AI Planning

15. Combining Heuristic Functions

"Not Orthogonal? Who Cares?"

Álvaro Torralba, Cosmina Croitoru

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Winter Term 2018/2019

Thanks to Prof. Jörg Hoffmann for slide sources

Agenda

We Need to *Combine* Heuristic Functions!

We have covered the 4 different methods currently known:

- Critical path heuristics: Done. → **Chapter 8**
- Delete relaxation: Basically done. → **Chapters 9 and 10**
- Abstractions: Done. → **Chapters 11–13**
- Landmarks: Done. → **Chapter 14**

→ Every h yields good performance only in some domains.

Can we exploit their complementary strengths?

Combining Lower-Bound Heuristics: Our Story So Far

Q: Say somebody gives you lower-bounds $h_1, \dots, h_n$. How can you always obtain a lower-bound h that dominates each of them?
A: By $h := \max_{i=1\dots n} h_i$.

Q: Say somebody gives you lower-bounds $h_1, \dots, h_n$. What would be much better than taking their $\max$?
A: Taking their sum.

But how to ensure the sum is still a lower bound?

- For PDBs h^{P_i}: Require the patterns P_i to be orthogonal.
- For elementary landmark $h^{\mathsf{LM}}_{L_i}$: Require the L_i to be orthogonal.

→ What about all the other possible h? And what about combinations across different methods? Is there something we can do *in general*?

The rest of this chapter points out that the answer is **"Yes!!!"**

A Curious Observation in the 15-Puzzle

1	2	3	4
5	6	7	

		3	
			8
9	10	11	12
13	14		15

$\rightarrow$ Is the sum of the abstraction heuristics admissible? No, because the same moves of tile 3 may be counted by both abstractions.

$\rightarrow$ But what if, on each side, I count only 0.5 moves? Then yes, because "duplicate moves" will be accounted for as cost 1.

Cost Partitioning in a Nutshell

$\rightarrow$ Cost partitionings distribute the cost of each action across a set of otherwise identical planning tasks. This technique can be used to admissibly combine arbitrary admissible heuristic functions.

Our Agenda for This Chapter

2. **Cost Partitioning:** We introduce the concept, illustrate it, and prove admissibility of the partitioned sum.
3. **Domination of Previous Orthogonality Criteria:** We prove that there always exists a cost partitioning dominating our orthogonality criteria for PDBs and LMs.
4. **How To Find a Cost Partitioning?** We prove that, for PDBs and LMs and their combination, we can always find *the best possible* cost partitioning in polynomial time.
5. **General Cost Partitioning:** Generalization that improves cost partitioning.
6. **ps. Landmarks and Hitting Sets:** Departing from the fully general combination technique of cost partitioning, we have a look back at LMs and consider a technique even stronger than cost partitioning for combining this particular class of heuristic functions.

Cost Partitioning

Definition (Cost Partitioning). *Let Π be a planning task with actions A and cost function c. An ensemble of functions $c_1, \dots, c_n : A \mapsto \mathbb{R}_0^+$ is a cost partitioning for Π if, for all $a \in A$, $\sum_{i=1}^{n} c_i(a) \leq c(a)$. The cost partitioning is full if, for all $a \in A$, $\sum_{i=1}^{n} c_i(a) = c(a)$.*

Notes and Notations:

- "$=$" (full cost partitioning) is more intuitive; but only "$\leq$" is required for admissibility, and some practical cost partitioning methods are more naturally described that way.
- If h is a heuristic for Π, then $h[c_i]$ denotes the same heuristic but computed on the modification of Π where c is replaced by c_i.
  $\rightarrow$ We assume that $h[c_i]$ is *defined*, for any h.
- If $h_1, \dots, h_n$ is an ensemble of heuristic functions for Π, then the partitioned sum of $h_1, \dots, h_n$ given $c_1, \dots, c_n$ is $\sum_{i=1}^{n} h_i[c_i]$, for which we use the short-hand $\sum h[c]$.

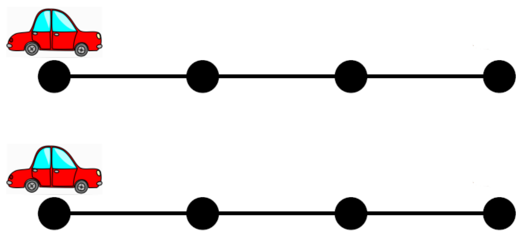
A Simple Example: Driving a Car

Planning task: Drive a car from left to right.

Heuristics: Two times the same heuristic. h_1: PDB for $P_1 = \{car\}$; h_2: PDB for $P_2 = \{car\}$.

P_1:

P_2:



Cost partitioning: For each action a, $c_1(a) = 0.2$ and $c_2(a) = 0.8$.

$\rightarrow h_1[c_1](I) + h_2[c_2](I) = 0.6 + 2.4 = 3 = h^*(I)$. Same for any other full cost partitioning.

Not So Simple: The Attack of the Zombie Tomatoes

Planning task: Goal: A and B both true. Initial state: A and B both false. Actions: $carA$ effect A cost 1; $carB$ effect B cost 1; $fancyCar$ effect A and B cost 1.5.

Heuristics: h_1: $h^{\mathsf{LM}}_{L_1}$ for $L_1 = \{carA, fancyCar\}$; h_2: $h^{\mathsf{LM}}_{L_2}$ for $L_2 = \{carB, fancyCar\}$.

$\rightarrow h^{\mathsf{LM}}_{L_1}(I) = h^{\mathsf{LM}}_{L_2}(I) = 1$. L_1 and L_2 are not orthogonal.

Cost partitioning: $c_1(carA) = 1$, $c_2(carA) = 0$; $c_1(carB) = 0$, $c_2(carB) = 1$; $c_1(fancyCar) = 0.75$, $c_2(fancyCar) = 0.75$. Then $h^{\mathsf{LM}}_{L_1}[c_1](I) = h^{\mathsf{LM}}_{L_2}[c_2](I) = 0.75$ so $h^{\mathsf{LM}}_{L_1}[c_1](I) + h^{\mathsf{LM}}_{L_2}[c_2](I) = 1.5 = h^*(I)$.

Partitioned Sums are Admissible

Theorem (Partitioned Sums are Admissible). *Let Π be a planning task, and let $h_1, \dots, h_n$ be heuristic functions for Π. If $c_1, \dots, c_n$ is a cost partitioning for Π, and if $h_i[c_i]$ is consistent and goal-aware for all i, then the partitioned sum $\sum h[c]$ is consistent and goal-aware, and thus also admissible and safe.*

Proof. Goal-awareness: Trivial because all component heuristics are goal-aware.

Consistency: We need to show that whenever $(s, a, t) \in T$, $\sum h[c](s) \leq \sum h[c](t) + c(a)$.

For all i, $h_i[c_i]$ is consistent. That is, $h_i[c_i](s) \leq h_i[c_i](t) + c_i(a)$ because the cost function underlying $h_i[c_i]$ is c_i (rather than c).

But then, $\sum h[c](s) = \sum_{i=1}^n h_i[c_i](s) \leq \sum_{i=1}^n (h_i[c_i](t) + c_i(a)) = \sum_{i=1}^n h_i[c_i](t) + \sum_{i=1}^n c_i(a)$. Since $c_1, \dots, c_n$ is a cost partitioning, $\sum_{i=1}^n c_i(a) \leq c(a)$ from which the claim follows.

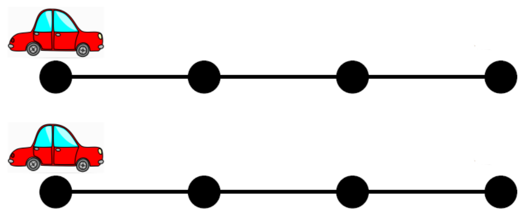
$\rightarrow$ Typical case: $h_i[c_i]$ is consistent and goal-aware because $h_i \in H_i$ where H_i is a family of heuristics (a class of heuristics h computed using the same framework, e.g. PDB heuristic) that are consistent and goal-aware.

Simple Again? Driving *Two* Cars

Planning task: Drive both cars from left to right, using actions $drive(carA, X, Y)$ and $drive(carB, X, Y)$ (unit costs).

$carA$:

$carB$:



Question!

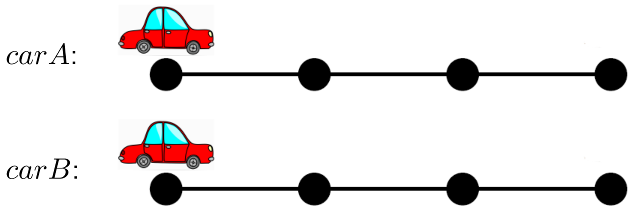
What is the value of $h^{\max}(I)$ for this task?

(A): 3 (B): 6

$\rightarrow h^{\max}(I) = 3$: As we consider single-fact subgoals only, the two cars are treated separately. Each of the respective goal facts costs 3, so (A) is correct.

Questionnaire

Planning task: Drive both cars from left to right, using actions $drive(carA, X, Y)$ and $drive(carB, X, Y)$ (unit costs).



Question!

$h^{\max}(I) = 3$. **Can we improve this using cost partitioning?**

→ Yes! Set $h_1 := h^{\max}$ and $h_2 := h^{\max}$. Then partition the costs so that all $carA$-moves have their full cost in h_1 and 0 cost in h_2, and all $carB$-moves have their full cost in h_2 and 0 cost in h_1. The resulting heuristic is $\sum h[c](I) = 6$.

(This kind of technique was first proposed by [Haslum *et al.* (2005)])

So What?

We can admissibly combine arbitrary heuristic functions.

→ But for the particular methods we have, is that any better than the admissible combinations we defined earlier?

Yes! (provided we manage to find the right cost partitionings)

- Given a collection $L_1, \dots, L_n$ of action sets, there always exists a cost partitioning that dominates the canonical (LM) heuristic, i.e., the best sum of orthogonal $h^{\mathsf{LM}}_{L_i}$.
- Given a pattern collection $P_1, \dots, P_n$, there always exists a cost partitioning that dominates the canonical (PDB) heuristic, i.e., the best sum of orthogonal h^{P_i}.
- In both settings, there are cases where the domination is strict.

Dominating Orthogonal Landmarks

Reminder: → Chapter 14

If $L \subseteq A$ is a disjunctive action landmark for s then $h^{\mathsf{LM}}_L(s) = \min\{c(a) \mid a \in L\}$; else, $h^{\mathsf{LM}}_L(s) = 0$.

Action sets $L_1, \dots, L_n \subseteq A$ are orthogonal if $L_i \cap L_j = \emptyset$ for $i \neq j$. Then, $\sum_{i=1}^n h^{\mathsf{LM}}_{L_i}$ is admissible.

Theorem (Cost Partitionings Can Dominate the Sum of Orthogonal Landmarks). *Let Π be a planning task, and let $L_1, \dots, L_n \subseteq A$ be orthogonal action sets. For each i and $a \in A$, define $c_i(a) := c(a)$ if $a \in L_i$, and $c_i(a) := 0$ otherwise. Then $c_1, \dots, c_n$ is a cost partitioning, and for all states s we have $\sum_{i=1}^n h^{\mathsf{LM}}_{L_i}(s) = \sum h[c](s)$.*

→ Orthogonality for landmarks is subsumed by "0/1" cost partitionings, putting the entire cost of each action into the landmark it is a member of.

Dominating Orthogonal Landmarks, ctd.

Proof. $c_1, \dots, c_n$ is a cost partitioning: $\sum_{i=1}^n c_i(a) \leq c(a)$ because, with $L_1, \dots, L_n$ being orthogonal, a is contained in at most one L_i.

For any s, $\sum_{i=1}^n h^{\mathsf{LM}}_{L_i}(s) = \sum h[c](s)$: By definition, $\sum h[c](s) = \sum_{i=1}^n h^{\mathsf{LM}}_{L_i}[c_i](s)$ so it suffices to show that, for each i, $h^{\mathsf{LM}}_{L_i}(s) = h^{\mathsf{LM}}_{L_i}[c_i](s)$. The latter holds because every action contained in L_i has its original cost in c_i.

Corollary (Cost Partitionings Can Dominate the Canonical LM Heuristic). *Let Π be a planning task, let $\mathcal{C}$ be a collection of action subsets, and let s be a state. Then there exists a cost partitioning for Π so that $h^{\mathcal{C}}(s) \leq \sum h[c](s)$. There are cases where $h^{\mathcal{C}}(s) < \sum h[c](s)$.*

Proof. "$\leq$": Apply theorem to the independent $\{L_1, \dots, L_k\} \subseteq \mathcal{C}$ yielding the maximum in s. "$<$": See next slide.

→ State-dependence: $h^{\mathcal{C}}$ selects the maximum additive $\{L_1, \dots, L_k\} \subseteq \mathcal{C}$ depending on the state. Hence we have to select the cost partitioning depending on the state. That is, we can't in general select a cost partitioning *once* and dominate $h^{\mathcal{C}}$ on *all* states. Example see next slide.

Strictly Dominating the Canonical LM Heuristic

- Goal: A and B both true.
- Initial state: A and B both false.
- Actions: $carA$ effect A cost 1; $carB$ effect B cost 1; $fancyCar$ effect A and B cost 1.5.
- Landmarks $L_1 = \{carA, fancyCar\}$ and $L_2 = \{carB, fancyCar\}$.

Reminder (cf. slide 12): $h^{\text{LM}}_{L_1}(I) = h^{\text{LM}}_{L_2}(I) = 1$, and L_1, L_2 are not orthogonal, so $h^{\mathcal{C}}(I) = \max(h^{\text{LM}}_{L_1}(I), h^{\text{LM}}_{L_2}(I)) = 1$. But, partitioning $c(fancyCar)$ evenly across L_1 and L_2, we get $\sum h[c](I) = 1.5 = h^*(I)$.

$\rightarrow$ This shows "<" on previous slide.

Regarding state-dependence: Consider the same L_1, L_2, and states $s = \{A\}$ and $s' = \{B\}$. Then $h^{\mathcal{C}}(s) = h^{\mathcal{C}}(s') = 1$.

$\rightarrow$ We cannot achieve the same based on a single cost partitioning, as $fancyCar$ would have to have cost ≥ 1 in both landmarks.

Dominating Orthogonal PDBs

Reminder: $\rightarrow$ Chapter 12

An action a affects a projection π_P if there exists a variable $v \in P$ on which eff_a is defined. Patterns $P_1, \dots, P_n$ are orthogonal if every action affects at most one P_i. Then, $\sum_{i=1}^n h^{P_i}$ is admissible.

Theorem (Cost Partitionings Can Dominate the Sum of Orthogonal PDBs). *Let Π be a planning task, and let $\{P_1, \dots, P_n\}$ be an orthogonal pattern collection. For each i and $a \in A$, define $c_i(a) := c(a)$ if a affects α_i, and $c_i(a) := 0$ otherwise. Then $c_1, \dots, c_n$ is a cost partitioning, and for all states s we have $\sum_{i=1}^n h^{P_i}(s) = \sum h[c](s)$.*

$\rightarrow$ Orthogonality for PDBs is subsumed by "0/1" cost partitionings, putting the entire cost of each action into the PDB it affects.

($\rightarrow$ Yes, this works for arbitrary abstractions, not just for PDBs.)

Dominating Orthogonal PDBs, ctd.

Proof. $c_1, \dots, c_n$ is a cost partitioning: $\sum_{i=1}^n c_i(a) \leq c(a)$ because $c_i(a)$ is 0 unless a affects α_i, which by prerequisite is the case for at most one i.

For any s, $\sum_{i=1}^n h^{\alpha_i}(s) = \sum h[c](s)$: By definition, $\sum h[c](s) = \sum_{i=1}^n h^{\alpha_i}[c_i](s)$ so it suffices to show that, for each i, $h^{\alpha_i}(s) = h^{\alpha_i}[c_i](s)$. The latter holds because every action that affects α_i has its original cost in c_i.

Corollary (Cost Partitionings Can Dominate the Canonical PDB Heuristic). *Let Π be a planning task, let $\mathcal{C}$ be a pattern collection, and let s be a state. Then there exists a cost partitioning for Π so that $h^{\mathcal{C}}(s) \leq \sum h[c](s)$. There are cases where $h^{\mathcal{C}}(s) < \sum h[c](s)$.*

Proof. "$\leq$": Apply theorem to the additive $\{P_1, \dots, P_k\} \subseteq \mathcal{C}$ yielding the maximum in s. "<": See next slide.

$\rightarrow$ State-dependence: $h^{\mathcal{C}}$ selects the maximum additive $\{P_1, \dots, P_k\} \subseteq \mathcal{C}$ depending on the state. Hence we have to select the cost partitioning depending on the state. That is, we can't in general select a cost partitioning *once* and dominate $h^{\mathcal{C}}$ on *all* states. Example see next slide.

Strictly Dominating the Canonical PDB Heuristic

- Goal: A and B both true.
- Initial state: A and B both false.
- Actions: $carA$ effect A cost 1; $carB$ effect B cost 1; $fancyCar$ effect A and B cost 1.5.
- Patterns $P_1 = \{A\}$ and $P_2 = \{B\}$.

$\rightarrow$ What is $h^{\mathcal{C}}(I)$? $h^{P_1}(I) = h^{P_2}(I) = 1$, and P_1, P_2 are not orthogonal. So $h^{\mathcal{C}}(I) = \max(h^{P_1}(I), h^{P_2}(I)) = 1$.

$\rightarrow$ Can we improve this using cost partitioning? Yes: Setting $c_1(fancyCar) = c_2(fancyCar) = 0.75$ we get $\sum h[c](I) = 1.5 = h^*(I)$.

$\rightarrow$ This shows "<" on previous slide.

State-dependence: Say $fancyCar$ has effect p (not affecting A or B), and p is a precondition of $carA$ and $carB$. Say $P_1 = \{p, A\}$ and $P_2 = \{p, B\}$. Consider the states $s = \{A\}$ and $s' = \{B\}$. We have $h^{\mathcal{C}}(s) = h^{\mathcal{C}}(s') = 2.5$.

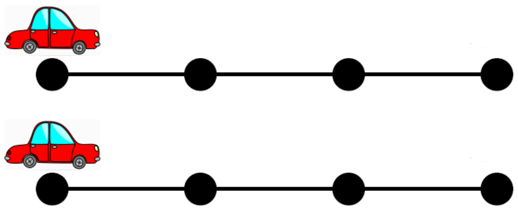
$\rightarrow$ We cannot achieve the same based on a single cost partitioning, as $fancyCar$ would have to have cost 1.5 in both patterns.

Questionnaire

Planning task: Drive both cars from left to right.

$P_1 =$ $\{carA\}$:

$P_2 =$ $\{carB\}$:



Question!

Which cost partitioning corresponds to the canonical PDB heuristic here?

→ All $carA$-moves have their full cost in h^{P_1} and 0 cost in h^{P_2}, and all $carB$-moves have their full cost in h^{P_2} and 0 cost in h^{P_1}.

→ Orthogonality corresponds to particular cost partitionings that can be computed very efficiently and that put action costs only where needed.

What is the Problem?

Given: A collection $h_1, \dots, h_n$ of admissible heuristics, and a state s.

Wanted: A cost partitioning $c_1, \dots, c_n$.

Number of candidates: Infinite.

→ Do all of these yield a good overall lower bound on $h^*(s)$? No! E.g., say $h^{P_1}(s) = 0$ and $h^{P_2}(s) = 100$ in the original task, where P_1 and P_2 are not additive. We *could* choose c_1, c_2 so that, for all a, $c_1(a) = c(a)$ and $c_2(a) = 0$. This would yield $\sum h[c](s) = 0$.

→ Many (most) cost partitionings are bad. Our challenge is to automatically find good ones.

→ The challenge is particularly vexing because ideally we want to do this for every search state s! (In particular, if we wish to dominate the canonical heuristics, cf. slides 21 and 18.)

Optimal Cost Partitioning

Definition (Optimal Cost Partitioning). *Let Π be a planning task, let $h_1, \dots, h_n$ be admissible heuristic functions for Π, and let s be a state. An optimal cost partitioning for s and $h_1, \dots, h_n$ is any cost partitioning $c_1, \dots, c_n$ for which $\sum h[c](s)$ is maximal.*

→ Optimal cost partitionings distribute costs in a way that yields the best possible lower bound, *for a given state*.

Question!

Does this definition sound completely impractical?

(A): Yes (B): No

→ Yes it does. However, it isn't! In many cases, we can compute an optimal cost partitioning efficiently.

Questionnaire

- V: $Glass_1, Glass_2 : \{Table, Hand\}$; $Empty_1, Empty_2 : \{0, 1\}$; $Wodka, Tomato : \{Bottle, Glass_1, Glass_2, Shaker\}$; $BloodyMary, Alive : \{0, 1\}$.
- Initial state I: $Glass_1 = Table$, $Glass_2 = Table$, $Empty_1 = 1$, $Empty_2 = 1$, $Wodka = Bottle$, $Tomato = Bottle$, $BloodyMary = 0$, $Alive = 1$.
- Goal G: $BloodyMary = 1$, $Alive = 1$.
- Actions A:
  $Take(x)$: pre $Glass_x = Table$; eff $Glass_x = Hand$
  $Drop(x)$: pre $Glass_x = Hand$; eff $Glass_x = Table$
  $Fill(x, y)$: pre $Glass_x = Hand$, $Empty_x = 1$, $y = Bottle$; eff $y = Glass_x$
  $Pour(x, Wodka)$: pre $Glass_x = Hand$, $Wodka = Glass_x$; eff $Wodka = Shaker$, $Empty_x = 1$
  $Pour(x, Tomato)$: pre $Glass_x = Hand$, $Tomato = Glass_x$; eff $Tomato = Shaker$, $Empty_x = 1$, $Alive = 0$
  $Shake()$: pre $Wodka = Shaker$, $Tomato = Shaker$; eff $BloodyMary = 1$

Heuristics: h^+; landmarks $\{Shake()\}$, $\{Pour(1, Wodka), Pour(2, Wodka)\}$, $\{Pour(1, Tomato), Pour(2, Tomato)\}$; PDB $h^{\{Tomato, BloodyMary, Alive\}}$.

Question!

What are the optimal cost partitionings for I here?

→ Any arbitrary cost partitioning is optimal for I, because $h^{\{Tomato, BloodyMary, Alive\}}(I) = \infty$ regardless of what the cost partitioning is.

Optimal Cost Partitioning for Landmarks

Theorem (Polynomial-Time Optimal Cost Partitioning for Landmarks). *Let Π be a planning task, let s be a state, and let $L_1, \dots, L_n$ be disjunctive action landmarks for s. Then an optimal cost partitioning for s and $h^{\mathsf{LM}}_{L_1}, \dots, h^{\mathsf{LM}}_{L_n}$ can be computed in time polynomial in $\|\Pi\|$ and n.*

Proof Sketch. The problem of finding an optimal cost partitioning $c_1, \dots, c_n$ can be formulated as a Linear Programming (LP) problem. We use LP variables $c_{i,a}$ encoding the partitioned costs, and variables h_{L_i} encoding the weight the final heuristic will count for the landmark L_i. Simple constraints ensure that $c_{i,a}$ is indeed a cost partitioning, and that the weights h_{L_i} are not larger than allowed. Maximizing $\sum_{i=1}^{n} h_{L_i}$ results in an optimal cost partitioning.

→ Selection of the cheapest action from a landmark L_i can be encoded into LP, giving a weight to each L_i. An optimal cost partitioning corresponds to an LP solution maximizing the summed-up weights.

→ Note: We assume here that the L_i are LMs for s. Corresponds to standard methods determining a set of LMs for each search state (cf. **Chapter 14**).

Optimal Cost Partitioning for Landmarks: Proof

LP variables:

- For all i and $a \in L_i$: $c_{i,a}$ [value to be assigned to $c_i(a)$].
- For all i: h_{L_i} [weight to be counted for LM L_i].

Maximize: $\sum_{i=1}^{n} h_{L_i}$ subject to **LP constraints:**

(i) For all $a \in \bigcup L_i$: $\sum_{L_i : a \in L_i} c_{i,a} \leq c(a)$. [Ensures that the solution corresponds to a cost partitioning.]

(ii) For all L_i and $a \in L_i$: $h_{L_i} \leq c_{i,a}$. [Ensures that the weight counted for each LM is at most the cost of its cheapest action.]

Let $c_{i,a}$ and h_{L_i} be the values in an optimal solution to this LP. Define $c_i := \{(a, c_{i,a}) \mid a \in A\}$. We show that $c_1, \dots, c_n$ is an optimal cost partitioning.

By (i) $c_1, \dots, c_n$ is a cost partitioning. By (ii) $h_{L_i} \leq \min_{a \in L_i} c_i(a) = h^{\mathsf{LM}}_{L_i}[c_i](s)$. As $\sum_{i=1}^{n} h_{L_i}$ is maximal and there are no other constraints on h_{L_i}, we have $h_{L_i} = \min_{a \in L_i} c_i(a)$ and thus $\sum_{i=1}^{n} h_{L_i} = \sum h[c](s)$.

Now let $c'_1, \dots, c'_n$ be any cost partitioning. Then we obtain a solution to the LP by defining $c'_{i,a} := c'_i(a)$ and $h'_{L_i} := \min_{a \in L_i} c'_i(a)$. Thus $\sum h[c'](s) \leq \sum h[c](s)$, QED.

The Tomato Zombies Strike Again!

- Goal: A and B both true.
- Initial state: A and B both false.
- Actions: cA effect A cost 1; cB effect B cost 1; fC effect A and B cost 1.5.
- Landmarks $L_1 = \{cA, fC\}$ and $L_2 = \{cB, fC\}$.

LP variables: $c_{1,cA}$, $c_{1,fC}$, $c_{2,cB}$, $c_{2,fC}$, h_{L_1}, h_{L_2}.

LP constraints:

(i) $c_{1,cA} \leq 1$; $c_{2,cB} \leq 1$; $c_{1,fC} + c_{2,fC} \leq 1.5$.

(ii) $h_{L_1} \leq c_{1,cA}$; $h_{L_1} \leq c_{1,fC}$; $h_{L_2} \leq c_{2,cB}$; $h_{L_2} \leq c_{2,fC}$.

Solution maximizing $h_{L_1} + h_{L_2}$: For example, $c_{1,cA} = 1$, $c_{1,fC} = 0.75$, $c_{2,cB} = 1$, $c_{2,fC} = 0.75$, $h_{L_1} = 0.75$, $h_{L_2} = 0.75$. In general, any assignment where $c_{1,fC} + c_{2,fC} = 1.5$, $c_{1,fC} \leq c_{1,cA} \leq 1$, $c_{2,fC} \leq c_{2,cB} \leq 1$, $h_{L_1} = c_{1,fC}$ and $h_{L_2} = c_{2,fC}$.

Optimal Cost Partitioning for PDBs

Theorem (Polynomial-Time Optimal Cost Partitioning for PDBs). *Let Π be a planning task, let s be a state, and let $\{P_1, \dots, P_n\}$ be a pattern collection. Then an optimal cost partitioning for s and $h^{P_1}, \dots, h^{P_n}$ can be computed in time polynomial in $\|\Pi\|$ and $\|\Theta^{P_1}\|, \dots, \|\Theta^{P_n}\|$.*

Proof Sketch. LP formulation: Constraints $\sum_{i=1}^{n} c_{i,a} \leq c(a)$ ensure that we get a cost partitioning. For each i, constraints $c_{i,s} = 0$ and $c_{i,t'} \leq c_{i,t} + c_{i,a}$ for each transition $t \xrightarrow{a} t'$ in Θ^{P_i} ensure that $c_{i,t}$ for any state t in Θ^{P_i} is at most the abstract cost to reach t from s (using the partitioned costs $c_{i,a}$). Constraints $h_{P_i} \leq c_{i,t}$ for all abstract goal states t in Θ^{P_i} ensure that the weight h_{P_i} counted for each P_i is at most the real abstract remaining cost of s. Maximizing $\sum_{i=1}^{n} h_{P_i}$ results in an optimal cost partitioning.

→ Cheapest paths in abstract state spaces can be encoded into LP, giving a weight to each PDB. An optimal cost partitioning corresponds to an LP solution maximizing the summed-up weights.

(→ Yes, this works for arbitrary abstractions, not just for PDBs.)

Killer-LP Tomato ZomPDBs

- Goal: A and B both true.
- Initial state: A and B both false.
- Actions: cA effect A cost 1; cB effect B cost 1; fC effect A and B cost 1.5.
- Patterns $P_1 = \{A\}$ and $P_2 = \{B\}$.

LP variables: $c_{1,cA}$, $c_{2,cA}$; $c_{1,cB}$, $c_{2,cB}$; $c_{1,fC}$, $c_{2,fC}$; $c_{1,\neg A}$, $c_{1,A}$; $c_{2,\neg B}$, $c_{2,B}$; h_{P_1}, h_{P_2}.

LP constraints:

- $c_{1,cA} + c_{2,cA} \leq 1$; $c_{1,cB} + c_{2,cB} \leq 1$; $c_{1,fC} + c_{2,fC} \leq 1.5$.
- $c_{1,\neg A} = 0$; $c_{1,A} \leq c_{1,\neg A} + c_{1,cA}$; $c_{1,A} \leq c_{1,\neg A} + c_{1,fC}$.
  $c_{2,\neg B} = 0$; $c_{2,B} \leq c_{2,\neg B} + c_{2,cB}$; $c_{2,B} \leq c_{2,\neg B} + c_{2,fC}$.
- $h_{P_1} \leq c_{1,A}$; $h_{P_2} \leq c_{2,B}$.

Solution maximizing $h_{P_1} + h_{P_2}$: For example, $c_{1,cA} = 1$, $c_{1,cB} = 0$, $c_{1,fC} = 0.75$, $c_{1,A} = 0.75$, $h_{P_1} = 0.75$; $c_{2,cA} = 0$, $c_{2,cB} = 1$, $c_{2,fC} = 0.75$, $c_{2,B} = 0.75$, $h_{P_2} = 0.75$.

Optimal Cost Partitionings for Landmarks *and* PDBs

Theorem (Polynomial-Time Optimal Cost Partitioning). *Let Π be a planning task, let s be a state, and let $h_1, \dots, h_n$ be heuristic functions for Π such that each h_i either is given by $h_i = h^{\mathsf{LM}}_{L_i}$ for a disjunctive action landmark for s, or is given by $h_i = h^{P_i}$ for a pattern P_i with abstract state space Θ^{P_i}. Then an optimal cost partitioning for s and $h_1, \dots, h_n$ can be computed in time polynomial in $\|\Pi\|$ and the size of the representation of $h_1, \dots, h_n$.*

Proof Sketch. Simply put all the LP variables and constraints described previously into a single formulation.

→ Selection of the cheapest action from a landmark L_i can be encoded into LP, giving a weight to each L_i. Cheapest paths in abstract state spaces can be encoded into LP, giving a weight to each PDB. An optimal cost partitioning corresponds to an LP solution maximizing the summed-up weights.

(→ Yes, this works for arbitrary abstractions, not just for PDBs.)

(Optimal) Cost Partitioning in Practice

Sometimes, optimal just isn't good enough: LPs can be solved in polynomial time, however may not be fast enough especially if we do it for every state during a search.

Some possible fixes:

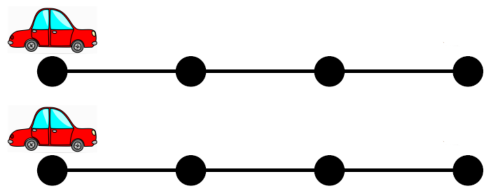
- Ditch all this and use our previous orthogonality criteria (not bad, really, at least at the moment). [Haslum *et al.* (2007)]
- Use uniform cost partitioning, distributing the cost of each action evenly over all LMs it is a member of/over all PDBs it affects (not that bad either). [Karpas and Domshlak (2009)]
- Just live with it and solve an LP in every search state (useful for highly challenging tasks if you got lots of time). [Katz and Domshlak (2010)]
- Solve an LP for initial state and/or sample states, use combination/selections of the resulting cost partitionings during search. [Katz and Domshlak (2010); Karpas *et al.* (2011)]
- For a set of abstractions, fix an order $\alpha_1, \dots \alpha_n$; saturate α_1, giving it enough costs to preserve h^{α_1}; then proceed for $\alpha_2, \dots \alpha_n$ with the left-over costs. [Seipp and Helmert (2014); Seipp *et al.* (2017)]

Questionnaire

Planning task: Drive both cars. (Can move only to the right)

$P_1 = \{carA\}$:

$P_2 = \{carB\}$:



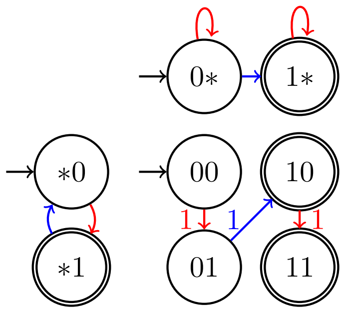
Question!

According to slide 32, the optimal cost partitioning LP has 22 variables (4 for the states t within each abstract state space, plus h_{P_1} and h_{P_2}, plus $c_{i,a}$ for all i and a). How many variables do we actually need to find an optimal cost partitioning?

→ We don't need $c_{i,a}$ if a does not affect P_i, removing 6 variables. If a affects only a single P_i then we don't need any $c_{i,a}$, so just 10. $c_{i,s} = 0$ is fixed so just 8.

→ Orthogonality criteria help to keep cost partitioning LPs small.

Where Optimal Cost Partitioning Is Not Good Enough



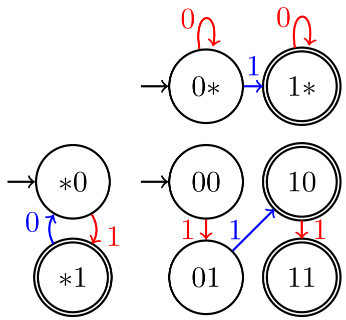
- Initial state: $\{A = 0, B = 0\}$.
- Goal: $\{A = 1\}$.
- Actions:
  cB pre $\{B = 0\}$ eff $\{B = 1\}$;
  cA pre $\{A = 0, B = 1\}$ eff
  $\{A = 1, B = 0\}$
- Patterns $P_1 = \{A\}$ and $P_2 = \{B\}$.

Question!

What is the optimal cost partitioning for I?

$h(I) = 0 + 1 = 1 < 2 = h^*(I)$

Where Optimal Cost Partitioning Is Not Good Enough



- Initial state: $\{A = 0, B = 0\}$.
- Goal: $\{A = 1\}$.
- Actions:
  cB pre $\{B = 0\}$ eff $\{B = 1\}$;
  cA pre $\{A = 0, B = 1\}$ eff
  $\{A = 1, B = 0\}$
- Patterns $P_1 = \{A\}$ and $P_2 = \{B\}$.

Question!

What is the optimal cost partitioning for I?

$h(I) = 0 + 1 = 1 < 2 = h^*(I)$

General Cost Partitioning

Cost functions usually non-negative

- Makes intuitively sense: original costs are non-negative
- But: not necessary for cost-partitioning!

Definition (General Cost Partitioning). *Let Π be a planning task with actions A and cost function c. An ensemble of functions $c_1, \dots, c_n : A \mapsto \mathbb{R}$ is a general cost partitioning for Π if, for all $a \in A$, $\sum_{i=1}^{n} c_i(a) \leq c(a)$.*

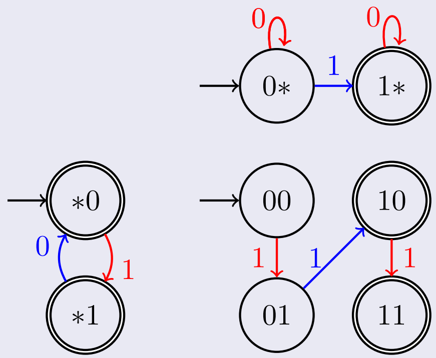
Theorem (General Partitioned Sums are Admissible). *Let Π be a planning task, and let $h_1, \dots, h_n$ be heuristic functions for Π. If $c_1, \dots, c_n$ is a general cost partitioning for Π, and if $h_i[c_i]$ is admissible for all i, then the partitioned sum $\sum h[c]$ is admissible.*

(Proof omitted.)

→More powerful than non-negative cost partitioning (optimal cost partitioning maximizes the objective value so removing constraints can only increase the heuristic value)

General Cost Partitioning: Example

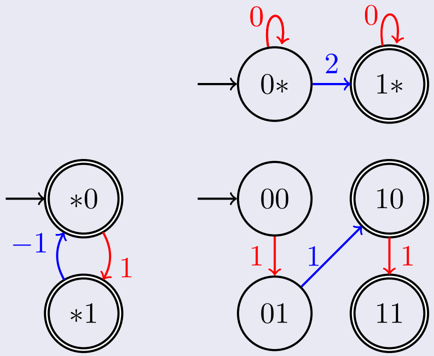
Example



Heuristic value: $0 + 1 = 1$

General Cost Partitioning: Example

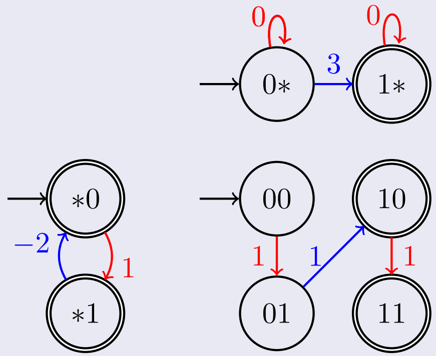
Example



Heuristic value: $0 + 2 = 2$

General Cost Partitioning: Example

Example



Heuristic value: $-\infty + 3 = -\infty$

Where Cost Partitioning Fails

- Goal: A, B, C true.
- Initial state: A, B, C false.
- Actions (unit cost):
  $carAB$ effect A and B;
  $carBC$ effect B and C;
  $carAC$ effect A and C.
- Landmarks $L_1 = \{carAB, carAC\}$,
  $L_2 = \{carAB, carBC\}$,
  $L_3 = \{carAC, carBC\}$.

Optimal cost partitioning: $h(I) = 1.5 < h^*(I)$ (for $h_{L_1} = h_{L_2} = h_{L_3} = 0.5$).

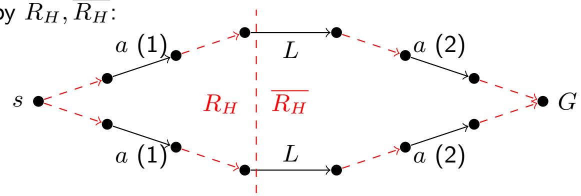
Minimum cost hitting set: $h(I) = 2 = h^*(I)$! E.g., $H := \{carAB, carAC\}$.

Hitting sets are admissible: *Let $L_1, \dots, L_n$ be disjunctive action landmarks for s. Let H be a minimum-cost hitting set. Then $\sum_{a \in H} c(a) \leq h^*(s)$.*
(Simply because by definition every plan must hit every L_i.)

From Landmarks to h^+!

Theorem. *Let s be a state, and let $L_1, \dots, L_n$ be the collection of all delete relaxation disjunctive action landmarks for s. Let H be a minimum-cost hitting set. Then $\sum_{a \in H} c(a) = h^+(s)$.*

Proof. "$\leq$": Every relaxed plan must hit every L_i. For "$\geq$", we prove that any hitting set H contains a relaxed plan. With $R_H := \{p \mid p$ *can be reached in delete relaxation using only* $H\}$, assume to the contrary that $G \not\subseteq R_H$. Choose 1 fact from the goal and each action precondition, using a fact outside R_H where possible. Consider the graph over facts with arcs (p, a, q) where $p \in pre_a$ and $q \in e\!f\!f_a$, and consider the cut L defined by $R_H, \overline{R_H}$:



L is a LM for s: We cannot reach the goal without using one of these actions. However, consider any $a \in H$. Case (1): If $pre_a \subseteq R_H$, then $add_a \subseteq R_H$ because $a \in H$. So $a \notin L$. Case (2): If $pre_a \not\subseteq R_H$, then we selected $p \in pre_a \setminus R_H$. So, again, $a \notin L$. Altogether, H does not hit L, in contradiction.

So What?

- Hitting sets over LMs were first proposed by [Bonet and Helmert (2010)].
- Hitting sets over LMs dominate the optimal cost partitioning. This is because, for any action a, the total weight (after cost partitioning) of all LMs a participates in is bounded by $c(a)$. So if we hit all LMs then we got an upper bound on the cost-partitioning heuristic.
- There are constructive methods to find "complete" sets of landmarks, i.e., methods which guarantee that the minimum-cost hitting set will deliver h^+. This is nowadays the state-of-the-art method to compute h^+ [Haslum *et al.* (2012)].
- A similar result does *not* hold for h^* even if we somehow found all (non-delete-relaxed) disjunctive action LMs. This is because, in the original planning task, we may have to apply the same action *more than once*.
- In practice, hitting sets over LMs tend to be computationally too expensive (for every state, apart from finding all the LMs we have to solve the **NP**-hard minimum-cost hitting set problem ...).

Summary

- A cost partitioning distributes the cost of each action across n otherwise identical planning tasks. This can be used to admissibly sum up *any* ensemble of admissible heuristic functions.
- For every state and ensemble of PDB heuristics, there exists a cost partitioning that dominates the canonical PDB heuristic; the domination can be strict.
- The same is true of the canonical LM heuristic.
- Optimal cost partitionings distribute action costs such that the lower bound for a given state is maximal.
- For PDBs and LMs, and for their combination, optimal cost partitionings can be computed in polynomial time by Linear Programming.
- In practice, computing optimal cost partitionings for every search state typically is too costly, and we need to approximate.

Historical Remarks

- The admissible combination of lower bounds has a long history. Famous instances pertain to additive PDBs in Game playing [Felner *et al.* (2004)].
- In planning, this story also started with additive PDBs [Edelkamp (2001); Haslum *et al.* (2007)], then was extended to h^m among others [Haslum *et al.* (2005)]. The intuition always was to design the heuristics in a way making them *independent*.
- When I was in some project meeting somewhere in about 2005, someone from outside the area said "But what if we count each move only half in each of the heuristics?". The remark was received with confusion, then forgotten about.
- Then Michael & Carmel [Katz and Domshlak (2008)] suddenly came along and told us we'd been looking at 0/1 cost partitionings all the time, and how to find optimal general ones efficiently using LP.
- Since then, various works towards making this practical, cf. slide 35.
- Cost partitioning is not specific to planning, can be applied anywhere!

Reading

- *Optimal Additive Composition of Abstraction-Based Admissible Heuristics* [Katz and Domshlak (2008)].

  Available at:

  `http://fai.cs.uni-saarland.de/katz/papers/icaps08b.pdf`

  Content: Original paper proposing cost partitioning, and showing that, for certain classes of heuristics, optimal cost partitionings can be computed in polynomial time using Linear Programming. Specifically, the paper established this for abstractions as handled in this course, as well as for implicit abstractions represented through planning task fragments identified based on the causal graph.

Reading, ctd.

- *Cost-Optimal Planning with Landmarks* [Karpas and Domshlak (2009)].

  Available at:

  http://iew3.technion.ac.il/~dcarmel/Papers/Sources/ijcai09a.pdf

  Content: The "alarm clock" waking LMs up to the modern age of cost-optimal planning (cf. → **Chapter 14**). Introduces cost partitioning for elementary landmarks heuristics, and the computation of optimal cost partitionings for such heuristics using Linear Programming. Introduces uniform cost partitioning, which is used in the experiments due to being more runtime-effective.

Reading, ctd.

- *Diverse and Additive Cartesian Abstraction Heuristics* [Seipp and Helmert (2014)].

  Available at:

  http://ai.cs.unibas.ch/papers/seipp-helmert-icaps2014.pdf

  Content: Introduces the current state of the art technique for cost partitioning with abstraction heuristics, saturated cost partitioning, which partitions costs according to what is actually needed to preserve the abstraction heuristic.

References I

Blai Bonet and Malte Helmert. Strengthening landmark heuristics via hitting sets. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, pages 329–334, Lisbon, Portugal, August 2010. IOS Press.

Stefan Edelkamp. Planning with pattern databases. In A. Cesta and D. Borrajo, editors, *Proceedings of the 6th European Conference on Planning (ECP'01)*, pages 13–24. Springer-Verlag, 2001.

Ariel Felner, Richard Korf, and Sarit Hanan. Additive pattern database heuristics. *Journal of Artificial Intelligence Research*, 22:279–318, 2004.

Patrik Haslum, Blai Bonet, and Héctor Geffner. New admissible heuristics for domain-independent planning. In Manuela M. Veloso and Subbarao Kambhampati, editors, *Proceedings of the 20th National Conference of the American Association for Artificial Intelligence (AAAI'05)*, pages 1163–1168, Pittsburgh, Pennsylvania, USA, July 2005. AAAI Press.

References II

Patrik Haslum, Adi Botea, Malte Helmert, Blai Bonet, and Sven Koenig. Domain-independent construction of pattern database heuristics for cost-optimal planning. In Adele Howe and Robert C. Holte, editors, *Proceedings of the 22nd National Conference of the American Association for Artificial Intelligence (AAAI'07)*, pages 1007–1012, Vancouver, BC, Canada, July 2007. AAAI Press.

Patrik Haslum, John Slaney, and Sylvie Thiébaux. Minimal landmarks for optimal delete-free planning. In Blai Bonet, Lee McCluskey, José Reinaldo Silva, and Brian Williams, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS'12)*, pages 353–357. AAAI Press, 2012.

Erez Karpas and Carmel Domshlak. Cost-optimal planning with landmarks. In Craig Boutilier, editor, *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, pages 1728–1733, Pasadena, California, USA, July 2009. Morgan Kaufmann.

Erez Karpas, Michael Katz, and Shaul Markovitch. When optimal is just not good enough: Learning fast informative action cost-partitionings. In Fahiem Bacchus, Carmel Domshlak, Stefan Edelkamp, and Malte Helmert, editors, *Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS'11)*, pages 122–129. AAAI Press, 2011.

References III

Michael Katz and Carmel Domshlak. Optimal additive composition of abstraction-based admissible heuristics. In Jussi Rintanen, Bernhard Nebel, J. Christopher Beck, and Eric Hansen, editors, *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS'08)*, pages 174–181. AAAI Press, 2008.

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

Jendrik Seipp and Malte Helmert. Diverse and additive Cartesian abstraction heuristics. In Steve Chien, Minh Do, Alan Fern, and Wheeler Ruml, editors, *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS'14)*. AAAI Press, 2014.

Jendrik Seipp, Thomas Keller, and Malte Helmert. Narrowing the gap between saturated and optimal cost partitioning for classical planning. In Satinder Singh and Shaul Markovitch, editors, *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*, pages 3651–3657. AAAI Press, February 2017.