahám, E., & Tacchella, A. (2020). Optimal planning modulo theories. In *Proc. IJCAI*, pp. 4128–4134.

Li, D., Scala, E., Haslum, P., & Bogomolov, S. (2018). Effect-abstraction based relaxation for linear numeric planning. In *Proc. IJCAI*, pp. 4787–4793.

Nakhost, H., Hoffmann, J., & Müller, M. (2012). Resource-constrained planning: A Monte Carlo random walk approach. In *Proc. ICAPS*, pp. 181–189.

Papadimitriou, C. H. (1981). On the complexity of integer programming. *Journal of the ACM*, *28*(4), 765–768.

Piacentini, C., Castro, M. P., Ciré, A. A., & Beck, J. C. (2018a). Compiling optimal numeric planning to mixed integer linear programming. In *Proc. ICAPS*, pp. 383–387.

Piacentini, C., Castro, M. P., Ciré, A. A., & Beck, J. C. (2018b). Linear and integer programming-based heuristics for cost-optimal numeric planning. In *Proc. AAAI*, pp. 6254–6261.

Pommerening, F., & Helmert, M. (2013). Incremental LM-cut. In *Proc. ICAPS*, pp. 162–170.

Pommerening, F., Helmert, M., Röger, G., & Seipp, J. (2015). From non-negative to general operator cost partitioning. In *Proc. AAAI*, pp. 3335–3341.

Pommerening, F., Röger, G., Helmert, M., & Bonet, B. (2014). LP-based heuristics for cost-optimal planning. In *Proc. ICAPS*, pp. 226–234.

Scala, E., Haslum, P., Magazzeni, D., & Thiébaux, S. (2017). Landmarks for numeric planning problems. In *Proc. IJCAI*, pp. 4384–4390.

Scala, E., Haslum, P., & Thiébaux, S. (2016a). Heuristics for numeric planning via subgoaling. In *Proc. IJCAI*, pp. 3228–3234.

Scala, E., Haslum, P., Thiebaux, S., & Ramirez, M. (2016b). Interval-based relaxation for general numeric planning. In *Proc. ECAI*, pp. 655–663.

Scala, E., Haslum, P., Thiébaux, S., & Ramírez, M. (2020). Subgoaling techniques for satisficing and optimal numeric planning. *J. Artif. Intell. Res.*, *68*, 691–752.

Scala, E., Ramírez, M., Haslum, P., & Thiébaux, S. (2016). Numeric planning with disjunctive global constraints via SMT. In *Proc. ICAPS*, pp. 276–284.

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

Wilhelm, A., Steinmetz, M., & Hoffmann, J. (2018). On stubborn sets and planning with resources. In *Proc. ICAPS*, pp. 288–297.

Yang, F., Culberson, J., Holte, R., Zahavi, U., & Felner, A. (2008). A general theory of additive state space abstractions. *J. Artif. Intell. Res.*, *32*, 631–662.

Zukerman, M., Jia, L., Neame, T. D., & Woeginger, G. J. (2001). A polynomially solvable special case of the unbounded knapsack problem. *Oper. Res. Lett.*, *29*(1), 13–16.