Simulation-Based Admissible Dominance Pruning

Álvaro Torralba, Jörg Hoffmann

HSDIP Workshop at ICAPS
June 8th, 2015

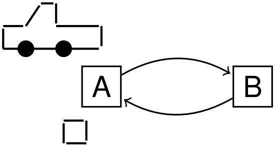
Motivation

- Cost-optimal planning: $(\mathcal{V}, \mathcal{O}, \mathcal{I}, \mathcal{G})$
- A^*+ admissible heuristic h(s): estimates distance to goal
- Pruning methods:
  1. Partial-order pruning
  2. Symmetries
  3. **Dominance pruning**

Dominance Pruning

Dominance Pruning

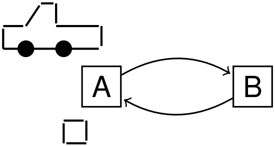
- Detect "better than" states



$\mathcal{V}$ = { at-T= {A, B}, at-P= {A, B, T} }
$\mathcal{I}$ = {at-T A, at-P A}
$\mathcal{G}$ = {at-P B}
$\mathcal{O}$ = {move-T (A, B),
move-T (B, A), load-P(A), . . . }

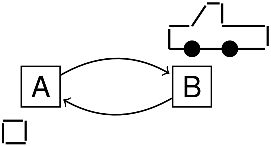
Dominance Pruning

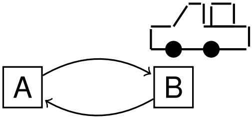
- Detect "better than" states



$$\mathcal{V} = \{ \text{at-T} = \{A, B\}, \text{at-P} = \{A, B, T\} \}$$
$$\mathcal{I} = \{\text{at-T } A, \text{at-P } A\}$$
$$\mathcal{G} = \{\text{at-P } B\}$$
$$\mathcal{O} = \{\text{move-T } (A, B), \text{move-T } (B, A), \text{load-P}(A), \dots \}$$

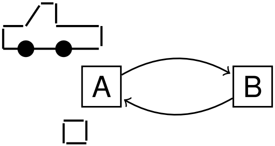
- What do you prefer?





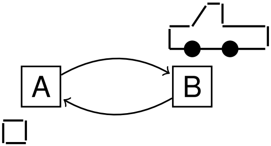
Dominance Pruning

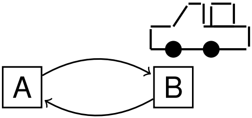
- Detect “better than” states



$\mathcal{V} =$ { at-T= {A, B}, at-P= {A, B, T} }
$\mathcal{I} =$ {at-T A, at-P A}
$\mathcal{G} =$ {at-P B}
$\mathcal{O} =$ {move-T (A, B), move-T (B, A), load-P(A), . . . }

- What do you prefer?





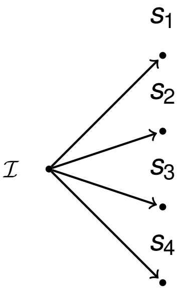
- Formally: relation of pair of states $s \preceq t$

Admissible Pruning

- t simulates s ($s \preceq t$) $\implies$ t is at least as good as s:

$$h^*(s) \geq h^*(t)$$

- If $g(t) \leq g(s)$ and $s \preceq t$ then s can be discarded

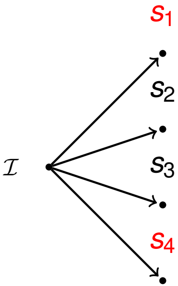


Admissible Pruning

- t simulates s ($s \preceq t$) $\implies$ t is at least as good as s:

$$h^*(s) \geq h^*(t)$$

- If $g(t) \leq g(s)$ and $s \preceq t$ then s can be discarded

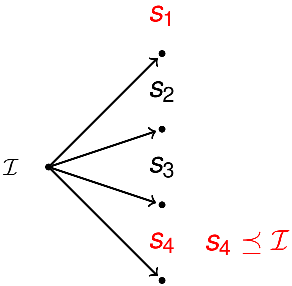


Admissible Pruning

- t simulates s ($s \preceq t$) $\implies$ t is at least as good as s:

$$h^*(s) \geq h^*(t)$$

- If $g(t) \leq g(s)$ and $s \preceq t$ then s can be discarded

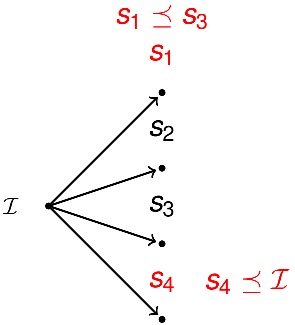


Admissible Pruning

- t simulates s ($s \preceq t$) $\implies$ t is at least as good as s:

$$h^*(s) \geq h^*(t)$$

- If $g(t) \leq g(s)$ and $s \preceq t$ then s can be discarded

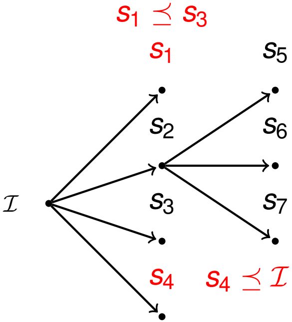


Admissible Pruning

- t simulates s ($s \preceq t$) $\implies$ t is at least as good as s:

$$h^*(s) \geq h^*(t)$$

- If $g(t) \leq g(s)$ and $s \preceq t$ then s can be discarded



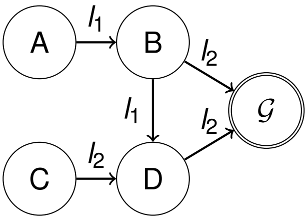
Challenges:

1. How to find good dominance relations?
2. How to efficiently check dominance?

Simulation Relation

Definition (Simulation)

A binary relation $\preceq \subseteq S \times S$ is a **simulation** for Θ if, whenever $s \preceq t$, **for every transition $s \xrightarrow{l} s'$ there exists $t \xrightarrow{l} t'$ s.t. $s' \preceq t'$**. We call $\preceq$ **goal-respecting** for Θ if, whenever $s \preceq t$, $s \in S_G$ **implies that** $t \in S_G$.



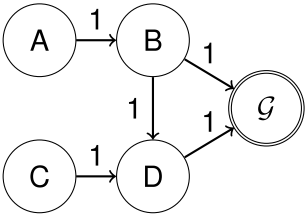
$D \preceq B$

Thm: A unique coarsest goal-respecting simulation always exists and can be computed in time polynomial in the size of Θ

Simulation Relation

Definition (Simulation)

A binary relation $\preceq \subseteq S \times S$ is a **simulation** for Θ if, whenever $s \preceq t$, **for every transition $s \xrightarrow{l} s'$ there exists $t \xrightarrow{l} t'$ s.t. $s' \preceq t'$**. We call $\preceq$ **goal-respecting** for Θ if, whenever $s \preceq t$, **$s \in S_G$ implies that $t \in S_G$**.



$D \preceq B$, $C \preceq A$

Cost-simulation: replace labels by their cost

$\rightarrow$ A cost-simulation on the state space of the planning task is a dominance relation

Thm: A unique coarsest goal-respecting simulation always exists and can be computed in time polynomial in the size of Θ

Compositional Approach

1. Consider a partition of the problem: $\Theta_1, \ldots, \Theta_k$
2. Compute a simulation for each part: $\preceq_1, \ldots, \preceq_k$
3. $\preceq$: $s \preceq t$ iff $\forall i \in [1, k]\ s_i \preceq_i t_i$

$\preceq$ is a cost-simulation

Compositional Approach

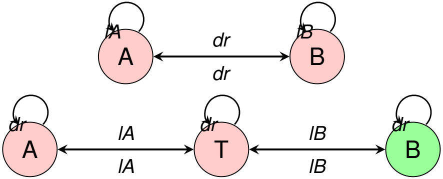
1. Consider a partition of the problem: $\Theta_1, \ldots, \Theta_k$
2. Compute a simulation for each part: $\preceq_1, \ldots, \preceq_k$
3. $\preceq$: $s \preceq t$ iff $\forall i \in [1, k]$ $s_i \preceq_i t_i$

$\preceq$ is a cost-simulation

In our example:

Θ^1:
(truck)

Θ^2:
(package)



Compositional Approach

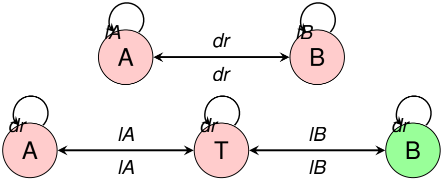
1. Consider a partition of the problem: $\Theta_1, \ldots, \Theta_k$
2. Compute a simulation for each part: $\preceq_1, \ldots, \preceq_k$
3. $\preceq$: $s \preceq t$ iff $\forall i \in [1, k]\ s_i \preceq_i t_i$

$\preceq$ is a cost-simulation

In our example:

Θ^1:
(truck)

Θ^2:
(package)

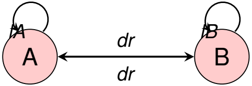


$A \preceq_2 T \preceq_2 B$

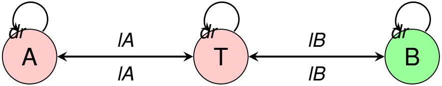
Not so fast

Definition of $s \preceq t$: For every $s \xrightarrow{l} s'$ there exists $t \xrightarrow{l} t'$ s.t. $s' \preceq t'$

Θ^1:
(truck)



Θ^2:
(package)



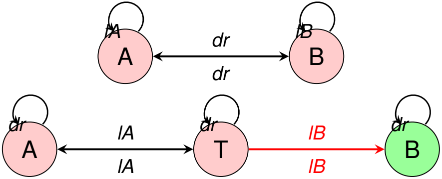
$A \preceq_2 T \preceq_2 B$

Not so fast

Definition of $s \preceq t$: For every $s \xrightarrow{l} s'$ there exists $t \xrightarrow{l} t'$ s.t. $s' \preceq t'$

Θ^1:
(truck)

Θ^2:
(package)



$A \preceq_2 T \preceq_2 B$

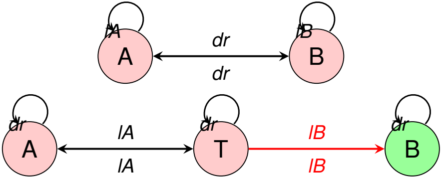
$T \not\preceq_2 B$: $T \xrightarrow{lB} B$ and there is no $B \xrightarrow{lB}$

Not so fast

Definition of $s \preceq t$: For every $s \xrightarrow{l} s'$ there exists $t \xrightarrow{l} t'$ s.t. $s' \preceq t'$

Θ^1:
(truck)

Θ^2:
(package)



$A \preceq_2 T \preceq_2 B$

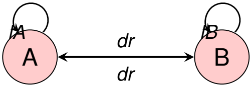
$T \not\preceq_2 B$: $T \xrightarrow{lB} B$ and there is no $B \xrightarrow{lB}$

$T \xrightarrow{lB} B$ and $T \xrightarrow{lA} A$ are simulated by $B \;\rightarrow$

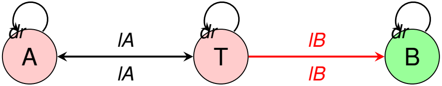
Not so fast

Definition of $s \preceq t$: For every $s \xrightarrow{l} s'$ there exists $t \xrightarrow{l} t'$ s.t. $s' \preceq t'$

Θ^1:
(truck)



Θ^2:
(package)



$$A \preceq_2 T \preceq_2 B$$

$$T \not\preceq_2 B\text{: } T \xrightarrow{lB} B \text{ and there is no } B \xrightarrow{lB}$$

$$T \xrightarrow{lB} B \text{ and } T \xrightarrow{lA} A \text{ are simulated by } B \xrightarrow{noop} B$$

lA, lB do not have useful effects in the rest of the problem (Θ^1)!

Label-Dominance Simulation

Definition (Label Dominance)

l' dominates l in Θ given $\preceq$ if for every $s \xrightarrow{l} s' \in \Theta$ there exists $s \xrightarrow{l'} t'$ s.t. $s' \preceq t'$

Definition (Label-Dominance Simulation)

A set $\mathcal{R} = \{\preceq_1, \ldots, \preceq_k\}$ of binary relations $\preceq_i \subseteq S_i \times S_i$ is a label-dominance simulation for $\{\Theta^1, \ldots, \Theta^k\}$ if, whenever $s \preceq_i t$:

- $s \in S_i^G$ implies that $t \in S_i^G$
- For every $s \xrightarrow{l} s'$ in Θ^i, there exists $t \xrightarrow{l'} t'$ in Θ^i s.t.:
  1. $s' \preceq_i t'$,
  2. $c(l') \leq c(l)$, and
  3. for all $j \neq i$, l' dominates l in Θ^j given $\preceq_j$

Label-Dominance Simulation: Theoretical Results

Theorem

A coarsest label-dominance simulation always exists and can be computed in polynomial time

For all i, set $\preceq_i := \{(s,t) \mid s,t \in S_i, s \notin S^i_G \text{ or } t \in S^i_G\}$
while ex. (i,s,t) s.t. not **Ok**(i,s,t) **do**
  Select one such triple (i,s,t)
  Set $\preceq_i := \preceq_i \setminus \{(s,t)\}$
return $\mathcal{R} := \{\preceq_1, \dots, \preceq_k\}$

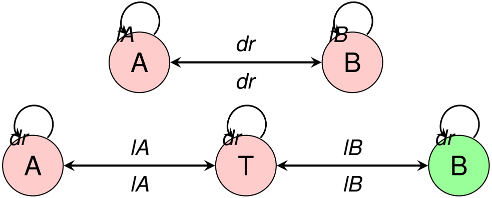
Theorem

Combination of $\{\preceq_1, \dots, \preceq_k\}$ *is a cost-simulation for the planning task* $\Theta_1 \otimes \cdots \otimes \Theta_k$

Computation of Label-Dominance Simulation

Θ^1:
(truck)

Θ^2:
(package)



Truck

$A \preceq_1 \{ B \}$
$B \preceq_1 \{ A \}$

noop simulates $\{lA, lB, dr\}$
dr simulates $\{lA, lB\}$

Package

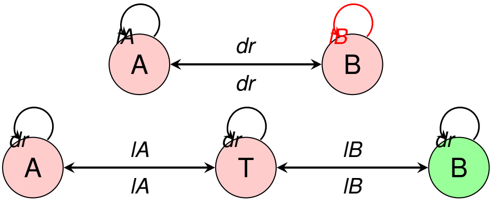
$A \preceq_2 \{ T, B \}$
$T \preceq_2 \{ A, B \}$
$B \preceq_2 \{ \}$

$(noop \equiv dr)$ simulate $\{lA\}$

Computation of Label-Dominance Simulation

Θ^1:
(truck)

Θ^2:
(package)



Truck

$A \preceq_1 \{\ B\ \}$
$B \preceq_1 \{\ \cancel{A}\ \}$

Package

$A \preceq_2 \{\ T, B\ \}$
$T \preceq_2 \{\ A, B\ \}$
$B \preceq_2 \{\ \}$

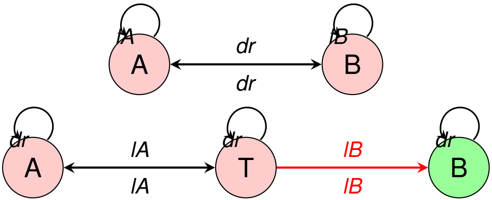
noop simulates $\{lA, lB, \cancel{dr}\}$
dr simulates $\{lA, \cancel{lB}\}$

$(noop \equiv dr)$ simulate $\{lA\}$

Computation of Label-Dominance Simulation

Θ^1:
(truck)

Θ^2:
(package)



Truck

$A \preceq_1 \{ B \}$
$B \preceq_1 \{ \cancel{A} \}$

noop simulates $\{lA, lB, \cancel{dr}\}$
dr simulates $\{lA, \cancel{lB}\}$

Package

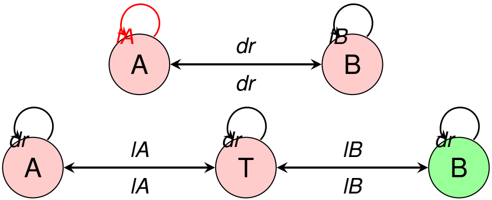
$A \preceq_2 \{ T, B \}$
$T \preceq_2 \{ \cancel{A}, B \}$
$B \preceq_2 \{ \}$

$(noop \equiv dr)$ simulate $\{\cancel{lA}\}$

Computation of Label-Dominance Simulation

Θ^1:
(truck)

Θ^2:
(package)



Truck

$A \preceq_1 \{ \cancel{B} \}$
$B \preceq_1 \{ \cancel{A} \}$

noop simulates $\{lA, lB, \cancel{dr}\}$
dr simulates $\{\cancel{lA}, \cancel{lB}\}$

Package

$A \preceq_2 \{ T, B \}$
$T \preceq_2 \{ \cancel{A}, B \}$
$B \preceq_2 \{ \}$

$(noop \equiv dr)$ simulate $\{\cancel{lA}\}$

Pruning: Implementation Details

- Insert in closed every state dominated by any expanded state
  $\rightarrow$ BDD B_g represents any state **expanded/dominated** with g

- When s is generated or expanded:
  1. Prune s if it is in $B_{g'}$ for some $g' \leq g(s)$

- When s is expanded:
  1. Insert all states dominated by s in $B_{g(s)}$

Pruning: Implementation Details

- Insert in closed every state dominated by any expanded state
  $\rightarrow$ BDD B_g represents any state **expanded/dominated** with g

- When s is generated or expanded:
  1. Prune s if it is in $B_{g'}$ for some $g' \leq g(s)$

- When s is expanded:
  1. Insert all states dominated by s in $B_{g(s)}$

- Safety Belt: Stop if no state is pruned after 1000 expansions
  - Don't waste time if no useful dominance relation has been found

Experimental Results

- M&S: Merge-DFP + bisimulation up to 100 000 transitions
- Pruning types:
  - A: Baseline without pruning
  - L: Label-dominance simulation
  - S: Simulation
  - B: Bisimulation
  - P: Partial-order reduction
- Heuristic: Blind or LM-cut

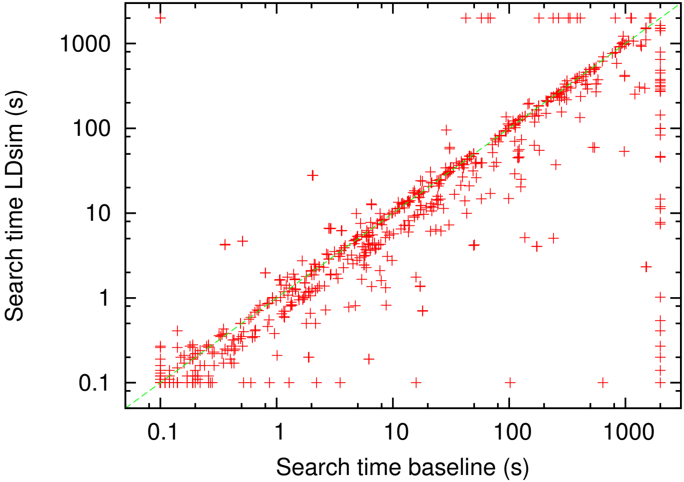
Experimental Results: Blind Search

Domain	#	Coverage A	Coverage L	Coverage S	Coverage B	Coverage P	Evaluations L	Evaluations S	Evaluations B	Evaluations P
Airport	50	**22**	-7	-7	**0**	-1	1.2	1.2	1	4.4
Driverlog	20	7	**+2**	0	0	0	15.8	2	2	1
Floortile11	20	2	**+4**	**+4**	0	0	177	177	1.8	1.3
Gripper	20	8	**+6**	**+6**	**+6**	0	53968	53968	28353	1
Logistics00	28	10	+6	0	0	0	32.7	3.1	1.2	1
Miconic	150	55	**+6**	-1	0	-5	58.3	8.7	3.4	1
NoMystery	20	8	**+10**	+1	+1	0	2497	128	29.1	1.1
OpenStack11	20	17	**+2**	**+2**	+1	0	2.1	2	1.8	2
ParcPrint11	20	6	+5	+3	+1	**+14**	869	10	1.5	21826
Rovers	40	6	**+2**	+1	0	+1	33.4	9.6	1.7	2
Satellite	36	6	0	0	0	0	72.9	35.3	9.9	10.7
TPP	30	6	0	0	0	0	6.5	3.4	1	1
Trucks	30	6	**+2**	0	0	0	24.8	21.9	2.8	1
VisitAll11	20	9	0	0	0	0	30	25.5	1	1
Woodwork11	20	3	**+9**	+5	+4	+6	1059	116	92.2	514
Zenotravel	20	8	**+1**	0	0	0	41.6	1.5	1.1	1
$\sum$	1271	605	**+57**	+16	+16	+8				

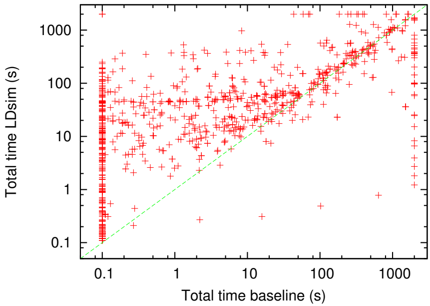
Experimental Results: LM-cut

Domain	#	Coverage A	Coverage L	Coverage S	Coverage B	Coverage P	Evaluations L	Evaluations S	Evaluations B	Evaluations P
Airport	50	28	-1	-1	-1	**+1**	1	1	1	4.7
Driverlog	20	13	0	0	0	0	1.9	1.2	1.2	1
Floortile11	20	7	**+1**	**+1**	0	0	6.4	6.4	1	1
Gripper	20	7	**+7**	**+7**	**+7**	0	14662	14662	10049	1
Logistics00	28	20	0	0	0	0	1.9	1.1	1.1	2.9
Miconic	150	141	0	0	0	0	2.1	1.5	1.1	1
NoMystery	20	14	**+6**	+3	0	0	6.5	3.1	1	1
OpenStack11	20	16	0	0	0	0	2.5	2.4	2.1	1.8
ParcPrint11	20	13	0	0	0	**+7**	5	1.2	1.1	1246
Rovers	40	7	**+2**	+1	+1	**+2**	6.1	3.8	1.2	4.4
Satellite	36	7	+3	+3	+3	**+4**	4.8	1.8	1.7	21.5
TPP	30	6	**+1**	**+1**	**+1**	0	1.2	1.1	1	1
Trucks	30	10	0	0	0	0	2.7	2.3	1	1
VisitAll11	20	10	**+1**	**+1**	0	0	7	6.8	1	1
Woodwork11	20	12	+5	+4	+4	**+7**	91.6	23.8	17	772
Zenotravel	20	13	0	0	0	0	3.6	1.6	1	1
$\sum$	1271	833	+20	+14	+17	**+38**				

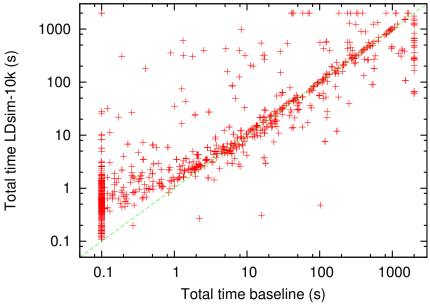
Experimental Results: Search Time



Experimental Results: Total Time



Max partition size:
100 000 transitions



Max partition size:
10 000 transitions

Conclusions

- Novel method of dominance pruning useful for many domains
- Overhead in computing the relation and comparing states during the search
- Future work:
  - Find coarser relations
  - Reduce overhead
  - Irrelevance pruning

    $\rightarrow$ SoCS talk on Thursday (joint session with ICAPS)!

Thank you for your attention!

Questions?