State-dependent Cost Partitionings for Cartesian Abstractions in Classical Planning (Extended Abstract)

Thomas Keller[1], Florian Pommerening[1], Jendrik Seipp[1], Florian Geißer[2], and Robert Mattmüller[2]

[1] University of Basel, Switzerland
{tho.keller, florian.pommerening, jendrik.seipp}@unibas.ch
[2] University of Freiburg, Germany
{geisserf, mattmuel}@informatik.uni-freiburg.de

Abstract. Abstraction heuristics are a popular method to guide optimal search algorithms in classical planning. *Cost partitionings* allow to sum heuristic estimates admissibly by partitioning action costs among the abstractions. We introduce *state-dependent* cost partitionings which take context information of actions into account, and show that an *optimal state-dependent cost partitioning* dominates its state-independent counterpart. We demonstrate the potential of state-dependent cost partitionings with a state-dependent variant of the recently proposed *saturated cost partitioning*, and show that it can sometimes improve not only over its state-independent counterpart, but even over the optimal state-independent cost partitioning. [3]

Keywords: AI planning, abstraction heuristics, cost partitioning, state-dependent cost partitioning

1 Introduction

Abstraction heuristics [2,14] are a popular method to guide optimal heuristic search algorithms in classical planning. Since a single abstraction often provides poor guidance, we would like to combine the information from *several* abstractions admissibly. This can be accomplished either by *maximizing* over a set of admissible heuristics, or even better, by *adding* admissible heuristics, provided that one can guarantee that the sum of heuristic values is still admissible. This can be guaranteed either by restricting oneself to additive abstractions [11,3], or by *cost partitioning* [7,8]. The latter approach counts only some fraction of the original cost of each action in each abstraction, such that the accumulated cost of each action over all abstractions does not exceed its original cost.

Interesting instances of cost partitioning include *optimal cost partitioning* that leads to highest possible accumulated costs per state, *general cost partitioning* [12] that also allows negative costs, and *saturated cost partitioning* [15],

[3] This extended abstract is based on an IJCAI 2016 paper by the same authors [9]. Full proofs can be found there and in an associated technical report [10].

where the cost partitioning is computed iteratively by "consuming" the minimum costs in each abstraction such that the costs of all shortest paths are preserved.

In this paper, we show that even more information can be extracted from a collection of abstractions if *context information* is taken into account and abstract action costs are allowed to differ from state to state. To that end, we define *state-dependent cost partitioning* and show that its optimal version dominates optimal state-independent cost partitioning. Since computing *optimal* state-dependent cost partitionings is usually infeasible, we also consider *saturated* state-dependent cost partitioning, which is cheaper to compute. Whereas saturated state-*independent* cost partitioning loses valuable information when maximizing over all transitions incurred by the same action, saturated state-*dependent* cost partitioning, where costs are consumed only in a given context, does not suffer from this loss of information.

Besides the definition of state-dependent cost partitioning, the major contribution of this paper is a complete analysis of theoretical dominance relationships between the four combinations of optimal and saturated, and state-dependent and state-independent cost partitionings.

2 Preliminaries

We consider SAS$^+$ planning tasks [1] Π with the usual components, i. e., variables $\mathcal{V}$, actions A, initial state s_I, and goal description $s_\star$. The set of states is denoted with S. Applicability of actions and action sequences to states as well as the result of their application is also defined as usual via preconditions and effects. In addition, we allow non-negative *action costs* to be specified by *cost functions* $c : A \to \mathbb{R}^+_0$. At several places in this paper, we are interested in costs that are based on modified cost functions. An important aspect of this work are *general* and *state-dependent* cost functions $c : A \times S \to \mathbb{R}$ that determine transition costs $c(a, s)$ that depend on the state s in addition to the action a that is applied. Since state-dependent cost functions are more general, we define the following concepts in terms of state-dependent instead of regular cost functions unless we want to emphasize that the cost function of the original task is used.

An action sequence $\pi = \langle a_1, \ldots, a_n \rangle$ is an s-plan if it is applicable in s and leads to a state satisfying the goal condition. It is a *plan* if it is an s_I-plan. The *cost* of s-plan π under cost function c is the sum of action costs along the induced state sequence $\langle s_0, \ldots, s_n \rangle$, i.e., $c(\pi, s) = \sum_{i=1}^{n} c(a_i, s_{i-1})$. It is *optimal* under c if it minimizes $c(\pi, s)$. A *heuristic function* h estimates the cost of an optimal s-plan under cost function c with values $h(s, c) \in \mathbb{R} \cup \{-\infty, \infty\}$. Note that we allow negative heuristic values to support general cost partitioning [12]. A heuristic h is called *admissible* if it never overestimates the true cost. A planning task Π and a cost function c induce a weighted labeled transition system $\mathcal{T}$ in the usual way. Edge weights in $\mathcal{T}$ are the (possibly state-dependent) action costs of the planning task.

The core idea of *abstraction heuristics* is to collapse several states into a single abstract state, which reduces the size of the transition system and allows the computation of abstract goal distances that can be used as admissible heuristic estimates in the original task. Given a planning task Π with induced transition system $\mathcal{T}$, we denote abstraction mappings from concrete to abstract states preserving initial state, goal states, and transitions, by α, and the induced abstract transition system by $\mathcal{T}^\alpha$. In defining the weight of an abstract transition in $\mathcal{T}^\alpha$ between abstract states t and u with transition label a, we follow Geißer et al. [4,5] and define it to be the *minimal* weight of all concrete transitions labeled with action a that start in a state s with $\alpha(s) = t$. Together with the fact that every plan in the concrete transition system is a plan in the abstract transition system, this ensures that the cost of each optimal abstract plan is an admissible heuristic estimate. Abstractions where all abstract states are Cartesian products of domain subsets of the state variables are called *Cartesian abstractions*. Since we only consider Cartesian abstractions here, we simply call them *abstractions* in the following.

3 State-dependent Cost Partitioning

Early work on additive admissible heuristics has mostly focused on techniques that allow to generate or identify heuristics that can be added up admissibly because each deals with a sub-problem of the planning task that can be regarded independently from the rest [3,6]. An equivalent view on these techniques is to regard them as cost partitionings [8] that distribute action costs such that each operator is assigned its full cost in one heuristic and a cost of zero in all other. However, cost partitionings are more general as costs can be distributed arbitrarily between the heuristics as long as the sum over the individual costs does not exceed the original cost. Given such a cost partitioning, heuristic values are then computed on a copy of the planning task where actions cost only the fraction of the actual action cost that is assigned to the heuristic. In this paper, we continue developing more accurate cost partitioning techniques by presenting state-dependent cost partitionings, a generalization where context information of applied actions is taken into account.

Definition 1 (State-dependent cost partitioning). *Let Π be a planning task. Then a general* state-dependent cost partitioning *for Π is a tuple $P = \langle c_1 \dots, c_n \rangle$, where $c_i : A \times S \to \mathbb{R}$ for $1 \leq i \leq n$ and $\sum_{i=1}^{n} c_i(a,s) \leq c(a)$ for all $s \in S$ and $a \in A$. If P is* state-independent*, i.e., if $c_i(a,s) = c_i(a,s')$ for all $s, s' \in S$, $a \in A$ and $1 \leq i \leq n$, then P is a general* state-independent cost partitioning *for Π.*

Let $h_1, \dots, h_n$ be admissible heuristics and $P = \langle c_1 \dots, c_n \rangle$ a cost partitioning. Then the corresponding *cost partitioning heuristic* is denoted as $h_P(s) = \sum_{i=1}^{n} h_i(s, c_i)$, where the sum is defined as ∞ if any term in the sum is ∞, even if another term is $-\infty$. We want to point out that the introduction of state-dependent cost functions does not break admissibility of h_P.

State-dependent cost partitionings differ from their state-independent counterpart in that each state-action pair can have its own cost instead of a cost that is shared among all possible applications of an action.

Definition 2 ($\mathbf{OCP_D}$ and $\mathbf{OCP_I}$). *Let $h_1, \ldots, h_n$ be admissible heuristics for a planning task Π, $\mathbb{P}_D$ the space of state-dependent cost partitionings and $\mathbb{P}_I \subseteq \mathbb{P}_D$ the space of state-independent cost partitionings for Π. The* optimal state-dependent cost partitioning ($\mathrm{OCP_D}$) *heuristic estimate for $h_1, \ldots, h_n$ in state s is $h^{ocp_D}(s) = \max_{P \in \mathbb{P}_D} h_P(s)$, and the* optimal state-independent cost partitioning ($\mathrm{OCP_I}$) *heuristic estimate for $h_1, \ldots, h_n$ is $h^{ocp_I}(s) = \max_{P \in \mathbb{P}_I} h_P(s)$.*

State-dependent cost partitionings allow the computation of more accurate heuristics estimates.

Theorem 1 ($\mathbf{OCP_D}$ dominates $\mathbf{OCP_I}$). *Let $h_1, \ldots, h_n$ be admissible heuristics for a planning task Π. Then $h^{ocp_D}(s) \geq h^{ocp_I}(s)$ for all $s \in S$. Moreover, there are planning tasks where the inequality is strict for at least one state.* □

Although Theorem 1 provides an encouraging result, its practical impact appears limited. This is mostly because the computation of an optimal state-dependent cost partitioning with a method designed for state-independent cost partitionings [8,12] would require a compilation with one action for each state-action pair, a number that is exponential in the number of state variables. Whereas there are techniques like context splitting [13] that allow to compute a more compact compilation, the worst-case exponential blowup cannot be avoided in general. We therefore turn our attention to saturated cost partitioning [15], a technique that is tractable in practice.

4 Saturated Cost Partitioning

Seipp and Helmert [15] introduced the concept of *cost saturation*. Iteratively, they compute an abstraction, reduce the action costs such that all goal distances are preserved, and use the remaining costs for subsequent abstractions. The result is known as a *saturated cost partitioning*. Due to the greediness of the procedure, the resulting cost partitioning usually provides poorer estimates than the optimal cost partitioning. However, we can compute the saturated cost partitioning much faster and more memory-efficiently. Following Seipp and Helmert [15] and extending their definition to potentially negative, but still state-independent action cost, we can define saturated state-independent cost partitioning as follows.

Definition 3 ($\mathbf{SCP_I}$). *Let Π be a planning task with cost function c and $\alpha_1, \ldots, \alpha_n$ abstractions. Let $\langle c_1, \ldots, c_n \rangle$ and $P = \langle \hat{c}_1, \ldots, \hat{c}_n \rangle$ be tuples of cost functions with the following properties: $c_1 = c$; $\hat{c}_i(a) = \max_{s \in S} h_i(\alpha_i(s)) - h_i(\alpha_i(s[a]))$, where h_i is the goal distance function of $\mathcal{T}^{\alpha_i}$ with cost function c_i; and $c_{i+1} = c_i - \hat{c}_i$. We call c_i the* remaining cost *for $\mathcal{T}^{\alpha_i}$, $\hat{c}_i$ the* saturated cost *of $\mathcal{T}^{\alpha_i}$ and P the* saturated state-independent cost partitioning ($\mathrm{SCP_I}$) *for $\alpha_1, \ldots, \alpha_n$.*

We denote the associated heuristic by h^{scp_I}. Seipp and Helmert [15] show that the saturated cost function preserves the goal distances of all abstract states in all abstractions, and is minimal among all distance-preserving cost functions. The same holds for the potentially negative cost partitioning that we use.

As state-independent cost functions do not allow that costs are assigned to actions in the context of the current state, saturated cost functions are computed by maximizing over all weights of transitions that are labeled with the same action. State-*dependent* cost partitioning offers an opportunity to overcome this weakness by allowing to reduce the costs of state-action pairs rather than actions.

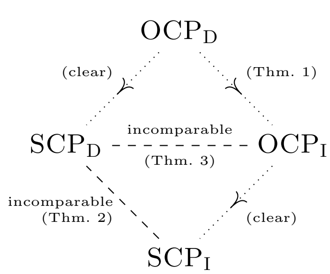
Definition 4 ($\mathbf{SCP_D}$). *Let Π be a planning task with cost function c and let $\alpha_1, \dots, \alpha_n$ be abstractions. Let $\langle c_1, \dots, c_n \rangle$ and $P = \langle \hat{c}_1, \dots, \hat{c}_n \rangle$ be tuples of cost functions with the following properties: $c_1(a, s) = c(a)$ for all $a \in A$ and $s \in S$; $\hat{c}_i(a, s) = h_i(\alpha(s)) - h_i(\alpha(s[a]))$, where h_i is the goal distance function of $\mathcal{T}^{\alpha_i}$ with cost function c_i; and $c_{i+1} = c_i - \hat{c}_i$. We call c_i the* remaining cost *for $\mathcal{T}^{\alpha_i}$, $\hat{c}_i$ the* saturated cost *of $\mathcal{T}^{\alpha_i}$ and P the* saturated state-dependent cost partitioning ($\mathrm{SCP_D}$) *for $\alpha_1, \dots, \alpha_n$.*

We denote the associated heuristic by h^{scp_D}. In analogy to Theorem 1, we might be tempted to expect a similar theoretical dominance of $\mathrm{SCP_D}$ over $\mathrm{SCP_I}$. However, it turns out that this is not the case due to the inaccuracy caused by the greediness of saturated cost partitionings.

Theorem 2 ($\mathbf{SCP_D}$ and $\mathbf{SCP_I}$ are incomparable). *There are planning tasks Π and Π' with states $s \in S$ and $s' \in S'$ such that $h^{scp_D}(s) > h^{scp_I}(s)$ and $h^{scp_I}(s') > h^{scp_D}(s')$.* □

In Theorems 1 and 2, we investigated the relationship between $\mathrm{OCP_D}$ and $\mathrm{OCP_I}$, and between $\mathrm{SCP_D}$ and $\mathrm{SCP_I}$. Dominance of $\mathrm{OCP_D}$ over $\mathrm{SCP_D}$ and of $\mathrm{OCP_I}$ over $\mathrm{SCP_I}$ is clear. What is left is comparing $\mathrm{SCP_D}$ to $\mathrm{OCP_I}$.

Theorem 3 ($\mathbf{SCP_D}$ and $\mathbf{OCP_I}$ are incomparable). *There are a planning tasks Π and Π' with states $s \in S$ and $s' \in S'$ such that $h^{scp_D}(s) > h^{ocp_I}(s)$ and $h^{ocp_I}(s') > h^{scp_D}(s')$.* □



The figure to the left shows a summary of our theoretical results (where $A \succ B$ means A dominates B). Optimal state-dependent cost partitioning combines the best of both worlds, but computing it is exponential. Saturated state-dependent cost partitioning may not always result in better heuristic estimates, but it has the potential to surpass optimal state-independent cost partitioning.

5 Conclusion

We generalized cost partitionings and showed that additional information can be extracted from a set of abstractions if context information of applied actions is taken into account. We showed that an optimal state-dependent cost partitioning dominates all state-independent cost partitionings and that there are planning tasks where the dominance is strict. As it is unclear how an optimal state-dependent cost partitioning can be computed efficiently in practice, we applied the idea to the efficiently computable saturated cost partitioning. We showed that saturated state-dependent cost partitioning does not dominate its state-independent sibling, but may still surpass optimal state-independent cost partitioning. Preliminary experimental results are generally in line with what our theoretical results suggest.

Acknowledgments. This work was supported by the European Research Council as part of the project "State Space Exploration: Principles, Algorithms and Applications" and by BMBF grant 02PJ2667 as part of the KARIS PRO project.

References

1. Bäckström, C., Nebel, B.: Complexity results for SAS$^+$ planning. Computational Intelligence 11(4), 625–655 (1995)
2. Culberson, J.C., Schaeffer, J.: Pattern databases. Computational Intelligence 14(3), 318–334 (1998)
3. Felner, A., Korf, R., Hanan, S.: Additive pattern database heuristics. JAIR 22, 279–318 (2004)
4. Geißer, F., Keller, T., Mattmüller, R.: Delete relaxations for planning with state-dependent action costs. In: Proc. IJCAI 2015. pp. 1573–1579 (2015)
5. Geißer, F., Keller, T., Mattmüller, R.: Abstractions for planning with state-dependent action costs. In: Proc. ICAPS 2016 (2016)
6. Haslum, P., Botea, A., Helmert, M., Bonet, B., Koenig, S.: Domain-independent construction of pattern database heuristics for cost-optimal planning. In: Proc. AAAI 2007. pp. 1007–1012 (2007)
7. Katz, M., Domshlak, C.: Structural patterns of tractable sequentially-optimal planning. In: Proc. ICAPS 2007. pp. 200–207 (2007)
8. Katz, M., Domshlak, C.: Optimal admissible composition of abstraction heuristics. AIJ 174(12–13), 767–798 (2010)
9. Keller, T., Pommerening, F., Seipp, J., Geißer, F., Mattmüller, R.: State-dependent cost partitionings for cartesian abstractions in classical planning. In: Proc. IJCAI 2016 (2016)
10. Keller, T., Pommerening, F., Seipp, J., Geißer, F., Mattmüller, R.: State-dependent cost partitionings for cartesian abstractions in classical planning: Full proofs. Tech. Rep. CS-2016-002, University of Basel, Switzerland (2016)
11. Korf, R.E., Felner, A.: Disjoint pattern database heuristics. AIJ 134(1–2), 9–22 (2002)
12. Pommerening, F., Helmert, M., Röger, G., Seipp, J.: From non-negative to general operator cost partitioning. In: Proc. AAAI 2015. pp. 3335–3341 (2015)
13. Röger, G., Pommerening, F., Helmert, M.: Optimal planning in the presence of conditional effects: Extending LM-Cut with context splitting. In: Proc. ECAI 2014. pp. 80–87 (2014)
14. Seipp, J., Helmert, M.: Counterexample-guided Cartesian abstraction refinement. In: Proc. ICAPS 2013. pp. 347–351 (2013)
15. Seipp, J., Helmert, M.: Diverse and additive Cartesian abstraction heuristics. In: Proc. ICAPS 2014. pp. 289–297 (2014)