Bridging the Gap between Abstractions and Critical-Path Heuristics via Hypergraphs

Marcel Steinmetz
CISPA Helmholtz Center for Information Security
Saarland Informatics Campus
Saarbrücken, Germany
steinmetz@cispa.saarland

Álvaro Torralba
Saarland University
Saarland Informatics Campus
Saarbrücken, Germany
torralba@cs.uni-saarland.de

Abstract

Abstractions and critical-path heuristics are among the most important families of admissible heuristics in classical planning. In this paper, we present a new family of heuristics, which we name *hyperabstractions*, given by the combination of the principal ideas underlying abstractions and critical-path heuristics. Hyperabstractions approximate goal distances through a mapping from states to *sets* of abstract states. The abstract transition behavior forms a relation between abstract states and sets of abstract states, and is formally represented via the notion of hypergraphs. We show that both abstractions and critical-path heuristics can naturally be expressed as members of this family. Moreover, we devise a method to construct hyperabstractions, using either a set of abstractions or a critical-path heuristic as a seed, in a way that guarantees that the resulting distance estimations dominate those of the input heuristics, sometimes even strictly. By finding suitable cost partitionings for hyperabstraction heuristics, this dominance result is preserved even in comparison to the additive combination of the input heuristics. Our experiments indicate the potential of this new class of heuristics, opening a wide range of future research topics.

Introduction

Heuristic search is a very popular method in classical planning. There is a huge body of research on how to automatically obtain good heuristic functions. Relating different families of heuristics is important to deepen the understanding of existing and to derive new heuristics. Helmert and Domshlak (2009) studied the relation between abstraction (Edelkamp 2001; 2006; Helmert et al. 2014), critical-path (Haslum and Geffner 2000), delete-relaxation and landmark heuristics (Bonet and Geffner 2001; Hoffmann, Porteous, and Sebastia 2004), showing that some of them are closely connected. However, the exact connection between general abstraction and critical-path heuristics is yet unclear.

We devise a new type of heuristics, *hyperabstractions*, that naturally generalizes abstraction and critical-path heuristics. Hyperabstractions are functions that map states to sets of *abstract concepts*. Heuristic values are computed as distances between sets of concepts in hypergraphs, concrete *interpretations* of hyperabstractions, with particular properties. Each hyperabstraction comes with multiple interpretations, differing in size and quality of the distance estimates. We show that every critical-path heuristic can be compiled into a dominating hyperabstraction heuristic in polynomial time. Moreover, for every set of abstraction heuristics, there exist polynomially constructable hyperabstraction heuristics dominating their combination via the maximum. Both results directly carry over to additive ensembles via the application of *cost partitionings*. While there exist polynomial algorithms computing the optimal cost partitioning for sets of abstractions (Katz and Domshlak 2008a), the complexity of optimal cost partitioning for critical-path heuristics is not known. In this paper, we show that finding the optimal cost-partitioning for critical-path heuristics is **NP**-hard. This result generalizes directly to hyperabstraction heuristics.

The experiments confirm the potential improvements in informativeness compared to abstraction and critical-path heuristics. Nevertheless, better construction methods for hyperabstraction heuristics remain an important question. All proofs are available in a TR (Steinmetz and Torralba 2019).

Preliminaries

We introduce the planning formalisms, the heuristic functions considered in this paper, as well as hypergraphs.

FDR Planning

A *FDR* planning task (Bäckström 1995) is a tuple $\Pi = \langle \mathcal{V}, \mathcal{A}, s_{\mathcal{I}}, \mathcal{G} \rangle$ of *variables* $\mathcal{V}$, each $v \in \mathcal{V}$ with finite domain $\mathcal{D}_v$; *actions* $\mathcal{A}$; a complete variable assignment $s_{\mathcal{I}}$, called the *initial state*; and a partial variable assignment $\mathcal{G}$, the *goal*. Each *action* $a \in \mathcal{A}$ defines a *precondition* pre_a and an *effect* eff_a, both partial variable assignments to $\mathcal{V}$, and a non-negative cost $c_a \in \mathbb{R}_0^+$. A *fact* is a variable value pair $\langle v, d \rangle$ where $v \in \mathcal{V}$ and $d \in \mathcal{D}_v$. We use conjunctions of facts and partial variable assignments interchangeably. Both are treated as sets of facts, using usual set operations for manipulation and comparison. Two conjunctions of facts C, C' are *compatible*, written $C \parallel C'$, if $C(v) = C'(v)$ for every $v \in \mathcal{V}$ where both are defined.

We denote by $\mathcal{S}$ the set of all states of Π. An action a is applicable in state s if $pre_a \parallel s$. $\mathcal{A}(s)$ gives the set of all actions applicable in s. For $a \in \mathcal{A}(s)$, the result of applying a in s, written $s[\![a]\!]$, is given by the variable assignments in s, overwritten by those in eff_a. The *state space* of Π is given by the labeled transition system $\Theta^{\Pi} = \langle \mathcal{S}, \mathcal{T}, \mathcal{L}, s_{\mathcal{I}}, \mathcal{S}_{\mathcal{G}} \rangle$ where the set of *transitions* contains $\langle s, a, s[\![a]\!] \rangle \in \mathcal{T}$ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$; *transition labels* $\mathcal{L} = \mathcal{A}$; and $\mathcal{S}_{\mathcal{G}}$ gives all *goal states*, i. e., $s_{\mathcal{G}} \in \mathcal{S}_{\mathcal{G}}$ if $\mathcal{G} \parallel s_{\mathcal{G}}$. A *plan* for s is a sequence of actions $\pi = \langle a_1, \dots, a_n \rangle$ that labels a path from s to some $s_{\mathcal{G}} \in \mathcal{S}_{\mathcal{G}}$. π is *optimal* if its summed-up action cost is minimal among all plans for s.

Heuristics

A *heuristic (function)* is a function $h\colon \mathcal{S} \mapsto \mathbb{R}_0^+ \cup \{\infty\}$ that maps states to an approximation of the cost-to-go to reach a goal state. The *perfect heuristic* h^* assigns each state s to the cost of an optimal plan for s, $h^*(s) = \infty$ if no plan for s exists. A heuristic h is *admissible* if $h(s) \leq h^*(s)$ for every $s \in \mathcal{S}$. h is *consistent* if $h(s) - h(t) \leq c(a)$ for every $\langle s, a, t \rangle \in \mathcal{T}$. h is *goal-aware* if $h(s_{\mathcal{G}}) = 0$ for all $s_{\mathcal{G}} \in \mathcal{S}_{\mathcal{G}}$. Every goal-aware and consistent heuristic is admissible. A heuristic h *dominates* another h', written $h \geq h'$, if $h(s) \geq h'(s)$ for all $s \in \mathcal{S}$. h *strictly dominates* h', written $h > h'$, if $h \geq h'$, and $h(s) > h'(s)$ for at least one state $s \in \mathcal{S}$.

In this paper, we will consider two families of admissible heuristics: *abstraction*, and *critical-path heuristics*.

Abstractions Abstraction heuristics compute the goal-distance in an *abstract state space*, dropping the distinction between some of the original states so to make the computation of h^* feasible. Formally, an *abstraction* is a surjective function $\alpha\colon \mathcal{S} \mapsto \mathcal{S}^{\alpha}$ of states to *abstract states*. α implicitly introduces an equivalence relation between states, where $s \sim_{\alpha} t$ iff $\alpha(s) = \alpha(t)$. For an abstract state $s^{\alpha} \in \mathcal{S}^{\alpha}$, we denote by $[s^{\alpha}]$ all states $s \in \mathcal{S}$ such that $\alpha(s) = s^{\alpha}$.

The *abstract state space* of Θ^{Π} induced by α is given by the labeled transition system $\Theta^{\alpha} = \langle \mathcal{S}^{\alpha}, \mathcal{T}^{\alpha}, \mathcal{A}, s^{\alpha}_{\mathcal{I}}, \mathcal{S}^{\alpha}_{\mathcal{G}} \rangle$ where $s^{\alpha}_{\mathcal{I}} = \alpha(s_{\mathcal{I}})$, $\mathcal{S}_{\mathcal{G}} = \{\alpha(s_{\mathcal{G}}) \mid s_{\mathcal{G}} \in \mathcal{S}_{\mathcal{G}}\}$, and $\mathcal{T}^{\alpha} = \{\langle \alpha(s), a, \alpha(t) \rangle \mid \langle s, a, t \rangle \in \mathcal{T}\}$. We denote by $h^*_{\Theta^{\alpha}}$ the function assigning every abstract state s^{α} in Θ^{α} to the minimal cost to reach any state in $\mathcal{S}^{\alpha}_{\mathcal{G}}$ from s^{α}. The *abstraction heuristic* associated with α is given by $h^{\alpha}(s) := h^*_{\Theta^{\alpha}}(\alpha(s))$.

Various techniques have been proposed to automatically construct the abstraction function α. Different approaches aim to be as accurate as possible while still practical to compute, e. g., symbolic representations of Θ^{α} (Edelkamp 2002; Torralba et al. 2017), or designing α so to compute $h^*_{\Theta^{\alpha}}$ without actually constructing Θ^{α} (Katz and Domshlak 2008b). In this paper, however, we consider only abstractions where the abstraction mapping α and its abstract state space can be computed and stored in an *explicit* form. Examples are pattern databases (PDBs) (Edelkamp 2001; Haslum et al. 2007), merge-and-shrink (MS) (Helmert et al. 2014), and Cartesian abstractions (Seipp and Helmert 2018).

Critical-Path Heuristics Critical-path heuristics estimate goal distance by breaking reasoning down to *atomic conjunctions*. Which conjunctions are treated as atomic, and thus how accurate the estimations are going to be, is controlled via the parameter $\mathcal{C}$. Let $\mathcal{C}$ be any set of conjunctions of facts. The *regression* of a conjunction C by an action a is defined if $C \parallel \mathit{eff}_a$ and $(C \setminus \mathit{eff}_a) \parallel \mathit{pre}_a$. If defined, the regression is given by $\mathit{Regr}(C, a) = (C \setminus \mathit{eff}_a) \cup \mathit{pre}_a$. Otherwise, we write $\mathit{Regr}(C, a) = \bot$. The *critical-path heuristic* over $\mathcal{C}$ is defined as $h^{\mathcal{C}}(s) = h^{\mathcal{C}}(s, \mathcal{G})$ where $h^{\mathcal{C}}(s, C) =$

$$\begin{cases} 0 & C \subseteq s \\ \min\limits_{a \in \mathcal{A}, \mathit{Regr}(C,a) \neq \bot} (c_a + h^{\mathcal{C}}(s, \mathit{Regr}(C, a))) & C \in \mathcal{C} \\ \max\limits_{C' \in \mathcal{C}, C' \subseteq C} h^{\mathcal{C}}(s, C') & \text{otherwise} \end{cases} \tag{1}$$

The most common method to choose $\mathcal{C}$ is enumerating all conjunctions of size up to m, where $m \in \mathbb{N}^+$ is a parameter. The resulting heuristic is denoted h^m (Haslum and Geffner 2000; Haslum 2009). Recent works used the flexibility of critical-path heuristics to refine the heuristic online, during search, by incrementally adding conjunctions to $\mathcal{C}$ (Steinmetz and Hoffmann 2017; Fickert and Hoffmann 2017).

Cost Partitioning Multiple admissible heuristics can trivially be combined in a way that preserves admissibility by taking the maximum. Summing up the individual estimations dominates the maximum, but is admissible only under particular conditions. A popular method satisfying such condition by construction is *cost partitioning* (Katz and Domshlak 2008a). Cost partitioning does not only allow the admissible combination of multiple heuristics, but can be used also to improve a single heuristic. A prominent example for the latter is the LM-cut heuristic (Helmert and Domshlak 2009).

Formally, let $c'\colon \mathcal{A} \mapsto \mathbb{R}_0^+$ be any cost function. We denote by $h[\![c']\!]$ the heuristic function h computed in the copy of Π whose cost function is replaced by c'. A *cost partitioning* is a tuple of cost functions $\mathbf{c} = \langle c_1, \dots, c_n \rangle$ such that for every action $a \in \mathcal{A}$, it holds that $\sum_{i=1}^{n} c_i(a) \leq c(a)$. Using cost partitionings, an ensemble of heuristics $h_1, \dots, h_n$ can be additively combined through $h_{1,\dots,n}[\![\mathbf{c}]\!] := \sum_{i=1}^{n} h_i[\![c_i]\!]$. $h_{1,\dots,n}[\![\mathbf{c}]\!]$ is admissible if all heuristics $h_1, \dots, h_n$ are admissible (Katz and Domshlak 2008a). A cost partitioning is applied to a single heuristic h by setting $h_i = h$ for all i. We denote the additive combination by $h[\![\mathbf{c}]\!]$ in that case.

Clearly, the estimates of $h_{1,\dots,n}[\![\mathbf{c}]\!]$ depend crucially on the distribution of the action costs over the individual heuristics. We say that a cost partitioning $\mathbf{c} = \langle c_1, \dots, c_n \rangle$ is *optimal* for $h_1, \dots, h_n$ in a state s, if $h_{1,\dots,n}[\![\mathbf{c}]\!](s) \geq h_{1,\dots,n}[\![\mathbf{c}']\!](s)$ for all other cost partitionings $\mathbf{c}'$ of size n. For a single heuristic h, the size of $\mathbf{c}$ is not fixed a priori. Hence, in the definition of optimality for single heuristics, the size restriction is dropped. For an ensemble of abstraction heuristics, an optimal cost partitioning can be found in polynomial time (Katz and Domshlak 2008a). For critical-path heuristics it was an open question whether it is possible to compute an optimal cost partitioning in polynomial time.

Hypergraphs

A *labeled weighted directed hypergraph* (Gallo, Longo, and Pallottino 1993) is given by a triple $\mathcal{H} = \langle \mathcal{N}, \mathcal{E}, \mathcal{L}, w \rangle$ consisting of a finite set of *nodes* $\mathcal{N}$, a finite set of *labels* $\mathcal{L}$, each $l \in \mathcal{L}$ associated with a *weight* $w(l) \in \mathbb{R}_0^+$, and a finite set of *hyperedges* $\mathcal{E} \subseteq 2^{\mathcal{N}} \times \mathcal{L} \times 2^{\mathcal{N}}$. A hyperedge e is a tuple $\langle T_e, l_e, H_e \rangle$ where T_e is the *tail* of e, H_e is the *head* e, and l_e is the *label*. e is called a *backward (*B*) hyperedge* if $|H_e| \leq 1$. e is a *strict* B*-hyperedge* if $|H_e| = 1$. e is a *(strict) forward (*F*) hyperedge* if $|T_e| \leq 1$ ($|T_e| = 1$). $\mathcal{H}$ is a *(strict)* B*-hypergraph* if it contains only (strict) B-hyperedges, and similarly, $\mathcal{H}$ is an *(strict)* F*-hypergraph* if all its edges are (strict) F-hyperedges. Hypergraphs satisfying the B- and F-conditions at the same time are called BF*-hypergraphs*. The *symmetric image* of $\mathcal{H}$ is given by the hypergraph $\widehat{\mathcal{H}} = \langle \widehat{\mathcal{N}}, \widehat{\mathcal{E}}, \widehat{\mathcal{L}} \rangle$ where $\widehat{\mathcal{N}} = \mathcal{N}$, $\widehat{\mathcal{L}} = \mathcal{L}$, and, for every $e \in \mathcal{E}$, $\widehat{\mathcal{E}}$ contains $\widehat{e} = \langle H_e, l_e, T_e \rangle$. Note that $\widehat{\mathcal{H}}$ is an F-hypergraph iff $\mathcal{H}$ is a B-hypergraph, and a B-hypergraph iff $\mathcal{H}$ is an F-hypergraph.

For the rest of the paper, we will exclusively consider B- and F-hypergraphs. Let $\mathcal{H}$ be a B-hypergraph, and $N, N' \subseteq \mathcal{N}$ be two subsets of nodes. The *minimal distance*, *distance* for short, from N to N' in $\mathcal{H}$ is given by the point-wise maximal function satisfying[1] $d^{\mathrm{B}}_{\mathcal{H}}(N, N') =$

$$\begin{cases} 0 & \text{if } N' \subseteq N \\ \min_{e \in \mathcal{E}, n' \in H_e} (w(l_e) + d^{\mathrm{B}}_{\mathcal{H}}(N, T_e)) & \text{if } N' = \{n'\} \\ \max_{n' \in N'} d^{\mathrm{B}}_{\mathcal{H}}(N, \{n'\}) & \text{otherwise} \end{cases} \quad (2)$$

Note the similarities between Equations (1) and (2). Both are indeed closely related. We will come back to this comparison when we later show how critical-path heuristics can be phrased in terms of this hypergraph notation.

In F-hypergraphs the distance $d^{\mathrm{F}}_{\mathcal{H}}$ is defined symmetrically. Using the symmetric image operator, it holds that $d^{\mathrm{F}}_{\mathcal{H}}(N, N') = d^{\mathrm{B}}_{\widehat{\mathcal{H}}}(N', N)$. We will omit the F and B superscript if the type is clear, and omit the sub- and superscripts all together if also $\mathcal{H}$ is clear from the context. Gallo et al. (1993) have shown a polynomial (in $|\mathcal{H}|$) algorithm to compute minimal distances in hypergraphs.

Hyperabstractions

In this section we introduce *hyperabstractions* and show how admissible heuristics can be derived from this notion. Consider any planning task $\Pi = \langle \mathcal{V}, \mathcal{A}, s_{\mathcal{I}}, \mathcal{G} \rangle$, and let $\Theta^{\Pi} = \langle \mathcal{S}, \mathcal{T}, \mathcal{A}, s_{\mathcal{I}}, \mathcal{S}_{\mathcal{G}} \rangle$ be its associated state space.

Definition 1 (Hyperabstraction) *A* hyperabstraction *is a function $\rho \colon \mathcal{S} \mapsto 2^{\mathcal{P}}$ mapping states to* abstract concepts*. $\mathcal{P}$ is the set of abstract concepts associated with ρ.*

Abstract concepts describe features of particular states. Propositional formulae over facts are natural variants of such features, but more complex structures such as for example those underlying abstract states qualify as well. Similarly to abstractions, for any $p \in \mathcal{P}$, we denote by $[p]$ the set of all states $s \in \mathcal{S}$ such that $p \in \rho(s)$. Note, however, that in contrast to abstract states, the same state may be represented by multiple abstract concepts. We chose the term *abstract concept* instead of *abstract state* to make this difference explicit.

[1]We assume that $\min(\emptyset) = \infty$ and $\max(\emptyset) = 0$.

Each state being possibly associated with multiple concepts, abstract transitions should no longer connect different concepts individually, but form a relation between *sets* of them. In principle, hypergraphs provide the possibility to define complex transition behavior between arbitrary sets of concepts. However, the distance metric for such general hypergraphs gets difficult to define, or expensive to compute (see e. g. Ausiello and Laura 2017), and are thus not particularly suited for our purpose: heuristic computation. In the following, we thus consider specifically the F- and B-hypergraph special cases. We propose two different *interpretations* of hyperabstractions accordingly. Let $\mathcal{H} = \langle \mathcal{N}, \mathcal{E}, \mathcal{L}, w \rangle$ by a hypergraph with nodes $\mathcal{N} = \mathcal{P}$, labels $\mathcal{L} = \mathcal{A}$, and weights $w = c$. We say that:

Definition 2 (F**-interpretation)** *$\mathcal{H}$ is an* F-interpretation *of ρ if $\mathcal{H}$ is an* F*-hypergraph, and for every transition $\langle s, a, t \rangle \in \mathcal{T}$ and for every $p_s \in \rho(s)$, $\mathcal{H}$ has a hyperedge $\langle \{p_s\}, a, P_t \rangle \in \mathcal{E}$ such that $P_t \subseteq \rho(t)$.*

Definition 3 (B**-interpretation)** *$\mathcal{H}$ is a* B-interpretation *of ρ if $\mathcal{H}$ is a* B*-hypergraph, and for every transition $\langle s, a, t \rangle \in \mathcal{T}$ and for every $p_t \in \rho(t)$, $\mathcal{H}$ has a hyperedge $\langle P_s, a, \{p_t\} \rangle \in \mathcal{E}$ such that $P_s \subseteq \rho(s)$.*

To ensure that distances obtained from hyperabstractions are admissible, all transitions in the original state space must be preserved by the abstract transition relation of the interpretations. Both definitions, however, leave open the exact choice of the sets P_t, respectively P_s, to do so for any particular transition. Defining *the*, unique, (F or B) interpretation associated with a hyperabstraction raises certain complications, as we will see below. The restriction of hyperedges to tails of size ≤ 1 (F-interpretation) and heads of size ≤ 1 (B-interpretation) leads to a very fundamental difference in how Θ^{Π}'s transitions are reflected in the hypergraphs. Given an abstract concept $p \in \mathcal{P}$ and action $a \in \mathcal{A}$, the hyperedges of F-interpretations enumerate consequences of the application of a in the context of p, i. e., possible effects of applying a in the states $[p]$. In contrast, the hyperedges of B-interpretations enumerate different conditions under which the application of a makes true p, i. e., results in a state in $[p]$. Phrased in common search terms, F-interpretations provide information in terms of *progression*, whereas B-interpretations provide information on the *regression*.

Heuristic functions associated with a hyperabstraction ρ are obtained directly from the distance metrics of the F- and B-interpretations of ρ. To make apparent the symmetric nature of F- and B-hyperabstraction heuristics, we define both heuristics as functions $h_\rho : (2^{\mathcal{S}} \times 2^{\mathcal{S}}) \mapsto \mathbb{R}_0^+ \cup \{\infty\}$, $h_\rho(S, T)$ yielding an approximation of the minimal cost of all paths in Θ^{Π} between any state $s \in S$ and any $t \in T$. Abusing the notation, we use the same heuristic symbol to denote $h_\rho(s) := h_\rho(\{s\}, \mathcal{S}_{\mathcal{G}})$. Let $\mathcal{H}^{\mathrm{F}}$ be any F-interpretation of ρ, and $\mathcal{H}^{\mathrm{B}}$ be any B-interpretation $\mathcal{H}^{\mathrm{B}}$ of ρ. We define:

Definition 4 (F**-hyperabstraction heuristic)** *The* F-hyperabstraction heuristic *associated with* ρ *and* $\mathcal{H}^{\mathrm{F}}$ *is given by* $h^{\mathrm{F}}_{\rho}[\mathcal{H}^{\mathrm{F}}](S,T) = d^{\mathrm{F}}_{\mathcal{H}^{\mathrm{F}}}(\bigcap_{s\in S}\rho(s), \bigcup_{t\in T}\rho(t))$.

Definition 5 (B**-hyperabstraction heuristic)** *The* B-hyperabstraction heuristic *associated with* ρ *and* $\mathcal{H}^{\mathrm{B}}$ *is given by* $h^{\mathrm{B}}_{\rho}[\mathcal{H}^{\mathrm{B}}](S,T) = d^{\mathrm{B}}_{\mathcal{H}^{\mathrm{B}}}(\bigcup_{s\in S}\rho(s), \bigcap_{t\in T}\rho(t))$.

We omit the hypergraph parameter if it is clear from the context, or unimportant for the discussion. The difference in F- versus B-hypergraph distance function requires to swap the set operators in the definitions. In the forward case, for any two sets $P_S, P_T \subseteq \mathcal{P}$, $d^{\mathrm{F}}(P_S, P_T)$ measures the maximal distance $d^{\mathrm{F}}(\{p_s\}, P_T)$ of any $p_s \in P_S$. Because of this maximization, to guarantee that $h^{\mathrm{F}}_{\rho}(S,T)$ admissibly estimates of the distance from any $s \in S$ to the states in T, P_S may only contain abstract concepts representing all states in S. On the other hand, P_T must contain every abstract concept representing some state in T so that $d^{\mathrm{F}}(\{p_s\}, P_T)$ admissibly approximates the cost to reach from $[p_s]$ any state in T. Symmetrically, $d^{\mathrm{B}}(P_S, P_T)$ is the maximum over $d^{\mathrm{B}}(P_S, \{p_t\})$ for all $p_t \in P_T$. To ensure the admissibility of h^{B}_{ρ}, P_T may hence only contain abstract concepts that represent all states in T, while P_S must contain every abstract concept representing any state in S.

Moreover, observe that, for any fixed set $P_T \subseteq \mathcal{P}$, $d^{\mathrm{F}}(\{p\}, P_T)$ can be precomputed individually for every abstract concept p. Since the heuristic computations $h^{\mathrm{F}}_{\rho}(s)$ will generate $d^{\mathrm{F}}(P_S, P_T)$ calls, changing only the P_S part, $h^{\mathrm{F}}_{\rho}(s)$ can be computed just based on lookups of the precomputed distances. This optimization is not possible for h^{B}_{ρ}, since the distance estimate $d^{\mathrm{B}}(P_S, P_T)$ requires the consideration of P_S as a whole, which however varies from state to state.

Before going into formal claims, consider the following example to get an intuitive understanding of the two hyperabstraction variants.

Example 1 ("Robot in a china shop") *Consider the following task. There are three variables: whether the robot R has entered the shop (T) or not (F), and the state of two vases V_1 and V_2 (clean C, held by the robot R, broken B). The (unit-cost) actions are:* enter *changing R from F to T,* $\mathrm{pickup}(V_i)$ *with precondition $\{V_i = C, R = T\}$ and effect $\{V_i = R\}$,* $\mathrm{drop}(V_i)$ *requiring that $\{V_i = R\}$ and setting $\{V_i = B, R = F\}$ in its effect, and* $\mathrm{smash}(V_i, V_j)$ *with precondition $\{R = T, V_i = R, V_j = C\}$ and effect $\{V_j = B\}$. The initial state is $s_{\mathcal{I}} = \{R = F, V_1 = C, V_2 = C\}$. The goal is $\mathcal{G} = \{R = T, V_1 = B, V_2 = B\}$. An optimal plan for $s_{\mathcal{I}}$ is given by the action sequence* $\langle \mathrm{enter}, \mathrm{pickup}(V_1), \mathrm{smash}(V_1, V_2), \mathrm{drop}(V_1), \mathrm{enter}\rangle$.

Let $\mathcal{P}$ be the set of all facts of Π. *Consider the hyperabstraction $\rho : \mathcal{S} \mapsto 2^{\mathcal{P}}$ that maps every state to the facts true in it. Figure 1 depicts a* F*- and a* B*-interpretation of ρ, omitting edge labels and self-loops for the sake of readability. Ignore the red part for this example.*

Both interpretations are constructed by connecting for every action the precondition and effect facts accordingly. Consider $\mathrm{drop}(V_1)$ *along with its hyperedges (those marked in blue).* $\mathrm{drop}(V_1)$ *affects V_1 and R. In $\mathcal{H}^{\mathrm{F}}$, applying this action in any state with $R{=}T$ will change the value of R to F. The hyperedge* $\langle\{R{=}T\}, \mathrm{drop}(V_1), \{R{=}F\}\rangle$ *represents every such transition. $V_1{=}B$ could be added to the head as well, which would strengthen the hyperedge, further constraining the set of successor states, and thus possibly increasing the distance estimates. Adding $V_1{=}B$ is however not required to satisfy Definition 2. Similarly, the hyperedge* $\langle\{V_1{=}R\}, \mathrm{drop}(V_1), \{R{=}F, V_1{=}B\}\rangle$ *covers the application of* $\mathrm{drop}(V_1)$ *in every state where $V_1{=}R$. The remaining abstract concepts either don't represent any state where* $\mathrm{drop}(V_1)$ *is applicable, e. g., $V_1{=}C$, or remain invariant, e. g., the V_2 facts. In $\mathcal{H}^{\mathrm{B}}$, every application of* $\mathrm{drop}(V_1)$ *requires $V_1{=}R$. The action always makes true the facts $R{=}F$ and $V_1{=}B$, encoded by the two depicted hyperedges.*

The resulting heuristic estimates are $h^{\mathrm{F}}_{\rho}[\mathcal{H}^{\mathrm{F}}](s_{\mathcal{I}}) = h^{\mathrm{B}}_{\rho}[\mathcal{H}^{\mathrm{B}}](s_{\mathcal{I}}) = 3$. There is only a single goal state, so both measure the distance from the set of facts $P_{\mathcal{I}} := s_{\mathcal{I}}$ to $P_{\mathcal{G}} := \mathcal{G}$. Figure 1 annotates every $p \in \mathcal{P}$ with the distances $d^{\mathrm{F}}_{\mathcal{H}^{\mathrm{F}}}(\{p\}, P_{\mathcal{G}})$ and $d^{\mathrm{B}}_{\mathcal{H}^{\mathrm{B}}}(P_{\mathcal{I}}, \{p\})$. Consider the fact $p = \langle V_1, R\rangle$. The heuristic values of both heuristics rely on $d^{\mathrm{F}}_{\mathcal{H}^{\mathrm{F}}}(\{p\}, P_{\mathcal{G}}) = d^{\mathrm{B}}_{\mathcal{H}^{\mathrm{B}}}(P_{\mathcal{I}}, \{p\}) = 2$. p's only outgoing hyperedge in $\mathcal{H}^{\mathrm{F}}$ leads to $R{=}F$ and $V_1{=}B$. The estimated distance for the former fact is 1, the latter is contained in $P_{\mathcal{G}}$. The maximum over both distances is 1, resulting in $d^{\mathrm{F}}_{\mathcal{H}^{\mathrm{F}}}(p, P_{\mathcal{G}}) = 2$. In $\mathcal{H}^{\mathrm{B}}$, the only hyperedge leading to p has $R{=}T$ and $V_1{=}C$ in its tail. The distance from $P_{\mathcal{I}}$ to the former is 1, the latter is 0. The maximum is again 1, yielding $d^{\mathrm{B}}_{\mathcal{H}^{\mathrm{B}}}(P_{\mathcal{I}}, \{p\}) = 2$. Notably, the F *and* B *distance functions account for different applications of* enter*: $d^{\mathrm{F}}_{\mathcal{H}^{\mathrm{F}}}$ observes that* enter *must be executed after* drop*; $d^{\mathrm{B}}_{\mathcal{H}^{\mathrm{B}}}$ sees that* enter *must be executed prior to* pickup.

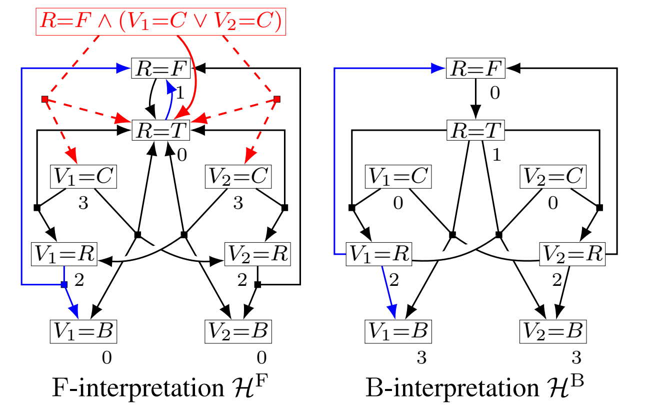

Figure 1: F- and B-interpretations for the task in Example 1.

As pointed out earlier, there might be different hypergraphs satisfying Definitions 2 or 3, differing in the design of the transition relation. Since the hyperabstraction heuristics compute their estimations based on the chosen hyperedges, the quality of the heuristics is strongly affected by those choices. Consider the red part of Figure 1, which introduces a new abstract concept p representing all states where the robot has not entered the shop and at least one of the vases is clear. Figure 1 shows two possible ways to represent the enter transitions leaving the states $[p]$: via the single solid hyperedge, or via the two dashed ones. Choosing the former, the information is lost that one of the vases was clear beforehand. In this case, $h^{\mathrm{F}}_{\rho}(s_{\mathcal{I}})$ stays 3. The latter also satisfies the F-interpretation definition: the application of enter in any state $s \in [p]$ changes the value of R to T. Moreover, one of $s(V_1)=C$ and $s(V_2)=C$ must have been true beforehand, both variables are not affected by enter. For the latter choice, we however obtain $h^{\mathrm{F}}_{\rho}(s_{\mathcal{I}}) = 4$.

Of course, one would like to always choose the hypergraph yielding the most accurate estimations relative to the abstract concepts at hand. Unfortunately this not is not feasible in general. Let $h^{\mathrm{F}*}_{\rho}$ and $h^{\mathrm{B}*}_{\rho}$ denote the "best" hyperabstraction heuristics one can obtain for ρ. This is, dropping the distinction between F and B for the remainder of this section, $h^{\mathrm{X}*}_{\rho}$ is defined such that, for every X-interpretation $\mathcal{H}$ of ρ, it holds that $h^{\mathrm{X}*}_{\rho} \geq h^{\mathrm{X}}_{\rho}[\mathcal{H}]$, and there is an X-interpretation $\mathcal{H}^*$ of ρ such that $h^{\mathrm{X}*}_{\rho} = h^{\mathrm{X}}_{\rho}[\mathcal{H}^*]$. Then

Theorem 1 *(i) $h^{\mathrm{X}*}_{\rho}$ is well-defined, i. e., for every ρ there exists an* X*-interpretation $\mathcal{H}^*$ of ρ as defined in the text, and (ii) given an* X*-interpretation $\mathcal{H}$ of ρ and state s, deciding whether $h^{\mathrm{X}}_{\rho}[\mathcal{H}](s) = h^{\mathrm{X}*}_{\rho}(s)$ is **NP**-hard.*

Proof (sketch). Consider the forward case. For (i), a desired F-interpretation $\mathcal{H}^*$ can be constructed by creating for every transition $\langle s, a, t\rangle$ and for every $p_s \in \rho(s)$, a hyperedge $\langle \{p_s\}, a, \rho(t)\rangle$. Regarding (ii), similar to the example given above, hyperedges can be used to resolve disjunctive conditions in atomic concepts. One can design Π_n and ρ_n such that such disjunctions are turned into transition nondeterminism, and thus exponentially many hyperedges must be considered to obtain $h^{\mathrm{X}*}_{\rho}$.

Despite this rather disappointing result, hyperabstraction heuristics remain admissible regardless of the interpretation. Moreover, we will see in the next sections that polynomially constructable interpretations suffice to show dominance over various existing heuristics from the literature.

Theorem 2 *For every task Π, hyperabstraction ρ, and* X*-interpretation $\mathcal{H}$ of ρ, $h^{\mathrm{X}}_{\rho}[\mathcal{H}]$ is consistent and goal-aware.*

Proof (sketch). Follows from the requirements on the structure of the hyperedges $\mathcal{E}$ imposed by Definitions 2 and 3.

Relation to Existing Techniques

Both abstraction and critical-path heuristic can naturally be written as hyperabstractions. The former boils down to a special case of hyperabstractions where every state is mapped to exactly one abstract concept. The latter can be seen as a hyperabstraction that maps every state to the set of atomic conjunctions true in that state.

This section goes beyond this simple observation, showing that critical-path heuristics can indeed be seen as one particular interpretation of a hyperabstraction. Moreover, we will establish the connection between abstractions and hyperabstractions, regarding hyperabstractions as a new way to admissibly combine multiple abstractions.

In both comparisons, we will provide dominance results for the non-additive case. Those results directly generalize to the additive combination of abstraction and critical-path heuristics through the use of appropriate cost partitionings. This promotes the quest for computing optimal cost partitionings for hyperabstractions. We close the section with a negative result, showing that finding optimal cost partitionings for hyperabstractions is **NP**-hard in general.

Critical-Path Heuristics

That critical-path heuristics are related to hypergraphs has been observed before, e. g., (Haslum 2006). However, this relation has so far not been spelled out formally and explicitly. We do this here using our notions of hyperabstractions. More specifically, we show how to construct, from $h^{\mathcal{C}}$, a hyperabstraction $\rho^{\mathcal{C}}$ and B-interpretation $\mathcal{H}^{\mathcal{C}}$ of $\rho^{\mathcal{C}}$ such that the equations underlying $h^{\mathcal{C}}$ and $h^{\mathrm{B}}_{\rho}[\mathcal{H}^{\mathcal{C}}]$ turn into the same.

Let $\mathcal{C}$ be any set of atomic conjunctions. Consider the hyperabstraction $\rho^{\mathcal{C}}$ with abstract concepts $\mathcal{P} = \mathcal{C}$, mapping every state to the atomic conjunctions satisfied in it. The hypergraph $\mathcal{H}^{\mathcal{C}}$ underlying $h^{\mathcal{C}}$'s distance computation is constructed as follows. $\mathcal{H}^{\mathcal{C}}$ contains one node for every atomic conjunction. For every conjunction $C \in \mathcal{C}$ and action $a \in \mathcal{A}$, $\mathcal{H}^{\mathcal{C}}$ contains a hyperedge $e^{C,a}$ iff $\mathit{Regr}(C, a)$ is defined, and $e^{C,a} = \langle T_{e^{C,a}}, a, \{C\}\rangle$ where $T_{e^{C,a}} = \{C' \in \mathcal{C} \mid C' \subseteq \mathit{Regr}(C, a)\}$. It is straightforward to verify that $d^{\mathrm{B}}_{\mathcal{H}^{\mathcal{C}}}$ is indeed equivalent to the recursive definition of $h^{\mathcal{C}}$. Observe that $\mathcal{H}^{\mathcal{C}}$ also satisfies Definition 3. Consider any transition $\langle s, a, t\rangle \in \mathcal{T}$, and any conjunction satisfied in t, $C_t \in \rho^{\mathcal{C}}(t)$. Clearly, $\mathit{Regr}(C_t, a)$ must be defined and $\mathit{Regr}(C_t, a) \subseteq s$. By the construction of $\mathcal{H}^{\mathcal{C}}$, there is an edge $e^{C_t,a} \in \mathcal{E}$ labeled with a and whose tail contains exactly the atomic conjunctions satisfied in $\mathit{Regr}(C_t, a)$. Thus $T_{e^{C_t,a}} \subseteq \rho^{\mathcal{C}}(s)$.

Theorem 3 *For every task Π, and set of atomic conjunctions $\mathcal{C}$, one can construct a* B*-interpretation $\mathcal{H}$ of $\rho^{\mathcal{C}}$ in polynomial time such that $h^{\mathrm{B}}_{\rho^{\mathcal{C}}}[\mathcal{H}] \geq h^{\mathcal{C}}$.*

This result immediately carries over to the comparison to additive critical-path heuristics (Haslum, Bonet, and Geffner 2005; Helmert and Domshlak 2009), i. e., is orthogonal to the application of cost-partitionings.

The definition of $\mathcal{H}^{\mathcal{C}}$ above is loosely related to the Π^m-compilation (Haslum 2009). The Π^m compilation explicitly encodes the satisfaction of atomic conjunctions C via new state variables π_C. To determine the satisfaction of C, Π^m introduces copies of actions a^f augmenting the precondition of a by additional context information, sets of facts f, very similar to the construction of $\mathcal{H}^{\mathcal{C}}$'s hyperedges. The Π^m compilation differs from the $\mathcal{H}^{\mathcal{C}}$ construction in that we consider arbitrary conjunctions. Moreover, Π^m introduces an action copy a^f for every action a and set of facts f of size of at most $m{-}1$, but sets true π_C for all conjunctions C guaranteed to be satisfied after the application of a in the context of f. In contrast, $\mathcal{H}^{\mathcal{C}}$ considers individually the satisfaction of each C by a via separate hyperedges $e^{C,a}$.

B-hyperabstractions generalize critical-path heuristics in two aspects: (1) they natively support more complex abstract states; (2) critical-path heuristics represent only one particular instantiation of the associated hyperabstraction. How to exploit (1) will be the topic of an upcoming section. Regarding (2), Theorem 1 already indicates that $\mathcal{H}^{\mathcal{C}}$ might not be the *optimal* B-interpretation for $\rho^{\mathcal{C}}$. In fact, under particular circumstances, it is possible to consider different B-interpretations, leading to higher heuristic estimates, while still staying within the polynomial (in $|\mathcal{C}|$) size bound. For instance, assume there are actions that affect variables without imposing preconditions on them. Such variables will be left unspecified in every regression over this action. Hence, conjunctions that contain assignments to these variables will never be part of the tails of the hyperedges generated for this action. To avoid this shortcoming, one can generate multiple hyperedges for an atomic conjunction action pair, instead of just a single one. Then in each of those hyperedges, one can augment the regression with an additional context, allowing to include more atomic conjunctions in the tails of the hyperedges, and thus may lead to higher heuristic estimates.

Theorem 4 *There exist Π, $\mathcal{C}$, and B-interpretations $\mathcal{H}$ of $\rho^{\mathcal{C}}$ where $|\mathcal{H}|$ is polynomially bounded in $|\mathcal{C}|$ but $h^{\mathrm{B}}_{\rho^{\mathcal{C}}}[\mathcal{H}] > h^{\mathcal{C}}$.*

The idea above is similar to *context-splitting* (Röger, Pommerening, and Helmert 2014). The latter is a technique that, given a propositional formula ϕ over facts and an action a, constructs a new task where a is split into two actions that are identical to a but one having the precondition $pre_a \wedge \phi$ and the other one $pre_a \wedge \neg\phi$. This split can be helpful to derive (admissible) additive heuristics, e. g., in the construction of cost partitionings the costs of both splits can be distributed independently to different heuristics. However, in contrast to context-splitting, hyperabstraction interpretations allow to split hyperedges much more specifically, independently of other hyperedges of the same action. Moreover, and much more crucially, hyperedges can be split over non-disjoint contexts. In the example in Figure 1, the two dashed hyperedges are not mutually exclusive: both are simultaneously representing the application of enter to the initial state.

Abstractions

Let $\alpha \colon \mathcal{S} \mapsto \mathcal{S}^{\alpha}$ be any abstraction, and Θ^{α} be its induced abstract state space. Labeled transition systems can be seen as a special case of labeled hypergraphs whose hyperedges have heads and tails with size of exactly one, i. e., satisfy the strict BF-hypergraph criterion. This has two immediate consequences: (1) Θ^{α} satisfies Definitions 2 and 3, by construction. (2) For strict BF-hypergraphs the forward distance $d^{\mathrm{F}}(\{n\}, N')$ boils down to the minimal graph distance from n to any $n' \in N'$. Therefore, $h^{\mathrm{F}}_{\alpha}[\Theta^{\alpha}](s) = d^{\mathrm{F}}_{\Theta^{\alpha}}(\{\alpha(s)\}, \mathcal{S}^{\alpha}_{\mathcal{G}}) = h^{*}_{\Theta^{\alpha}}(\alpha(s)) = h^{\alpha}(s)$, i. e., abstraction heuristics can be interpreted directly as F-hyperabstraction heuristics. Moreover, in the construction of any F-interpretation $\mathcal{H}$ of α, there are exactly two possible hyperedges to represent any transition $\langle s, a, t\rangle \in \mathcal{T}$: either $\langle \{\alpha(s)\}, a, \{\alpha(t)\}\rangle$ or $\langle \{\alpha(s)\}, a, \emptyset\rangle$. The former is selected by Θ^{α}, and clearly carries more information than the latter, i. e., $h^{\alpha} = h^{\mathrm{F}*}_{\alpha}$. However, although Θ^{α} being a B-interpretation of α, h^{α} is in general not equivalent to $h^{\mathrm{B}}_{\alpha}[\Theta^{\alpha}]$. In strict BF-hypergraphs, $d^{\mathrm{B}}(\{n\}, N')$ gives the *maximum* of the minimal graph distances from n to any $n' \in N'$. To account for that, h^{B} considers in N' only those abstract concepts representing *all* goal states. In case of the abstraction α, this means that $h^{\mathrm{B}}_{\alpha}[\Theta^{\alpha}] = h^{\alpha}$ only if (*) $|\mathcal{S}^{\alpha}_{\mathcal{G}}| = 1$, but $h^{\mathrm{B}}_{\alpha}[\Theta^{\alpha}](s) = d^{\mathrm{B}}_{\mathcal{H}}\Theta^{\alpha}(\{\alpha(s)\}, \emptyset) = 0 \leq h^{\alpha}$ if α maps any two goal states to different abstract states. Note that (*) necessarily holds for tasks in transition normal form (TNF) (Pommerening and Helmert 2015).

Single abstractions hence constitute trivial instantiations of our concepts of F-hyperabstraction heuristics, and under certain circumstances also B-hyperabstraction heuristics. The converse is however not true. Helmert et al. (2014) have shown planning tasks where the critical-path heuristic h^2 is perfect, but, unless $\mathbf{P} = \mathbf{NP}$, it is not possible to construct any M&S abstraction α with $h^{\alpha} = h^{*}$ in polynomial time. Given the observations from the previous section, this results directly carries over to B-hyperabstraction heuristics in general. It turns out that the hypergraph characteristics are actually not required to show this relation:

Theorem 5 *There exist families of tasks Π_n and polynomially size-bounded forward and backward hyperabstraction heuristics such that $h^{\mathrm{F}}_{\rho^{\mathrm{F}}}[\mathcal{H}^{\mathrm{F}}] = h^{\mathrm{B}}_{\rho^{\mathrm{B}}}[\mathcal{H}^{\mathrm{B}}] = h^{*}$ and both $\mathcal{H}^{\mathrm{F}}$ and $\mathcal{H}^{\mathrm{B}}$ are BF-hypergraphs, but unless **$\mathbf{P} = \mathbf{NP}$**, it is not possible to construct any M&S abstraction α such that $h^{\alpha} = h^{*}$ in polynomial time.*

Proof (sketch). The proof is based on a family of tasks representing CNF formulas. Given a formula ϕ, the corresponding task is designed so that states represent possible Boolean variable assignments, $h^{*}(s) = 1$ iff s does not satisfy the formula, and $h^{*}(s) = 0$ otherwise. Using this, the evaluation of ϕ can be encoded as F and B hyperabstraction heuristics, even restricted to BF-hypergraphs. However, if it were possible to construct a M&S abstraction α such that $h^{\alpha} = h^{*}$, for arbitrary ϕ, in time polynomial in the size of ϕ, then we could solve SAT for any CNF formula in polynomial time through an inspection of α.

Note that even though this claim considers only M&S abstractions, it also applies to less general cases like PDBs.

So far, we have seen that hyperabstractions are strictly more powerful than single abstractions, even restricted to BF-hypergraphs, and thus without making use of the possibility to define actual hyperedges. We next show that single hyperabstractions also dominate the combination of collections of abstractions. Unfortunately, we cannot report any result for the other direction, i. e., it remains unclear whether a single hyperabstraction can be always compiled into an equivalent collection of abstractions, in polynomial time. Let $\alpha_1, \dots, \alpha_n$ be any set of abstractions. Consider the hyperabstraction whose abstract concepts $\mathcal{P}$ are given by the union of all $\mathcal{S}^{\alpha_i}$, combining all the abstraction functions into the single hyperabstraction function: $\rho(s) = \{\alpha_1(s), \dots, \alpha_n(s)\}$. Consider the BF-hypergraph $\mathcal{H}$, given by the union of all induced abstract state spaces $\Theta^{\alpha_1}, \dots, \Theta^{\alpha_n}$. $\mathcal{H}$ satisfies Definitions 2 and 3 similarly to the single abstraction case.

Moreover, since abstract states of different abstractions are not connected, the forward distance in $\mathcal{H}$ from any abstract state to the abstract goal states remains the same as in the abstract state's corresponding abstraction, i. e., it holds that $d^{\mathrm{F}}_{\mathcal{H}}(\{s^{\alpha_i}\}, \bigcup_i \mathcal{S}^{\alpha_i}_{\mathcal{G}}) = d^{\mathrm{F}}_{\mathcal{H}}(\{s^{\alpha_i}\}, \mathcal{S}^{\alpha_i}_{\mathcal{G}})$, and therefore $h^{\mathrm{F}}_{\rho}[\mathcal{H}](s) = d^{\mathrm{F}}_{\mathcal{H}}(\{\alpha_1(s), \dots, \alpha_n(s)\}, \bigcup_i \mathcal{S}^{\alpha_i}_{\mathcal{G}}) = \max_{1\leq i\leq n} d^{\mathrm{F}}_{\mathcal{H}}(\{\alpha_i(s)\}, \mathcal{S}^{\alpha_i}_{\mathcal{G}}) = \max_{1\leq i \leq n} h^{\alpha_i}(s)$.

In the construction of $\mathcal{H}$, we still did not make use of the expressiveness of hypergraphs. The consideration of hypergraphs instead of simple transition systems allows drawing connections *between* the different abstractions. Consider for example two projections, one on variable v, one on u. If there is an action that modifies both v and u, then every hyperedge labeled by this action can be extended by connections to states in both projections. Moreover, adding more abstract states to the tail (in the B case) and to the head (in the F case), can never cause a decrease in distance. Thus adding such connections can only be beneficial for the resulting heuristic. In other words, hyperabstractions can be seen as a new method to admissibly combine a set of abstractions, which dominates just taking the maximum:

Theorem 6 *Let $\alpha_1, \dots, \alpha_n$ be any abstractions. Consider ρ as described in the text. One can always construct an* F*-interpretation $\mathcal{H}$ of ρ in time polynomial in the size of $\Theta^{\alpha_1}, \dots, \Theta^{\alpha_n}$ such that $h^{\mathrm{F}}_{\rho,\mathcal{H}} \geq \max_{1\leq i\leq n} h^{\alpha_i}$. There are cases where this dominance holds strictly. If Π is in TNF, the same holds for backward hyperabstraction heuristics.*

This result applies to the additive combination of abstractions as well. In particular, given any set of abstraction heuristics that are additively combined via a cost partitioning $\mathbf{c}$, Theorem 6 can easily be extended to show the existence of a *single* hyperabstraction heuristic that, under the application of the same $\mathbf{c}$, dominates the additive ensemble:

Corollary 1 *For every ensemble of abstractions $\alpha_1, \dots, \alpha_n$, a single* F*-hyperabstraction heuristic $h^{\mathrm{F}}_{\rho}[\mathcal{H}]$ can be constructed in polynomial time (in the size of the abstract state space) such that it holds, for every cost partitioning $\mathbf{c}$, that $h^{\mathrm{F}}_{\rho}[\mathcal{H}][\![\mathbf{c}]\!] \geq h_{\alpha_1,\dots,\alpha_n}[\![\mathbf{c}]\!]$. If Π is in TNF, the same holds for* B *hyperabstraction heuristics.*

Optimal Cost Partitioning

Plugging multiple cost functions into a single hyperabstraction heuristic provides a very powerful formalism able to dominate arbitrary additive ensembles of abstraction heuristics as well as the application of cost partitionings to critical-path heuristics. This raises the question of whether we can do anything more with additive hyperabstractions. To answer this question to at least some extent, we next show that in contrast to additive abstractions, it is hard to compute the optimal cost partitioning for hyperabstraction heuristics. The proof works via a detour to the critical-path heuristic h^1:

Theorem 7 *Optimal cost partitioning for h^1 is **NP**-hard.*

Since critical-path heuristics can be seen as particular B-interpretations of a hyperabstraction, this result immediately carries over to backward hyperabstraction heuristics. Moreover, due to the symmetric nature of the definitions of B- and F-hyperabstraction heuristics, it is not difficult to extend the proof for h^1 to work also for F-interpretations. We obtain:

Corollary 2 *Optimal cost partitioning for hyperabstraction heuristics in general is **NP**-hard.*

Practical Construction of Hyperabstractions

Hyperabstraction heuristics consist of two components: (1) the hyperabstraction function ρ, and (2) an interpretation of ρ as a hypergraph. Regarding (1), given the vast literature in the automatic construction of abstraction heuristics in classical planning, it is natural to define ρ as the union of abstraction functions $\alpha_1, \dots, \alpha_k$. Each individual abstraction α_i can for example be a PDB, Cartesian, or M&S abstraction. The considered abstractions do not have to be of the same type, though. The set of abstract concepts are given by $\mathcal{P} = \bigcup_{i\in[1,k]} \mathcal{S}^{\alpha_i}$. Each state s is mapped to its corresponding abstract state in each abstraction, i. e., $\rho(s) := \{\alpha_1(s), \dots, \alpha_k(s)\}$.

Regarding (2), we next show a generic method to generate B-hyperedges so to obtain different B-interpretations for ρ. The construction of F-interpretations is omitted for the sake of brevity, but works analogously. To provide a unified algorithm that supports all the different abstraction variants, we treat abstract states as the equivalence relations they induce. The algorithm operates directly on these sets of states. In an actual implementation, these sets can of course not be enumerated explicitly, each abstract state possibly representing an exponential number of real states. However, the specific operations used by the algorithm can be implemented efficiently for various kinds of abstractions (particularly Cartesian and PDBs), taking into account the concrete structure and representation of abstract states. For computing the tail of the B-hyperedges, a notion for the regression of sets of states is required. Let $T \subseteq \mathcal{S}$ be any set of states, and $a \in \mathcal{A}$ be any action. We define the regression of T over a as $\mathit{Regr}(T, a) = \{s \in \mathcal{S} \mid \langle s, a, t\rangle \in \mathcal{T}, t \in T\}$.

Consider any action $a \in \mathcal{A}$. We next describe the method, Algorithm 1, to generate the B-hyperedges for this action. To satisfy Definition 3, we must add, for every abstract state $t^{\alpha_i} \in \mathcal{P}$ where $\mathit{Regr}([t^{\alpha_i}], a) \neq \emptyset$, at least one representative hyperedge $e = \langle T_e, a, \{t^{\alpha_i}\}\rangle$. Different interpretations may be constructed differing in how many such edges e to consider. There are two extremes:

(i) A single hyperedge that represents all transitions into any state in $[t^{\alpha_i}]$. In this case, $T_e = \{s^{\alpha} \in \mathcal{P} \mid \mathit{Regr}([t^{\alpha_i}], a) \subseteq [s^{\alpha}]\}$ includes the maximal number of abstract states, satisfying Definition 3.

(ii) One hyperedge for every concrete transition into any state in $[t^{\alpha_i}]$, i. e., for every state $s \in \mathit{Regr}([t^{\alpha_i}], a)$, the hyperedge $e_{s,a,t^{\alpha_i}}$ with tail $T_{e_{s,a,t^{\alpha_i}}} = \rho(s)$.

(i) gives the simplest "reasonable" choice of a B-interpretation of ρ, yet pays this simplicity through possible information loss. The red part in Figure 1 shows an example.

Algorithm 1: Generic algorithm for generating all B-hyperedges with head $t^{\alpha_i} \in \mathcal{S}^{\alpha_i} \subseteq \mathcal{P}$, and action $a \in \mathcal{A}$. To obtain F-interpretations, the regression operation must be substituted by progression, and the head and the tail of the hyperedges must be swapped.

```
Input: Abstractions $\alpha_1, \dots, \alpha_k$,
       abstract state $t^{\alpha_i} \in \mathcal{S}^{\alpha_i}$, action $a \in \mathcal{A}$
Output: B-hyperedges $\mathcal{E}(t^{\alpha_i}, a)$
1  $X \leftarrow$ SelectSplit($\{\alpha_1, \dots, \alpha_k\}, t^{\alpha_i}, a$) ;
2  $Y \leftarrow$ SelectOthers($\{\alpha_1, \dots, \alpha_k\}, t^{\alpha_i}, a$) ;
3  $\mathcal{E}(t^{\alpha_i}, a) \leftarrow \emptyset$ ;
4  if $X = \emptyset$ then
5      $T_e \leftarrow \{s^\alpha \in \mathcal{S}^\alpha \mid \alpha \in Y, Regr([t^{\alpha_i}], a) \subseteq [s^\alpha]\}$;
6      $\mathcal{E}(t^\alpha, a) \leftarrow \{\langle T_e, a, \{t^{\alpha_i}\}\rangle\}$
7  foreach $S_X \in \prod_{\alpha \in X} \mathcal{S}^\alpha$ do
8      $S_{ctxt} \leftarrow Regr([t^{\alpha_i}], a) \cap \bigcap_{s^\alpha \in S_X} [s^\alpha]$ ;
9      if $S_{ctxt} \neq \emptyset$ then
10         $S_Y \leftarrow \{s^\alpha \in \mathcal{S}^\alpha \mid \alpha \in Y, S_{ctxt} \subseteq [s^\alpha]\}$ ;
11         $T_e \leftarrow S_X \cup S_Y$;
12         $\mathcal{E}(t^\alpha, a) \leftarrow \mathcal{E}(t^\alpha, a) \cup \{\langle T_e, a, \{t^\alpha\}\rangle\}$ ;
13 return $\mathcal{E}(t^\alpha, a)$ ;
```

On the contrary, (ii) provides the most informative hypergraph possible, but the number of hyperedges is worst-case exponential in the number of abstract states.

To trade-off between heuristic accuracy and computational cost, Algorithm 1 allows to interpolate between the two extremes by means of the methods `SelectSplit` and `SelectOthers`, both choosing a subset of the input abstractions. The abstractions given by `SelectSplit` are used to determine transitions for which to generate separate hyperedges as in (ii). To do so, the hyperedges are built so that each $e \in \mathcal{E}(t^{\alpha_i}, a)$ contains $S_X \subseteq T_e$ for some choice S_X of abstract states for all abstractions selected by `SelectSplit`. Each e is supposed to cover only those transitions $s[\![a]\!] \in [t^{\alpha_i}]$ where s is jointly represented by all abstract states in S_X. To ensure that Definition 3 is satisfied, Algorithm 1 computes the collection $\mathcal{E}(t^{\alpha_i}, a)$ in an exhaustive manner, considering, and possibly creating a hyperedge, for every combination of abstract states $S_X \in \prod_{\alpha \in X} \mathcal{S}^\alpha$. For the remaining abstractions $\alpha \notin X$, in order to preserve that every generated hyperedge e with set S_X indeed covers all applications $s[\![a]\!]$ relevant to S_X, we may include in the tail of e an abstract state $s^\alpha \in \mathcal{S}^\alpha$ only if s^α represents *all* the states s with such transition (line 10). The method `SelectOthers` determines the abstractions for which to do this analysis, and thus which abstractions to take into account in the computation of the tails in addition to X.

Table 1 shows three particular choices of `SelectSplit` and `SelectOthers`, resulting in: (a) the critical-path heuristic h^m, considering as abstractions the projections onto all variable subsets of size up to m; (b) the maximum over multiple abstractions, and (c) a new combination, which guarantees to dominate both of them.

	`SelectSplit`	`SelectOthers`
(a) h^m	$\emptyset$	$\{\alpha_1, \dots, \alpha_k\}$
(b) $\max_{j \in [1,k]} h^{\alpha_j}$	$\{\alpha_i\}$	$\emptyset$
(c) combination	$\{\alpha_i\}$	$\{\alpha_1, \dots, \alpha_k\}$

Table 1: Different instantiations of Algorithm 1.

Experiments

Given the extensive research on the construction of state-of-the-art abstractions and critical-path heuristics, e. g., (Seipp and Helmert 2018; Helmert et al. 2014; Franco et al. 2017; Steinmetz and Hoffmann 2017), we do not expect to beat those configurations. The goal of our evaluation is to verify whether the theoretical dominance results also show in practice. For that, we investigate whether applying the hyperabstraction construction on top of a set of abstractions is more informative than taking their maximum. Our implementation is in Fast Downward (Helmert 2006). We are using all IPC STRIPS benchmarks of the optimal tracks. The experiments were performed on a cluster of Intel E5-2660 machines running at 2.20 GHz, restricting CPU time to 30 minutes and memory to 4 GB. We use $PDB(m)$: pattern databases of size up to $m \in \{1, 2, 3\}$; as well as Cartesian abstractions constructed via the CEGAR approach (Seipp and Helmert 2018) as seed abstractions. The hyperabstraction interpretations are computed using Algorithm 1 with the parameters as shown in Table 1(c).

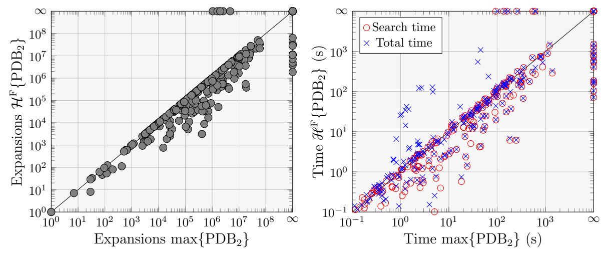

Figure 2: Comparison of the F-hyperabstraction based on $PDB(2)$ versus taking their maximum in terms of expansions until the last f-layer and runtime.

Figure 2 compares the systematic PDBs of size 2 against the F-interpretation of the corresponding hyperabstraction. The F-interpretation of abstraction heuristics can provide more informative estimates, reducing the number of expansions in some cases compared to taking the maximum among them. This result can be expected to carry over to the additive scenario where the same cost-partitioning is used for abstractions and hyperabstractions. Moreover, F-interpretations do not have a large overhead over abstraction heuristics in evaluation time, though the comparison in total time shows that the construction is more expensive.

Applying the hyperabstraction construction on the CEGAR abstractions did not have a considerable effect on the heuristic accuracy, regardless of considering F- or B-hyperabstraction heuristics. An explanation is that the construction of the individual Cartesian abstractions focuses on different parts of the planning task.

Conclusion

In this paper we introduced hyperabstractions, a new family of heuristics that generalizes abstractions and critical-path heuristics, dominating most admissible heuristics in the literature. Hyperabstractions map each state to a set of abstract states. This is related to more general notions of abstractions used in model-checking, where the same state may be mapped into multiple abstract predicates (Cousot and Cousot 1977; Ball, Podelski, and Rajamani 2001). Previous multimapping abstractions for planning resulted in less informed estimates than taking the maximum over several abstraction heuristics (Pang and Holte 2011). We circumvent this by computing distances in an hypergraph instead, ensuring that hyperabstractions dominate the corresponding abstraction heuristics, sometimes strictly.

Hyperabstractions have potential for obtaining stronger heuristics than previous families of heuristics, e.g., as a novel method to combine different abstractions, or by explicitly reasoning about the hypergraphs underlying the computation of critical-path heuristics. However, it is yet unknown how to exploit their full potential. Promising lines for future research include finding automatic methods to derive hyperabstractions (i.e., identifying cases where there may be oportunities to combine abstractions via an hyperabstraction), and additive ensembles thereof.

Acknowledgments. Thanks to the anonymous reviewers for their very helpful comments. This work has been partially supported by German Federal Ministry of Education and Research (BMBF) through funding for CISPA (grant no. 16KIS0656), and the German Research Foundation (DFG), under grant HO 2169/5-1. The FAI research group at Saarland University has received support by DFG grant 389792660 as part of TRR 248 (see https://perspicuous-computing.science).

References

Ausiello, G., and Laura, L. 2017. Directed hypergraphs: Introduction and fundamental algorithms—a survey. *Theoretical Computer Science* 658:293 – 306.

Bäckström, C. 1995. Expressive equivalence of planning formalisms. *Artificial Intelligence* 76(1–2):17–34.

Ball, T.; Podelski, A.; and Rajamani, S. K. 2001. Boolean and Cartesian abstraction for model checking C programs. In *Proc. of TACAS'01*, 268–283.

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence* 129(1–2):5–33.

Cousot, P., and Cousot, R. 1977. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 5th ACM Symposium on Principles of Programming Languages (POPL'77)*, 238–252.

Edelkamp, S. 2001. Planning with pattern databases. In *Proc. of ECP'01*, 13–24.

Edelkamp, S. 2002. Symbolic pattern databases in heuristic search planning. In *Proc. of AIPS'02*, 274–283.

Edelkamp, S. 2006. Automated creation of pattern database search heuristics. In *Proc. of MoChArt'06*, 35–50.

Fickert, M., and Hoffmann, J. 2017. Complete local search: Boosting hill-climbing through online heuristic-function refinement. In *Proc. ICAPS'17*.

Franco, S.; Torralba, A.; Lelis, L. H.; and Barley, M. 2017. On creating complementary pattern databases. In *Proc. of IJCAI'17*.

Gallo, G.; Longo, G.; and Pallottino, S. 1993. Directed hypergraphs and applications. *Discrete Applied Mathematics* 42(2):177–201.

Haslum, P., and Geffner, H. 2000. Admissible heuristics for optimal planning. In *Proc. of AIPS'00*, 140–149.

Haslum, P.; Botea, A.; Helmert, M.; Bonet, B.; and Koenig, S. 2007. Domain-independent construction of pattern database heuristics for cost-optimal planning. In *Proc. of AAAI'07*, 1007–1012.

Haslum, P.; Bonet, B.; and Geffner, H. 2005. New admissible heuristics for domain-independent planning. In *Proc. of AAAI'05*, 1163–1168.

Haslum, P. 2006. Improving heuristics through relaxed search - an analysis of TP4 and HSP*a in the 2004 planning competition. *Journal of Artificial Intelligence Research* 25:233–267.

Haslum, P. 2009. $h^m(P) = h^1(P^m)$: Alternative characterisations of the generalisation from $h^{\max}$ to h^m. In *Proc. of ICAPS'09*, 354–357.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In *Proc. of ICAPS'09*, 162–169.

Helmert, M.; Haslum, P.; Hoffmann, J.; and Nissim, R. 2014. Merge & shrink abstraction: A method for generating lower bounds in factored state spaces. *Journal of the Association for Computing Machinery* 61(3):16:1–16:63.

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

Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered landmarks in planning. *Journal of Artificial Intelligence Research* 22:215–278.

Katz, M., and Domshlak, C. 2008a. Optimal additive composition of abstraction-based admissible heuristics. In *Proc. of ICAPS'08*, 174–181.

Katz, M., and Domshlak, C. 2008b. Structural patterns heuristics via fork decomposition. In *Proc. of ICAPS'08*, 182–189.

Pang, B., and Holte, R. C. 2011. State-set search. In *Proc. of SOCS'11*.

Pommerening, F., and Helmert, M. 2015. A normal form for classical planning tasks. In *Proc. ICAPS'15*, 188–192.

Röger, G.; Pommerening, F.; and Helmert, M. 2014. Optimal planning in the presence of conditional effects: Extending lm-cut with context splitting. In *Proc. of ECAI'14*, 765–770.

Seipp, J., and Helmert, M. 2018. Counterexample-guided Cartesian abstraction refinement for classical planning. *Journal of Artificial Intelligence Research* 62:535–577.

Steinmetz, M., and Hoffmann, J. 2017. State space search nogood learning: Online refinement of critical-path dead-end detectors in planning. *Artificial Intelligence* 245:1–37.

Steinmetz, M., and Torralba, Á. 2019. Bridging the gap between abstractions and critical-path heuristics via hypergraphs. Technical report. Available at http://fai.cs.uni-saarland.de/steinmetz/papers/icaps19-tr.pdf.

Torralba, Á.; Alcázar, V.; Kissmann, P.; and Edelkamp, S. 2017. Efficient symbolic search for cost-optimal planning. *Artificial Intelligence* 242:52–79.