Focusing on What Really Matters: Irrelevance Pruning in M&S

Álvaro Torralba, Peter Kissmann
Saarland University, Germany

SoCS 2015, June 11
Session with ICAPS 2015

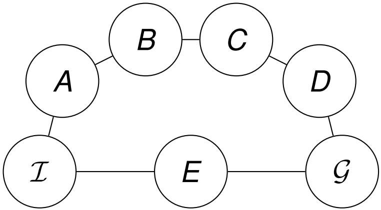
Motivation: Irrelevance Pruning

- Last Tuesday: h^2-based preprocessor
  - Simplify the task in a preprocessing step
  - Remove operators that cannot possibly belong to any plan
  - Very useful!!!!
- Today: Can we simplify the tasks even further?

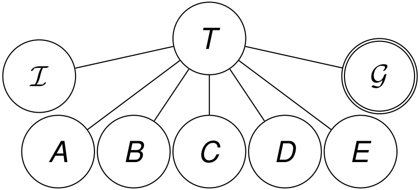
Motivation: Irrelevance Pruning

- Last Tuesday: h^2-based preprocessor
  - Simplify the task in a preprocessing step
  - Remove operators that cannot possibly belong to any plan
  - Very useful!!!!

- Today: Can we simplify the tasks even further?



(Truck)

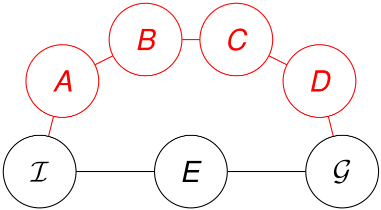


(Package)

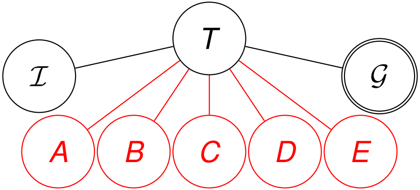
Motivation: Irrelevance Pruning

- Last Tuesday: h^2-based preprocessor
  - Simplify the task in a preprocessing step
  - Remove operators that cannot possibly belong to any plan
  - Very useful!!!!

- Today: Can we simplify the tasks even further?



(Truck)

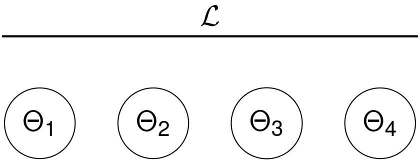


(Package)

Merge-and-Shrink Heuristic

An admissible abstraction heuristic for cost-optimal planning

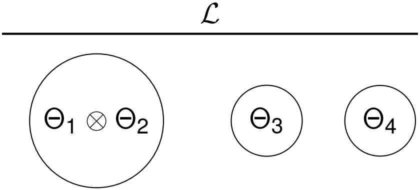
1. Start with the projection over variables: v_1, v_2, v_3, v_4
2. Merge: replace Θ_i and Θ_j by their product
3. Shrink: replace Θ_i by its abstraction $\alpha(\Theta_i)$



Merge-and-Shrink Heuristic

An admissible abstraction heuristic for cost-optimal planning

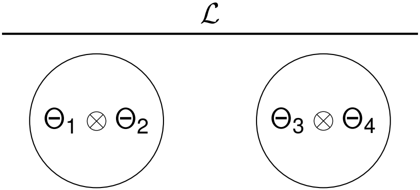
1. Start with the projection over variables: v_1, v_2, v_3, v_4
2. Merge: replace Θ_i and Θ_j by their product
3. Shrink: replace Θ_i by its abstraction $\alpha(\Theta_i)$



Merge-and-Shrink Heuristic

An admissible abstraction heuristic for cost-optimal planning

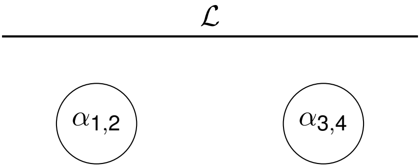
1. Start with the projection over variables: v_1, v_2, v_3, v_4
2. Merge: replace Θ_i and Θ_j by their product
3. Shrink: replace Θ_i by its abstraction $\alpha(\Theta_i)$



Merge-and-Shrink Heuristic

An admissible abstraction heuristic for cost-optimal planning

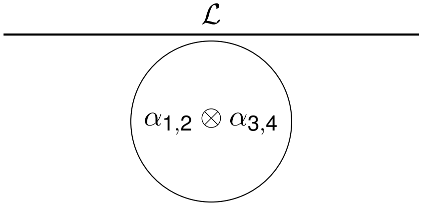
1. Start with the projection over variables: v_1, v_2, v_3, v_4
2. Merge: replace Θ_i and Θ_j by their product
3. Shrink: replace Θ_i by its abstraction $\alpha(\Theta_i)$



Merge-and-Shrink Heuristic

An admissible abstraction heuristic for cost-optimal planning

1. Start with the projection over variables: v_1, v_2, v_3, v_4
2. Merge: replace Θ_i and Θ_j by their product
3. Shrink: replace Θ_i by its abstraction $\alpha(\Theta_i)$

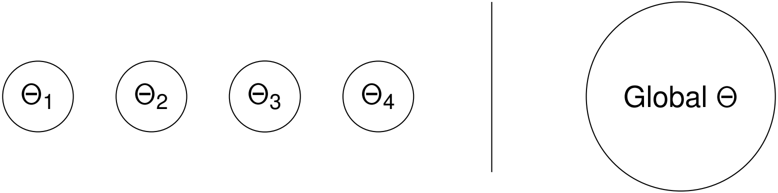


Simulation-Based Dominance Pruning

Label-dominance simulation (Torralba and Hoffmann, IJCAI 2015):

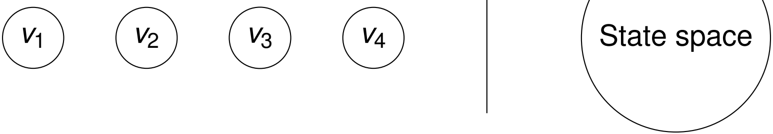
1. Use M&S to compute a partition of the problem: $\{\Theta_1, \ldots, \Theta_k\}$
2. Compute label-dominance simulation relation: $\{\preceq_1, \ldots, \preceq_k\}$
   - Label dominance: l dominates l' in Θ_i if for any $s \xrightarrow{l} t$ exists $s \xrightarrow{l'} t'$ s.t. $t \preceq t'$
   - State dominance $s \preceq t$: For any $s \xrightarrow{l} s'$, exists $t \xrightarrow{l'} t'$ s.t.:
     - $t \preceq t'$
     - $c(l') \leq c(l)$
     - l' dominates l in the rest of the problem
3. In A^*, prune any s s.t. $s \preceq t$, $g(s) \geq g(t)$ for some t

Merge-and-Shrink Framework (Sievers et al. 2014)

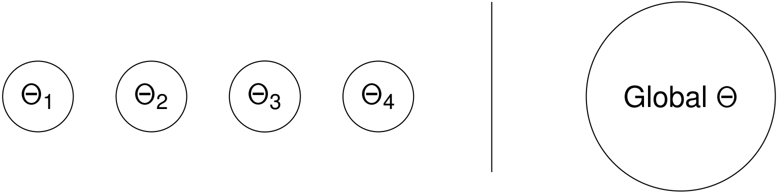


Merge-and-Shrink Framework (Sievers et al. 2014)

FDR task: $\langle \mathcal{V}, \mathcal{O}, \mathcal{I}, \mathcal{G} \rangle$



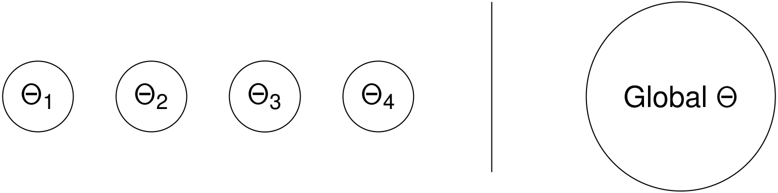
Merge-and-Shrink Framework (Sievers et al. 2014)



M&S: Framework for transformation of planning tasks

Operation	
Merge	
Shrink	
Exact Label Reduction	
Bisimulation shrinking	
Reachability pruning	

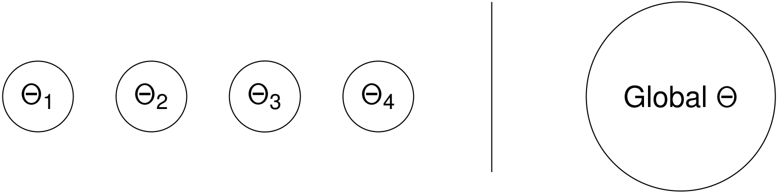
Merge-and-Shrink Framework (Sievers et al. 2014)



M&S: Framework for transformation of planning tasks

Operation	Transformation to global LTS
Merge	None
Shrink	Abstraction
Exact Label Reduction	Change labels
Bisimulation shrinking	Preserves h^*
Reachability pruning	Keeps reachable/solvable part

Merge-and-Shrink Framework (Sievers et al. 2014)



M&S: Framework for transformation of planning tasks

Operation	Transformation to global LTS
Merge	None
Shrink	Abstraction
Exact Label Reduction	Change labels
Bisimulation shrinking	Preserves h^*
Reachability pruning	Keeps reachable/solvable part
Subsumed transition pruning	Preserves h^*

Plan Preserving Transformations of Planning Tasks

$$\Pi \xrightarrow{\text{Plan preserving}} \Pi'$$

- Plan preserving:
  1. Does not add any new optimal plan to the task
  2. At least one optimal plan for the original task is preserved ($h^*(\mathcal{I})$)

- Unreachable/dead-end pruning is plan preserving

- In this paper: **subsumed transition pruning**
  - $\rightarrow$ remove transitions from M&S transition systems
  - $\rightarrow$ globally h-preserving ($h^*(s)$ for every s)

Subsumed Transition Pruning

Definition (Subsumed transition)

$s_i \xrightarrow{l} t_i$ is **subsumed** by $s_i \xrightarrow{l'} t_i'$ if:

1. $t_i \preceq t_i'$ and
2. $c(l') \leq c(l)$ and
3. l' dominates l in all Θ_j for $j \neq i$.

Thm: Remove subsumed transitions is globally h-preserving

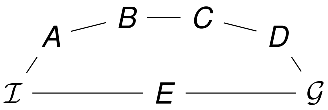
Subsumed Transition Pruning

Definition (Subsumed transition)

$s_i \xrightarrow{l} t_i$ is **subsumed** by $s_i \xrightarrow{l'} t'_i$ if:

1. $t_i \preceq t'_i$ and
2. $c(l') \leq c(l)$ and
3. l' dominates l in all Θ_j for $j \neq i$.

Thm: Remove subsumed transitions is globally *h*-preserving



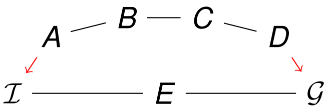
Subsumed Transition Pruning

Definition (Subsumed transition)

$s_i \xrightarrow{l} t_i$ is **subsumed** by $s_i \xrightarrow{l'} t'_i$ if:

1. $t_i \preceq t'_i$ and
2. $c(l') \leq c(l)$ and
3. l' dominates l in all Θ_j for $j \neq i$.

Thm: Remove subsumed transitions is globally h-preserving



$\mathcal{I} \rightarrow A$ is subsumed by $\mathcal{I} \rightarrow E$

$\mathcal{G} \rightarrow D$ is subsumed by $\mathcal{G} \rightarrow E$

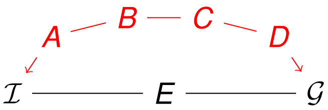
Subsumed Transition Pruning

Definition (Subsumed transition)

$s_i \xrightarrow{l} t_i$ is **subsumed** by $s_i \xrightarrow{l'} t_i'$ if:

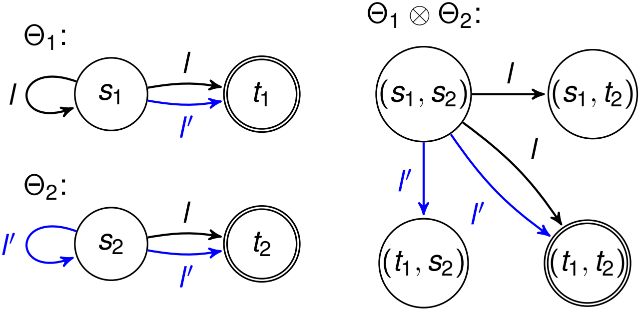
1. $t_i \preceq t_i'$ and
2. $c(l') \leq c(l)$ and
3. l' dominates l in all Θ_j for $j \neq i$.

Thm: Remove subsumed transitions is globally h-preserving

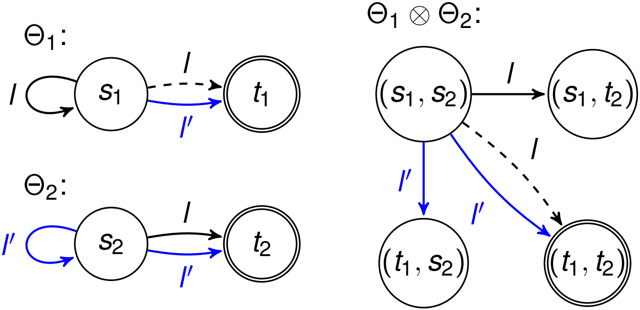


$\mathcal{I} \rightarrow A$ is subsumed by $\mathcal{I} \rightarrow E$
$\mathcal{G} \rightarrow D$ is subsumed by $\mathcal{G} \rightarrow E$
A, B, C, D become unreachable

Example: Subsumed Transition Pruning

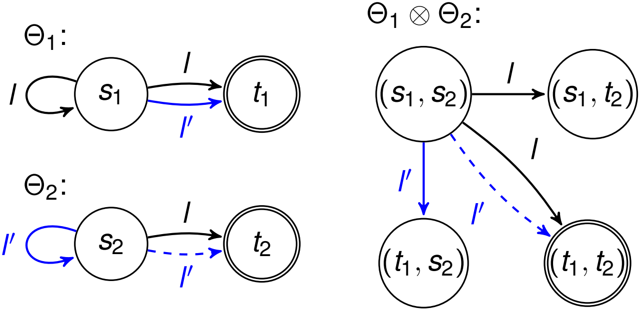


Example: Subsumed Transition Pruning



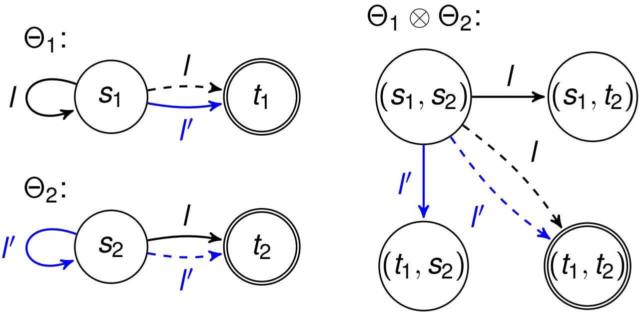
- $s_1 \xrightarrow{l} t_1$ is subsumed by $s_1 \xrightarrow{l'} t_1$

Example: Subsumed Transition Pruning



- $s_1 \xrightarrow{l} t_1$ is subsumed by $s_1 \xrightarrow{l'} t_1$
- $s_2 \xrightarrow{l'} t_2$ is subsumed by $s_2 \xrightarrow{l} t_2$

Example: Subsumed Transition Pruning



- $s_1 \xrightarrow{l} t_1$ is subsumed by $s_1 \xrightarrow{l'} t_1$
- $s_2 \xrightarrow{l'} t_2$ is subsumed by $s_2 \xrightarrow{l} t_2$

Don't remove a transition if the label dominance changes!

Taking Advantage of Plan-Preserving Transformations

1. Search task Π' instead of Π
   - implementation overhead (future work)

Taking Advantage of Plan-Preserving Transformations

1. Search task Π' instead of Π
   - implementation overhead (future work)

2. Remove dead operators:
   - after subsumed transition and unreachability pruning

Taking Advantage of Plan-Preserving Transformations

1. Search task Π' instead of Π
   - implementation overhead (future work)

2. Remove dead operators:
   - after subsumed transition and unreachability pruning

3. M&S heuristics: If Θ' is a plan-preserving transformation of Θ, abstractions of Θ' are not admissible for Θ

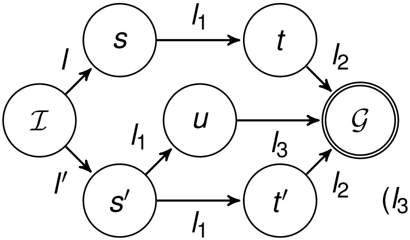
Taking Advantage of Plan-Preserving Transformations

1. Search task Π' instead of Π
   - implementation overhead (future work)
2. Remove dead operators:
   - after subsumed transition and unreachability pruning
3. M&S heuristics: If Θ' is a plan-preserving transformation of Θ, abstractions of Θ' are not admissible for Θ
   - It's not a bug, it's a feature!!! $\rightarrow$ less expanded states
   - **globally admissible** (preserve h^* in at least one optimal plan) $\Rightarrow$ A^*returns optimal solutions

Subsumed transition pruning + unreachability analysis must be applied before any shrinking (except bisimulation)

Similarity Shrinking

- Shrink s, t iff $s \preceq t$ and $t \preceq s$
- Globally h-preserving $\Rightarrow$ derives perfect heuristics
- Coarser than bisimulation (s and s' are similar but not bisimilar)



$(l_3 \preceq l_2)$

Redundant with subsumed transition pruning (mod label reduction)

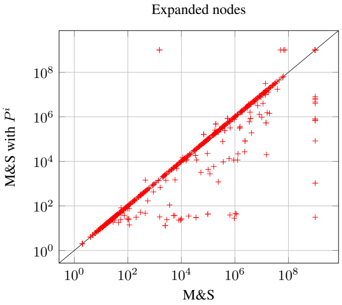
Experiments

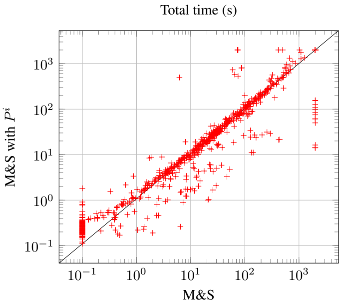
- Configuration P^i:
  - Incremental computation: recompute simulation after each merge
  - No label reduction, no shrinking

- Preprocess successful in 1463 of 1612 tasks
- Takes around 100s but up to 500-1000s in larger tasks

  $\Rightarrow$ Suitable for optimal but not for satisficing planning

Experiments: M&S Heuristic





Experiments: Removing Irrelevant Operators

	% pruned operators			Coverage LM-cut			
Domain	P^i	h^2	$h^2 + P^i$	-	P^i	h^2	$h^2 + P^i$
Floortile11	28	38	38	7	+1	+7	+7
Logistics00	67	0	67	20	+1	0	+1
NoMystery	49	23	49	14	+4	0	+4
ParcPrint11	77	70	79	13	+6	+4	+6
Rovers	71	0	71	7	+3	0	+2
Satellite	50	0	50	7	+2	0	+2
TPP	25	56	61	6	+1	0	+1
Trucks	90	38	90	10	+1	0	+1
Woodwk11	89	51	88	12	+8	+3	+8
Total (1612)	32	23	42	833	+29	+46	+65

+13 problems for symbolic bidirectional uniform-cost search (over 964)

Experiments: Comparison with State of the Art

HHJ (Haslum, Helmert, and Jonsson ICAPS 2013)

- Analyzes path subsumption in DTGs
- Current implementation only applicable to unary domains

	Operators		LM-Cut		
Domain	P^i	HHJ	–	P^i	HHJ
Blocksworld	0.01	**0.81**	28	28	**35**
Driverlog	**0.05**	**0.05**	13	13	**14**
Logistics00	**0.65**	0.52	20	**21**	**21**
Logistics98	**0.38**	0.09	6	6	6
Miconic	**0.58**	0.57	141	**142**	**142**
Total			208	210	**218**

Conclusions

Take home messages:

1. M&S is suitable for transformation of planning tasks
2. Simulation relations useful for:
   - Subsumed transition pruning $\rightarrow$ very good in practice!
   - Similarity shrinking:
     - perfect shrinking better than bisimulation but...
     - redundant with subsumed transition pruning + bisimulation
3. Irrelevance pruning greatly simplifies planning tasks

Future work:

- Extensions of label-dominance simulation
- Path subsumption
- More types of problem transformations

Thanks for your attention!

Questions?