Cost Partitioning Heuristics for Stochastic Shortest Path Problems

Thorsten Klößner[1], Florian Pommerening[2], Thomas Keller[2], Gabriele Röger[2]

[1]Saarland University, Saarland Informatics Campus, Germany
[2]University of Basel, Switzerland
kloessner@cs.uni-saarland.de, {florian.pommerening, tho.keller, gabriele.roeger}@unibas.ch

Abstract

In classical planning, cost partitioning is a powerful method which allows to combine multiple admissible heuristics while retaining an admissible bound. In this paper, we extend the theory of cost partitioning to probabilistic planning by generalizing from deterministic transition systems to stochastic shortest path problems (SSPs). We show that fundamental results related to cost partitioning still hold in our extended theory. We also investigate how to optimally partition costs for a large class of abstraction heuristics for SSPs. Lastly, we analyze occupation measure heuristics for SSPs as well as the theory of approximate linear programming for reward-oriented Markov decision processes. All of these fit our framework and can be seen as cost-partitioned heuristics.

Introduction

In classical planning, the aim is to determine a sequence of *deterministic* actions that leads from the given initial state to a goal state. Applying action a in state s always leads to the same successor state, which is known in advance.

Stochastic shortest path problems (SSPs) are a class of Markov decision processes that generalize the classical planning setting to *probabilistic* action outcomes: applying an action in a state can lead to different successor states. The agent cannot control the successor state but only knows the probability distribution from which it is drawn by the environment. Since successor states are not deterministic, a solution is not simply a *sequence* of actions but instead a *policy* that maps each reachable state to the action to apply.

In both settings, we are only interested in *optimal* solutions that minimize the (expected) cost accumulated before reaching the goal. In classical planning, this can be achieved by using the A^* algorithm (Hart, Nilsson, and Raphael 1968) with an *admissible* heuristic, i.e. a distance estimator that maps every state to a lower bound of its actual goal distance. For SSPs, suitable search algorithms that guarantee optimality if combined with an admissible heuristic include RTDP (Barto, Bradtke, and Singh 1995), Labeled RTDP (Bonet and Geffner 2003) and LAO* (Hansen and Zilberstein 2001).

Operator cost partitioning (Katz and Domshlak 2010) is a technique to combine multiple admissible heuristics while retaining an admissible bound. The idea is to distribute the original action costs among several cost functions, one for each heuristic. Each heuristic is then evaluated wrt. its associated cost function and the sum of these estimates is an admissible estimate wrt. the true cost function. Operator cost partitioning has been generalized to support negative costs (Pommerening et al. 2015) and to the more fine-grained *transition cost partitioning* (Keller et al. 2016).

The SSP heuristics by Trevizan, Thiébaux, and Haslum (2017) can likewise be perceived as heuristic combination: the *projection occupation measure* heuristic h^{pom} considers occupation measures from several projections to single variables and combines them in a single linear program by means of tying constraints. The *regrouped operator-counting* heuristic h^{roc} combines net-change constraints from several projections by means of regrouping constraints.

Klößner and Hoffmann (2021) transferred PDB abstraction heuristics (Korf 1997; Culberson and Schaeffer 1998) to SSPs. They identified an additivity criterion that guarantees that the sum of estimates of certain PDBs is admissible. In contrast to cost partitioning in the classical setting, this does not allow to admissibly sum up *arbitrary* PDB heuristics but requires *orthogonality* of the individual heuristics for additivity in the spirit of the work of Edelkamp (2001) and Korf and Felner (2002) in classical planning.

Approximate linear programming (ALP) (Guestrin et al. 2003) is a technique for infinite-horizon discounted-reward MDPs, which can be cast as SSPs. ALP approximates the optimal value function with a linear weighted sum of basis functions, which is structurally very similar to the *potential heuristics* (Pommerening et al. 2015) known in classical planning. Potential heuristics over indicator functions of abstract states can be related to a cost partitioning over these abstractions (Pommerening, Helmert, and Bonet 2017).

The aim of this paper is to transfer the idea of cost partitioning to SSPs and to gain a better understanding of the theoretical connections between the different SSP techniques.

To support *negative costs* in cost partitions, we suitably extend some SSP notions. We show that cost partitioning for SSPs preserves fundamental results from classical planning. In classical planning, we can efficiently compute an *optimal* cost partition for abstractions (Katz and Domshlak 2010). We establish an analogous result for a class of abstraction heuristics for SSPs. Afterwards we analyze existing SSP heuristics in the light of cost partitioning. We show that h^{pom} is an optimal cost partitioning heuristic over atomic projections and contribute to the open question whether $h^{\text{roc}} = h^{\text{pom}}$. For ALP we show an analogous relationship as known for potential heuristics and cost partitioning.

Background

Probabilistic Planning Tasks

We consider planning tasks with stochastic effects in finite-domain representation (Trevizan, Thiébaux, and Haslum 2017). A *probabilistic planning task* is given by a 5-tuple $\langle V, A, s_0, s_\star, c\rangle$, where V is a finite and non-empty set of *variables* and each variable v has a finite *domain* D_v of at least two values. W.l.o.g. we assume that all variable domains are disjoint. A *partial state* is an assignment s that maps each variable $v \in V$ to $s(v) \in D_v \cup \{\bot\}$. If $s(v) \neq \bot$ we say s is defined on v. We define $V(s)$ as the variables for which s is defined. A *state* s is a partial state with $V(s) = V$. For partial states s, t, we write $t \subseteq s$ if $s(v) = t(v)$ for all $V(t)$. The *application* of partial state e on partial state s is defined as $\mathrm{appl}(s, e)(v) = s(v)$ if $e(v) = \bot$ and $\mathrm{appl}(s, e)(v) = e(v)$ otherwise. For the finite and non-empty set A of *actions*, each $a \in A$ is associated with a *precondition* $\mathrm{pre}(a)$, which is a partial state, a finite, non-empty set $\mathrm{Eff}(a)$ of *effects*, which are partial states, and an *effect probability distribution* $\mathrm{Pr}_a : \mathrm{Eff}(a) \to [0, 1]$. State s_0 is the *initial state* and state $s_\star$ the single *goal state*. Finally, $c : S \times A \to \mathbb{R}_0^+$ is a state-dependent, non-negative action *cost function*.

Stochastic Shortest Path Problems

A *stochastic shortest path problem* (SSP, Bertsekas and Tsitsiklis 1991) is a 6-tuple $\langle S, A, T, s_0, s_\star, c\rangle$ where S is a finite and non-empty set of states, A is a finite and non-empty set of actions, $T : S \times A \times S \to [0, 1]$ is a transition probability function, $s_0 \in S$ is the initial state, $s_\star$ is the goal state, and $c : S \times A \to \mathbb{R}_0^+$ is a state-dependent action cost assignment.

For all state action pairs (s, a), either $T(s, a, t) = 0$ for all $t \in S$, or $\sum_{t\in S} T(s, a, t) = 1$ in which case a is *applicable* in s. We denote the set of actions applicable in s by $A(s)$, and the set of states in which a is applicable by $A^{-1}(a)$. We assume $A(s) \neq \emptyset$ for every s for simplicity and without loss of generality (for states without applicable actions, we can add zero-cost self loops).[1]

A probabilistic planning task $\Pi = \langle V, A, s_0, s_\star, c\rangle$ induces the SSP $\langle S, A, T, s_0, s_\star, c\rangle$ in which S are the states of Π, and the transition function T is defined by $T(s, a, t) = 0$ if $\mathrm{pre}(a) \not\subseteq s$ and otherwise by

$$T(s, a, t) := \sum_{\substack{e\in\mathrm{Eff}(a)\\ \mathrm{appl}(s,e)=t}} \mathrm{Pr}_a(e).$$

In other words, a is not applicable in s if s does not satisfy the precondition of a, otherwise $T(s, a, t)$ is the total probability of an outcome which leads to t when applied on s.

A (deterministic and stationary) *policy* $\pi : S \to A$ maps each state s to an applicable action $\pi(s) \in A(s)$. It is s-*proper* if the probability of reaching the goal $s_\star$ from s is 1 when selecting actions according to π.

The *policy state-value function* J^π maps each state s to the expected cost of reaching the goal state when following policy π. If π is not s-proper, we define $J^\pi(s) := \infty$. The policy state-value for the states $S' \subseteq S$ where π is s-proper is defined through the operator $\mathcal{B}^\pi : (S' \to \mathbb{R}) \to (S' \to \mathbb{R})$ specified as $(\mathcal{B}^\pi x)(s) := 0$ if $s = s_\star$, and otherwise as

$$(\mathcal{B}^\pi x)(s) := c(s, \pi(s)) + \sum_{t\in S} T(s, \pi(s), t) \cdot x(t).$$

This operator is a contraction mapping and thus has a unique fixpoint $\bar{x}$. For any x, we have $\lim_{k\to\infty}(\mathcal{B}^\pi)^k x = \bar{x}$. For $s \in S'$, the policy state-value is defined by $J^\pi(s) := \bar{x}(s)$.

The *optimal state-value function* is the point-wise minimum $J^\star = \min_\pi J^\pi$, which always exists. Policy π is *optimal* if $J^\pi = J^\star$. Based on the Bellman optimality equation (Bellman 1957), the optimal state-value $J^\star(\tilde{s})$ for some state of interest $\tilde{s} \in S$ can be expressed as the optimal objective value of LP 1, if we treat the optimal objective value of an unbounded linear program (LP) as ∞. The LP is always feasible, but is unbounded if there is no $\tilde{s}$-proper policy.

$$\begin{aligned}
&\text{Maximize } \mathsf{y}_{\tilde{s}} \text{ subject to} && \text{(LP 1)}\\
&\quad \mathsf{y}_{s_\star} = 0 && (1)\\
&\quad \mathsf{y}_s \leq c(s, a) + \sum_{t\in S} T(s, a, t) \cdot \mathsf{y}_t && (2)\\
&\qquad\qquad \text{for all } s \in S \text{ and } a \in A(s) &&\\
&\text{where all variables } (\mathsf{y}_s)_{s\in S} \text{ are unrestricted.} &&
\end{aligned}$$

Occasionally, we will need to refer to the values of $J^\star$ and J^π under a different cost function c', and write $J^\star_{c'}$ and $J^\pi_{c'}$.

SSPs with Negative Costs

In this paper, we also consider heuristics that are evaluated on probabilistic planning tasks with negative costs, but the SSPs defined so far only support non-negative costs. We will now discuss why this is a problem and how we generalize SSPs to general real-valued cost functions $c : S \times A \to \mathbb{R}$.

With non-negative costs, it can be shown that there always is an optimal *stationary, deterministic* policy, in contrast to *non-stationary* policies that permit taking different actions when encountering the same state several times. The intuition is that with non-negative costs one always wants to pursue the same "expected most direct path" to the goal to avoid incurring unnecessary costs. In the presence of negative costs, there can be areas of the state space where the agent can remain indefinitely, accumulating unbounded negative costs before eventually changing the behaviour to reach the goal. Depending on the cost function it might even pay off to leave the goal state, incur some negative costs and then return to the goal state, which is why we do not consider an

[1] Compared to other definitions (e.g., Guillot and Stauffer 2020), we do not require an absorbing goal state, which can be simulated by a goal state where the only applicable action is a zero-cost self loop. We discuss the reasons for this in the next section.

absorbing goal state. The expected cost of a non-stationary policy that exploits such a cycle can be made arbitrarily low.

Guillot and Stauffer (2020) also consider SSPs with negative costs but exclude SSPs with *transition cycles* with negative cost, which capture this behaviour. A transition cycle is a solution x to the probabilistic flow equations

$$\sum_{s\in S, a\in A(s)} T(s,a,t)x_{s,a} = \sum_{a\in A(t)} x_{t,a} \quad \text{for all } t\in S \qquad (3)$$

over non-negative variables $x_{s,a}$ for $s \in S$ and $a \in A(s)$, where $x_{s,a} > 0$ at least once. The equations enforce that the incoming (left) and outgoing (right) flow is balanced for all states. Given a solution x, all states s where an action a with $x_{s,a} > 0$ exists are part of a cycle, as all flow that leaves s must come back to s again. We denote the set of these states by S_x. Analogously, the actions a which transport flow for a state s are those with $x_{s,a} > 0$, denoted $A_x(s)$.

Figure 1 shows an example (ignore that parts are dashed for now) containing several transition cycles. For example $\{1, 2\}$ and $\{14, 15\}$ are the states of a transition cycle, but so are $\{3, 4, 5\}$ and $\{s_\star, 7, 10\}$. The last example shows that the states of a transition cycle do not have to be connected, because a solution may contain several disconnected flows.

A *simple transition cycle* (or just *simple cycle*) prevents this behaviour and requires (*i*) A positive amount of flow is transmitted between every two states participating in the flow exchange. In other words, the graph with vertices S_x and edges according to $A_x(s)$ for vertex s is strongly connected. (*ii*) $|A_x(s)| = 1$ for every state $s \in S_x$, so only one action may be used to forward the flow. Essentially, simple cycles capture a stationary policy π_x which chooses the unique action $a \in A_x(s)$ for all $s \in S_x$. In doing so, π_x stays in S_x forever, and is always able to reach any state in the simple cycle with positive probability.

Guillot and Stauffer forbid SSPs with *negative-cost* simple cycles, which satisfy

$$\sum_{s\in S, a\in A(s)} c(s,a)\cdot x_{s,a} < 0. \qquad (4)$$

Intuitively, this inequality tells us that the expected cost paid by π_x in the long run is negative, so π_x accumulates unbounded negative costs. Guillot and Stauffer show that one can decompose any transition cycle into simple cycles, so it is sufficient to handle simple cycles with negative cost.

This exclusion is not possible with the approaches we consider since we allow arbitrary cost functions for the same SSP. This SSP can have no negative-cost simple cycles under one cost function, but these may exist under another cost function. For this reason, we allow negative-cost simple cycles, but define $J^\star(s) := -\infty$ if there is a stationary s-proper policy that reaches (a state in) a negative-cost simple cycle with non-zero probability and $J^\star(s) := \min_\pi J^\pi(s)$ otherwise.

We define $J^{\mathrm{LP}}(\tilde{s})$ as the objective value of LP 1 if it is bounded feasible, as ∞ if it is unbounded, and as $-\infty$ if it is infeasible. When we consider a cost function c' different from c in this LP, we use the notation $J^{\mathrm{LP}}_{c'}(\tilde{s})$. Unfortunately, the characterization of $J^\star(\tilde{s})$ as the objective value of LP 1 is not correct anymore when negative-cost simple cycles are allowed, as the following theorem shows.

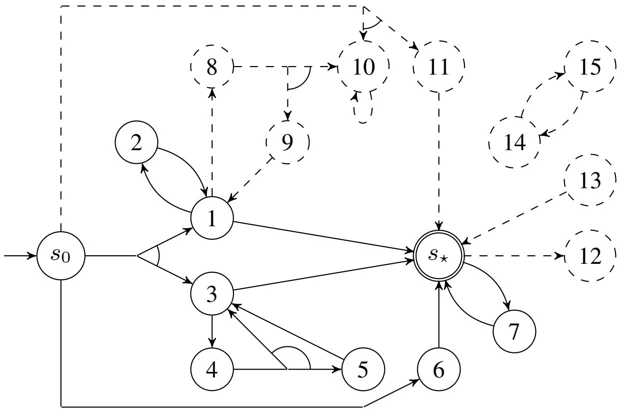

Figure 1: Example SSP with irrelevant states and actions marked with dashes.

Theorem 1 $J^{\mathrm{LP}}(\tilde{s}) = -\infty$ *if and only if there exists* any *negative-cost simple cycle.*

Proof. First of all, if there is a variable assignment y satisfying constraints (2) but not (1), then we can construct a solution $\mathsf{y}'_s := \mathsf{y}_s - \mathsf{y}_{s_\star}$ satisfying both. We can therefore ignore (1) when arguing about feasibility. Likewise, we can treat the objective as constant zero which also does not influence feasibility. Consider the dual of this modified LP:

$$\text{Minimize} \sum_{s\in S, a\in A(s)} c(s,a)\cdot \mathsf{x}_{s,a} \text{ subject to} \qquad \text{(LP 2)}$$

$$\sum_{s\in S, a\in A(s)} T(s,a,t)\mathsf{x}_{s,a} = \sum_{a\in A(t)} \mathsf{x}_{t,a} \quad \text{for all } t\in S$$

where all variables $(\mathsf{x}_{s,a})_{s\in S, a\in A(s)}$ are non-negative.

The constraints of this LP are exactly equation (3) used to define transition cycles, and the objective is exactly the left hand side of (4). This LP is feasible because setting all variables to zero is a solution. Since this dualization is feasible, LP 1 is infeasible exactly when this dualization is unbounded. It is easy to see that the above dualization is unbounded if and only if there is a negative-cost transition cycle. Guillot and Stauffer show that every negative-cost transition cycle can be decomposed into simple cycles, one of which has to have negative cost. □

This is problematic, as $J^\star(\tilde{s})$ is only equal to $-\infty$ if a negative-cost simple cycle is actually reachable from $\tilde{s}$ along a stationary $\tilde{s}$-proper policy. Fortunately, we can strengthen the formulation to recover $J^\star(\tilde{s})$ as the objective value. To this end, we define the set of states *relevant* for state s as those states which are reachable from s by a non-stationary s-proper policy. Note that we deviate from stationary policies for this definition. Likewise, we define the applicable actions of state t relevant for s as those actions that never transition to a state irrelevant for s. Figure 1 shows an example of an SSP where irrelevant states and actions for s_0 are pruned. For example, state 10 has no 10-proper policy, so it is irrelevant. While there are 11-proper and 13-proper policies, there is no s_0-proper policy that reaches those states, so they are also irrelevant. Note that there is no stationary s_0-proper policy that reaches state 7, but there is a non-stationary one, so this state is relevant.

Intuitively, relevant states can be reached with non-zero probability "on the way to the goal", i.e., there is a non-stationary policy that reaches the relevant state and reaches the goal with probability 1. Irrelevant states and irrelevant applicable actions do not affect possible solutions for $\tilde{s}$, so we can safely prune these states and transitions from the LP.

Theorem 2 *The objective value of LP 1 in which irrelevant states and irrelevant applicable actions are pruned is* $J^\star(\tilde{s})$.

Proof. Let $J^\star(\tilde{s}) = \infty$. Then there is no stationary $\tilde{s}$-proper policy. There is no non-stationary $\tilde{s}$-proper policy either, as stationary policies are sufficient in terms of goal probability maximization. No state is relevant by definition, so the pruned LP has no constraints left and must be unbounded.

Let $J^\star(\tilde{s}) = -\infty$. Then there is a stationary $\tilde{s}$-proper policy π which can reach a state t in a negative-cost simple cycle x. We can construct a non-stationary $\tilde{s}$-proper policy from π which initially uses π_x for the states in the cycle instead. Eventually, the policy reverts back to π for t so the cycle is eventually left. Therefore, all states and actions in this cycle are relevant and the LP remains infeasible.

Let $J^\star(\tilde{s}) \in \mathbb{R}$. Consider the SSP in which irrelevant states and actions are pruned. In this case, all negative-cost simple cycles are pruned, and every remaining state t must have a t-proper policy, as the state can be visited by an $\tilde{s}$-proper policy from $\tilde{s}$. For this class of SSPs, Guillot and Stauffer show that LP 1 computes $J^\star(\tilde{s})$, so it is left to argue that this SSP transformation preserves $J^\star(\tilde{s})$. By definition of relevance, any $\tilde{s}$-proper policy π can only reach relevant states from $\tilde{s}$, so the value $J^\pi(\tilde{s})$ can by definition not depend on irrelevant states or actions. □

To see how our definition deals with negative-cost simple cycles, consider the example in Figure 1 again. Assume all irrelevant states have been removed (the dashed parts) and all remaining actions have a cost of -1. Then the stationary policy π that reaches $s_\star$ via state 6 has the value $J^\pi(s_0) = -2$. However, the value $J^\star(s_0)$ is $-\infty$ because $s_\star$ is part of a negative-cost simple cycle $\{s_\star, 7\}$, so the stationary policy π, which has finite cost, could be extended to non-stationary ones of arbitrarily low costs which also reach the state $s_\star$.

Heuristics

A *heuristic* for an SSP with general costs is a function h which accepts a state $s \in S$ and a cost function c' and provides an estimate $h(s, c') \in \mathbb{R} \cup \{-\infty, \infty\}$ for $J^\star_{c'}(s)$. If c' is the original cost function c, we will simply write $h(s)$ instead of $h(s, c)$. In definitions, this means the defined heuristic is only defined for the original cost function.

We define addition over $\mathbb{R} \cup \{-\infty, \infty\}$ such that a sum is ∞ if one of its parts is ∞ (even if another is $-\infty$) and that a sum is $-\infty$ if one of its parts is $-\infty$ and no part is ∞; and the multiplication over $\mathbb{R} \cup \{-\infty, \infty\}$ such that $c \cdot \infty = \infty$ and $c \cdot (-\infty) = -\infty$ for any finite constant $c > 0$.

We say that the heuristic h is *admissible* iff $h(s) \leq J^\star(s)$ for every state $s \in S$. Furthermore, h is *generally admissible* if $h(s, c') \leq J^\star_{c'}(s)$ for every state $s \in S$ and cost function c'. Lastly, h is π-admissible if $h(s) \leq J^\pi(s)$ for all $s \in S$.

Abstraction Heuristics

An important class of heuristics is based on abstractions of the SSP. In classical planning, an abstraction is commonly induced by a surjective abstraction mapping $\alpha : S \to S_\alpha$, where S_α is a finite set, establishing the abstract states. To define the semantics of such an abstraction mapping for SSPs, we need to specify the abstract SSP which is induced by the abstraction mapping. Contrary to classical planning this is not possible for all abstraction mappings. For example, say α maps s to σ and both t_1 and t_2 to τ. If $T(s, a, t_1) \neq T(s, a, t_2)$ then $T(\sigma, a, \tau)$ is not well-defined.

Klößner and Hoffmann (2021) show that the transition probability is well-behaved in the special case of *projections* of probabilistic planning tasks. A projection function for a subset of variables $P \subseteq V$ maps a state s to its projection on P. We consider a class of abstractions that is more general and, for example, also covers Cartesian abstractions. We only require that the abstraction mapping α is such that for each abstract state $\sigma \in S_\alpha$, the expression $\sum_{t \in \alpha^{-1}(\tau)} T(s, a, t)$ evaluates to the same value for any concrete state $s \in \alpha^{-1}(\sigma)$ in which a is applicable. This implies that the following transition function T_α is well-defined on states $\sigma, \tau \in S_\alpha$ and actions $a \in A$:

$$T_\alpha(\sigma, a, \tau) := \begin{cases} \sum\limits_{t \in \alpha^{-1}(\tau)} T(s, a, t) & \exists s \in \alpha^{-1}(\sigma) . a \in A(s) \\ 0 & \text{otherwise} \end{cases}$$

where the state $s \in \alpha^{-1}(\sigma)$ is chosen as any concrete state satisfying $a \in A(s)$. Such an abstraction function induces the abstract SSP $\langle S_\alpha, A, T_\alpha, \alpha(s_0), \alpha(s_\star), \alpha(c) \rangle$, where

$$\alpha(c)(\sigma, a) := \min_{\substack{s \in \alpha^{-1}(\sigma) \\ \text{s.t. } a \in A(s)}} c(s, a)$$

This definition is equivalent to the definition for projections given by Klößner and Hoffmann when α is a projection mapping and c is state-independent.

We can define two heuristics based on an abstraction function α: the *abstraction heuristic* $h^\alpha(s, c') := J^\star_{\alpha(c')}(\alpha(s))$ maps each state to the optimal value of its abstract state in the induced abstract SSP where the cost function is $\alpha(c')$, while the *LP heuristic* $h^\alpha_{\mathrm{LP}}(s, c') := J^{\mathrm{LP}}_{\alpha(c')}(\alpha(s))$ maps each state to the objective value computed by LP 1 for the abstract SSP with cost function $\alpha(c')$. As we have seen, $h^\alpha(s, c') \geq h^\alpha_{\mathrm{LP}}(s, c')$ in general because $h^\alpha_{\mathrm{LP}}(s, c')$ becomes $-\infty$ in the presence of any negative-cost transition cycle in the abstraction. However, as seen in Theorem 2, we can compute $h^\alpha(s, c')$ using the version of LP 1 where irrelevant abstract states and actions are pruned.

We claim that h^α (and by extension also h^α_{LP}) is generally admissible. Essentially, we can prove that a solution for the pruned version of LP 1 for the abstract SSP and $\alpha(s)$ can be transformed to a solution with equal objective value for the pruned version of LP 1 for the original SSP and s. We refer to the technical report (Klößner et al. 2022) for the proof details.

General Cost-Partitioning for SSPs

In classical planning, general cost-partitioning (Katz and Domshlak 2010; Pommerening et al. 2015) is an approach that allows to admissibly sum up arbitrary generally admissible heuristics. Here, the participating heuristics are computed under alternative cost functions which have the property that the sum of the costs of each operator in the alternative cost functions is below the original operator cost. In the following, we generalize the concepts behind general cost partitioning to SSPs and verify that basic results from classical planning still hold. Afterwards, we investigate the special case of optimal cost-partitioning for SSPs and abstraction heuristics. We start with the formal definition of transition and operator cost partitions.

Definition 1 *A* transition cost partition *(for cost function c) is a finite family of cost functions $c_i : S \times A \to \mathbb{R}$ for $i \in I$ (where I is an index set) satisfying $\sum_{i \in I} c_i(s, a) \le c(s, a)$ for all $s \in S$ and $a \in A$. An* operator cost partition *is a transition cost partition which additionally satisfies $c_i(s, a) = c_i(t, a)$ for every $s, t \in S$, $a \in A$ and $i \in I$. We define $\mathcal{P}_I$ as the set of transition cost partitions over the index set I.*

Given a finite family of heuristics $\mathcal{H} = (h_i)_{i \in I}$ and a cost partition $P \in \mathcal{P}_I$, we define the transition cost partitioning heuristic *$h^{\mathcal{H},P}(s)$ for $\mathcal{H}$ and P by*

$$h^{\mathcal{H},P}(s) := \sum_{i \in I} h_i(s, c_i).$$

Recall that sums involving ∞ evaluate to ∞ and sums involving $-\infty$ but no ∞ evaluate to $-\infty$. Structurally, this definition matches the corresponding definitions known from classical planning. We now show that if $\mathcal{H}$ is a family of generally admissible heuristics, $h^{\mathcal{H},P}(s)$ is admissible, which generalizes the same result established in classical planning. In this setting, we can make use of the fact that the cost of a plan in the original problem is greater or equal than the summed up costs of the same plan in the versions of the problem with alternative costs. We first formulate the analogous statement for policies and verify its correctness.

Lemma 1 *Let $(c_i)_{i \in I}$ be a transition cost partition and let π be any policy. Then $h(s) := \sum_{i \in I} J^\pi_{c_i}(s)$ is π-admissible.*

Proof. Let $S' \subseteq S$ be the set of states s for which π is s-proper. If $s \notin S'$, we have $J^\pi(s) = \infty \ge h(s)$. It is left to show that $h(s) \le J^\pi(s)$ for $s \in S'$. To this end, let us denote the restriction of h to S' by h'.

First of all, note that h' must be real-valued. We conclude the proof by showing that $h' \le \mathcal{B}^\pi h'$. Because $\mathcal{B}^\pi$ is a contraction mapping and increases monotonically, we then have $h' \le \lim_{k \to \infty} (\mathcal{B}^\pi)^k h' = J^\pi|_{S'}$. Now let $s \in S'$. For $s = s_\star$, we have $h(s) = 0 = (\mathcal{B}^\pi h)(s)$. Otherwise:

$$\begin{aligned} h(s) &= \sum_{i \in I} J^\pi_{c_i}(s) \\ &= \sum_{i \in I} \left[c_i(s, \pi(s)) + \sum_{t \in S} T(s, \pi(s), t) J^\pi_{c_i}(t) \right] \\ &= \sum_{i \in I} c_i(s, \pi(s)) + \sum_{t \in S} T(s, \pi(s), t) h(t) \\ &\le c(s, \pi(s)) + \sum_{t \in S} T(s, \pi(s), t) h(t) \end{aligned}$$

□

With this lemma, we now advance to prove our claim.

Theorem 3 *Let $P = (c_i)_{i \in I}$ be a transition cost partition and let $\mathcal{H} = (h_i)_{i \in I}$ be a finite family of generally admissible heuristics. Then the heuristic $h^{\mathcal{H},P}$ is admissible.*

Proof. Let $s \in S$. If $J^\star(s) = \infty$ the claim is trivial. Now assume $J^\star(s) = -\infty$. By definition, there is (a state in) a negative-cost simple cycle x that is reachable by an s-proper policy π with non-zero probability. The existence and reachability of the cycle is independent of the cost function. Since c_i is a cost partition and x has negative cost under the original cost function, we have

$$\sum_{i \in I} \sum_{t \in S, a \in A(t)} c_i(t, a) x_{t,a} \le \sum_{t \in S, a \in A(t)} c(t, a) x_{t,a} < 0.$$

Consequentially, $\sum_{t \in S, a \in A(t)} c_i(t, a) x_{t,a} < 0$ for at least one $i \in I$, which means that $J^\star_{c_i}(s) = -\infty$. Since h_i is generally admissible, also $h_i(s, c_i) = -\infty$. For all $j \ne i$ we know that $h_j(s, c_j) \ne \infty$ because π is s-proper (independent of the cost function) and h_j is generally admissible. Therefore, $h^{\mathcal{H},P}(s) = -\infty$.

Lastly, let $J^\star(s) \in \mathbb{R}$ and let $\pi^\star \in \arg\min_\pi J^\pi(s)$ be an optimal policy for s. First of all, we have

$$h^{\mathcal{H},P}(s) = \sum_{i \in I} h_i(s, c_i) \le \sum_{i \in I} J^\star_{c_i}(s) \le \sum_{i \in I} J^{\pi^\star}_{c_i}(s)$$

by general admissibility of the heuristics. Note the last step is an inequality because $\pi^\star$ may be optimal for the cost function c, but may become suboptimal when the cost function is changed. By Lemma 1, $h^{\mathcal{H},P}(s) \le J^{\pi^\star}(s) = J^\star(s)$. □

Naturally, we are interested in transition cost partitions P which maximize the heuristic estimates for a state s. Given the family of heuristics $\mathcal{H} = (h_i)_{i \in I}$, we define the optimal transition cost partitioning heuristic as

$$h^{\text{OTCP}}_{\mathcal{H}}(s) := \max_{P \in \mathcal{P}_I} \{ h^{\mathcal{H},P}(s) \}.$$

We note that the set might not have a maximum if transition cost partitions for arbitrarily high heuristic values exist. In that case, we define $h^{\text{OTCP}}_{\mathcal{H}}(s) := \infty$. However, if $h^{\mathcal{H},P}(s) = h^{\text{OTCP}}_{\mathcal{H}}(s)$, we say that P is an *optimal transition cost partition* for s. Likewise, we define the optimal operator cost partitioning heuristic $h^{\text{OOCP}}_{\mathcal{H}}(s)$.

Optimal Cost Partitioning for Abstractions

In classical planning, optimal (transition) cost partitioning can be pursued for many classes of heuristics by solving a linear program, including abstraction heuristics. We now consider the following linear program for a set of abstractions $\mathcal{A}$ and a state of interest $\tilde{s}$ for which we want an estimate for the optimal state value. We will analyze this LP and sketch the required modifications such that it computes the value of the optimal transition cost partitioning heuristic $h^{\mathrm{OTCP}}_{\mathcal{H}}(\tilde{s})$ for the abstraction heuristics $\mathcal{H} = (h^{\alpha}_{\mathrm{LP}})_{\alpha \in \mathcal{A}}$.

$$\text{Maximize} \sum_{\alpha \in \mathcal{A}} \mathsf{y}_{\alpha(\tilde{s})} \text{ subject to} \tag{LP 3}$$

$$\mathsf{y}_{\alpha(s_\star)} = 0 \quad \text{for all } \alpha \in \mathcal{A} \tag{5}$$

$$\mathsf{y}_{\alpha(s)} \leq \mathsf{c}_{\alpha s a} + \sum_{t \in S} T(s,a,t)\mathsf{y}_{\alpha(t)} \quad \text{for all } \alpha \in \mathcal{A}, s \in S, a \in A(\alpha(s)) \tag{6}$$

$$\sum_{\alpha \in \mathcal{A}} \mathsf{c}_{\alpha s a} \leq c(s,a) \qquad \text{for all } a \in A \tag{7}$$

where all variables are unrestricted.

In this linear program, the variables $\mathsf{c}_{\alpha s a}$ model the part of $c(s,a)$ that is attributed to abstraction α. Constraints (7) ensure that these values form a transition cost partition. If the cost variables are fixed, the remaining constraints for different abstractions do not depend on each other, so the remaining LP can be seen as a sum of independent LPs (one for each abstraction). These LPs (constraints (5) and (6) for one abstraction α with the objective value $\mathsf{y}_{\alpha(\tilde{s})}$) compute exactly $J^{\mathrm{LP}}(\alpha(\tilde{s}))$ under the cost function encoded in $\mathsf{c}_{\alpha s a}$.

As we have seen in Theorem 1, $J^{\mathrm{LP}}(\alpha(\tilde{s}))$ is not necessarily equal to $J^\star(\alpha(\tilde{s}))$, as the former becomes $-\infty$ also if the corresponding abstract SSP for α contains a negative-cost cycle which is irrelevant for $\tilde{s}$. Thus, not every transition cost partition, even if it induces a finite heuristic value, corresponds to an assignment of $\mathsf{c}_{\alpha s a}$ which is part of a solution. This is only the case if the transition cost partition introduces no negative-cost cycles in any abstract SSP. We call such transition cost partitions *feasible*.

To illustrate why this is an issue, consider the example shown in Figure 2 with initial state $s_0 = \langle 1, 1 \rangle$. There is no $\alpha(s_0)$-proper policy for both of the abstractions, so both of them have $h^\alpha(s_0, c') = \infty$ under any cost function c' and we get $h^{\mathcal{H},P}(s_0) = \infty$ for any transition cost partition P. Hence, we also have $h^{\mathrm{OTCP}}_{\mathcal{H}}(s_0) = \infty$. But no matter how we distribute the costs for a, one abstraction will have a negative-cost simple cycle, so there is no feasible transition cost partition. Therefore, the respective instantiation of LP 3 is infeasible, which represents a heuristic value of $-\infty$.

Without modifications to it, LP 3 only maximizes over feasible transition cost partitions. In this sense, the LP above computes an optimal *feasible* transition cost partitioning over the abstraction heuristics. We now consider the LP where constraints (5) and (6) for the abstract SSPs are restricted to the relevant actions and states of the corresponding abstractions. If the LP is evaluated for such pruned abstractions, it computes the optimal transition cost partitioning of the abstraction heuristics h^α.

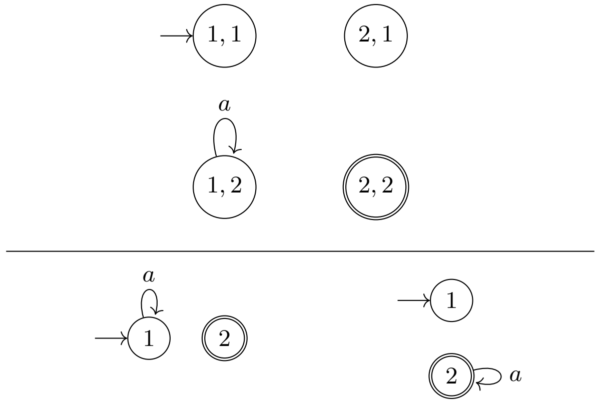

Figure 2: A deterministic SSP with variables $v, w \in \{1, 2\}$ and action costs $c(\langle 1, 2 \rangle, a) = -1$ (top), and its projections onto $\{v\}$ (bottom left) and $\{w\}$ (bottom right).

To see why this is the case, note that a transition cost partition P with $h^{\mathcal{H},P}(\tilde{s}) \in \mathbb{R}$ does not introduce a relevant negative-cost transition cycle in any of the abstract SSPs. Therefore, the LP will now have a corresponding solution with the same objective value as $h^{\mathcal{H},P}(\tilde{s})$. In the case where $h^{\mathcal{H},P}(\tilde{s}) = \infty$, one abstraction α has no $\tilde{s}$-proper policy, so all its transitions are irrelevant and would be pruned. The transition cost partition that assigns all costs to this abstraction and zero costs elsewhere then induces a solution. Because the variable $\mathsf{y}_{\alpha(\tilde{s})}$ is in the objective and left without constraints, the objective value can always be increased and the LP is unbounded.

Note that the transition cost partitioning LP contains $\mathcal{O}(|S||A||\mathcal{A}|)$ constraints, so it is generally not practical to evaluate it. The constraints can be simplified if we partition operator costs instead of transition costs, i.e., if we further require that all costs of one action in one abstraction are the same. We can do so by replacing the variables $\mathsf{c}_{\alpha s a}$ with state-independent variables $\mathsf{c}_{\alpha a}$. As we required that $\sum_{t \in \alpha^{-1}(\tau)} T(s,a,t)$ is constant for any concrete state $s \in \alpha^{-1}(\sigma)$, we can then replace $\sum_{t \in S} T(s,a,t)\mathsf{y}_{\alpha(t)}$ with $\sum_{\tau \in S_\alpha} T_\alpha(\sigma, a, \tau)\mathsf{y}_\tau$ for $\sigma = \alpha(s)$. In this case, only a single constraint of type (6) for every abstract non-goal state is introduced. The number of constraints then is in $\Theta(\sum_{\alpha \in \mathcal{A}} |S_\alpha||A|)$ and the size of the LP is polynomial in the size of the abstract SSPs.

Relationship to Occupation Measure Heuristics

We now establish a link between optimal operator cost partitioning and the LP-based occupation measure heuristics (Trevizan, Thiébaux, and Haslum 2017), which are the equivalent of operator-counting constraints (Pommerening et al. 2014) for SSPs. An occupation measure gives the expected number of times an action is executed by a policy.

Since these heuristics were only formulated on planning tasks with state-independent action costs, we assume in this section that for action a, $c(s,a)$ is the same for every state s

and use $c(a)$ as an abbreviation for this common value.

The projection occupation measure heuristic h^{pom} utilizes atomic projections, which are the projections onto the single variable sets, and the dual LP formulation for solving SSPs (d'Epenoux 1963; Altman 1999), which essentially casts the problem of finding the optimal value function to a probabilistic flow problem in which the flow leaving the abstract state d via action $a \in A(d)$ is represented by the LP variable $\mathsf{x}^{\text{pom}}_{d,a}$. The inflow and outflow of the abstract state $d \in D_v$ of variable $v \in V$ in this problem are defined as

$$\text{in}(d) := \sum_{\substack{d' \in D_v \\ a \in A(d')}} T_v(d', a, d)\mathsf{x}^{\text{pom}}_{d',a} \quad \text{out}(d) := \sum_{a \in A(d)} \mathsf{x}^{\text{pom}}_{d,a}$$

where T_v denotes the transition function of the atomic projection with respect to v. With this, $h^{\text{pom}}(\tilde{s})$ is defined as the optimal objective value of LP 4, which uses Iverson brackets notation.

$$\text{Minimize} \sum_{a \in A} \mathsf{X}^{\text{pom}}_a c(a) \text{ subject to} \qquad \text{(LP 4)}$$

$$\text{out}(d) - \text{in}(d) = [d = \tilde{s}(v)] - [d = s_\star(v)] \quad \text{for all } v \in V \text{ and } d \in D_v \qquad (8)$$

$$\mathsf{X}^{\text{pom}}_a = \sum_{d \in D_v \cap A^{-1}(a)} \mathsf{x}^{\text{pom}}_{d,a} \quad \text{for all } a \in A \text{ and } v \in V \qquad (9)$$

where all variables are non-negative.

The flow constraints (8) enforce that the flow conservation principle applies for all states except for the initial and goal state. The initial state produces one unit of flow, which the goal state absorbs, unless both are equal in which case no flow is produced at all. Lastly, the tying constraints (9) enforce that the occupation measure $\mathsf{X}^{\text{pom}}_a$ of an action a is the same across all atomic projections.

At first sight, LP 4 seems to have nothing to do with operator cost partitioning. However, consider the dualization:

$$\text{Maximize} \sum_{v \in V} \mathsf{y}_{\tilde{s}(v)} - \mathsf{y}_{s_\star(v)} \text{ subject to}$$

$$\mathsf{y}_{s_\star(v)} \leq 0 \qquad \text{for all } v \in V$$

$$\mathsf{y}_d \leq \mathsf{c}_{va} + \sum_{d' \in D_v} T_v(d, a, d')\mathsf{y}_{d'} \quad \text{for all } v \in V, d \in D_v \text{ and } a \in A(d)$$

$$\sum_{v \in V} \mathsf{c}_{va} \leq c(a) \qquad \text{for all } a \in A$$

where all variables are unrestricted.

This LP is equivalent to the optimal feasible operator cost partitioning LP for atomic abstractions, since we can always enforce $\mathsf{y}_{s_\star(v)} = 0$ through the transformation $\mathsf{y}'_d := \mathsf{y}_d - \mathsf{y}_{s_\star(v)}$ without changing the objective value, which shows Theorem 4 below.

Theorem 4 *h^{pom} computes an optimal feasible operator cost partitioning over atomic projections.*

Relationship to Regrouped Operator Counting

Trevizan, Thiébaux, and Haslum also introduce the regrouped operator counting heuristic h^{roc}. It uses net-change constraints known from classical planning (Pommerening et al. 2014) on the all-outcomes determinization of the task, in which a possible effect can be freely selected.

We will now show that h^{roc} also computes an optimal feasible operator cost partitioning for a syntactical restriction of planning tasks called transition normal form (TNF) where $V(\text{pre}(a)) = V(e)$ for all actions a and effects $e \in \text{Eff}(a)$. With this assumption, each fact pair $\langle v, d\rangle$ induces three disjoint, exhaustive classes of action-effect pairs: Those which always consume (AC), those which always produce (AP), and those which never change the fact (NC):

$$\begin{aligned}
\text{AC}_{v=d} &:= \{(a,e) \mid e(v) \neq d, \text{pre}(a)(v) = d\} \\
\text{AP}_{v=d} &:= \{(a,e) \mid e(v) = d, \text{pre}(a)(v) \neq d\} \\
\text{NC}_{v=d} &:= \{(a,e) \mid e(v) = \text{pre}(a)(v)\} \cup \\
&\quad\ \{(a,e) \mid e(v) \neq d \neq \text{pre}(a)(v)\}
\end{aligned}$$

The possible net-change of a fact $\langle v, d\rangle$ when going from state $\tilde{s}$ to the goal $s_\star$ is described by $\text{pnc}^{\tilde{s}\to s_\star}_{v=d} = [d = s_\star(v)] - [d = \tilde{s}(v)]$.

With this, the heuristic $h^{\text{roc}}(s)$ is defined as the optimal objective value of LP 5.

$$\text{Minimize} \sum_{a \in A} \mathsf{X}^{\text{roc}}_a c(a) \text{ subject to} \qquad \text{(LP 5)}$$

$$\sum_{(a,e) \in \text{AP}_{v=d}} \mathsf{x}^{\text{roc}}_{a,e} - \sum_{(a,e) \in \text{AC}_{v=d}} \mathsf{x}^{\text{roc}}_{a,e} = \text{pnc}^{\tilde{s}\to s_\star}_{v=d} \quad \text{for all } v \in V, d \in D_v \qquad (10)$$

$$\mathsf{X}^{\text{roc}}_a = \frac{\mathsf{x}^{\text{roc}}_{a,e}}{\Pr_a(e)} \qquad \text{for all } a \in A, e \in \text{Eff}(a) \qquad (11)$$

where all variables are non-negative.

In this LP, the variable $\mathsf{x}^{\text{roc}}_{a,e}$ models the expected amount of times a is applied and the effect e occurs. Constraints (10) are the net-change constraints, which are divided into lower-bounding and upper-bounding net-change constraints in the original formulation of Trevizan, Thiébaux, and Haslum (2017). The simplified constraints in LP 5 are equivalent for tasks in transition normal form. The regrouping constraints (11) enforce that the expected number of times an effect of an action occurs in a solution is proportional to its probability: if the occupation measure of an action a is $\mathsf{X}^{\text{roc}}_a$, then the effect e occurs $\Pr_a(e) \cdot \mathsf{X}^{\text{roc}}_a$ times in expectation.

While Trevizan et al. show that h^{pom} dominates h^{roc}, they raise the question whether these two heuristics are in fact equal. We will now show that this conjecture is valid for tasks in TNF. A similar result was already shown by Pommerening et al. (2015) between the optimal operator cost partitioning heuristic over atomic projections in classical planning and the state equation heuristic h^{SEQ} (Bonet and van den Briel 2014), which can be seen as a special case of h^{roc} for deterministic problems.

Theorem 5 *For tasks in TNF, $h^{\text{roc}} = h^{\text{pom}}$.*

Proof. Let $\langle \mathsf{X}^{\mathrm{roc}}, \mathsf{x}^{\mathrm{roc}} \rangle$ be a solution for LP 5. We construct a solution for LP 4 with equal objective value. Define

$$\mathsf{X}^{\mathrm{pom}}_{a} := \sum_{e \in \mathrm{Eff}(a)} \mathsf{x}^{\mathrm{roc}}_{a,e}$$

$$\mathsf{x}^{\mathrm{pom}}_{d,a} := \begin{cases} \frac{\mathsf{x}^{\mathrm{roc}}_{a,e}}{\mathrm{Pr}_a(e)} & \text{if } d = s_0(v) \text{ or } v \in V(\mathrm{pre}(a)) \\ 0 & \text{otherwise} \end{cases}$$

where v is the variable where $d \in D_v$ and e is any effect in $\mathrm{Eff}(a)$. Freely choosing e is possible because of the regrouping constraints (11).

The objective value is clearly the same. Consider the tying constraints (9). Let $a \in A$ and $v \in V$. If $v \in V(\mathrm{pre}(a))$, then the right hand side $\sum_{d \in D_v \cap A^{-1}(a)} \mathsf{x}^{\mathrm{pom}}_{d,a}$ collapses to the single value $\mathsf{x}^{\mathrm{pom}}_{\mathrm{pre}(a)(v),a}$ as a is not applicable in any other abstract state. Likewise, if $v \notin V(\mathrm{pre}(a))$, then a is applicable in all abstract states but the sum collapses to the single value $\mathsf{x}^{\mathrm{pom}}_{s_0(v),a}$ because $\mathsf{x}^{\mathrm{pom}}_{d,a} = 0$ for all $d \neq s_0(v)$. For this single value $\mathsf{x}^{\mathrm{pom}}_{d,a}$ we derive

$$\mathsf{x}^{\mathrm{pom}}_{d,a} = (\sum_{e \in \mathrm{Eff}(a)} \mathrm{Pr}_a(e)) \frac{\mathsf{x}^{\mathrm{roc}}_{a,e}}{\mathrm{Pr}_a(e)} = \sum_{e \in \mathrm{Eff}(a)} \mathsf{x}^{\mathrm{roc}}_{a,e} = \mathsf{X}^{\mathrm{pom}}_{a},$$

so the tying constraints are satisfied.

Next, we consider the flow constraints (8). Let $v \in V$ and assume $d \in D_v \setminus \{s_0(v)\}$. The outgoing flow then is $\mathrm{out}(d) = \sum_{a \in A(d)} \mathsf{x}^{\mathrm{pom}}_{d,a}$ but for all actions a with $v \notin V(\mathrm{pre}(a))$ the term is 0. This leaves only actions a with $\mathrm{pre}(a)(v) = d$. For every such action we can then rewrite $\mathsf{x}^{\mathrm{pom}}_{d,a}$ in the same way as before as $\sum_{e \in \mathrm{Eff}(a)} \mathsf{x}^{\mathrm{roc}}_{a,e}$. We get

$$\mathrm{out}(d) = \sum_{\substack{a \in A, e \in \mathrm{Eff}(a) \\ \mathrm{pre}(a)(v) = d}} \mathsf{x}^{\mathrm{roc}}_{a,e} = \sum_{(a,e) \in \mathrm{AC}_{v=d}} \mathsf{x}^{\mathrm{roc}}_{a,e} + \sum_{\substack{(a,e) \in \mathrm{NC}_{v=d} \\ \mathrm{pre}(a)(v) = d}} \mathsf{x}^{\mathrm{roc}}_{a,e}.$$

For $d = s_0(v)$, the outflow is the same, except that we sum over actions a with $\mathrm{pre}(a)(v) \in \{\bot, d\}$ in the respective sums.

Finally, we consider the inflow $\mathrm{in}(d)$ into an abstract state $d \in D_v$. We denote the inflow coming from a value $d' \neq d$ by $\mathrm{in}(d', d) = \sum_{a \in A(d')} T(d', a, d) \mathsf{x}^{\mathrm{pom}}_{d',a}$. Note that we have $T(d', a, d) = 0$ unless there is an effect $e \in \mathrm{Eff}(a)$ with $e(v) = d$. We assume TNF, so we conclude $v \in V(\mathrm{pre}(a))$. This means we can simplify the inflow from value $d' \neq d$ to

$$\mathrm{in}(d', d) = \sum_{\substack{a \in A \\ \mathrm{pre}(a)(v) = d'}} \sum_{\substack{e \in \mathrm{Eff}(a) \\ e(v) = d}} \mathrm{Pr}_a(e) \cdot \mathsf{x}^{\mathrm{pom}}_{d',a}.$$

Now, acknowledge that when we accumulate the inflows of all $d' \neq d$, we sum over every action effect pair $(a, e) \in \mathrm{AP}_{v=d}$ exactly once. Thus, we have

$$\sum_{\substack{d' \in D(v) \\ d' \neq d}} \mathrm{in}(d', d) = \sum_{(a,e) \in \mathrm{AP}_{v=d}} \mathrm{Pr}_a(e) \cdot \mathsf{x}^{\mathrm{pom}}_{d',a} = \sum_{(a,e) \in \mathrm{AP}_{v=d}} \mathsf{x}^{\mathrm{roc}}_{a,e}.$$

For $\mathrm{in}(d, d)$, we consider two cases. If $d \neq s_0(v)$, then again $\mathsf{x}^{\mathrm{pom}}_{d,a} = 0$ unless $v \in V(\mathrm{pre}(a))$ and we have

$$\mathrm{in}(d, d) = \sum_{\substack{a \in A \\ \mathrm{pre}(a)(v) = d}} \sum_{\substack{e \in \mathrm{Eff}(a) \\ e(v) = d}} \mathsf{x}^{\mathrm{pom}}_{a,e} = \sum_{\substack{(a,e) \in \mathrm{NC}_{v=d} \\ \mathrm{pre}(a)(v) = d}} \mathsf{x}^{\mathrm{roc}}_{a,e}.$$

In the case $s = s_0(v)$, we instead sum over actions a with $\mathrm{pre}(a)(v) \in \{\bot, d\}$ in the sums.

All in all, we obtain that

$$\mathrm{out}(d) - \mathrm{in}(d) = \sum_{(a,e) \in \mathrm{AC}_{v=d}} \mathsf{x}^{\mathrm{roc}}_{a,e} - \sum_{(a,e) \in \mathrm{AP}_{v=d}} \mathsf{x}^{\mathrm{roc}}_{a,e}$$

and derive $\mathrm{out}(d) - \mathrm{in}(d) = -\mathrm{pnc}^{s' \to s_\star}_{v=d}$ from the net-change constraints (10). It is straightforward to show that the flow constraints (8) are satisfied by enumerating all possible cases for the flow constraints and $\mathrm{pnc}^{s' \to s_\star}_{v=d}$. We omit this final step due to lack of space. □

Approximate Linear Programming and Potential Heuristics

We now analyze the connection of cost partitioning to approximate linear programming (ALP), a technique introduced for infinite-horizon discounted-reward MDPs by Guestrin et al. (2003).

ALP approximates the optimal value function $J^\star(s)$ of an MDP with a weighted sum $h(s) = \sum_{f \in \mathcal{F}} w_f f(s)$ of some *basis functions* $f : S \to \mathbb{R}$. It then optimizes the weights w_f instead of optimizing $J^\star$ directly. The values $h(s)$ are an admissible estimate for $J^\star$. The same idea was developed independently for classical planning under the name *potential heuristics* (Pommerening et al. 2015) where the basis functions are called *features*. Pommerening, Helmert, and Bonet (2017) showed a connection of potential heuristics where the features are indicator functions of abstract states to the transition cost partitioning over these abstractions. Both classical planning and discounted-reward MDPs are special cases of SSPs in the sense that they can be compiled into equivalent SSPs (Mausam and Kolobov 2012). We show that the link between potential heuristics and transition cost partitioning in classical planning extends to SSPs and a corresponding extension of ALP.

The following LP optimizes the weights for approximate linear programming in an SSP $\langle S, A, T, s_0, s_\star, c \rangle$ with basis functions $\mathcal{F}$. It optimizes relative to a given *state relevance function* ρ, which specifies the importance of a high estimate for each state.

$$\text{Maximize} \sum_{s \in S} \rho(s) \sum_{f \in \mathcal{F}} \mathsf{w}_f f(s) \text{ subject to} \qquad \text{(LP 6)}$$

$$\sum_{f \in \mathcal{F}} \mathsf{w}_f f(s_\star) = 0 \qquad (12)$$

$$\sum_{f \in \mathcal{F}} \mathsf{w}_f f(s) \leq c(s, a) + \sum_{t \in S} T(s, a, t) \sum_{f \in \mathcal{F}} \mathsf{w}_f f(t) \qquad (13)$$

for all $s \in S$ and $a \in A(s)$

where all variables $(\mathsf{w}_f)_{f \in \mathcal{F}}$ are unrestricted.

In the technical report, we show that the above LP computes equivalent solutions to the LP by Guestrin et al. (2003) when used on the compilation. When compiling an MDP into an SSP, a new state is added to serve as the SSP's goal state. We thus have to extend the state relevance function ρ

and the basis functions $f \in \mathcal{F}$ to this new state. The LPs are equivalent if all functions map the new state to 0.

Both Guestrin et al. (2003) and Pommerening, Helmert, and Bonet (2017) show how to use bucket elimination to reduce the size of this LP in factored state spaces and make the computation fixed-parameter tractable. The same technique can be applied for SSPs, but we keep the simpler LP above as the LP size does not matter for our theoretical analysis.

We now focus our attention to basis functions that are indicator functions of abstract states, i.e., for an abstraction $\alpha : S \to S_\alpha$, we consider one basis function f_σ for each abstract state $\sigma \in S_\alpha$ that is defined as

$$f_\sigma(s) = \begin{cases} 1 & \text{if } \alpha(s) = \sigma \\ 0 & \text{otherwise.} \end{cases}$$

We consider a set of abstractions $\mathcal{A}$ and assume that the sets of abstract states are pairwise disjoint so an abstract state uniquely identifies its abstraction. For each state s and each abstraction α, exactly one of the basis functions associated with α has the value 1 while all others have 0. This simplifies the LP computed for ALP as follows (the formal proof for equivalence can be found in the technical report):

$$\text{Maximize} \sum_{s \in S} \rho(s) \sum_{\alpha \in \mathcal{A}} \mathsf{w}_{\alpha(s)} \text{ subject to} \qquad \text{(LP 7)}$$

$$\mathsf{w}_{\alpha(s_\star)} = 0 \qquad \text{for all } \alpha \in \mathcal{A} \qquad (14)$$

$$\sum_{\alpha \in \mathcal{A}} \mathsf{w}_{\alpha(s)} \leq c(s,a) + \sum_{t \in S} T(s,a,t) \sum_{\alpha \in \mathcal{A}} \mathsf{w}_{\alpha(t)} \qquad (15)$$

$$\text{for all } s \in S \text{ and } a \in A(s)$$

where all variables $(\mathsf{w}_\sigma)_{\sigma \in S^\alpha, \alpha \in \mathcal{A}}$ are unrestricted.

The final step is to show that LP 7 computes an optimal feasible transition cost partitioning over the abstractions in $\mathcal{A}$ that is optimized according to ρ.

Theorem 6 *Let $\mathcal{A}$ be a set of abstraction mappings for an SSP, $\mathcal{F} = \{f_\sigma \mid \alpha \in \mathcal{A}, \sigma \in S_\alpha\}$ be a set of basis functions and ρ be a state-relevance function.*

ALP with basis functions $\mathcal{F}$ computes a feasible transition cost partitioning over the heuristics $\mathcal{H} = (h^\alpha_{\mathrm{LP}})_{\alpha \in \mathcal{A}}$ that is optimal according to ρ. In particular, ALP computes $h^{\mathrm{OTCP}}_{\mathcal{H}}(\tilde{s})$ if $\rho(\tilde{s}) = 1$ and $\rho(s) = 0$ for $s \neq \tilde{s}$.

Proof. For a solution w of LP 7 we can show that $\mathsf{v} = \mathsf{w}$ and $\mathsf{c}_{\alpha s a} = \mathsf{w}_{\alpha(s)} - \sum_{t \in S} T(s,a,t)\mathsf{w}_{\alpha(t)}$ is a solution to LP 3. Constraint (6) trivializes and constraint (7) follows directly from (15), while (5) follows from (14).

Likewise, for any solution y, c of LP 3, we can show that $\mathsf{w} = \mathsf{y}$ is a solution to LP 7. This can be seen by summing constraint (6) for all abstractions and then using (7) to show that (15) is satisfied. Constraint (5) clearly implies (14).

If we maximize the value of LP 3 according to ρ, then both transformations maintain their objective value. This proves that the LPs are equivalent for the maximization according to ρ. If the relevance function ρ simplifies the objective to just maximize the cost partitioned value of $\tilde{s}$, ALP computes $h^{\mathrm{OTCP}}_{\mathcal{H}}(\tilde{s})$. □

Theorem 6 considers a given set of abstractions so it can be used to interpret cost-partitioned abstraction heuristics as ALP. But note that it can also be used in the other direction: if the basis functions are indicator functions that can be grouped into classes where in each state exactly one function in the class is 1 and all others are 0, then these classes can be seen as abstractions. Each such abstraction has one abstract state per indicator function in the class and maps concrete states s to the abstract state of the indicator function f with $f(s) = 1$. ALP over such basis functions can then be interpreted as a cost partitioning over these abstractions.

These relations are the same as (and can be seen as an extension of) the relations between potential heuristics and transition cost partitioning in classical planning. The conceptual difference between ALP and $h^{\mathrm{OTCP}}_{\mathcal{H}}$ is that ALP is only optimized once for the state relevance function ρ, so it is fast to evaluate but corresponds to a suboptimal cost partitioning for $\tilde{s}$ in general.

Conclusion

We developed a theory of cost partitioning applicable to stochastic shortest path problems. We defined transition cost partitioning over a set of SSP heuristics and showed that it inherits admissibility from its input heuristics. We demonstrated that both occupation measure heuristics (Trevizan, Thiébaux, and Haslum 2017) and approximate linear programming (Guestrin et al. 2003) compute optimal cost partitions under certain conditions.

Our contribution opens up a variety of research directions for future work. Adapting the various cost-partitioning techniques known from classical planning to the SSP setting could lead to strong SSP heuristics, and it would be interesting to see if the trade-offs between the time spent on optimizing a cost partition and the time saved by having a stronger heuristic is similar as in the classical setting. While we focused on optimal ways to partition the costs here, suboptimal methods like saturated cost partitioning (Seipp, Keller, and Helmert 2020) are the state of the art in classical planning. Extending such suboptimal but fast techniques to SSPs would be an interesting way forward. Both ALP and potential heuristics also use bucket-elimination techniques to efficiently deal with abstractions to more than one variable. Comparing the approaches could potentially lead to further transfer of knowledge between the two areas.

Acknowledgments

Thorsten Klößner received funding from DFG grant 389792660 as part of TRR 248 (see https://perspicuous-computing.science). Florian Pommerening, Thomas Keller and Gabriele Röger have received funding for this work from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 817639). Moreover, this research was partially supported by TAILOR, a project funded by the EU Horizon 2020 research and innovation programme under grant agreement no. 952215.

References

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Barto, A. G.; Bradtke, S. J.; and Singh, S. P. 1995. Learning to Act Using Real-Time Dynamic Programming. *Artificial Intelligence*, 72(1–2): 81–138.

Bellman, R. E. 1957. *Dynamic Programming*. Princeton University Press.

Bertsekas, D. P.; and Tsitsiklis, J. N. 1991. An Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research*, 16: 580–595.

Bonet, B.; and Geffner, H. 2003. Labeled RTDP: Improving the Convergence of Real-Time Dynamic Programming. In Giunchiglia, E.; Muscettola, N.; and Nau, D., eds., *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 2003)*, 12–21. AAAI Press.

Bonet, B.; and van den Briel, M. 2014. Flow-based Heuristics for Optimal Planning: Landmarks and Merges. In Chien, S.; Fern, A.; Ruml, W.; and Do, M., eds., *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS 2014)*, 47–55. AAAI Press.

Culberson, J. C.; and Schaeffer, J. 1998. Pattern Databases. *Computational Intelligence*, 14(3): 318–334.

d'Epenoux, F. 1963. A Probabilistic Production and Inventory Problem. *Management Science*, 10(1): 98–108.

Edelkamp, S. 2001. Planning with Pattern Databases. In Cesta, A.; and Borrajo, D., eds., *Proceedings of the Sixth European Conference on Planning (ECP 2001)*, 84–90. AAAI Press.

Guestrin, C.; Koller, D.; Parr, R.; and Venkataraman, S. 2003. Efficient Solution Algorithms for Factored MDPs. *Journal of Artificial Intelligence Research*, 19: 399–468.

Guillot, M.; and Stauffer, G. 2020. The Stochastic Shortest Path Problem: A polyhedral combinatorics perspective. *European Journal of Operational Research*, 285(1): 148–158.

Hansen, E. A.; and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129(1–2): 35–62.

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.

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

Keller, T.; Pommerening, F.; Seipp, J.; Geißer, F.; and Mattmüller, R. 2016. State-dependent Cost Partitionings for Cartesian Abstractions in Classical Planning. In Kambhampati, S., ed., *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 3161–3169. AAAI Press.

Klößner, T.; and Hoffmann, J. 2021. Pattern Databases for Stochastic Shortest Path Problems. In Ma, H.; and Serina, I., eds., *Proceedings of the 14th Annual Symposium on Combinatorial Search (SoCS 2021)*, 131–135. AAAI Press.

Klößner, T.; Pommerening, F.; Keller, T.; and Röger, G. 2022. Cost Partitioning Heuristics for Stochastic Shortest Path Problems: Technical Report. Technical Report CS-2022-001, University of Basel, Department of Mathematics and Computer Science.

Korf, R. E. 1997. Finding Optimal Solutions to Rubik's Cube Using Pattern Databases. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI 1997)*, 700–705. AAAI Press.

Korf, R. E.; and Felner, A. 2002. Disjoint Pattern Database Heuristics. *Artificial Intelligence*, 134(1–2): 9–22.

Mausam; and Kolobov, A. 2012. *Planning with Markov Decision Processes: An AI Perspective*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Pommerening, F.; Helmert, M.; and Bonet, B. 2017. Higher-Dimensional Potential Heuristics for Optimal Classical Planning. In Singh, S.; and Markovitch, S., eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*, 3636–3643. AAAI Press.

Pommerening, F.; Helmert, M.; Röger, G.; and Seipp, J. 2015. From Non-Negative to General Operator Cost Partitioning. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, 3335–3341. AAAI Press.

Pommerening, F.; Röger, G.; Helmert, M.; and Bonet, B. 2014. LP-based Heuristics for Cost-optimal Planning. In Chien, S.; Fern, A.; Ruml, W.; and Do, M., eds., *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling (ICAPS 2014)*, 226–234. AAAI Press.

Seipp, J.; Keller, T.; and Helmert, M. 2020. Saturated Cost Partitioning for Optimal Classical Planning. *Journal of Artificial Intelligence Research*, 67: 129–167.

Trevizan, F. W.; Thiébaux, S.; and Haslum, P. 2017. Occupation Measure Heuristics for Probabilistic Planning. In Barbulescu, L.; Frank, J.; Mausam; and Smith, S. F., eds., *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling (ICAPS 2017)*, 306–315. AAAI Press.