Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Higher-Dimensional Potential Heuristics for Optimal Classical Planning

Florian Pommerening, Malte Helmert
University of Basel, Switzerland
{florian.pommerening, malte.helmert}@unibas.ch

Blai Bonet
Universidad Simón Bolívar, Venezuela
bonet@ldc.usb.ve

Abstract

Potential heuristics for state-space search are defined as weighted sums over simple state features. *Atomic* features consider the value of a single state variable in a factored state representation, while *binary* features consider joint assignments to two state variables. Previous work showed that the set of all admissible and consistent potential heuristics using *atomic* features can be characterized by a compact set of linear constraints. We generalize this result to *binary* features and prove a hardness result for features of higher dimension. Furthermore, we prove a tractability result based on the treewidth of a new graphical structure we call the *context-dependency graph*. Finally, we study the relationship of potential heuristics to *transition cost partitioning*. Experimental results show that binary potential heuristics are significantly more informative than the previously considered atomic ones.

Introduction

Potential heuristics (Pommerening et al. 2015) are a family of declarative heuristic functions for state-space search. They fix the form of the heuristic to be a weighted sum over a set of features. Conditions on heuristic values such as admissibility and consistency can then be expressed as constraints on feature weights.

Previous work on admissible potential heuristics is limited to *atomic* features, which consider the value of a single state variable in a factored state representation. Pommerening et al. (2015) show that admissible and consistent potential heuristics over atomic features can be characterized by a compact set of linear constraints. Seipp, Pommerening, and Helmert (2015) introduce several objective functions to select the *best* potential heuristic according to different criteria. They showed that a small collection of diverse potential heuristics closely approximates the state equation heuristic (SEQ) (van den Briel et al. 2007; Bonet 2013; Bonet and van den Briel 2014). Since the quality of the SEQ heuristic is an upper limit on the quality of *any combination* of potential heuristics over atomic features (Pommerening et al. 2015), more complex features are needed to significantly improve heuristic quality of potential heuristics.

Additionally, theoretical analyses of common planning benchmarks (Chen and Giménez 2007; 2009; Lipovetzky and Geffner 2012; Seipp et al. 2016) suggest that potential heuristics for *binary* features (that consider the joint assignment of two state variables rather than valuations over single state variables as in the atomic case) could already lead to a significant increase in accuracy in many planning domains.

In this paper we generalize known results about potential heuristics with atomic features to those with larger features. After introducing some notation, we show that admissible and consistent potential heuristics for binary features are also characterized by a compact set of linear constraints. We then prove that such a compact characterization is not possible in the general case of features mentioning three or more variables. However, we show that compact representations are still possible for "sparse" features, as measured by the treewidth (Dechter 2003) of a new graphical structure we call the context-dependency graph. Finally, we generalize a known relation between atomic potential heuristics and optimal cost partitioning and show that potential heuristics correspond to optimal transition cost partitionings (Keller et al. 2016), i. e., cost partitionings that distribute the cost of each transition instead of each operator.

Background

We consider SAS$^+$ planning tasks (Bäckström and Nebel 1995) in transition normal form (TNF) (Pommerening and Helmert 2015). A planning task is a tuple $\Pi = \langle \mathcal{V}, \mathcal{O}, s_{\mathrm{I}}, s_\star \rangle$ with the following components. $\mathcal{V}$ is a finite set of *variables* where each $V \in \mathcal{V}$ has a finite domain $\mathit{dom}(V)$. A pair $\langle V, v \rangle$ of a variable $V \in \mathcal{V}$ and one of its values $v \in \mathit{dom}(V)$ is called a *fact*. Partial variable assignments p map a subset of variables $\mathit{vars}(p) \subseteq \mathcal{V}$ to values in their domain. Where convenient, we also treat them as sets of facts. A partial variable assignment s with $\mathit{vars}(s) = \mathcal{V}$ is called a *state* and $\mathcal{S}$ is the set of all states. The state s_{I} is the *initial state* of Π and the state $s_\star$ is the *goal state*. (Note that in TNF, there is a single goal state.) We call a partial variable assignment p *consistent* with a state s if s and p agree on all variables in $\mathit{vars}(p)$. The set $\mathcal{O}$ is a finite set of *operators* o, each with a *precondition* $\mathit{pre}(o)$, an *effect* $\mathit{eff}(o)$, and a *cost* $\mathit{cost}(o)$, where $\mathit{pre}(o)$ and $\mathit{eff}(o)$ are both partial variable assignments and $\mathit{cost}(o)$ is a non-negative integer. The restriction to tasks in TNF means that we can assume that $\mathit{vars}(\mathit{pre}(o)) = \mathit{vars}(\mathit{eff}(o))$. We denote this set of variables by $\mathit{vars}(o)$. Considering only tasks in TNF does

not limit generality, since there is an efficient transformation from general SAS$^+$ tasks into equivalent tasks in TNF (Pommerening and Helmert 2015).

An operator o is *applicable* in state s if s is consistent with $pre(o)$. Applying o in s results in the state $s[\![o]\!]$ with $s[\![o]\!][V] = \mathit{eff}(o)[V]$ for all $V \in \mathit{vars}(o)$ and $s[\![o]\!][V] = s[V]$ for all other variables. An operator sequence $\pi = \langle o_1, \ldots, o_n \rangle$ is applicable in state s if there are states $s = s_0, \ldots, s_n$ such that o_i is applicable in s_{i-1} and $s_{i-1}[\![o_i]\!] = s_i$. We write $s[\![\pi]\!]$ for s_n. If $s[\![\pi]\!] = s_\star$, we call π an *s-plan*. If $s = s_\mathrm{I}$, we call it a *plan*. The cost of π under a cost function cost' is $\sum_{i=1}^{n} \mathit{cost}'(o_i)$. An s-plan π with minimal $\mathit{cost}'(\pi)$ among all s-plans is called *optimal* and we write its cost as $h^*(s, \mathit{cost}')$, or $h^*(s)$ if $\mathit{cost}' = \mathit{cost}$.

A *heuristic function* h maps states to values in $\mathbb{R} \cup \{-\infty, \infty\}$. It is *admissible* if $h(s) \leq h^*(s)$ for all $s \in \mathcal{S}$, *goal-aware* if $h(s_\star) \leq 0$, and *consistent* if $h(s) \leq \mathit{cost}(o) + h(s[\![o]\!])$ for all $s \in \mathcal{S}$ and $o \in \mathcal{O}$ that are applicable in s.

A task Π induces a weighted, labeled transition system $\mathrm{TS}_\Pi = \langle \mathcal{S}, \mathcal{T}, s_\mathrm{I}, \{s_\star\} \rangle$ with the set of states $\mathcal{S}$, the initial state s_I, the single goal state $s_\star$ and set of transitions $\mathcal{T}$: for each $s \in \mathcal{S}$ and $o \in \mathcal{O}$ that is applicable in s, there is a transition $s \xrightarrow{o} s[\![o]\!]$ labeled with o and weighted with $\mathit{cost}(o)$. Shortest paths in TS_Π correspond to optimal plans for Π.

A conjunction of facts is called a *feature* and the number of conjuncts is called its *size*. Features of size 1 and 2 are called *unary* and *binary* features. We say a feature f *is true* in a state s (written as $s \vDash f$) if all its facts are in s. A *weight function* for features $\mathcal{F}$ is a function $w : \mathcal{F} \to \mathbb{R}$. The *potential* of a state s under a weight function w is

$$\varphi(s) = \sum_{f \in \mathcal{F}} w(f)[s \vDash f],$$

where the bracket is an indicator function (Knuth 1992). We call φ the *potential heuristic* for features $\mathcal{F}$ and weights w. Its *dimension* is the size of a largest feature $f \in \mathcal{F}$.

Two-Dimensional Potential Heuristics

Two-dimensional potential heuristics only consider atomic and binary features. We use that consistent heuristics are admissible iff they are goal-aware (Russell and Norvig 1995).

Let $\Pi = \langle \mathcal{V}, \mathcal{O}, s_\mathrm{I}, s_\star, \mathit{cost} \rangle$ be a planning task in TNF and $\mathrm{TS}_\Pi = \langle \mathcal{S}, \mathcal{T}, s_\mathrm{I}, \{s_\star\} \rangle$ its transition system. A potential heuristic φ over features $\mathcal{F}$ is goal-aware and consistent iff it satisfies the following constraints:

$$\varphi(s_\star) \leq 0, \tag{1}$$

$$\varphi(s) - \varphi(s') \leq \mathit{cost}(o) \qquad \text{for } s \xrightarrow{o} s' \in \mathcal{T}. \tag{2}$$

This set of constraints has exponential size as there is one constraint for each transition $s \xrightarrow{o} s'$ in $\mathcal{T}$.

Constraint (1) is a linear constraint over the weights, which is easy to see since $\varphi(s_\star) = \sum_{f \in \mathcal{F}} w(f)[s_\star \vDash f]$.

Next, let $o \in \mathcal{O}$ be a fixed operator and consider constraint (2). Replacing $\varphi(s)$ and $\varphi(s')$ by their definitions, we get the equivalent constraint

$$\sum_{f \in \mathcal{F}} w(f)([s \vDash f] - [s' \vDash f]) \leq \mathit{cost}(o) \tag{3}$$

for all transitions $s \xrightarrow{o} s' \in \mathcal{T}$. We abbreviate the change of a feature's truth value ($[s \vDash f] - [s[\![o]\!] \vDash f]$) as $\Delta_o(f, s)$.

We partition the set of features into three subsets: *irrelevant* features $\mathcal{F}^{\mathrm{irr}}$ have no variables in common with $\mathit{vars}(o)$, *context-independent* features $\mathcal{F}^{\mathrm{ind}}$ mention only variables in $\mathit{vars}(o)$, and the remaining *context-dependent* features $\mathcal{F}^{\mathrm{ctx}}$ mention one variable from $\mathit{vars}(o)$ and another variable not in $\mathit{vars}(o)$. We write $\Delta_o^{\mathrm{irr}}(s)$ for $\sum_{f \in \mathcal{F}^{\mathrm{irr}}} w(f)\Delta_o(f, s)$ and analogously $\Delta_o^{\mathrm{ind}}(s)$ and $\Delta_o^{\mathrm{ctx}}(s)$.

The truth value of an irrelevant feature never changes by applying o in some state. Thus, $\Delta_o^{\mathrm{irr}}(s) = 0$ for all states s.

For a context-independent feature f, the effect of applying o in s is completely determined by o: f holds in s iff f is entailed by the precondition, and in $s[\![o]\!]$ iff it is entailed by the effect. Thus, $\Delta_o(f, s) = [\mathit{pre}(o) \vDash f] - [\mathit{eff}(o) \vDash f]$ for every state s in which o is applicable. Clearly, $\Delta_o(f, s)$ and $\Delta_o^{\mathrm{ind}}(s)$ do not depend on the state s for $f \in \mathcal{F}^{\mathrm{ind}}$ and we write $\Delta_o(f)$ and Δ_o^{ind}.

A feature $f \in \mathcal{F}^{\mathrm{ctx}}$ is a conjunction $f = f_o \wedge f_{\bar{o}}$ where f_o is a fact over a variable in $\mathit{vars}(o)$ and $f_{\bar{o}}$ is a fact over a variable in $\mathcal{V}_{\bar{o}} = \mathcal{V} \setminus \mathit{vars}(o)$. If o is applied in a state s with $s \nvDash f_{\bar{o}}$, then $s \nvDash f$ and $s[\![o]\!] \nvDash f$, so $\Delta_o(f, s) = 0$. For the remaining features, we know that $f_{\bar{o}}$ is present in both s and $s[\![o]\!]$ and the truth value of f_o is solely determined by o. Thus $\Delta_o(f, s) = \Delta_o(f_o)[s \vDash f_{\bar{o}}]$. If o is applicable in s,

$$\Delta_o^{\mathrm{ctx}}(s) = \sum_{V \in \mathcal{V}_{\bar{o}}} \sum_{\substack{f \in \mathcal{F}^{\mathrm{ctx}} \\ f = f_o \wedge f_{\bar{o}} \\ \mathit{vars}(f_{\bar{o}}) = \{V\}}} w(f)\Delta_o(f, s) \tag{4}$$

$$= \sum_{V \in \mathcal{V}_{\bar{o}}} \sum_{\substack{f \in \mathcal{F}^{\mathrm{ctx}} \\ f = f_o \wedge f_{\bar{o}}}} w(f)\Delta_o(f_o)[f_{\bar{o}} = \langle V, s[V] \rangle] \tag{5}$$

$$\leq \sum_{V \in \mathcal{V}_{\bar{o}}} \sum_{\substack{f \in \mathcal{F}^{\mathrm{ctx}} \\ f = f_o \wedge f_{\bar{o}}}} w(f)\Delta_o(f_o)[f_{\bar{o}} = \langle V, v_V^* \rangle] \tag{6}$$

where

$$v_V^* = \operatorname*{arg\,max}_{v \in \mathit{dom}(V)} \sum_{\substack{f \in \mathcal{F}^{\mathrm{ctx}} \\ f = f_o \wedge f_{\bar{o}}}} w(f)\Delta_o(f_o)[f_{\bar{o}} = \langle V, v \rangle].$$

If we denote the inner sum in (6) with b_V^o, then

$$\varphi(s) - \varphi(s') \leq \Delta_o^{\mathrm{ind}} + \sum_{V \in \mathcal{V}_{\bar{o}}} b_V^o \tag{7}$$

for all transitions $s \xrightarrow{o} s' \in \mathcal{T}$. Therefore, if $\Delta_o^{\mathrm{ind}} + \sum_{V \in \mathcal{V}_{\bar{o}}} b_V^o \leq \mathit{cost}(o)$ for all operators o, then φ is consistent. Conversely, if φ is consistent, then $\varphi(s) - \varphi(s[\![o]\!]) \leq \mathit{cost}(o)$ for operator o and the states s in which o is applicable. In particular, for states s^* such that $s^*[V] = \mathit{pre}(o)[V]$ for $V \in \mathit{vars}(o)$, and $s^*[V] = v_V^*$ otherwise. It is then not difficult to check that the inequality in (7) is tight for such states s^*. Hence, φ is consistent iff $\Delta_o^{\mathrm{ind}} + \sum_{V \in \mathcal{V}_{\bar{o}}} b_V^o \leq \mathit{cost}(o)$ for all operators o.

Putting everything together, we see that the constraint (3) for a fixed operator o is equivalent to the constraints

$$\Delta_o^{\mathrm{ind}} + \sum_{V \in \mathcal{V}_{\bar{o}}} z_V^o \leq \mathit{cost}(o), \tag{8}$$

$$z^o_V \geq \sum_{\substack{f \in \mathcal{F}^{\text{ctx}} \\ f = f_o \wedge \langle V, v \rangle}} w(f)\Delta_o(f_o) \quad \text{for } V \in \mathcal{V}_{\bar{o}}, v \in \mathit{dom}(V) \quad (9)$$

where z^o_V is a new variable that upper bounds b^o_V. This set of constraints has $O(|\mathcal{V}|d)$ constraints where d bounds the size of the variable domains, while each constraint has size $O(|\mathcal{F}| + |\mathcal{V}|)$. The set of constraints is over the variables $\{w(f) : f \in \mathcal{F}\} \cup \{z^o_V : o \in \mathcal{O}, V \in \mathcal{V}_{\bar{o}}\}$.

Theorem 1. *Let $\mathcal{F}$ be a set of features of size at most 2 for a planning task Π. The set of solutions to the constraints* (1)*, and* (8)–(9) *for each operator, projected to w, corresponds to the set of weight functions of admissible and consistent potential heuristics for Π over $\mathcal{F}$. The total number of constraints is $O(|\mathcal{O}||\mathcal{V}|d)$, where d bounds the size of variable domains, while each constraint has size $O(|\mathcal{F}| + |\mathcal{V}|)$.*

High-Dimensional Potential Heuristics

In this section we show that a general result like Theorem 1 is not possible, unless NP equals P, for sets of features of dimension 3 or more, but we identify classes of problems on which potential heuristics can be characterized compactly.

Intractability

Theorem 1 allows one to answer many interesting questions in polynomial time about potential heuristics of dimension 2. In particular, by solving a single LP one can test whether a given potential heuristic is consistent and/or goal-aware. We use this idea to show that no general result like Theorem 1 is possible for potential heuristics of dimension 3, by making a reduction of non-3-colorability (a decision problem that is complete for coNP (Garey and Johnson 1979)) into the problem of testing whether a potential heuristic of dimension 3 is consistent.

Let $G = \langle V, E \rangle$ be an undirected graph. We first construct, in polynomial time, a planning task $\Pi = \langle \mathcal{V}, \mathcal{O}, s_{\text{I}}, s_\star \rangle$ in TNF and a potential heuristic φ of dimension 3 such that G is not 3-colorable iff φ is consistent. The task Π has $|V| + 1$ variables: one variable C_v for the color of each vertex $v \in V$ that can be either red, blue, or green, and one "master" binary variable denoted by M.

For every vertex $v \in V$ and pair of different colors $c, c' \in \mathit{dom}(C_v)$, there is a unique operator $o_{v,c,c'}$ of zero cost that changes C_v from c to c' when $M = 0$. For the variable M, there is a unique operator o_M, also of zero cost, that changes M from 0 to 1. These are all the operators in the task Π.

Each state $s \in \mathcal{S}$ encodes a coloring of G, where the color of vertex v is the value $s[C_v]$ of the state variable C_v. The initial state s_{I} is set to an arbitrary coloring but with the master variable set to 0; e. g., $s_{\text{I}}[M] = 0$ and $s_{\text{I}}[C_v] = \mathit{red}$ for every vertex $v \in V$. The goal state $s_\star$ is also set to an arbitrary coloring but with $s_\star[M] = 1$; e. g., $s_\star[M] = 1$ and $s_\star[C_v] = \mathit{red}$ for every vertex $v \in V$.

The potential heuristic φ of dimension 3 is constructed as follows. For features f with $\mathit{vars}(f) = \{M, C_u, C_v\}$ such that $\{u, v\} \in E$ is an edge in the graph, let its weight $w(f) = -1$ when $f[M] = 1$ and $f[C_u] \neq f[C_v]$, and $w(f) = 0$ otherwise. For the feature $f_M = \langle M, 1 \rangle$ of dimension 1, let $w(f_M) = |E| - 1$. The weight $w(f)$ for all other features f is set to 0.

Let us now reason about the states of the task Π and the values assigned to them by the heuristic. Let s be a state for Π. If $s[M] = 0$, then $\varphi(s) = 0$. If $s[M] = 1$, then no operator is applicable at s and $\varphi(s) \geq -1$, with $\varphi(s) = -1$ iff s encodes a 3-coloring of G, as the feature f_M contributes a value of $|E| - 1$ to $\varphi(s)$, while the features corresponding to edges contribute a value of $-|E|$ when s encodes a coloring.

Let us consider a transition $s \xrightarrow{o} s' \in \mathcal{T}$. Clearly, $s[M] \leq s'[M]$ as no operator decreases the value of M. If $s[M] = s'[M] = 0$, then $\varphi(s) = \varphi(s') = 0$. If $s[M] = 0$ and $s'[M] = 1$, then $\varphi(s) \leq \varphi(s')$ iff s' does not encode a 3-coloring of G. The case $s[M] = s'[M] = 1$ is not possible as no operator is applicable in states with $s[M] = 1$. Therefore, since all operator costs are equal to zero, φ is consistent iff there is no transition $s \xrightarrow{o} s'$ with $s[M] = 0$, $s'[M] = 1$ and s' encoding a 3-coloring of G, and the latter iff the graph G is not 3-colorable.

Finally, observe that testing whether a potential function φ is inconsistent can be done in non-deterministic polynomial time: guess a state s and an operator o, and check whether $\varphi(s) > \varphi(s[\![o]\!]) + \mathit{cost}(o)$.

Theorem 2. *Let $\mathcal{F}$ be a set of features for a planning task Π, and let φ be a potential heuristic over $\mathcal{F}$. Testing whether φ is consistent is coNP-complete.*

Parametrized Tractability

We first give an algorithm for maximizing a sum of functions using linear programming, and then apply it to characterize high-dimensional potential heuristics.

Maximizing a Sum of Functions. Let $\mathcal{X}$ be a set of finite-domain variables. We extend the notation $\mathit{dom}(\mathcal{X})$ to mean the set of variable assignments over $\mathcal{X}$. For an assignment $\nu \in \mathit{dom}(\mathcal{X})$ and a *scope* $S \subseteq \mathcal{X}$, we use $\nu|_S$ to describe the restriction of ν to S. Let Ψ be a set of scoped functions $\langle S, \psi \rangle$ with $S \subseteq \mathcal{X}$ and $\psi : S \to \mathbb{V}$ for a set of values $\mathbb{V}$. For now, think of $\mathbb{V}$ as the real numbers $\mathbb{R}$, but we will later generalize this. We only require that maximization and addition is defined on values in $\mathbb{V}$, that both operations are commutative and associative, and that $\max\{a + c, b + c\} \equiv c + \max\{a, b\}$ for all $a, b, c \in \mathbb{V}$. We are interested in the value $\mathit{Max}(\Psi) = \max_{\nu \in \mathit{dom}(\mathcal{X})} \sum_{\langle S, \psi \rangle \in \Psi} \psi(\nu|_S)$.

Computing $\mathit{Max}(\Psi)$ is the goal of constraint optimization for extensional constraints, an important problem in AI. It is challenging because the number of valuations in $\mathit{dom}(\mathcal{X})$ is exponential in the number $|\mathcal{X}|$ of variables. Bucket elimination (Dechter 2003) is a well-known algorithm to compute $\mathit{Max}(\Psi)$. For reasons that will become clear later in this section, we describe the bucket elimination algorithm in a slightly unusual way: in our formulation, the algorithm generates a system of equations, and its output can be extracted from the (uniquely defined) solution to these equations. The system of equations makes use of auxiliary variables $\mathit{Aux}_1, \ldots, \mathit{Aux}_m$ that take values from $\mathbb{V}$. The generated equations have the form $\mathit{Aux}_i = \max_{j \in \{1, \ldots, k_i\}} e_{i,j}$,

where $e_{i,j}$ is a sum that contains only values from $\mathbb{V}$ or the variables $Aux_1, \ldots, Aux_{i-1}$. Solutions to the system of equations guarantee that $Aux_m \equiv Max(\Psi)$.

Bucket Elimination We now describe the general algorithm and state its correctness (without proof due to lack of space). Its execution depends on an order $\sigma = \langle X_1, \ldots, X_n \rangle$ of the variables in $\mathcal{X}$. The algorithm operates in stages which are enumerated in a decreasing manner, starting at stage $n+1$ and ending at stage 0:

- Stage $n+1$ (Initialization). Start with a set $\{B_i\}_{i=0}^n$ of *empty* buckets. Place each $\langle S, \psi \rangle \in \Psi$ into the bucket B_i if X_i is the largest variable in S, according to σ, or into the bucket B_0 if $S = \emptyset$.
- Stages $i = n, \ldots, 1$ (Elimination). Let $\langle S_j, \psi_j \rangle$ for $j \in \{1, \ldots, k_i\}$ be the scoped functions currently in bucket B_i. Construct the scope $S_{X_i} = (\bigcup_{j \in \{1,\ldots,k_i\}} S_j) \setminus \{X_i\}$ and the function $\psi_{X_i} : S_{X_i} \to \mathbb{V}$ that represents the contribution of all functions that depend on X_i. The definition of ψ_{X_i} is added to the generated system of equations by adding one auxiliary variable $Aux_{X_i,\nu}$ for every $\nu \in dom(S_{X_i})$ which represents the value $\psi_{X_i}(\nu)$:

  $$Aux_{X_i,\nu} = \max_{x_i \in dom(X_i)} \sum_{j \in \{1,\ldots,k_i\}} \psi_j(\nu_{x_i}|_{S_j})$$

  where $\nu_{x_i} = \nu \cup \{X_i \mapsto x_i\}$ extends the variable assignment ν with $X_i \mapsto x_i$. If ψ_j is a function in Ψ, then $\psi_j(\nu_{x_i}|_{S_j})$ is an element of $\mathbb{V}$. Otherwise ψ_j is a previously defined function $\psi_{X_{i'}}$ for $i' > i$ and its value for $\nu' = \nu_{x_i}|_{S_j}$ is represented by $Aux_{X_{i'},\nu'}$.

  The newly defined function ψ_{X_i} no longer depends on X_i but depends on all variables in S_{X_i}, so $\langle S_{X_i}, \psi_{X_i} \rangle$ is added to bucket B_j if X_j is the largest variable in S_{X_i} according to σ or to B_0 if $S_{X_i} = \emptyset$. Observe that $j < i$ because S_{X_i} only contains variables from scopes where X_i is the largest variable and S_{X_i} does not contain X_i.
- Stage 0 (Termination). Let $\langle S_j, \psi_j \rangle$ for $j \in \{1, \ldots, k\}$ be the scoped functions currently in bucket B_0. Add the auxiliary variable Aux_Ψ and the equation $Aux_\Psi = \sum_{j \in \{1,\ldots,k\}} \psi_j$ analogously to the elimination step. (All S_j are empty and the maximum is over $dom(\emptyset) = \{\emptyset\}$.)

Example. Consider $\Psi = \{\langle \{X\}, f \rangle, \langle \{X, Y\}, g \rangle\}$ over the binary variables $\mathcal{X} = \{X, Y\}$. Bucket elimination generates the following system of equations for the variable order $\sigma = \langle X, Y \rangle$.

$$\begin{aligned}
Aux_1 = Aux_{Y,\{X \mapsto 0\}} &= \max_{y \in \{0,1\}} g(0, y) \\
&= \max\{g(0,0), g(0,1)\} \\
Aux_2 = Aux_{Y,\{X \mapsto 1\}} &= \max_{y \in \{0,1\}} g(1, y) \\
&= \max\{g(1,0), g(1,1)\} \\
Aux_3 = Aux_{X,\emptyset} &= \max_{x \in \{0,1\}} (f(x) + Aux_{Y,\{X \mapsto x\}}) \\
&= \max\{f(0) + Aux_1, f(1) + Aux_2\} \\
Aux_4 = Aux_\Psi &= Aux_{X,\emptyset} = Aux_3
\end{aligned}$$

Bucket Elimination for Linear Expressions As a generalization of the bucket elimination algorithm, consider $\mathbb{V}$ to be the following set $\mathbb{E}$ of mathematical expressions over a set of variable symbols $\mathcal{Y}$. For every $Y \in \mathcal{Y}$ and $r \in \mathbb{R}$, the expressions Y, r, and rY are in $\mathbb{E}$. If a and b are elements of $\mathbb{E}$, then the expressions $(a+b)$ and $\max\{a, b\}$ are elements of $\mathbb{E}$. There are no additional elements in $\mathbb{E}$. An assignment $f : \mathcal{Y} \to \mathbb{R}$ that maps variables to values can be extended to $\mathbb{E}$ in the straight-forward way. Two expressions $a, b \in \mathbb{E}$ are equivalent if $f(a) = f(b)$ for all assignments f. An expression is *linear* if it does not mention $\max$.

Clearly, maximization and addition are commutative, associative and satisfy $\max\{a+c, b+c\} \equiv c + \max\{a, b\}$ for all expressions $a, b, c \in \mathbb{E}$. Bucket elimination therefore generates a system of equations $Aux_i = \max_{j \in \{1,\ldots,k_i\}} e_{i,j}$, where all $e_{i,j}$ are sums over expressions and variables $Aux_{i'}$ with $i' < i$. Since a variable is a mathematical expression, the whole result can be seen as a system of equations over the variables $\mathcal{Y} \cup \{Aux_1, \ldots, Aux_m\}$. If additionally all functions in Ψ only produce linear expressions over $\mathcal{Y}$, then in the resulting system all $e_{i,j}$ are linear expressions over $\mathcal{Y} \cup \{Aux_1, \ldots, Aux_m\}$.

Consider the example problem again. We define f and g so they map to linear expressions over $\mathcal{Y} = \{a, b\}$:

$f(x)$	$x=0$	$x=1$
	$3a-2b$	$4a+2b$

$g(x,y)$	$x=0$	$x=1$
$y=0$	$8a$	$-3b$
$y=1$	$7b$	0

In the resulting system of equations all elements in the maxima are linear expressions over the variables $\{a, b, Aux_1, Aux_2, Aux_3, Aux_4\}$:

$$\begin{aligned}
Aux_1 &= \max\{8a, 7b\} \\
Aux_2 &= \max\{-3b, 0\} \\
Aux_3 &= \max\{3a - 2b + Aux_1, 4a + 2b + Aux_2\} \\
Aux_4 &= Aux_3
\end{aligned}$$

Bucket elimination guarantees that $Aux_4 \equiv Max(\Psi)$ for any value of a and b in $\mathbb{E}$. We argue that this system of equations can be solved by an LP solver.

Theorem 3. *Let $\mathcal{Y}$ and $\{Aux_1, \ldots Aux_m\}$ be disjoint sets of variables. Let $P_{\max}$ be the system of equations $Aux_i = \max_{j \in \{1,\ldots,k_i\}} e_{i,j}$ for $i \in \{1, \ldots, m\}$, where $e_{i,j}$ is a linear expression over variables $\mathcal{Y} \cup \{Aux_1, \ldots, Aux_{i-1}\}$. Let P_{LP} be the set of linear constraints $Aux_i \geq e_{i,j}$ for $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, k_i\}$.*

Every solution of $P_{\max}$ is a solution of P_{LP} and for every solution f of P_{LP} there is a solution f' of $P_{\max}$ with $f'(Y) = f(Y)$ for all $Y \in \mathcal{Y}$ and $f'(Aux) \leq f(Aux)$ for all $Aux \notin \mathcal{Y}$.

As a corollary, we can use this result to represent a "symbolic" version of the bucket elimination algorithm with unknowns $\mathcal{Y}$ as an LP. (Note that the constraints generated by the bucket elimination algorithm have exactly the form required by the theorem if functions in Ψ produce linear expressions.) This LP has the property that for every assignment to the unknowns $\mathcal{Y}$ there exists a feasible solution, and the values of Aux_m in these feasible solutions are exactly the set of numbers greater or equal to $Max(\Psi)$ for the given assignment to $\mathcal{Y}$. (For simplicity, it would be preferable if *only*

Max(Ψ) itself resulted in a feasible assignment to *Aux*$_m$, but we will see that the weaker property where *Aux*$_m$ may overestimate *Max*(Ψ) is sufficient for our purposes.)

We denote the set of constraints for this LP by $P_{\text{LP}(\Psi,\sigma)}$. This LP can be solved in time that is polynomial in the size of $P_{\text{LP}(\Psi,\sigma)}$, so to bound the complexity, we have to consider the number and size of the constraints in $P_{\text{LP}(\Psi,\sigma)}$.

Dechter (2003) defines the *dependency graph* of a problem Ψ over variables $\mathcal{X}$ as the undirected graph $G(\Psi) = \langle \mathcal{X}, E \rangle$ with set of vertices given by the variables $\mathcal{X}$ and an edge $\langle X, X' \rangle \in E$ iff $X \neq X'$ and there is a scoped function $\langle S, \psi \rangle$ in Ψ with $\{X, X'\} \subseteq S$. Given an undirected graph G and an order of its nodes σ, a *parent* of a node n is a neighbor of n that precedes n in σ. Dechter defines the *induced graph* of G along σ as the result of processing each node of G in descending order of σ and for each node connecting each pair of its parents if they are not already connected. The induced width of G along σ then is the maximal number of parents of a node in the induced graph of G along σ.

If there are n variables in $\mathcal{X}$ and each of their domains is bounded by d, then eliminating variable X_i adds one equation *Aux*$_{X_i,\nu} = \max_{j \in dom(X_i)} e_{i,j}$ for each valuation ν of the scope S_{X_i} (variables relevant for X_i). The size of this scope is limited by the induced width $w(\sigma)$, so the number of valuations is limited by $d^{w(\sigma)}$. As there are n buckets to eliminate, the number of auxiliary variables in the LP can thus be bounded by $O(nd^{w(\sigma)})$. Each such variable occurs in $|dom(X_i)| \leq d$ constraints of the form *Aux*$_{X_i,\nu} \geq e_{i,j}$ in $P_{\text{LP}(\Psi,\sigma)}$, so there are $O(nd^{w(\sigma)+1})$ constraints.

Theorem 4. *Let Ψ be a set of scoped functions over the variables in $\mathcal{X}$ that map to linear expressions over $\mathcal{Y}$. Let σ be an ordering for $\mathcal{X}$.*

Then $P_{\text{LP}(\Psi,\sigma)}$ has $O(|\mathcal{Y}| + |\mathcal{X}|d^{w(\sigma)})$ variables and $O(|\mathcal{X}|d^{w(\sigma)+1})$ constraints, where $d = \max_{X \in \mathcal{X}} |dom(X)|$ and $w(\sigma)$ is the induced width of $G(\Psi)$.

The smallest possible induced width of $G(\Psi)$ along any order σ is called the induced width of $G(\Psi)$ and equals the treewidth of $G(\Psi)$ (Dechter 2003). Unfortunately, finding the induced width or a minimizing order is NP-hard. However, it is fixed-parameter tractable (Downey and Fellows 1999) with the treewidth as the parameter (Bodlaender 1996).

High-Dimensional Potential Functions. Of the two conditions that characterize goal-aware and consistent potential heuristics, constraint (2) is the most challenging to test as it really denotes an exponential number of constraints, one per state. The constraint is equivalent to

$$\begin{aligned} cost(o) &\geq \max_{s \vDash pre(o)} (\varphi(s) - \varphi(s[\![o]\!])) \\ &= \max_{s \vDash pre(o)} \sum_{f \in \mathcal{F}} w(f) \Delta_o(f, s) \\ &= \max_{\nu \in dom(\mathcal{V}_{\bar{o}})} \sum_{f \in \mathcal{F}} w(f) \Delta_o(f, s_{o,\nu}) \end{aligned}$$

for all operators $o \in \mathcal{O}$ where the state $s_{o,\nu}$ is $s_{o,\nu} = pre(o) \cup \nu$.

As done before, for a fixed operator $o \in \mathcal{O}$, we partition $\mathcal{F}$ as $\mathcal{F} = \mathcal{F}^{\text{irr}} \cup \mathcal{F}^{\text{ind}} \cup \mathcal{F}^{\text{ctx}}$ and consider the functions $\Delta_o^{\text{irr}}(s)$, $\Delta_o^{\text{ind}}(s)$ and $\Delta_o^{\text{ctx}}(s)$ for states s on which the operator o is applicable. We have $\Delta_o^{\text{irr}}(s) = 0$ and $\Delta_o^{\text{ind}}(s) = \Delta_o^{\text{ind}}$ as the value of the latter is independent of the state s. Therefore, the constraint (2) is equivalent to

$$cost(o) \geq \Delta_o^{\text{ind}} + \max_{\nu \in dom(\mathcal{V}_{\bar{o}})} \Delta_o^{\text{ctx}}(s_{o,\nu}) \tag{10}$$

where $\Delta_o^{\text{ctx}}(s_{o,\nu}) = \sum_{f \in \mathcal{F}^{\text{ctx}}} w(f) \Delta_o(f, s_{o,\nu})$. Observe that for $f \in \mathcal{F}^{\text{ctx}}$, the value of $\Delta_o(f, s_{o,\nu})$ is equal to

$$[pre(o) \cup s_{o,\nu}|_{\mathcal{V}_f \setminus \mathcal{V}_o} \vDash f] - [\mathit{eff}(o) \cup s_{o,\nu}[\![o]\!]|_{\mathcal{V}_f \setminus \mathcal{V}_o} \vDash f].$$

Let us define the functions ψ_o^f that maps partial assignments $\nu \in dom(\mathcal{V}_f \setminus \mathcal{V}_o)$ to expressions in $\{0, w(f), -w(f)\}$ as

$$\psi_o^f(\nu) = w(f)([pre(o) \cup \nu \vDash f] - [\mathit{eff}(o) \cup \nu \vDash f]).$$

For operator $o \in \mathcal{O}$, we define $\Psi_o = \{\psi_o^f : f \in \mathcal{F}^{\text{ctx}}\}$. Then

$$\begin{aligned} \max_{\nu \in dom(\mathcal{V}_{\bar{o}})} \Delta_o^{\text{ctx}}(s_{o,\nu}) &= \max_{\nu \in dom(\mathcal{V}_{\bar{o}})} \sum_{f \in \mathcal{F}^{\text{ctx}}} \psi_o^f(s_{o,\nu}|_{\mathcal{V}_f \setminus \mathcal{V}_o}) \\ &= \mathit{Max}(\Psi_o). \end{aligned}$$

Constraint (10) is then equivalent to

$$cost(o) \geq \Delta_o^{\text{ind}} + \mathit{Max}(\Psi_o).$$

Applying Theorem 4 to Ψ_o along an ordering σ_o for the variables $\mathcal{V}$ in Π, we obtain a set of constraints $P_{\text{LP}(\Psi_o,\sigma_o)}$ that characterize *Max*(Ψ_o). We can replace the above constraint by $cost(o) \geq \Delta_o^{\text{ind}} + \mathit{Aux}_{\Psi_o}$ and the constraints in $P_{\text{LP}(\Psi_o,\sigma_o)}$. The constraints $P_{\text{LP}(\Psi_o,\sigma_o)}$ allow setting *Aux*$_{\Psi_o}$ higher than necessary, but this is never beneficial as long as *Aux*$_{\Psi_o}$ does not occur in the objective.

Theorem 5. *Let $\mathcal{F}$ be a set of features for a planning task Π, and let $\{\sigma_o\}_{o \in \mathcal{O}}$ be an indexed collection of orderings of the variables in Π (one ordering σ_o for each operator $o \in \mathcal{O}$). Then, the set of solutions to*

$$\varphi(s_\star) \leq 0, \tag{11}$$

$$\Delta_o^{\text{ind}} + \mathit{Aux}_{\Psi_o} \leq cost(o), \quad \text{for each operator } o \in \mathcal{O}, \tag{12}$$

$$P_{\text{LP}(\Psi_o,\sigma_o)} \quad \text{for each operator } o \in \mathcal{O} \tag{13}$$

(projected on the feature weights) corresponds to the set of weight functions of admissible and consistent potential heuristics for Π over $\mathcal{F}$.

We finish the section by bounding the number and size of the constraints in $P_{\text{LP}(\Psi_o,\sigma_o)}$. We define the *context-dependency graph* $G(\Pi, \mathcal{F}, o)$ for a planning task Π, features $\mathcal{F}$ and an operator o as follows: the vertices are the variables of Π and there is an edge between V and V' with $V \neq V'$ iff there is a feature $f \in \mathcal{F}$ with $vars(f) \cap vars(o) \neq \emptyset$ and $\{V, V'\} \subseteq vars(f) \setminus vars(o)$.

Theorem 6. *Let $\mathcal{F}$ be a set of features for a planning task Π, let o be an operator of Π, and let σ_o an ordering on the variables in Π. The number of constraints in $P_{\text{LP}(\Psi_o,\sigma_o)}$ is $O(nd^{w(\sigma_o)+1})$ where n is the number of variables, d bounds the size of the variable domains, and $w(\sigma_o)$ is the induced width of $G(\Pi, \mathcal{F}, o)$ along the ordering σ_o. The number of variables in $P_{\text{LP}(\Psi_o,\sigma_o)}$ is $O(|\mathcal{F}| + nd^{w(\sigma_o)})$.*

By combining the constraints from the different operators according to Theorem 5, we thus obtain the following fixed-parameter tractability result.

Corollary 1. *Let $\mathcal{F}$ be a set of features for a planning task Π. Define w^* as the maximum treewidth of all context-dependency graphs for Π and $\mathcal{F}$, and define d as the maximum domain size of all variables of Π.*

Computing a set of linear constraints that characterize the admissible and consistent potential heuristics with features $\mathcal{F}$ for Π is fixed-parameter tractable with parameter $\max(w^*, d)$.

We remark that the general result (again) implies a polynomial characterization of potential heuristics where all features have dimension at most 2. In this case, no feature can simultaneously include a variable mentioned in a given operator o and two further variables not mentioned in o, and hence all context-dependency graphs are devoid of edges. In edge-free graphs, all orderings have width 0, and hence for each operator o, $P_{\mathrm{LP}(\Psi_o, \sigma_o)}$ has $O(nd)$ constraints and $O(|\mathcal{F}| + n)$ variables. The parameter w^* is 0 in this case.

Relation to Cost Partitioning

Operator cost partitioning (Katz and Domshlak 2007; Yang et al. 2008; Katz and Domshlak 2010) is a technique to make the sum of several heuristics admissible by distributing the cost of each operator between them. Katz and Domshlak (2010) show how the *optimal operator cost partitioning* (OCP) can be computed in polynomial time for a large family of abstraction heuristics. Pommerening et al. (2015) show that an admissible and consistent potential heuristic over all atomic features that achieves the maximal initial heuristic value corresponds to an OCP over atomic projection heuristics. Here, we extend this result to all potential heuristics that use the abstract states of a set of abstractions as features. To that end, we first discuss a generalization of OCP, which we call *optimal transition cost partitioning* (TCP) (Keller et al. 2016)[1].

Definition 1. *Let Π be a planning task with cost function cost and transitions $\mathcal{T}$. A set of* transition cost functions *$cost_i : \mathcal{T} \to \mathbb{R}$, $1 \leq i \leq n$, is a* transition cost partitioning *if*

$$\sum_{i=1}^{n} cost_i(s \xrightarrow{o} s') \leq cost(o) \qquad \text{for all } s \xrightarrow{o} s' \in \mathcal{T}.$$

To use transition cost partitioning, the notion of operator cost functions must be extended to transition cost functions with possibly negative values. For example, the cost of an s-plan $\pi = \langle o_1, \ldots, o_n \rangle$ under transition cost function *cost′* is $\sum_{i=1}^{n} cost'(s_{i-1} \xrightarrow{o_i} s_i)$, where $s = s_0, \ldots, s_n = s[\![\pi]\!]$ are the states visited by the plan π. The cheapest plan cost under *cost′* can now be negative or even $-\infty$ (in the case of negative cost cycles).

Proposition 1 (Keller et al. (2016))**.** *Let Π be a planning task, $P = \langle cost_1, \ldots, cost_n \rangle$ be a transition cost partitioning for Π, and $h_1, \ldots, h_n$ be admissible heuristics for Π. The heuristic $h_P(s) = \sum_{i=1}^{n} h_i(s, cost_i)$ is admissible.*

[1]Keller et al. use *state-dependent cost partitioning*, which may be confused with partitionings that are re-optimized for each state.

Definition 2. *Let Π be a planning task, $h_1, \ldots, h_n$ be admissible heuristics for Π, and $\mathcal{P}$ be the set of all transition cost partitionings. The* optimal transition cost partitioning heuristic *h^{TCP} is $h^{\mathrm{TCP}}(s) = \max_{P \in \mathcal{P}} h_P(s)$.*

Operator cost partitionings and the OCP heuristic are special cases where all cost functions satisfy $cost(t_1) = cost(t_2)$ for all pairs of transitions t_1, t_2 labeled with the same operator. The TCP heuristic thus dominates the OCP heuristic. The paper by Keller et al. contains examples where this dominance is strict.

The linear program defined by Katz and Domshlak (2010) to compute an optimal OCP for a set of abstraction heuristics can be extended to transition cost partitionings.

Formally, an abstraction heuristic is based on a homomorphic mapping $\alpha : \mathcal{S} \to \mathcal{S}^\alpha$ that maps states of a planning task Π to *abstract states* of an *abstract transition system* $\mathrm{TS}^\alpha = \langle \mathcal{S}^\alpha, \mathcal{T}^\alpha, \alpha(s_{\mathrm{I}}), \{\alpha(s_\star)\} \rangle$, such that for every transition $s \xrightarrow{o} s'$ of Π, $\mathcal{T}^\alpha$ contains the transition $\alpha(s) \xrightarrow{o} \alpha(s')$ with the same weight. For a collection $\mathcal{A}$ of abstractions, the optimal TCP can be encoded using two kinds of variables. The variable $h(s)$ represents the goal distance for each abstract state $s \in \mathcal{S}^\alpha$ of each $\alpha \in \mathcal{A}$ (we assume the states are uniquely named). The variable $c^\alpha(t)$ represents the cost of a transition $t \in \mathcal{T}$ that is attributed to abstraction α. Using linear constraints over these variables, we can express that the variables $h(s)$ do not exceed the true goal distance under the cost function encoded in c^α and that the cost functions respect the cost partitioning property:

$$h(\alpha(s_\star)) = 0 \qquad \text{for } \alpha \in \mathcal{A} \tag{14}$$

$$h(\alpha(s)) - h(\alpha(s')) \leq c^\alpha(s \xrightarrow{o} s') \qquad \text{for } \alpha \in \mathcal{A} \text{ and } s \xrightarrow{o} s' \in \mathcal{T} \tag{15}$$

$$\sum_{\alpha \in \mathcal{A}} c^\alpha(s \xrightarrow{o} s') \leq cost(o) \qquad \text{for } s \xrightarrow{o} s' \in \mathcal{T} \tag{16}$$

For a set of abstractions $\mathcal{A}$, we define the set $\mathcal{F}_\mathcal{A} = \bigcup_{\alpha \in \mathcal{A}} \mathcal{S}^\alpha$ of features for $\mathcal{A}$ with the interpretation that a state s has the feature $s' \in \mathcal{S}^\alpha$ iff $\alpha(s) = s'$. In the special case where α is a projection to k variables, the features $\mathcal{S}^\alpha$ correspond to conjunctions of k facts. We want to show that TCP heuristics over $\mathcal{A}$ and admissible and consistent potential heuristics over features $\mathcal{F}_\mathcal{A}$ have the same maximal heuristic value.

Proposition 2. *Let s be a state of a planning task Π and $\mathcal{A}$ be a set of abstractions of Π. The set of solutions for constraints (14)–(16) that maximize $\sum_{\alpha \in \mathcal{A}} h(\alpha(s))$ (projected to h) is equal to the set of solutions for constraints (1)–(2) for $\mathcal{F}_\mathcal{A}$ that maximize $\varphi(s)$.*

Proof: The important observation for this proof is that we can write $\varphi(s) = \sum_{f \in \mathcal{F}_\mathcal{A}} w(f)[s \models f]$ as $\varphi(s) = \sum_{\alpha \in \mathcal{A}} w(\alpha(s))$.

Assume we have an optimal solution to constraints (14)–(16) and set $w = h$. Obviously, constraint (14) implies constraint (1): if all features that are present in the goal state have a weight of 0, then $\varphi(s_\star) = 0$. Summing constraint (15) over all abstractions for a given transition, results in $\sum_{\alpha \in \mathcal{A}} h(\alpha(s)) - h(\alpha(s')) \leq \sum_{\alpha \in \mathcal{A}} c^\alpha(s \xrightarrow{o} s')$. Together with constraint (16) this implies constraint (2).

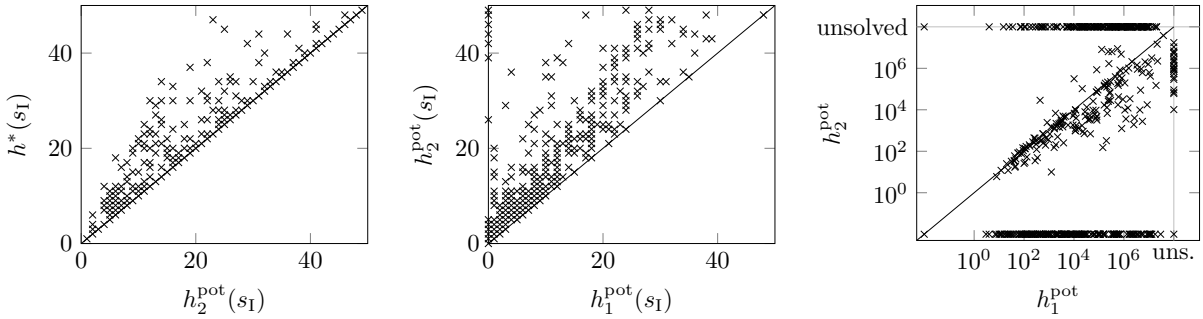

Figure 1: Initial heuristic values below 50 of h_1^{pot}, h_2^{pot}, and h^*. The final plot shows the number of expansions in an A* search with h_1^{pot} and h_2^{pot} (without expansions in the last f-layer).

For the other direction, assume we have an optimal solution to constraints (1)–(2), i. e., that φ is an admissible and consistent potential heuristic with weight function w. Set $h = w$ and $c^\alpha(s \xrightarrow{o} s') = w(\alpha(s)) - w(\alpha(s'))$.

We also assume that $w(\alpha(s_\star)) = 0$ for all $\alpha \in \mathcal{A}$. If this is not the case, consider the weight function w' with $w'(\alpha(s)) = w(\alpha(s)) - w(\alpha(s_\star))$ for all $\alpha \in \mathcal{A}$ and $s \in \mathcal{S}$. Let φ' be the induced potential function with $\varphi'(s) = \sum_{\alpha\in\mathcal{A}} w'(\alpha(s)) = \varphi(s) - \varphi(s_\star)$ We know from constraint (1) that $\varphi(s_\star) \leq 0$, so φ' dominates φ. It also still satisfies constraints (1) and (2), so φ' must be an optimal solution that we can use instead of φ. Thus, constraint (14) is satisfied.

Constraint (15) is trivially satisfied. Constraint (16) is also satisfied, which can be seen by replacing $c^\alpha(s \xrightarrow{o} s')$ by its definition: the inequality $\sum_{\alpha\in\mathcal{A}} w(\alpha(s)) - w(\alpha(s')) = \varphi(s) - \varphi(s') \leq \mathit{cost}(o)$ is identical to constraint (2). □

Theorem 7. *Let Π be a planning task and $\mathcal{A}$ a set of abstractions for Π. For every state s of Π, let $h^{\mathrm{pot}}_{\mathcal{F}_\mathcal{A},\max s}$ be an admissible and consistent potential function with maximal value for s. Then $h^{\mathrm{pot}}_{\mathcal{F}_\mathcal{A},\max s}(s) = h^{\mathrm{TCP}}_{\mathcal{A}}(s)$.*

Evaluation

We implemented one- and two-dimensional admissible potential heuristics (called h_1^{pot} and h_2^{pot} in the following) in the Fast Downward planning system (Helmert 2006) and evaluated them on the tasks from the optimal tracks of IPC 1998–2014 using limits of 2 GB and 24 hours for memory and run time. We use this fairly high time limit, as we are focusing on heuristic quality, not evaluation time. The linear programs for h_2^{pot} can become quite big, but we are confident that approximations with smaller representations exist.

Our first concern is heuristic accuracy in the initial state. In 437 out of 696 cases (63%) where we could determine both $h_2^{\mathrm{pot}}(s_\mathrm{I})$ and $h^*(s_\mathrm{I})$, the h_2^{pot} value is perfect. These perfect values occur in 42 of the 51 domains. The first plot in Figure 1 shows initial heuristic values for $h_2^{\mathrm{pot}}(s_\mathrm{I})$ vs. h^*. There are still some heuristic values that are far from the optimum, suggesting that using features of size three or larger might be necessary in some domains.

The second plot in Figure 1 compares initial heuristic values of h_1^{pot} and h_2^{pot}. As expected, larger features frequently achieve higher heuristic values and the initial heuristic value is perfect less frequently. Out of the 696 tasks mentioned above, only 110 (16%) have a perfect h_1^{pot} value. There are also several cases in which $h_1^{\mathrm{pot}}(s_\mathrm{I})$ is 0, while h_2^{pot} reports values as high as 48. The last plot shows the number of expansions in an A* search. Using h_2^{pot} almost always leads to fewer expansions than h_1^{pot}. The exceptions are mostly from the domain Blocksworld. The tasks on the x-axis of the plot are those where h_2^{pot} is perfect.

Theorem 7 shows that h_2^{pot} computes a TCP that is optimal in the initial state. We know that there are tasks where the optimal TCP results in a higher heuristic value than the optimal OCP. But does this also occur often in practice? To answer this question, we implemented optimal cost partitioning for projections and ran it for the set of abstractions that contains all projections to one and two variables; the resulting heuristic is denoted by h_2^{OCP}. Out of the 724 cases where we can compute both $h_2^{\mathrm{pot}}(s_\mathrm{I})$ and $h_2^{\mathrm{OCP}}(s_\mathrm{I})$, $h_2^{\mathrm{pot}}(s_\mathrm{I})$ is higher than $h_2^{\mathrm{OCP}}(s_\mathrm{I})$ in 312 cases (43%). In 329 of the remaining cases (46%) $h_2^{\mathrm{OCP}}(s_\mathrm{I})$ is already perfect, so h_2^{pot} cannot be higher. The advantage of TCP over OCP thus seems to be ubiquitous in practice.

Comparing an A* search using h_2^{OCP} to one using h_2^{pot} we notice two conflicting effects. First, as we have seen, the potential heuristic is perfect for more instances and even if it is not, its value often exceeds the value of h_2^{OCP}. Second, the linear program of h_2^{pot} is evaluated only once, while the optimal OCP is re-computed in every state. The latter is beneficial in some cases (h_2^{pot} is guaranteed to be no worse than h_2^{OCP} only at the initial state), but this re-computation takes too much time in practice. Diversification approaches (Seipp et al. 2016) can be used to boost the value of potential heuristics on states other than the initial state.

Conclusion

We have shown that highly accurate potential heuristics of dimension 2 can be computed in polynomial time. Optimized in each state, these heuristics correspond to an optimal TCP over atomic and binary projections, which dominates the optimal OCP. Generating admissible potential

heuristics of higher dimension is not tractable in general, but possible if the variables are not too interdependent.

To make higher-dimensional potential heuristics more practically relevant, methods to cheaply approximate the optimal TCP are needed. One promising direction is to look into a way of identifying a good subset of features, because usually not all features are necessary to maximize the heuristic value. Features could be statically selected, or learned dynamically during the search. Using diversification techniques to select a set of multiple admissible potential heuristics also sounds promising.

Acknowledgments

This work was supported by the European Research Council as part of the project "State Space Exploration: Principles, Algorithms and Applications".

References

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

Bodlaender, H. L. 1996. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing* 25(6):1305–1317.

Bonet, B., and van den Briel, M. 2014. Flow-based heuristics for optimal planning: Landmarks and merges. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS 2014)*, 47–55. AAAI Press.

Bonet, B. 2013. An admissible heuristic for SAS^+ planning obtained from the state equation. In Rossi, F., ed., *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, 2268–2274.

Chen, H., and Giménez, O. 2007. Act local, think global: Width notions for tractable planning. In Boddy, M.; Fox, M.; and Thiébaux, S., eds., *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling (ICAPS 2007)*, 73–80. AAAI Press.

Chen, H., and Giménez, O. 2009. On-the-fly macros. In *Logic, Language, Information and Computation*, volume 5514 of *Lecture Notes in Computer Science*, 155–169. Springer-Verlag.

Dechter, R. 2003. *Constraint Processing*. Morgan Kaufmann.

Downey, R. G., and Fellows, M. R. 1999. *Parameterized Complexity*. Springer.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. Freeman.

Helmert, M. 2006. The Fast Downward planning system. *Journal of Artificial Intelligence Research* 26:191–246.

Katz, M., and Domshlak, C. 2007. Structural patterns heuristics: Basic idea and concrete instance. In *ICAPS 2007 Workshop on Heuristics for Domain-Independent Planning: Progress, Ideas, Limitations, Challenges*.

Katz, M., and Domshlak, C. 2010. Optimal admissible composition of abstraction heuristics. *Artificial Intelligence* 174(12–13):767–798.

Keller, T.; Pommerening, F.; Seipp, J.; Geißer, F.; and Mattmüller, R. 2016. State-dependent cost partitionings for cartesian abstractions in classical planning. In Kambhampati, S., ed., *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 3161–3169.

Knuth, D. E. 1992. Two notes on notation. *American Mathematical Monthly* 99(5):403–422.

Lipovetzky, N., and Geffner, H. 2012. Width and serialization of classical planning problems. In De Raedt, L.; Bessiere, C.; Dubois, D.; Doherty, P.; Frasconi, P.; Heintz, F.; and Lucas, P., eds., *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, 540–545. IOS Press.

Pommerening, F., and Helmert, M. 2015. A normal form for classical planning tasks. In Brafman, R.; Domshlak, C.; Haslum, P.; and Zilberstein, S., eds., *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS 2015)*, 188–192. AAAI Press.

Pommerening, F.; Helmert, M.; Röger, G.; and Seipp, J. 2015. From non-negative to general operator cost partitioning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, 3335–3341. AAAI Press.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence — A Modern Approach*. Prentice Hall.

Seipp, J.; Pommerening, F.; Röger, G.; and Helmert, M. 2016. Correlation complexity of classical planning domains. In Kambhampati, S., ed., *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 3242–3250.

Seipp, J.; Pommerening, F.; and Helmert, M. 2015. New optimization functions for potential heuristics. In Brafman, R.; Domshlak, C.; Haslum, P.; and Zilberstein, S., eds., *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS 2015)*, 193–201. AAAI Press.

van den Briel, M.; Benton, J.; Kambhampati, S.; and Vossen, T. 2007. An LP-based heuristic for optimal planning. In Bessiere, C., ed., *Proceedings of the Thirteenth International Conference on Principles and Practice of Constraint Programming (CP 2007)*, volume 4741 of *Lecture Notes in Computer Science*, 651–665. Springer-Verlag.

Yang, F.; Culberson, J.; Holte, R.; Zahavi, U.; and Felner, A. 2008. A general theory of additive state space abstractions. *Journal of Artificial Intelligence Research* 32:631–662.