From Non-Negative to General Operator Cost Partitioning

Florian Pommerening and **Malte Helmert** and **Gabriele Röger** and **Jendrik Seipp**
University of Basel
Basel, Switzerland
{florian.pommerening,malte.helmert,gabriele.roeger,jendrik.seipp}@unibas.ch

Abstract

Operator cost partitioning is a well-known technique to make admissible heuristics additive by distributing the operator costs among individual heuristics. Planning tasks are usually defined with non-negative operator costs and therefore it appears natural to demand the same for the distributed costs. We argue that this requirement is not necessary and demonstrate the benefit of using *general* cost partitioning. We show that LP heuristics for operator-counting constraints are cost-partitioned heuristics and that the state equation heuristic computes a cost partitioning over atomic projections. We also introduce a new family of *potential heuristics* and show their relationship to general cost partitioning.

Introduction

Heuristic search is commonly used to solve classical planning tasks. Optimal planning requires admissible heuristics, which estimate the cost to the goal without overestimation. If several admissible heuristics are used, they must be combined in a way that guarantees admissibility. By evaluating each heuristic on a copy of the task with a suitably reduced cost function, *operator cost partitioning* (Katz and Domshlak 2010) allows summing heuristic estimates admissibly.

Previous work on operator cost partitioning required all cost functions to be non-negative. We show that this restriction is not necessary to guarantee admissibility and that dropping it can lead to more accurate heuristic estimates. Moreover, we demonstrate that when allowing negative operator costs, heuristics based on the recently proposed *operator-counting constraints* (Pommerening et al. 2014b) can be interpreted as a form of optimal cost partitioning. This includes the *state equation heuristic* (Bonet and van den Briel 2014), which was previously thought (Bonet 2013) to fall outside the four main categories of heuristics for classical planning: abstractions, landmarks, delete-relaxations and critical paths (Helmert and Domshlak 2009). We show that the state equation heuristic computes a general optimal cost partitioning over atomic projection heuristics. In addition, we introduce *potential heuristics*, a new family of heuristics with a close relationship to general operator cost partitioning. Finally, we empirically demonstrate the positive influence of allowing negative costs in cost partitioning and the strong performance of potential heuristics.

SAS$^+$ Planning and Heuristics

We consider SAS$^+$ planning (Bäckström and Nebel 1995) with operator costs, where a *task* is given as a tuple $\Pi = \langle \mathcal{V}, \mathcal{O}, s_{\mathrm{I}}, s_\star, \mathit{cost} \rangle$. Each V in the finite set of *variables* $\mathcal{V}$ has a finite domain $\mathit{dom}(V)$. A *state* s is a variable assignment over $\mathcal{V}$. A *fact* is a pair of a variable and one of its values. For a partial variable assignment p we use $\mathit{vars}(p)$ to refer to the set of variables on which p is defined and where convenient we consider p to be a set of facts $\{\langle V, p[V] \rangle \mid V \in \mathit{vars}(p)\}$. Each *operator* o in the finite set $\mathcal{O}$ has a *precondition* $\mathit{pre}(o)$ and an *effect* $\mathit{eff}(o)$, which are both partial variable assignments over $\mathcal{V}$. Operator o is *applicable* in a state s if s and $\mathit{pre}(o)$ are consistent, i. e., they do not assign a variable to different values. Applying o in s results in state $s[\![o]\!]$ with

$$s[\![o]\!][V] = \begin{cases} \mathit{eff}(o)[V] & \text{if } V \in \mathit{vars}(\mathit{eff}(o)) \\ s[V] & \text{otherwise.} \end{cases}$$

An operator sequence $\pi = \langle o_1, \ldots, o_n \rangle$ is applicable in state s_0 if there are states $s_1, \ldots, s_n$ such that for $1 \leq i \leq n$, operator o_i is applicable in s_{i-1} and $s_i = s_{i-1}[\![o_i]\!]$. We refer to the resulting state s_n by $s_0[\![\pi]\!]$.

The *initial state* s_{I} is a complete and the *goal description* $s_\star$ a partial variable assignment over $\mathcal{V}$. For a state s, an *s-plan* is an operator sequence π that is applicable in s and for which $s[\![\pi]\!]$ and $s_\star$ are consistent. An s_{I}-plan is called a *solution* or *plan* for the task.

The function $\mathit{cost} : \mathcal{O} \to \mathbb{R}_0^+$ assigns a non-negative *cost* to each operator. Throughout the paper, we will vary planning tasks by considering different cost functions, so the following definitions refer to arbitrary cost functions cost', which need not be equal to cost.

The cost of a plan $\pi = \langle o_1, \ldots, o_n \rangle$ under cost function cost' is $\mathit{cost}'(\pi) = \sum_{i=1}^{n} \mathit{cost}'(o_i)$. An *optimal s-plan under cost function cost'* is a plan π that minimizes $\mathit{cost}'(\pi)$ among all s-plans. Its cost is denoted with $h^*(s, \mathit{cost}')$. If there is no s-plan then $h^*(s, \mathit{cost}') = \infty$. A *heuristic* h is a function that produces an estimate $h(s, \mathit{cost}') \in \mathbb{R}_0^+ \cup \{\infty\}$ of the optimal cost $h^*(s, \mathit{cost}')$. If $\mathit{cost}' = \mathit{cost}$, these notations can be abbreviated to $h^*(s)$ and $h(s)$. In cases where we

do not need to consider other cost functions, we will define heuristics only for *cost*.

We say that $c \in \mathbb{R}_0^+ \cup \{\infty\}$ is an *admissible heuristic estimate* for state s and cost function *cost'* if $c \leq h^*(s, \mathit{cost}')$. A heuristic function is *admissible* if $h(s, \mathit{cost}')$ is an admissible heuristic estimate for all states s and cost functions *cost'*. A heuristic h is *goal-aware* if $h(s, \mathit{cost}') = 0$ for every state s consistent with $s_\star$ and cost function *cost'*. It is *consistent* if $h(s, \mathit{cost}') \leq \mathit{cost}'(o) + h(s[\![o]\!], \mathit{cost}')$ for every state s, operator o applicable in s and cost function *cost'*. Heuristics that are goal-aware and consistent are admissible, and admissible heuristics are goal-aware. The A* algorithm (Hart, Nilsson, and Raphael 1968) generates optimal plans when equipped with an admissible heuristic.

Non-Negative Cost Partitioning

A set of admissible heuristics is called *additive* if their sum is admissible. A famous early example of additive heuristics are *disjoint pattern database heuristics* for the sliding-tile puzzle (Korf and Felner 2002), which were later generalized to classical planning (e. g., Haslum et al. 2007).

Cost partitioning (Katz and Domshlak 2007; Yang et al. 2008; Katz and Domshlak 2010) is a technique for making arbitrary admissible heuristics additive. The idea is that each heuristic may only account for a fraction of the actual operator costs, so that the total cost cannot exceed the optimal plan cost.

Definition 1 (Non-negative cost partitioning)**.** *Let Π be a planning task with operators $\mathcal{O}$ and cost function cost. A* non-negative cost partitioning *for Π is a tuple $\langle \mathit{cost}_1, \ldots, \mathit{cost}_n \rangle$ where $\mathit{cost}_i : \mathcal{O} \to \mathbb{R}_0^+$ for $1 \leq i \leq n$ and $\sum_{i=1}^n \mathit{cost}_i(o) \leq \mathit{cost}(o)$ for all $o \in \mathcal{O}$.*

Proposition 1 (Katz and Domshlak 2010)**.** *Let Π be a planning task, let $h_1, \ldots, h_n$ be admissible heuristics for Π, and let $P = \langle \mathit{cost}_1, \ldots, \mathit{cost}_n \rangle$ be a non-negative cost partitioning for Π. Then $h_P(h_1, \ldots, h_n, s) = \sum_{i=1}^n h_i(s, \mathit{cost}_i)$ is an admissible heuristic estimate for s.*

Among other contributions, Katz and Domshlak (2010) showed how to compute an *optimal* (i. e., best possible) cost partitioning for a given state and a wide class of abstraction heuristics in polynomial time. However, for reasons of efficiency, non-optimal cost partitioning such as *zero-one cost partitioning* (e. g., Edelkamp 2006), *uniform cost partitioning* (e. g., Karpas and Domshlak 2009) or *post-hoc optimization* (Pommerening, Röger, and Helmert 2013) is also commonly used.

General Cost Partitioning

We argue that there is no compelling reason to restrict operator costs to be non-negative when performing cost partitioning, and we will remove this restriction in the following, permitting *general* (possibly negative) cost functions $\mathit{cost}' : \mathcal{O} \to \mathbb{R}$. This means that we must generalize some concepts and notations that relate to operator costs.

Shortest paths in weighted digraphs that permit negative weights are well-defined except in the case where there exists a cycle of negative overall cost that is incident to a path from the source state to the goal. So we can retain our definition of optimal plans except for this case, in which plans of arbitrarily low cost exist and we set $h^*(s, \mathit{cost}') = -\infty$.

Heuristic functions may now map to the set $\mathbb{R} \cup \{-\infty, \infty\}$. The definitions of admissibility and consistency remain unchanged, but the notion of *goal-aware* heuristics must be amended to require $h(s, \mathit{cost}') \leq 0$ instead of $h(s, \mathit{cost}') = 0$ for goal states s to retain the result that a consistent heuristic is goal-aware iff it is admissible. (Even if we are already in a goal state, the cheapest path to a goal state may be a non-empty path with negative cost.)

With these preparations, we can show the analogue of Proposition 1 for general cost partitioning:

Theorem 1. *Let Π be a planning task with operators $\mathcal{O}$ and cost function cost, and let $P = \langle \mathit{cost}_1, \ldots, \mathit{cost}_n \rangle$ be a* general cost partitioning *for Π, i. e., $\mathit{cost}_1, \ldots, \mathit{cost}_n$ are general cost functions with $\sum_{i=1}^n \mathit{cost}_i(o) \leq \mathit{cost}(o)$ for all $o \in \mathcal{O}$.*

Let $h_1, \ldots, h_n$ be admissible heuristics for Π. Then $h_P(h_1, \ldots, h_n, s) = \sum_{i=1}^n h_i(s, \mathit{cost}_i)$ is an admissible heuristic estimate for every state s. If any term in the sum is ∞, the sum is defined as ∞, even if another term is $-\infty$.

Proof: Consider an arbitrary state s. If Π has no s-plan then $h^*(s) = \infty$ and every estimate is admissible.

We are left with the case where Π has an s-plan. Then all $h_i(s, \mathit{cost}_i)$ are finite or $-\infty$ because an admissible heuristic can only produce ∞ for states without plans, no matter what the cost function is. Consider the case where $h^*(s) > -\infty$. Let $\pi = \langle o_1, \ldots, o_k \rangle$ be an optimal s-plan for Π. We get:

$$
\begin{aligned}
h_P(h_1, \ldots, h_n, s) &= \sum_{i=1}^n h_i(s, \mathit{cost}_i) \leq \sum_{i=1}^n h^*(s, \mathit{cost}_i) \\
&\leq \sum_{i=1}^n \sum_{j=1}^k \mathit{cost}_i(o_j) = \sum_{j=1}^k \sum_{i=1}^n \mathit{cost}_i(o_j) \\
&\leq \sum_{j=1}^k \mathit{cost}(o_j) = h^*(s).
\end{aligned}
$$

In the case where $h^*(s) = -\infty$, there exists a state s' on a path from s to a goal state with an incident negative-cost cycle π', i. e., π' with $s'[\![\pi']\!] = s'$ and $\mathit{cost}(\pi') < 0$. From the cost partitioning property, we get $\sum_{i=1}^n \mathit{cost}_i(\pi') \leq \mathit{cost}(\pi') < 0$, and hence $\mathit{cost}_j(\pi') < 0$ for at least one $j \in \{1, \ldots, n\}$. This implies $h^*(s, \mathit{cost}_j) = -\infty$ and hence $h_P(h_1, \ldots, h_n, s) = -\infty$, concluding the proof. □

General cost partitioning can result in negative heuristic values even if the original task uses only non-negative operator costs, but of course for such tasks it is always admissible to use the heuristic $h = \max(h_P, 0)$ instead.

Figure 1 illustrates the utility of general cost partitioning with a small example. The depicted task Π has two binary variables V_1 and V_2, both initially 0, and the goal is to set V_1 to 1. Both operators o_1 and o_2 have cost 1. The projections to V_1 and V_2 above and to the left of Π serve as two example abstractions of Π. Since abstractions preserve transitions their abstract goal distances induce admissible heuristics. If we want to add the heuristics for Π^{V_1} and Π^{V_2} admissibly, we have to find a cost partitioning $P = \langle \mathit{cost}_1, \mathit{cost}_2 \rangle$. If we

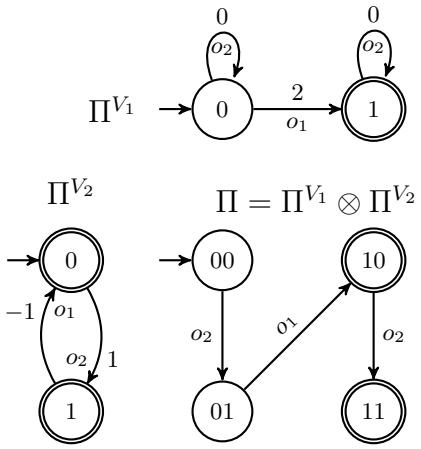

Figure 1: Example task Π with two binary variables V_1 and V_2 and goal $s_\star[V_1] = 1$. Above and to the left of the original task we show the projections to V_1 and V_2.

only consider non-negative cost partitionings it makes sense to use the full costs in $cost_1$ because V_2 is not a goal variable and therefore all abstract goal distances in Π^{V_2} are 0. This yields $h_P(h^{V_1}, h^{V_2}, s_\mathrm{I}) = h^{V_1}(s_\mathrm{I}, cost_1) + h^{V_2}(s_\mathrm{I}, cost_2) = cost(o_1) + 0 = 1$.

If we allow negative cost partitionings, however, we can assign a cost of -1 to o_1 in Π^{V_2}, allowing us to increase its costs to 2 in Π^{V_1}. The resulting cost partitioning shown in Figure 1 yields $h_P(h^{V_1}, h^{V_2}, s_\mathrm{I}) = h^{V_1}(s_\mathrm{I}, cost_1) + h^{V_2}(s_\mathrm{I}, cost_2) = 2 + 0 = 2$, a perfect heuristic estimate for the example.

An *optimal cost partitioning* is one that achieves the highest heuristic value for the given heuristics and state:

Definition 2. *Let $\Pi = \langle \mathcal{V}, \mathcal{O}, s_\mathrm{I}, s_\star, cost \rangle$ be a planning task and let $\mathcal{P}_n$ be the set of general cost partitionings for Π with n elements. The set of* optimal general cost partitionings *for admissible heuristics $h_1, \ldots, h_n$ in a state s is*

$$\mathrm{OCP}(h_1, \ldots, h_n, s) = \arg\max_{P \in \mathcal{P}_n} h_P(h_1, \ldots, h_n, s)$$

The optimal general cost partitioning heuristic estimate *for admissible heuristics $h_1, \ldots, h_n$ in a state s is*

$$h^{\mathrm{OCP}}(h_1, \ldots, h_n, s) = \max_{P \in \mathcal{P}_n} h_P(h_1, \ldots, h_n, s).$$

In general, there can be infinitely many optimal cost partitionings. Analogously to OCP and h^{OCP}, we define the non-negative variants OCP+ and $h^{\mathrm{OCP+}}$ based on the set $\mathcal{P}_n^+$ of non-negative cost partitionings for Π with n elements.

We emphasize that knowing the state s is critical for determining an optimal cost partitioning. In general, there exists no single cost partitioning that is optimal for all states.

Connection to Operator-Counting Constraints

We now study the connection between general cost partitioning and certain heuristics based on linear programming, originally introduced by Pommerening et al. (2014b).

Consider a set C of linear inequalities over non-negative *operator-counting* variables of the form Count_o for each operator o. An operator sequence π *satisfies* C if the inequalities in C are satisfied by setting each variable Count_o to the number of occurrences of o in π. If s is a state such that C is satisfied by every s-plan, C is called an *operator-counting constraint* for s. Examples of operator-counting constraints include landmark constraints, net change constraints, post-hoc optimization constraints and optimal cost partitioning constraints (Pommerening et al. 2014b).[1]

We write the inequalities of an operator-counting constraint C as $coeffs(C)\,\mathbf{Count} \geq bounds(C)$ for a coefficient matrix $coeffs(C)$, operator-counting variable vector $\mathbf{Count}$ and a bounds vector $bounds(C)$.

If $\mathcal{C}$ is a set of operator-counting contraints for a state s and $cost'$ is a cost function, then $\mathrm{IP}_\mathcal{C}(cost')$ denotes the following integer program: minimize $\sum_{o \in \mathcal{O}} cost'(o)\mathsf{Count}_o$ subject to $\mathcal{C}$ and $\mathbf{Count} \geq \mathbf{0}$. The objective value of this integer program, which we denote by $h^{\mathrm{IP}}_\mathcal{C}(cost')$, is an admissible heuristic estimate for s under cost function $cost'$. We write $\mathrm{LP}_\mathcal{C}(cost')$ and $h^{\mathrm{LP}}_\mathcal{C}(cost')$ for the LP relaxation of the integer program and the objective value of this LP, which is also an admissible heuristic estimate for s under $cost'$. (Note that even though our notations omit the state s, the constraints $\mathcal{C}$ of course generally depend on the state s.)

It turns out that there is an intimate link between operator-counting constraints and general cost partitioning. Given a set of operator-counting constraints, its LP heuristic estimate equals the optimal general cost partitioning over the LP heuristics for each *individual* constraint:

Theorem 2. *Let $\mathcal{C}$ be a set of operator-counting constraints for a state s. Then*

$$h^{\mathrm{LP}}_\mathcal{C}(cost) = h^{\mathrm{OCP}}((h^{\mathrm{LP}}_{\{C\}})_{C \in \mathcal{C}}, s).$$

Proof sketch (full proof in Pommerening et al. 2014a): In $\mathrm{LP}_\mathcal{C}(cost)$ we can introduce *local copies* LCount^C_o of Count_o for every constraint $C \in \mathcal{C}$ by adding equations $\mathsf{LCount}^C_o = \mathsf{Count}_o$ for all $o \in \mathcal{O}$. We then replace all occurrences of Count_o in this constraint by LCount^C_o.

The resulting LP minimizes $\sum_{o \in \mathcal{O}} cost(o)\mathsf{Count}_o$ subject to $coeffs(C)\mathbf{LCount}^C \geq bounds(C)$, $\mathbf{LCount}^C = \mathbf{Count}$, $\mathbf{LCount}^C \geq \mathbf{0}$ for all $C \in \mathcal{C}$, and $\mathbf{Count} \geq \mathbf{0}$. The dual of this LP has the same objective value and contains one non-negative variable Dual^C_i for each inequality and one unbounded variable Cost^C_o for each equation:

$$\text{Maximize} \sum_{C \in \mathcal{C}} bounds(C) \cdot \mathbf{Dual}^C \text{ subject to}$$
$$\left.\begin{aligned} coeffs(C)^\top \mathbf{Dual}^C &\leq \mathbf{Cost}^C \\ \mathbf{Dual}^C &\geq \mathbf{0} \end{aligned}\right\} \text{ for all } C \in \mathcal{C}$$
$$\sum_{C \in \mathcal{C}} \mathsf{Cost}^C_o \leq cost(o) \text{ for all } o \in \mathcal{O}$$

[1] Operator-counting constraints may also use auxiliary variables, ignored here for simplicity. We refer to a technical report (Pommerening et al. 2014a) for detailed proofs also covering this case.

The first two inequalities are exactly the dual constraints of $\text{LP}_{\{C\}}(\mathbf{Cost}^C)$, and the objective function is exactly the sum of dual objective functions for these LPs. The remaining inequality ensures that $\mathbf{Cost}^C$ defines a general cost partitioning. As we maximize the sum of individual heuristic values over all possible general cost partitionings, the result is the optimal general cost partitioning of the component heuristics. □

The proof is constructive in the sense that it shows a way to compute an optimal cost partitioning for the given state from a dual solution of the original LP heuristic.

Theorem 3. *Let $\mathcal{C}$ be a set of operator-counting constraints for a state s. Let d be a dual solution of $\text{LP}_{\mathcal{C}}(cost)$. Then the general cost partitioning*

$$cost^C(o) = coeffs(C)^\top d(\mathbf{Dual}^C)$$

is optimal for the heuristics $h^{\text{LP}}_{\{C\}}$ for $C \in \mathcal{C}$ in state s.

Proof: Every optimal solution of the LP in the proof for Theorem 2 contains an optimal cost partition in the variables Cost^C_o. If we take an optimal solution and replace the value of Cost^C_o with $coeffs(C)^\top d(\mathbf{Dual}^C)$, no value of Cost^C_o can increase. All inequalities are still satisfied and the objective value does not change. □

Net-Change Constraints

We now turn to the state equation heuristic h^{SEQ} (Bonet and van den Briel 2014) as an application of Theorem 2. Pommerening et al. (2014b) showed that this heuristic dominates the optimal non-negative cost-partitioning over projections to goal variables. We use their formalization as lower bound net-change constraints that compare the number of times a fact is produced to the number of times it is consumed on the path from a given state s to the goal. For each fact $\langle V, v\rangle$, we obtain one inequality:

$$\sum_{\substack{o \text{ always} \\ \text{produces } \langle V,v\rangle}} \mathsf{Count}_o + \sum_{\substack{o \text{ sometimes} \\ \text{produces } \langle V,v\rangle}} \mathsf{Count}_o - \sum_{\substack{o \text{ always} \\ \text{consumes } \langle V,v\rangle}} \mathsf{Count}_o \geq 1\{\text{if } s_\star[V] = v\} - 1\{\text{if } s[V] = v\}$$

An operator-counting constraint can consist of any number of linear inequalities. For our purposes, it is useful to group the inequalities for all facts that belong to the same state variable V into one constraint C_V. The state equation heuristic is then the LP heuristic for the set $\mathcal{C} = \{C_V \mid V \in \mathcal{V}\}$ consisting of one constraint for each state variable.

To apply Theorem 2, we must first understand which heuristics are defined by each *individual* constraint C_V and arbitrary cost function $cost'$.

Theorem 4. *Let Π be a planning task, and let V be one of its state variables. Let h^V denote the* atomic abstraction heuristic *based on projecting each state s to $s[V]$. Then*

$$h^{\text{LP}}_{\{C_V\}}(cost') = h^V(s, cost')$$

Proof: Consider the planning task Π^V obtained by projecting Π to V (i. e., discarding all aspects of its description related to other variables). It is easy to see that $h^{\text{LP}}_{\{C_V\}}(cost')$ corresponds to h^{SEQ} with cost function $cost'$ in Π^V and that $h^V(s, cost')$ in the original task is $h^*(s, cost')$ in Π^V.

The "$\leq$" part of the proof then follows from the admissibility of h^{SEQ} for Π^V. For the "$\geq$" part, Pommerening et al. (2014b) showed that h^{SEQ} dominates the optimal cost partitioning over atomic abstraction heuristics for goal variables. It can easily be verified that their proof also works for the case where negative costs are permitted. If V is a goal variable and we apply this result to Π^V, this optimal cost partitioning is a (trivial) partitioning over a single heuristic h^V and hence equal to $h^V(s, cost')$. If V is a non-goal variable, a slight adaptation is needed, for which we refer to the technical report (Pommerening et al. 2014a). □

We can now specify the connection between the state equation heuristic and abstraction heuristics more precisely:

Theorem 5. *The state equation heuristic is the optimal general cost partitioning of projections to all single variables,*

$$h^{\text{SEQ}}(s) = h^{\text{OCP}}((h^V)_{V\in\mathcal{V}}, s).$$

Proof: We apply Theorem 2 with the set of constraints $\mathcal{C} = \{C_V \mid V \in \mathcal{V}\}$. Theorem 4 establishes the connection to projections and Pommerening et al. (2014b) show that $h^{\text{SEQ}}(s) = h^{\text{LP}}_{\mathcal{C}}(cost)$. □

This answers a question raised by Bonet (2013): how does h^{SEQ} relate to the four families of heuristics identified by Helmert and Domshlak (2009)? We now see that we can interpret it as an abstraction heuristic within the framework of general cost partitioning. We also note that with Theorem 3 we can extract an optimal cost partitioning from the dual solution of the LP for h^{SEQ}.

The concept of *fluent merging* (van den Briel, Kambhampati, and Vossen 2007; Seipp and Helmert 2011) shows an interesting connection to projections to more than one variable. Merging state variables (fluents) X and Y means introducing a new state variable Z that captures the joint behaviour of X and Y. Theorem 4 shows that the LP heuristic for net change constraints of variable Z is perfect for the projection to Z and thus also for the projection to $\{X, Y\}$.

Operator-counting constraints for merging a set of facts $\mathcal{M}$ of two variables were introduced by Bonet and van den Briel (2014). They show that these constraints have perfect information for the merged variable if $\mathcal{M} = dom(X) \times dom(Y)$. We can now see that these constraints define the projection to two variables. Bonet and van den Briel further suggest the use of mutexes and note a resemblance of the corresponding abstraction heuristics to *constrained* pattern databases (Haslum, Bonet, and Geffner 2005). We conjecture that partial merges correspond to further abstractions of these projections. If this conjecture holds, Theorem 2 shows that the state equation heuristic extended with constraints for merged variables calculates a general cost partitioning over the corresponding abstractions.

Potential Heuristics

In this section we introduce a new family of heuristics called *potential heuristics* and show their relation to general cost partitioning. Potential heuristics associate a numerical *potential* with each fact. The heuristic value for a state s is then simply the sum of potentials of all facts in s.

Definition 3. *Let Π be a planning task with variables $\mathcal{V}$ and facts $\mathcal{F}$. A* potential function *is a function* $\mathrm{pot} : \mathcal{F} \to \mathbb{R}$. *The* potential heuristic for pot *maps each state to the sum of potentials of the facts of s:*

$$h^{\mathrm{pot}}(s) = \sum_{V \in \mathcal{V}} \mathrm{pot}(\langle V, s[V] \rangle)$$

We simplify our presentation by assuming that all variables in the effect of an operator o also occur in its precondition ($\mathit{vars}(\mathit{eff}(o)) \subseteq \mathit{vars}(\mathit{pre}(o))$) and there is a unique goal state ($\mathit{vars}(s_\star) = \mathcal{V}$). The definitions and proofs are generalized to arbitrary tasks in the technical report (Pommerening et al. 2014a).

A potential heuristic h^{pot} is goal-aware if and only if $h^{\mathrm{pot}}(s_\star) = \sum_{V \in \mathcal{V}} \mathrm{pot}(\langle V, s_\star[V] \rangle) \leq 0$. It is consistent if and only if $h^{\mathrm{pot}}(s) \leq \mathit{cost}(o) + h^{\mathrm{pot}}(s[\![o]\!])$ for every state s and every operator o applicable in s. This condition can be simplified as follows because the potentials of all facts not changed by an effect cancel out:

$$\begin{aligned}
\mathit{cost}(o) &\geq \sum_{V \in \mathcal{V}} \mathrm{pot}(\langle V, s[V] \rangle) - \sum_{V \in \mathcal{V}} \mathrm{pot}(\langle V, s[\![o]\!][V] \rangle) \\
&= \sum_{V \in \mathit{vars}(\mathit{eff}(o))} \mathrm{pot}(\langle V, s[V] \rangle) - \sum_{V \in \mathit{vars}(\mathit{eff}(o))} \mathrm{pot}(\langle V, s[\![o]\!][V] \rangle) \\
&= \sum_{V \in \mathit{vars}(\mathit{eff}(o))} (\mathrm{pot}(\langle V, \mathit{pre}(o)[V] \rangle) - \mathrm{pot}(\langle V, \mathit{eff}(o)[V] \rangle))
\end{aligned}$$

The resulting inequality is no longer state-dependent and is *necessary and sufficient* for the consistency of h^{pot}.

Goal-aware and consistent potential heuristics can thus be compactly classified by a set of linear inequalities. Goal-aware and consistent heuristics are also admissible, so we can use an LP solver to optimize any linear combination of potentials and transform the solution into an admissible and consistent potential heuristic.

Definition 4. *Let f be a solution to the following LP:*

Maximize opt *subject to* $\sum_{V \in \mathcal{V}} \mathsf{P}_{\langle V, s_\star[V] \rangle} \leq 0$ *and* $\sum_{V \in \mathit{vars}(\mathit{eff}(o))} (\mathsf{P}_{\langle V, \mathit{pre}(o)[V] \rangle} - \mathsf{P}_{\langle V, \mathit{eff}(o)[V] \rangle}) \leq \mathit{cost}(o)$ *for all $o \in \mathcal{O}$, where the objective function* opt *can be chosen arbitrarily.*

Then the function $\mathrm{pot}_{\mathrm{opt}}(\langle V, v \rangle) = f(\mathsf{P}_{\langle V, v \rangle})$ *is the* potential function optimized for opt *and* $h^{\mathrm{pot}}_{\mathrm{opt}}$ *is the* potential heuristic optimized for opt.

Proposition 2. *For any objective function* opt*, the heuristic $h^{\mathrm{pot}}_{\mathrm{opt}}$ is admissible and consistent, and it maximizes* opt *among all admissible and consistent potential heuristics.*

As an example, we consider the potential heuristic optimized for the heuristic value of the initial state:

$$\mathrm{opt}_{s_{\mathrm{I}}} = \sum_{V \in \mathcal{V}} \mathsf{P}_{\langle V, s_{\mathrm{I}}[v] \rangle}$$

It turns out that this heuristic is closely linked to the heuristics we discussed in the preceding sections:

Proposition 3. $h^{\mathrm{pot}}_{\mathrm{opt}_{s_{\mathrm{I}}}}(s_{\mathrm{I}}) = h^{\mathrm{SEQ}}(s_{\mathrm{I}})$.

Proof sketch (full proof in Pommerening et al. 2014a): The linear programs solved for $h^{\mathrm{pot}}_{\mathrm{opt}_{s_{\mathrm{I}}}}(s)$ and the state equation heuristic in the initial state are each other's duals. This can be seen with the substitution $\mathsf{P}_{\langle V, v \rangle} = \mathsf{X}_{\langle V, s_\star[V] \rangle} - \mathsf{X}_{\langle V, v \rangle}$ where the non-negative variables $\mathsf{X}_{\langle V, v \rangle}$ are the dual variables of the state equation heuristic LP. □

Together with Theorem 5, it follows that the potential heuristic approach offers an alternative way of performing an optimal general cost partitioning for single-variable abstraction heuristics. Compared to previous cost-partitioning approaches, the linear programs in Definition 4 are extremely simple in structure and very easy to understand. Our discussion also gives us a better understanding of what these heuristics compute: the *best possible* (for a given optimization function) admissible and consistent heuristic that can be represented as a weighted sum of indicator functions for the facts of the planning task.

Using the state equation heuristic requires solving an LP for every state evaluated by a search algorithm. It offers the best possible potential heuristic value for every single state. Alternatively, we can trade off accuracy for computation time by only computing *one* potential function (for example optimized for the initial state) and then using the induced, very quickly computable heuristic for the complete search. We will investigate this idea experimentally in the following section.

Evaluation

We implemented the state equation heuristic and the potential heuristic that optimizes the heuristic value of the initial state in the Fast Downward planning system (Helmert 2006). All our experiments were run on the set of tasks from optimal tracks of IPC 1998–2011, limiting runtime to 30 minutes and memory usage to 2 GB. Each task ran on a single core of an Intel Xeon E5-2660 processor (2.2 GHz). Linear programs were solved with IBM's CPLEX v12.5.1.

Optimal Cost Partitioning for Projections

Previous experiments (Pommerening et al. 2014b) showed that the state equation heuristic (h^{SEQ}) outperforms the optimal non-negative cost partitioning of projections to goal variables ($h^{\mathrm{OCP+}}_{\mathrm{Goal}_1}$). To explain this difference we compare these two heuristics to the optimal cost partitioning over all projections to single variables using non-negative ($h^{\mathrm{OCP+}}_{\mathrm{All}_1}$) and general ($h^{\mathrm{OCP}}_{\mathrm{All}_1}$) cost partitioning. The heuristics $h^{\mathrm{OCP}}_{\mathrm{All}_1}$ and h^{SEQ} compute the same value, but the encoding of $h^{\mathrm{OCP}}_{\mathrm{All}_1}$ is much larger because it contains one constraint for each transition of each projection and one variable for each abstract state. Likewise, the heuristics $h^{\mathrm{OCP+}}_{\mathrm{Goal}_1}$ and $h^{\mathrm{OCP+}}_{\mathrm{All}_1}$ compute the same value because non-goal variables cannot contribute to the heuristic with non-negative cost partitioning.

Table 1 shows the number of solved tasks (coverage) for the tested configurations. If only non-negative costs are considered ($h^{\mathrm{OCP+}}_{\mathrm{All}_1}$), using all variables is useless and reduces

	non-negative costs	general costs
Singleton goal patterns	$h^{\text{OCP+}}_{\text{Goal}_1}$: 500	$h^{\text{OCP}}_{\text{Goal}_1}$: 505
All singleton patterns	$h^{\text{OCP+}}_{\text{All}_1}$: 442	$h^{\text{OCP}}_{\text{All}_1}$: 490 h^{SEQ}: 630

Table 1: Coverage for different variants of optimal cost partitioning.

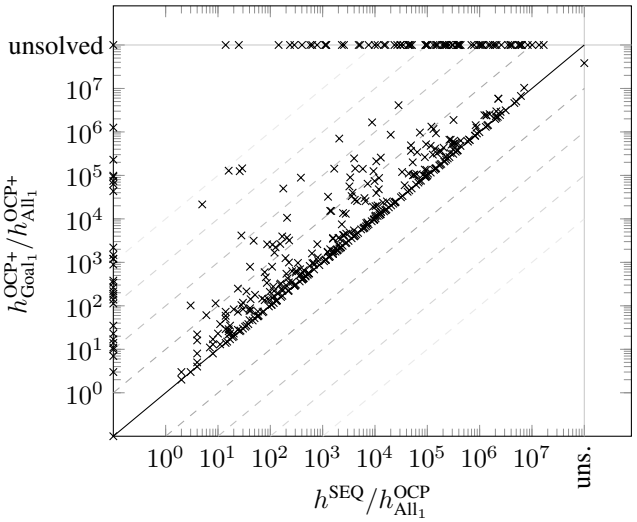

Figure 2: Number of expansions (excluding last f-layer) for optimal cost partitioning of projections to single variables with non-negative and general costs. Points above the diagonal represent tasks for which general operator cost partitioning needs fewer expansions.

coverage in 22 out of 44 domains and from 500 to 442 solved tasks in total. Allowing negative costs ($h^{\text{OCP}}_{\text{All}_1}$) recovers most, but not all of this loss. The LPs for general cost partitioning are harder to solve because more interaction with non-goal variables is possible. This decreases coverage by 1 in six domains and by 2 in one domain. On the other hand, general cost partitioning results in fewer expansions before the last f-layer in 25 domains (not shown), which results in better coverage in 13 of them. Compared to $h^{\text{OCP+}}_{\text{Goal}_1}$ using general operator cost partitioning and projections to all variables does not pay off overall because the resulting LPs get too large. A notable exception are the domains FreeCell and ParcPrinter where $h^{\text{OCP}}_{\text{All}_1}$ frequently calculates the perfect heuristic values for the initial state. Figure 2 shows that using general operator cost partitioning substantially reduces the number of expansions.

Due to the smaller LP size h^{SEQ} solves more tasks than $h^{\text{OCP}}_{\text{All}_1}$ in 39 domains, an overall coverage increase of 140. This shows that the main reason for h^{SEQ}'s superior performance is the more compact representation, though the general cost partition also plays an important role, as the comparison of $h^{\text{OCP+}}_{\text{All}_1}$ and $h^{\text{OCP}}_{\text{All}_1}$ shows.

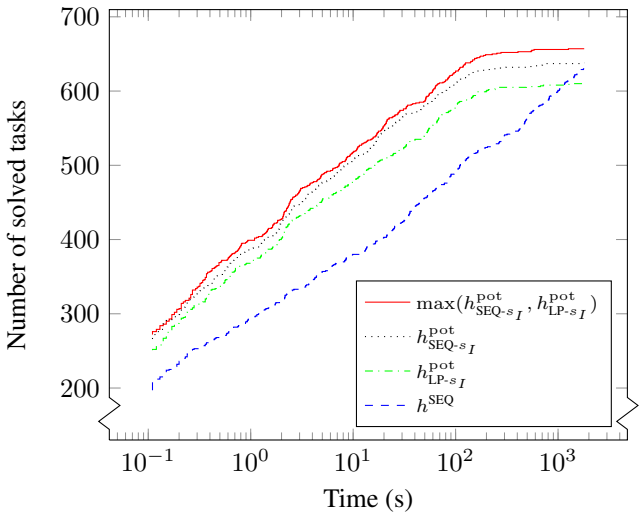

Figure 3: Number of solved tasks per time for the potential and the state equation heuristic.

Potential Heuristics

We experimented with two ways to compute potential heuristics optimized for the initial state. One solves the linear program in Definition 4 ($h^{\text{pot}}_{\text{LP-}s_I}$) directly, while the other computes the state equation heuristic and extracts the potentials from its dual using Theorem 3 ($h^{\text{pot}}_{\text{SEQ-}s_I}$). Both ways are guaranteed to result in the same heuristic estimate for the initial state, but since there are usually infinitely many optimal potential functions for a given state, they can differ widely on other states.

The heuristic $h^{\text{pot}}_{\text{SEQ-}s_I}$ has higher coverage than $h^{\text{pot}}_{\text{LP-}s_I}$ in 17 domains and lower coverage in 4 domains. Overall, the former solves 638 tasks and the latter 610 tasks. Since potential heuristics are fast to compute, we exploited their complementary strengths by evaluating the maximum of both heuristics, which solves all tasks solved by either of them except two. This results in a coverage of 657 tasks, a large improvement over the state equation heuristic which optimizes the cost partitioning for every state.

The main advantage of potential heuristics is that they are extremely fast to evaluate. This can be seen in Figure 3, which compares the number of tasks that could be solved within a given time by the state equation heuristic and the potential heuristics. With the maximum of both potential heuristics 600 tasks are solved in the first minute, compared to 17 minutes to solve 600 tasks for h^{SEQ}.

Conclusions

We showed that the traditional restriction to non-negative cost partitioning is not necessary and that heuristics can benefit from permitting operators with negative cost. In addition, we demonstrated that heuristics based on operator-counting constraints compute an optimal general cost partitioning. This allows for a much more compact representa-

tion of cost-partitioning LPs, and we saw that the state equation heuristic can be understood as such a compact form of expressing an optimal cost partitioning over projections. We believe that an extension to other cost-partitioned heuristics is a promising research direction for future work.

We also introduced potential heuristics as a fast alternative to optimal cost partitioning. By computing a heuristic parameterization only once and sticking with it, they obtain very fast state evaluations, leading to significant coverage increases and a more than 10-fold speedup over the state equation heuristic. We think that the introduction of potential heuristics opens the door for many interesting research avenues and intend to pursue this topic further in the future.

Acknowledgments

This work was supported by the Swiss National Science Foundation (SNSF) as part of the project "Abstraction Heuristics for Planning and Combinatorial Search" (AHPACS).

References

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

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.

Edelkamp, S. 2006. Automated creation of pattern database search heuristics. In *Proceedings of the 4th Workshop on Model Checking and Artificial Intelligence (MoChArt 2006)*, 35–50.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2):100–107.

Haslum, P.; Botea, A.; Helmert, M.; Bonet, B.; and Koenig, S. 2007. Domain-independent construction of pattern database heuristics for cost-optimal planning. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI 2007)*, 1007–1012. AAAI Press.

Haslum, P.; Bonet, B.; and Geffner, H. 2005. New admissible heuristics for domain-independent planning. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI 2005)*, 1163–1168. AAAI Press.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In Gerevini, A.; Howe, A.; Cesta, A.; and Refanidis, I., eds., *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS 2009)*, 162–169. AAAI Press.

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

Karpas, E., and Domshlak, C. 2009. Cost-optimal planning with landmarks. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, 1728–1733.

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.

Korf, R. E., and Felner, A. 2002. Disjoint pattern database heuristics. *Artificial Intelligence* 134(1–2):9–22.

Pommerening, F.; Helmert, M.; Röger, G.; and Seipp, J. 2014a. From non-negative to general operator cost partitioning: Proof details. Technical Report CS-2014-005, University of Basel, Department of Mathematics and Computer Science.

Pommerening, F.; Röger, G.; Helmert, M.; and Bonet, B. 2014b. LP-based heuristics for cost-optimal planning. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS 2014)*, 226–234. AAAI Press.

Pommerening, F.; Röger, G.; and Helmert, M. 2013. Getting the most out of pattern databases for classical planning. In Rossi, F., ed., *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, 2357–2364.

Seipp, J., and Helmert, M. 2011. Fluent merging for classical planning problems. In *ICAPS 2011 Workshop on Knowledge Engineering for Planning and Scheduling*, 47–53.

van den Briel, M.; Kambhampati, S.; and Vossen, T. 2007. Fluent merging: A general technique to improve reachability heuristics and factored planning. In *ICAPS 2007 Workshop on Heuristics for Domain-Independent Planning: Progress, Ideas, Limitations, Challenges*.

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.