# Optimal Additive Composition of Abstraction-based Admissible Heuristics

Michael Katz    Carmel Domshlak

Technion, IE&M

ICAPS-08

# Context

Abstraction-based Admissible Heuristics for Cost-Optimal Classical Planning

## Classical Planning

Planning task is 5-tuple $\langle V, A, \mathcal{C}, s^0, G\rangle$:

- $V$: finite set of finite-domain state variables
- $A$: finite set of actions of form $\langle \text{pre}, \text{eff}\rangle$ (preconditions/effects; partial variable assignments)
- $\mathcal{C} : A \mapsto \mathbb{R}^{0+}$ captures action cost
- $s^0$: initial state (variable assignment)
- $G$: goal description (partial variable assignment)

# Context

Abstraction-based Admissible Heuristics for Cost-Optimal Classical Planning



## Cost-Optimal Planning

**Given:** planning task $\Pi = \langle V, A, s^0, G \rangle$

**Find:** operator sequence $a_1 \ldots a_n \in A^*$
transforming $s^0$ into some state $s_n \supseteq G$,
while minimizing $\sum_{i=1}^{n} \mathcal{C}(a_i)$

**Approach:** A* $+$ admissible heuristic $h : S \mapsto \mathbb{R}^{0+}$

# Context

Abstraction-based Admissible Heuristics for Cost-Optimal Classical Planning

**Abstraction heuristics**

Heuristic estimate is goal distance in abstracted state space $S'$

| | | |
|---|---|---|
| Examples: | $h^+$ (ignore-deletes) | NP-hard |
| | PDBs (pattern databases) | [...,Ed01,...] |
| | constrained PDBs | [HBG05] |
| | merge-and-shrink | [HHH07] |
| | structural patterns | [KD08] |
| Examples NOT: | $h^m$ | [HG00] |
| | some LP relaxations | [...] |

# Composing Multiple Heuristics

"Two heads are better than one"



## State of the art

Input: problem Π, admissible heuristics $h_1, \ldots, h_m$, state $s$

MAX use $\max_{i=1}^{m} h_i(s|\Pi)$

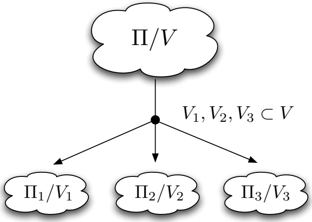
ADD use $\sum_{i=1}^{m} h_i(s|\Pi_i)$ for some transformations $\Pi_1, \ldots, \Pi_m$ of Π

# Projections

Widely-exploited idea: projections
$\rightsquigarrow$ map states to abstract states with perfect hash function

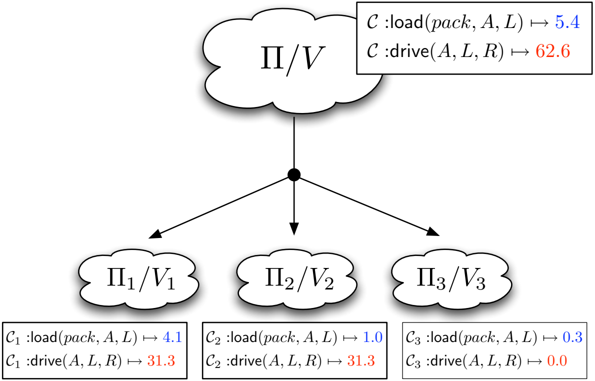
**Definition (projection)**

Projection $\Pi^{[V']}$ to variables $V' \subseteq V$: homomorphism $\alpha$ where $\alpha(s) = \alpha(s')$ iff $s$ and $s'$ agree on $V'$




Each $a \in A$ satisfies $\mathcal{C}(a) \geq \sum_{i=1}^{m} \mathcal{C}_i(a^{[V_i]})$

# Action-Cost Partitioning: Back to Projections




need selecting a good action-cost partition

$\rightsquigarrow$ optimal action-cost partition?

# Optimizing Action-Cost Partitioning



## Pitfalls

- ☹ infinite space of choices
- ☹ decision process should be fully unsupervised
- ☹ decision process should be state-dependent

$\leadsto$ *"determining which abstractions [action-cost partitions] will produce additives that are better than max over standards is still a big research issue."* (Yang *et al.*, JAIR, 2008)

# Composing Multiple Heuristics

"Two heads are better than one"



## State of the art

Input: problem $\Pi$, admissible heuristics $h_1, \ldots, h_m$, state $s$

MAX use $\max_{i=1}^{m} h_i(s|\Pi)$

ADD use $\sum_{i=1}^{m} h_i(s|\Pi_i)$ for some transformations $\Pi_1, \ldots, \Pi_m$ of $\Pi$

## Major Question

MAX or ADD, and if ADD, then add what?

Here we aim at answering this question

# Some Basic Formalism

## Transition graph

TG-structure $\mathcal{T} = (S, L, Tr, s^0, S^\star)$:

- $S$: finite set of states
- $L$: finite set of transition labels
- $Tr \subseteq S \times L \times S$: labelled transitions
- $s^0 \in S$: initial state
- $S^\star \subseteq S$: goal states

Transition graph $\langle \mathcal{T}, \varpi \rangle$:

- $\mathcal{T}$: TG-structure with labels $L$
- transition cost function $\varpi : L \mapsto \mathbb{R}^{0+}$

(Transition graph of planning task defined in the obvious way.)

# Some Basic Formalism

## Transition graph

TG-structure $\mathcal{T} = (S, L, Tr, s^0, S^\star)$:

- $S$: finite set of states
- $L$: finite set of transition labels
- $Tr \subseteq S \times L \times S$: labelled transitions
- $s^0 \in S$: initial state
- $S^\star \subseteq S$: goal states

Transition graph $\langle \mathcal{T}, \varpi \rangle$:

- $\mathcal{T}$: TG-structure with labels $L$
- transition cost function $\varpi : L \mapsto \mathbb{R}^{0+}$

(Transition graph of planning task defined in the obvious way.)

# Additive Abstractions



**Definition (additive abstractions)**

Additive abstraction of transition graph $\langle \mathcal{T}, \varpi \rangle$ is $\{\langle\langle \mathcal{T}_i, \varpi_i \rangle, \alpha_i \rangle\}_{i=1}^{m}$ where

- $\langle \mathcal{T}_i, \varpi_i \rangle$: transition graph
- $\alpha_i$ maps states of $\mathcal{T}$ to states of $\mathcal{T}_i$ such that
  - initial state maps to initial state
  - goal states map to goal states
- holds $\sum_{i=1}^{m} d(\alpha_i(s), \alpha_i(s')) \leq d(s, s')$

$h(s) = \sum_{i=1}^{m} d(\alpha_i(s), S_i^{\star})$ is (trivially) admissible

# Here We Go

## Main Idea

Instead of searching each abstract transition graph $\langle \mathcal{T}_i, \varpi_i \rangle$ given an action cost partition and using dynamic programming

1. compile SSSP problem over each TG-structure $\mathcal{T}_i$ into a linear program $\mathscr{L}_i$ with action-costs being free variables
2. combine $\mathscr{L}_1, \ldots, \mathscr{L}_m$ with additivity constraints $\mathcal{C}(a) \geq \sum_{i=1}^{m} \mathcal{C}_i(a^{[V_i]})$
3. solution of the joint LP $\leadsto$ $\leadsto h(s)$ under optimal action-cost partition

# Here We Go



## Main Idea

Instead of searching each abstract transition graph $\langle \mathcal{T}_i, \varpi_i \rangle$ given an action cost partition and using dynamic programming

1. compile SSSP problem over each TG-structure $\mathcal{T}_i$ into a linear program $\mathscr{L}_i$ with action-costs being free variables
2. combine $\mathscr{L}_1, \ldots, \mathscr{L}_m$ with additivity constraints $\mathcal{C}(a) \geq \sum_{i=1}^{m} \mathcal{C}_i(a^{[V_i]})$
3. solution of the joint LP $\rightsquigarrow$
$\rightsquigarrow$ $h(s)$ under optimal action-cost partition

# Single-Source Shortest Paths: LP Formulation



## LP formulation

Given: digraph $G = (N, E)$, source node $v \in N$

LP variables: $d(v') \rightsquigarrow$ shortest-path length from $v$ to $v'$

LP:

$$\max_{\vec{d}} \sum_{v'} d(v')$$

$$\text{s.t. } d(v) = 0$$

$$d(v'') \leq d(v') + w(v', v''), \ \ \forall (v', v'') \in E$$

# Step 1: Compile SSSP over $\mathcal{T}_i$ into $\mathscr{L}_i$

## LP formulation

Given: TG-structure $\mathcal{T}_i$, state $s$

LP variables: $\{d(s') \mid s' \in S_i\} \cup \{d(S_i^\star)\} \cup \{w(a,i)\}$

LP:

$$\max \quad d(S_i^\star)$$

$$\text{s.t.} \quad \begin{cases} d(s') \leq d(s'') + w(a,i), & \forall \langle s'', a, s' \rangle \in Tr_i \\ d(s') = 0, & s' = s^{[V_i]} \\ d(S_i^\star) \leq d(s'), & s' \in S_i^\star \end{cases}$$

# Step 2: Properly combine $\{\mathscr{L}_i\}_{i=1}^m$

## LP formulation

**Given:** TG-structures $\{\mathcal{T}_i\}_{i=1}^m$, state $s$

**LP variables:** $\bigcup_{i=1}^m \{d(s') \mid s' \in S_i\} \cup \{d(S_i^\star)\} \cup \{w(a,i)\}$

**LP:**

$$\max \sum_{i=1}^m d(S_i^\star)$$

$$\text{s.t.} \quad \forall i \quad \begin{cases} d(s') \leq d(s'') + w(a,i), & \forall \langle s'', a, s' \rangle \in Tr_i \\ d(s') = 0, & s' = s^{[V_i]} \\ d(S_i^\star) \leq d(s'), & s' \in S_i^\star \end{cases}$$

$$\forall a \in A : \quad \sum_{i=1}^m w(a,i) \leq \mathcal{C}(a)$$

# Optimizing Action-Cost Partitioning: Generalization

General theory of LP-optimizable ensembles
of additive heuristic functions

- Warning: Any reduction to LP is not enough
  $\leadsto$ requires (surprising) relation between polyhedron and planning problem

# Optimizing Action-Cost Partitioning: Generalization

General theory of LP-optimizable ensembles
of additive heuristic functions

- Warning: Any reduction to LP is not enough
- Works as above for
  - projection and variable-domain abstraction (PDB) heuristics
  - constrained PDBs heuristics (Haslum *et al.*, 2005)
  - merge-and-shrink abstractions (Helmert *et al.*, 2007)

# Optimizing Action-Cost Partitioning: Generalization

General theory of LP-optimizable ensembles
of additive heuristic functions

- Warning: Any reduction to LP is not enough
- Works as above for
  - projection and variable-domain abstraction (PDB) heuristics
  - constrained PDBs heuristics (Haslum *et al.*, 2005)
  - merge-and-shrink abstractions (Helmert *et al.*, 2007)
- Suitable poly-size LPs $\mathscr{L}_i$ exist also for
  1. fork-decomposition structural patterns (Katz & Domshlak, 2008)
  2. tree-COP reducible fragments of tractable cost-optimal planning (Katz & Domshlak, 2007)
  3. ...

# Summary

## State of the art

Input: problem $\Pi$, admissible heuristics $h_1, \ldots, h_m$, state $s$

MAX use $\max_{i=1}^{m} h_i(s|\Pi)$

ADD use $\sum_{i=1}^{m} h_i(s|\Pi_i)$ for some transformations $\Pi_1, \ldots, \Pi_m$ of $\Pi$

## Major Question

MAX or ADD, and if ADD, then add what?

For many (all ?) abstraction-based heuristics,
the question is closed