Relaxed Decision Diagrams for Delete-Free Planning

Augusto B. Corrêa Florian Pommerening Guillem Francès

University of Basel, Switzerland

Agenda

Introduction
- Motivation
- Delete-Free Planning
- Variable Assignment Problems
- Decision Diagrams

Π^+ as a Decision Diagram

Relaxed Decision Diagrams

Experiments

Conclusion

Motivation

- New state space of a delete-free task Π^+
- Compute h^+ in this new state space
- Use it to solve Π^+

Delete-Free Planning

In a delete-free planning task Π^+ we have:

- a set of facts;
- a set of actions in the format $a = (\text{pre}_a, \text{add}_a)$;
- a cost function;
- a initial state s_0; and
- a set of goal facts;

Delete-Free Planning

In a delete-free planning task Π^+ we have:

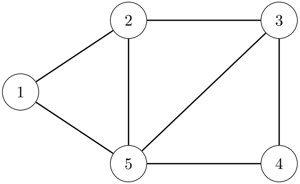
- a set of facts;
- a set of actions in the format $a = (\text{pre}_a, \text{add}_a)$;
- a cost function;
- a initial state s_0; and
- a set of goal facts;

Variable Assignment Problems

Variable Assignment Problems

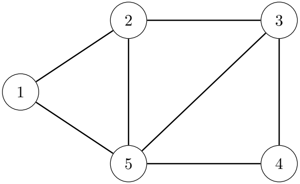
- Given a *set of variables* V
- Assign each variable to a value
- Some assignments are solutions
- Every assignment has a cost

Example: The Independent Set Problem[1]



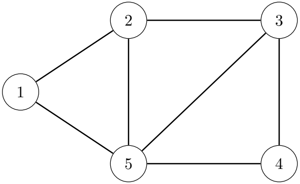
Example: The Independent Set Problem[1]

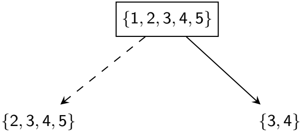
$\boxed{\{1,2,3,4,5\}}$



[1]Example from ICAPS 2016 tutorial "*Decision Diagrams for Discrete Optimization*" from J. Hooker

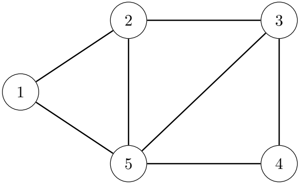
Example: The Independent Set Problem[1]

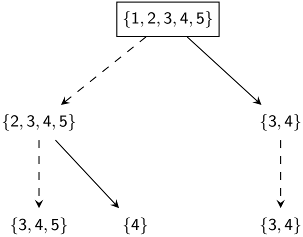




[1] Example from ICAPS 2016 tutorial "*Decision Diagrams for Discrete Optimization*" from J. Hooker

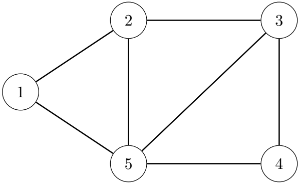
Example: The Independent Set Problem[1]

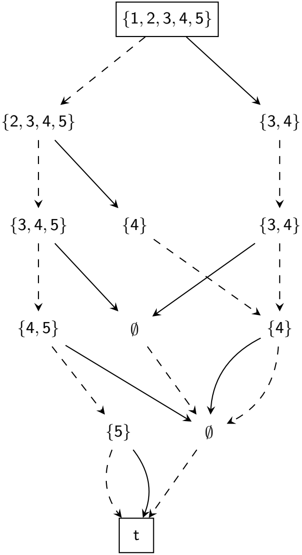




[1]Example from ICAPS 2016 tutorial "*Decision Diagrams for Discrete Optimization*" from J. Hooker

Example: The Independent Set Problem[1]





- every path in the decision diagram represents a solution
- path with maximum cost = maximum independent set

[1] Example from ICAPS 2016 tutorial "*Decision Diagrams for Discrete Optimization*" from J. Hooker

Decision Diagrams

Decision Diagrams

- a graph G divided in n layers $L_1, L_2, \ldots, L_n$
- a root node r and a terminal node t
- a set of variables V with finite domain D_v for all $v \in V$
- a cost function $w : V \times D_v \to \mathbb{R}$

Decision Diagrams

- a graph G divided in n layers $L_1, L_2, \ldots, L_n$
- a root node r and a terminal node t
- a set of variables V with finite domain D_v for all $v \in V$
- a cost function $w : V \times D_v \to \mathbb{R}$

Remark

Two nodes in a same layer can select different variables to assign.

Agenda

How to represent Π^+ as a decision diagram?

How to represent Π^+ as a decision diagram?

- The set of actions A is the set of variables
- Assign each action to 0 or 1
- Each node represents a state*
- A path from r to t is a plan
- Cost of a plan = cost of the actions assigned to 1

Properties of a Node

Properties of a Node

- Each node $g \in G$ has two properties:
    - $F(g)$, a set of facts (a state)
    - $N(g)$, a set of forbidden actions

Constructing a Decision Diagram for Π^+

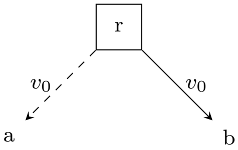
Constructing a Decision Diagram for Π^+

$$s_0 := \{\}$$
$$A := \{v_0, \ldots, v_n\}$$

Constructing a Decision Diagram for Π^+

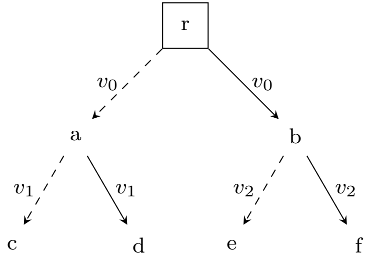
$$
\begin{aligned}
s_0 &:= \{\} \\
A &:= \{v_0, \ldots, v_n\} \\
r &:= (F(r) = s_0, N(r) = \emptyset)
\end{aligned}
$$

Constructing a Decision Diagram for Π^+



$$
\begin{aligned}
s_0 &:= \{\} \\
A &:= \{v_0, \ldots, v_n\} \\
r &:= (F(r) = s_0, N(r) = \emptyset) \\
a &:= (F(a) = s_0, N(a) = \{v_0\}) \\
b &:= (F(b) = s_0 \cup \mathrm{add}_{v_0}, N(b) = \{v_0\})
\end{aligned}
$$

Constructing a Decision Diagram for Π^+



$$
\begin{aligned}
s_0 &:= \{\} \\
A &:= \{v_0, \dots, v_n\} \\
r &:= (F(r) = s_0, N(r) = \emptyset) \\
a &:= (F(a) = s_0, N(a) = \{v_0\}) \\
b &:= (F(b) = s_0 \cup \mathrm{add}_{v_0}, N(b) = \{v_0\}) \\
c &:= (F(c) = s_0, N(c) = \{v_0, v_1\}) \\
d &:= (F(d) = s_0 \cup \mathrm{add}_{v_1}, N(d) = \{v_0, v_1\}) \\
e &:= (F(e) = s_0 \cup \mathrm{add}_{v_0}, N(e) = \{v_0, v_2\}) \\
f &:= (F(f) = s_0 \cup \mathrm{add}_{v_0} \cup \mathrm{add}_{v_2}, N(f) = \{v_0, v_2\})
\end{aligned}
$$

Some Caveats

Some Caveats

- Merge nodes representing goal states into t
- A path from r to t represents a partial assignment

Selecting an Applicable Action

An action $a \in A$ can be selected by a node g iff:

Selecting an Applicable Action

An action $a \in A$ can be selected by a node g iff:

- a is not forbidden: $a \notin N(g)$
- a is applicable: $\mathrm{pre}_a \subseteq F(g)$
- a is not useless: $\mathrm{add}_a \not\subseteq F(g)$

Agenda

Relaxed Decision Diagrams

- Decision diagrams can be exponentially large
- We can create a relaxed version with bounded size

Limited Width

- Limit the width (number of nodes) of the layers to ω
- If $|L_i| > \omega$, we combine nodes on L_i
- We need to find a good way to combine the nodes

Limited Width

- Limit the width (number of nodes) of the layers to ω
- If $|L_i| > \omega$, we combine nodes on L_i
- We need to find a good way to combine the nodes

Two conditions (for minimization problems)

- All solutions to the original problem must be preserved
- No solution cost can be overestimated

Limited Width

- Limit the width (number of nodes) of the layers to ω
- If $|L_i| > \omega$, we combine nodes on L_i
- We need to find a good way to combine the nodes

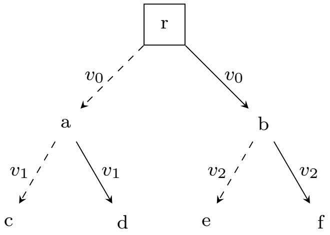
Two conditions (for minimization problems)

- All solutions to the original problem must be preserved
- No solution cost can be overestimated

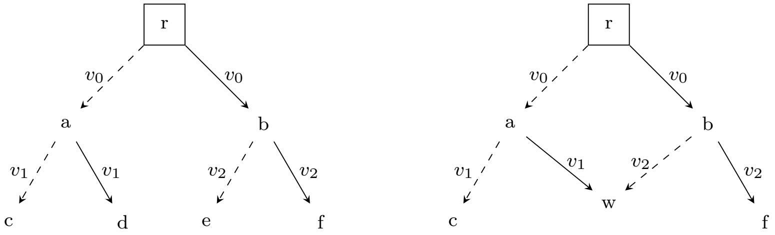
If both conditions hold, then the combination is admissible

Construction of a Relaxed Decision Diagram

Construction of a Relaxed Decision Diagram



Construction of a Relaxed Decision Diagram



Relaxed Decision Diagram for Π^+

- How to combine nodes in a same layer?

Relaxed Decision Diagram for Π^+

- How to combine nodes in a same layer?
- Combine a pair of nodes u, v into a single node w

Relaxed Decision Diagram for Π^+

- How to combine nodes in a same layer?
- Combine a pair of nodes u, v into a single node w
- Use the following combination function

$$F(w) = F(u) \cup F(v)$$
$$N(w) = N(u) \cap N(v)$$

Relaxed Decision Diagram for Π^+

- How to combine nodes in a same layer?
- Combine a pair of nodes u, v into a single node w
- Use the following combination function

$$F(w) = F(u) \cup F(v)$$
$$N(w) = N(u) \cap N(v)$$

Theorem

This combination function is admissible.

Relaxed Decision Diagram for Π^+

- How to combine nodes in a same layer?
- Combine a pair of nodes u, v into a single node w
- Use the following combination function

$$F(w) = F(u) \cup F(v)$$
$$N(w) = N(u) \cap N(v)$$

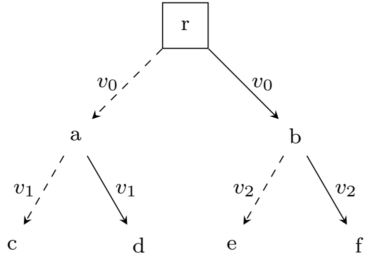
Theorem

This combination function is admissible.

Observation

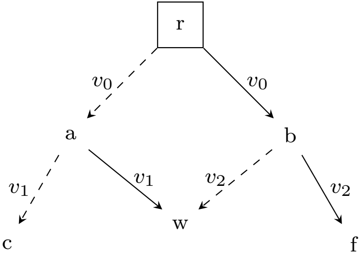
The shortest path from r to t is a lower bound for h^+.

Constructing a Decision Diagram for Π^+



$$s_0 := \{\}$$
$$A := \{v_0, \ldots, v_n\}$$
$$r := (F(r) = s_0, N(r) = \emptyset)$$
$$a := (F(a) = s_0, N(a) = \{v_0\})$$
$$b := (F(b) = s_0 \cup \mathrm{add}_{v_0}, N(b) = \{v_0\})$$
$$c := (F(c) = s_0, N(c) = \{v_0, v_1\})$$
$$d := (F(d) = s_0 \cup \mathrm{add}_{v_1}, N(d) = \{v_0, v_1\})$$
$$e := (F(e) = s_0 \cup \mathrm{add}_{v_0}, N(e) = \{v_0, v_2\})$$
$$f := (F(f) = s_0 \cup \mathrm{add}_{v_0} \cup \mathrm{add}_{v_2}, N(f) = \{v_0, v_2\})$$

Constructing a Decision Diagram for Π^+



$$s_0 := \{\}$$
$$A := \{v_0, \dots, v_n\}$$
$$r := (F(r) = s_0, N(r) = \emptyset)$$
$$a := (F(a) = s_0, N(a) = \{v_0\})$$
$$b := (F(b) = s_0 \cup \mathrm{add}_{v_0}, N(b) = \{v_0\})$$
$$c := (F(c) = s_0, N(c) = \{v_0, v_1\})$$
$$w := (F(w) = s_0 \cup \mathrm{add}_{v_0} \cup \mathrm{add}_{v_1}, N(w) = \{v_0\})$$
$$f := (F(f) = s_0 \cup \mathrm{add}_{v_0} \cup \mathrm{add}_{v_2}, N(f) = \{v_0, v_2\})$$

Selecting Nodes to Combine

- Lower bound obtained depends on the nodes combined

Selecting Nodes to Combine

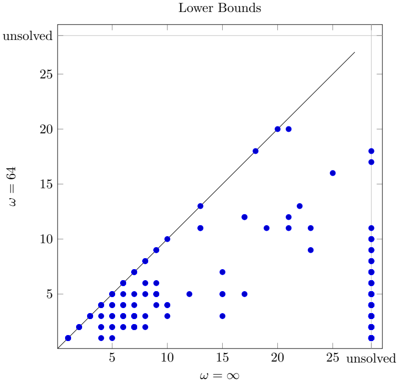
- Lower bound obtained depends on the nodes combined
- Two functions for nodes selection:
  - Hamming distance between nodes
  - Random Selection

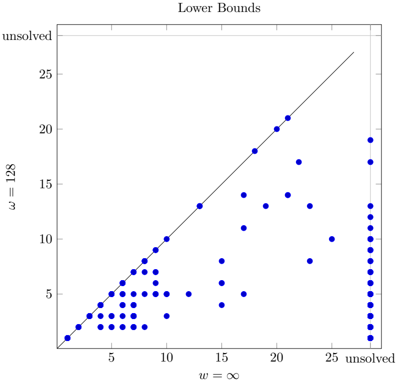
Agenda

Experimental Setup

- Using Fast Downward with 3.6GB and 30 minutes per task
- Considering all tasks as unit-cost
- LM-cut to compute h^+
- Six different values for ω:
  - 64, 128, 256, 512, 1024 and ∞
- Compared two nodes selection functions:
  - Hamming Distance
  - Random Node Selection

Hamming Distance with different ω

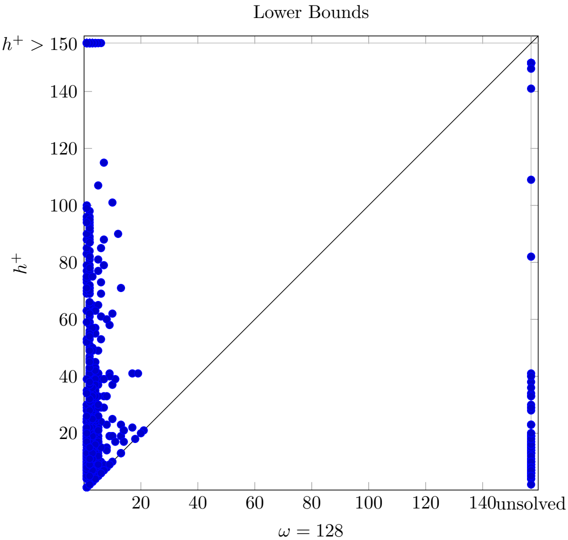




unsolved = diagram construction did not terminate

= no lower bound found

Hamming Distance w/ $\omega = 128$ vs h^+



Coverage

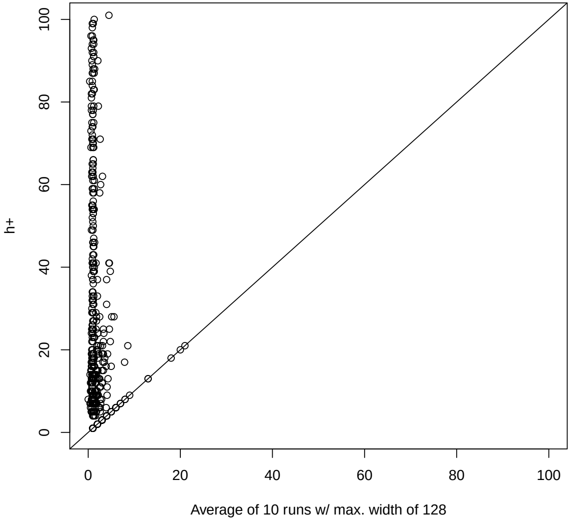
How many instances we could find an optimal plan?

	h^+	$\omega = \infty$	$\omega = 64$	$\omega = 128$	$\omega = 256$	$\omega = 512$	$\omega = 1024$
# **valid plans** (1667)	**921**	**171**	**135**	**155**	**146**	**130**	**122**
# **of domains** (57)	55	28	21	28	28	23	19

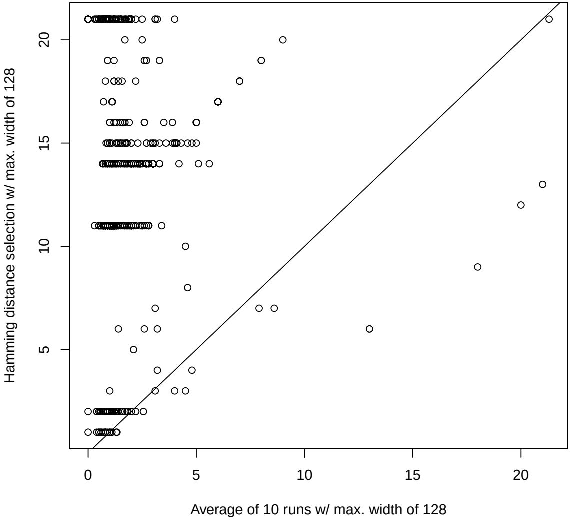
Selecting Random Nodes

- change the nodes selection function to a random selection
- results report the avg. over 10 runs
  - only for $\omega = \{128, 256\}$
- with $\omega = 128$ the number of optimal plans drops to 131.4
- with $\omega = 256$ the number of optimal plans drops to 139.9

Random Selection w/ $\omega = 128$ vs h^+



Lower bounds w/ different nodes selection



Agenda

Conclusion

Contributions:

- Proposed a new state space representation for delete-free tasks
- Analyzed theoretically some properties of this new state space
- Experiments showed this technique is not yet informative

Conclusion

Contributions:

- Proposed a new state space representation for delete-free tasks
- Analyzed theoretically some properties of this new state space
- Experiments showed this technique is not yet informative

Future Work:

- Better strategies for action selection
  - e.g., greedily select landmark actions whenever possible
- Better strategies for node selection
  - e.g., consider g-values to avoid "shortcuts"
- Refinement strategies
  - e.g., start with a diagram with width equals to 1 and split combined nodes

Conclusion

Contributions:

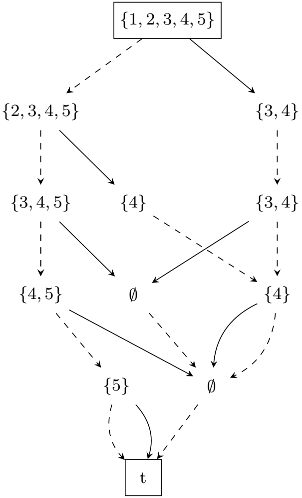
- Proposed a new state space representation for delete-free tasks
- Analyzed theoretically some properties of this new state space
- Experiments showed this technique is not yet informative

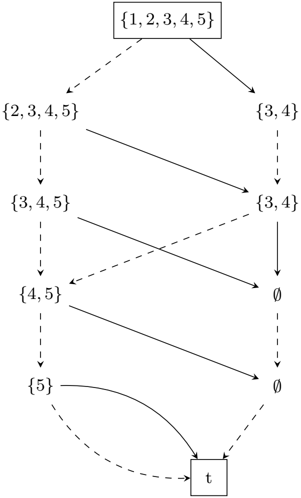
Future Work:

- Better strategies for action selection
  - e.g., greedily select landmark actions whenever possible
- Better strategies for node selection
  - e.g., consider g-values to avoid "shortcuts"
- Refinement strategies
  - e.g., start with a diagram with width equals to 1 and split combined nodes

Thank you!

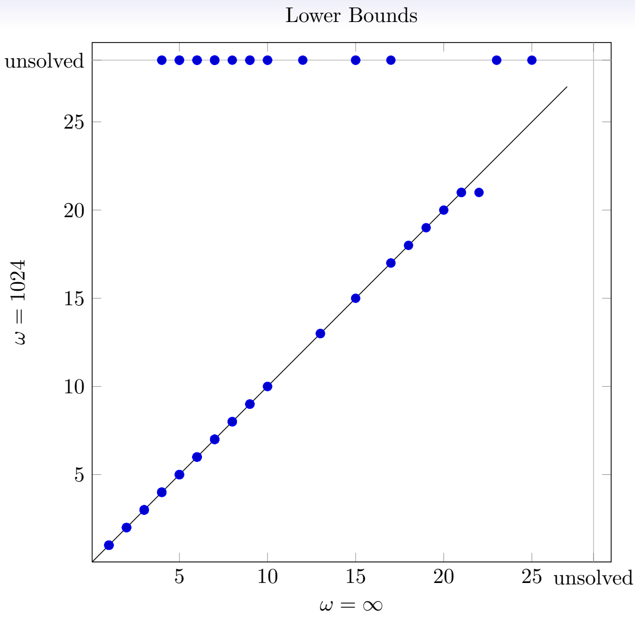
Example: The Independent Set Problem (again)





Lower Bounds on Different Domains

Domain	h^+	$\omega = \infty$	$\omega = 64$	$\omega = 128$	$\omega = 256$	$\omega = 512$	$\omega = 1024$
airport (7)	114	114	98	104	109	113	114
blocks (1)	6	6	3	4	4	5	6
driverlog (1)	6	6	6	6	6	6	6
ged-opt14-strips (3)	3	3	3	3	3	3	3
logistics00 (7)	106	106	74	80	97	106	106
miconic (12)	66	66	56	57	59	62	66
movie (8)	56	56	20	26	40	56	56
parcprinter-08-strips (3)	38	38	27	32	35	38	37
pegsol-08-strips (2)	8	8	7	7	8	8	8
psr-small (49)	155	155	155	155	155	155	155
rovers (4)	33	33	19	24	26	32	33
sokoban-opt08-strips (3)	20	20	17	19	20	20	20
sokoban-opt11-strips (2)	11	11	8	10	11	11	11
storage (6)	27	27	21	23	24	26	27
tidybot-opt11-strips (1)	4	4	4	4	4	4	4
tpp (4)	34	34	32	34	34	34	34
transport-opt08-strips (1)	5	5	3	3	3	5	5
visitall-opt11-strips (4)	17	17	13	17	17	17	17
zenotravel (2)	5	5	3	3	4	5	5
Sum (120)	**714**	**714**	**569**	**611**	**659**	**706**	**713**


Lower Bounds
unsolved
25
20
15
10
5
$\omega = 1024$
5
10
15
20
25
unsolved
$\omega = \infty$