Structural Patterns Heuristics via Fork Decomposition

Michael Katz   Carmel Domshlak

Technion, IE&M

ICAPS-08

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

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

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

Cost-Optimal Planning

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

Find: operator sequence $a_1 \dots a_n \in A^*$
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'

Well-known: projection (pattern database) heuristics
Here we: both generalize and enhance them

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

Transition Graphs

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.)

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

Transition Graphs

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.)

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

(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
- α_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 $\sum_{i=1}^{m} d(\alpha_i(s), \alpha_i(s')) \leq d(s, s')$

Abstraction heuristic:

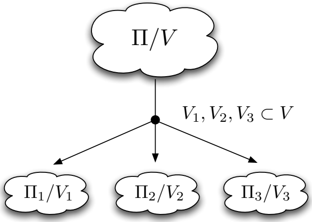
$h(s) = \sum_{i=1}^{m} d(\alpha_i(s), S_i^\star)$ is (trivially) admissible

Projections

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

Definition (projection)

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



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

Problems of Projections

No tricks: abstract spaces are searched exhaustively

$\rightsquigarrow$ must keep number of reflected variables in each projection small ($\leq O(\log(|V|))$)

$\rightsquigarrow$ (often) price in heuristic accuracy in long-run

Introduction

Abstractions

Projections

Structural Abstractions

Performance

Summary

Structural Abstraction Heuristics: Main Idea

Objective

(Katz & D, 2008a):

Instead of perfectly reflecting a few state variables, reflect many (up to $\Theta(|V|)$) state variables, BUT

- ♠ guarantee abstract space can be searched (implicitly) in poly-time

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

Structural Abstraction Heuristics: Main Idea

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

Objective

(Katz & D, 2008a):

Instead of perfectly reflecting a few state variables, reflect many (up to $\Theta(|V|)$) state variables, BUT

- ♠ guarantee abstract space can be searched (implicitly) in poly-time

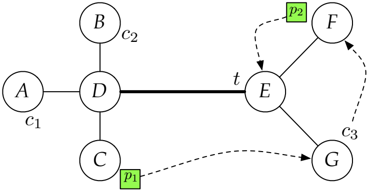
How

Abstracting Π by an instance of a tractable fragment of cost-optimal planning

- ☹ not many such known tractable fragments
- ☺ should find more, and useful for us!

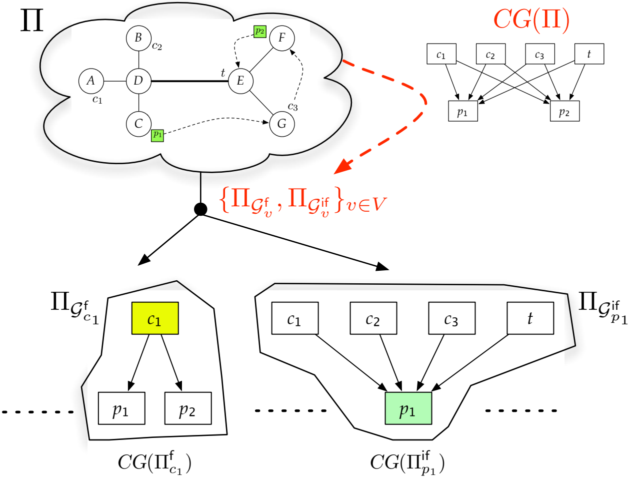
Running Example

Adapted from Malte Helmert



$$
\begin{aligned}
V &= \{p_1, p_2, c_1, c_2, c_3, t\} \\
dom(p_1) &= dom(p_2) = \{A, B, C, D, E, F, G, c_1, c_2, c_3, t\} \\
dom(c_1) &= dom(c_2) = \{A, B, C, D\} \\
dom(c_3) &= \{E, F, G\} \\
dom(t) &= \{D, E\} \\
s^0, G &\mapsto \text{see picture} \\
A &\mapsto \text{loads, unloads, single-segment movements}
\end{aligned}
$$

Fork-Decomposition (Additive Abstractions)



+ ensuring proper action cost partitioning

Works?
Problem!

Forks and Inverted Forks are Hard ...

- ☹ Even non-optimal planning for problems with fork and inverted fork causal graphs is NP-complete (Domshlak & Dinitz, 2001).
- ☹ Even if the domain-transition graphs of all variables are strongly connected, optimal planning for forks and inverted forks remains NP-hard (Helmert, 2003-04).

$\rightsquigarrow$ Shall we give up?

Tractable Cases of Planning with Forks

Theorem (forks)

Cost-optimal planning for fork problems with root $r \in V$ is poly-time if

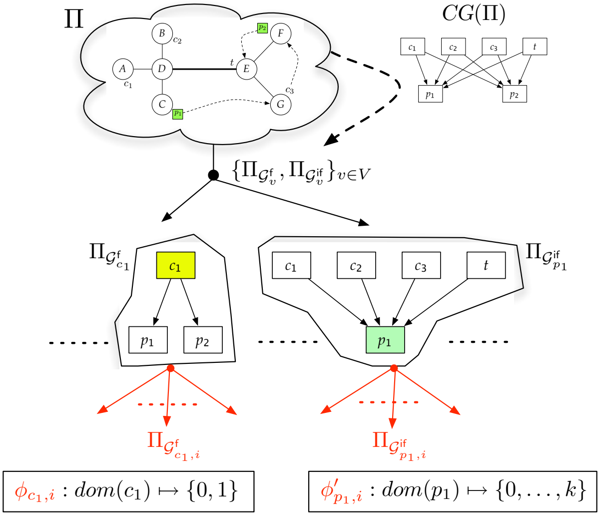
(i) $|dom(r)| = 2$, *or*

(ii) *for all* $v \in V$*, we have* $|dom(v)| = O(1)$,

Theorem (inverted forks)

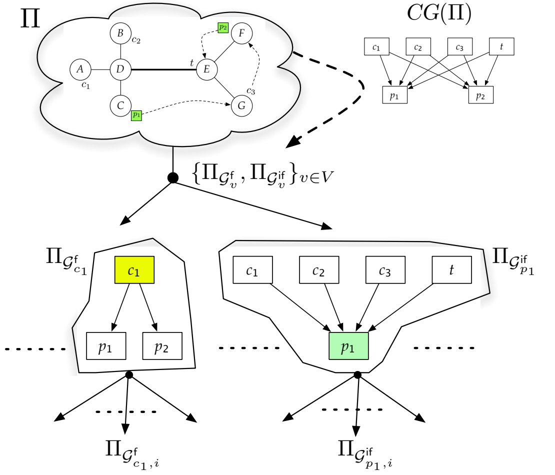
Cost-optimal planning for (1-dependent) inverted fork problems with root $r \in V$ is poly-time if $|dom(r)| = O(1)$.

Mixing Causal-Graph & Variable-Domain Decompositions



+ ensuring proper action cost partitioning

Back to our example



$\phi_{c_1,i} : dom(c_1) \mapsto \{0, 1\}$

$\phi'_{p_1,i} : dom(p_1) \mapsto \{0, 1, 2\}$

Informative?

(Intractable) Fork Decomposition

$d(s^0, S_G) = 19$ $h_{\mathsf{max}} = 8$ $h^2 = 13$ $h^{\mathfrak{FI}} = 15$

- h_{max} (Bonet & Geffner, 2001)
- h^2 (Haslum & Geffner, 2000)

Informative?

(Intractable) Fork Decomposition

$d(s^0, S_G) = 19 \quad h_{\mathsf{max}} = 8 \quad h^2 = 13 \quad h^{\mathfrak{FI}} = 15$

(Tractable) Fork + Variable-Domains Decomposition

$d(s^0, S_G) = 19 \quad h_{\mathsf{max}} = 8 \quad h^2 = 13 \quad h^{\mathfrak{FI}} = 16$

Hmm ... what?

Further abstraction gives a more precise estimate??

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

Informative?

(Intractable) Fork Decomposition

$d(s^0, S_G) = 19 \quad h_{\mathsf{max}} = 8 \quad h^2 = 13 \quad h^{\mathfrak{FJ}} = 15$

(Tractable) Fork + Variable-Domains Decomposition

$d(s^0, S_G) = 19 \quad h_{\mathsf{max}} = 8 \quad h^2 = 13 \quad h^{\mathfrak{FJ}} = 16$

Hmm … yes, that is possible!

Variable-domains abstraction may eliminate certain dependencies between the variables
$\leadsto$ less dependencies $\leadsto$ less action representatives $\leadsto$ less action cost erosion $\leadsto$ (potentially) higher estimate

Performance Evaluation

Option 1: Empirical evaluation

Implement h, plug into A*, test (comparatively) on standard benchmark suites

- ☺ standard approach, per-problem-instance comparison
- ☹ no conclusions *a la*
  "h expands fewer nodes than h' on a benchmark suite X"

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

Performance Evaluation

Option 1: Empirical evaluation

Implement h, plug into A*, test (comparatively) on standard benchmark suites

Option 2: Asymptotic performance analysis (Helmert and Mattmüller, 2008)

Given suite $\mathcal{D}$ and heuristic h, find a value $\alpha(h, \mathcal{D}) \in [0, 1]$ such that

(i) for all states s in all problems $\Pi \in \mathcal{D}$,
$h(s) \geq \alpha(h, \mathcal{D}) \cdot h^*(s) + o(h^*(s))$

(ii) there exist $\{\Pi_n\}_{n \in \mathbb{N}} \subseteq \mathcal{D}$ and solvable states $\{s_n\}_{n \in \mathbb{N}}$ with $s_n \in \Pi_n$, $\lim_{n \to \infty} h^*(s_n) = \infty$, and
$h(s_n) \leq \alpha(h, \mathcal{D}) \cdot h^*(s_n) + o(h^*(s_n))$

Asymptotic Performance Ratios
Selected benchmark suites

Domain	h^+	h^k	h^{PDB}	$h^{\mathsf{PDB}}_{\mathsf{add}}$	$h^{\mathcal{F}}$	$h^{\mathcal{I}}$	$h^{\mathcal{FI}}$
GRIPPER	2/3	0	0	2/3	2/3	1/2	2/3
LOGISTICS	3/4	0	0	1/2	1/2	1/2	1/2
BLOCKSWORLD	1/4	0	0	0	0	0	0
MICONIC	6/7	0	0	1/2	5/6	1/2	1/2
SATELLITE	1/2	0	0	1/6	1/6	1/6	1/6

ratios for h^+, h^k, h^{PDB}, $h^{\mathsf{PDB}}_{\mathsf{add}}$ are by Helmert and Mattmüller, 2008.

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

Asymptotic Performance Ratios
Selected benchmark suites

Domain	h^+	h^k	h^{PDB}	$h^{\mathsf{PDB}}_{\mathsf{add}}$	$h^{\mathfrak{F}}$	$h^{\mathfrak{I}}$	$h^{\mathfrak{FI}}$
Gripper	2/3	0	0	2/3	2/3	1/2	2/3
Logistics	3/4	0	0	1/2	1/2	1/2	1/2
Blocksworld	1/4	0	0	0	0	0	0
Miconic	6/7	0	0	1/2	5/6	1/2	1/2
Satellite	1/2	0	0	1/6	1/6	1/6	1/6

$h^{\mathsf{PDB}}_{\mathsf{add}}$: optimal, manually-selected set of projections

$h^{\mathfrak{FI}}$: non-parametric set of abstractions
basic variable-domain abstractions to binary/ternary

Summary

Introduction
Abstractions
Projections
Structural Abstractions
Performance
Summary

What we do

From small projections to large structural abstractions

Future work

- more tractability results for cost-optimal planning!
- optimization of variable-domains abstraction
- approximation-oriented structural patterns
- ...
- implementation and empirical evaluation