Correlation Complexity and Different Notions of Width

Simon Dold

University of Basel

June 2, 2021

University of Basel

Classical Planning

SAS$^+$ Planning Task $\Pi = \langle V, I, O, \gamma \rangle$

- State variables V with finite domain
- Initial state I
- Operators O with precondition and effect
- Goal γ

Classical Planning

Task induces a graph called state space

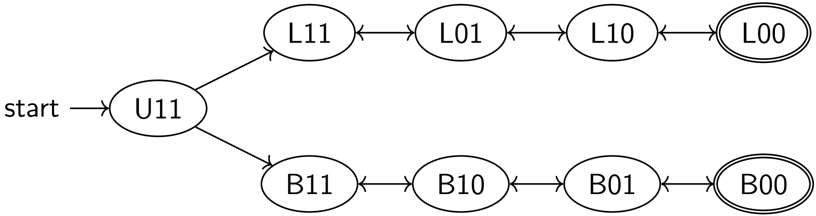
- Nodes correspond to states
- Arcs correspond to operators

Example

Little/big endian binary countdown

$$
\begin{aligned}
V =& \{v, b_0, b_1\} \\
dom(v) =& \{\text{undecided}, \text{little endian}, \text{big endian}\} \\
dom(b_0) =& dom(b_1) = \{0, 1\} \\
I =& \{v \mapsto \text{undecided}, b_0 \mapsto 1, b_1 \mapsto 1\} \\
\gamma =& \{b_0 \mapsto 0, b_1 \mapsto 0\} \\
O =& \{\langle\{v \mapsto \text{undecided}\}, \{v \mapsto \text{little endian}\}\rangle, \\
& \langle\{v \mapsto \text{undecided}\}, \{v \mapsto \text{big endian}\}\rangle, \\
& \langle\{v \mapsto \text{big endian}, b_0 \mapsto 1, b_1 \mapsto 1\}, \{b_1 \mapsto 0\}\rangle, \\
& \dots\}
\end{aligned}
$$

Example



Heuristic

A **heuristic** h assigns a value to each state.
Lower values for 'better' states.

Simple Hill-climbing

`Simple Hill-climbing` is a heuristic search algorithm.

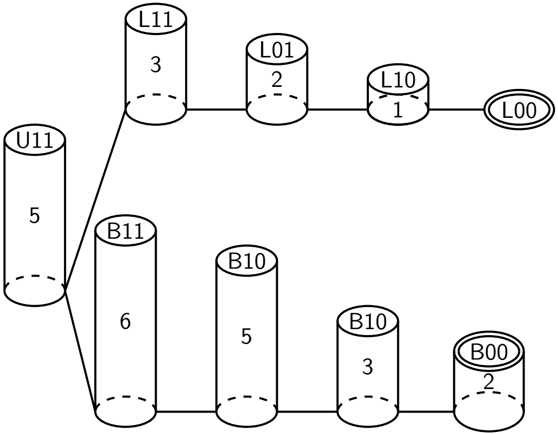
$s := I$
while $\gamma \nsubseteq s$ **do**
  if $\exists s' \in succ(s)$ **with** $h(s') < h(s)$ **then**
    $s := s'$
  else
    return fail
return s

Simple Hill-climbing

`Simple Hill-climbing` is guaranteed to find a goal state if the heuristic is **descending** and **dead-end avoiding** (DDA).

- Descending: each reachable, solvable (non-goal) state has an improving successor.
- Dead-end avoiding: Only solvable successors are improving.

DDA Heuristic



Potential Heuristic

Weighted count of the partial assignments that agree with the given state.

$$h^{pot}(s) = \sum_{p \in \mathcal{P}} (w(p) \cdot [p \subseteq s])$$

- $\mathcal{P}$ set of all possible partial assignments
- w weight function

Potential Heuristic

Weighted count of the partial assignments that agree with the given state.

$$h^{pot}(s) = \sum_{p \in \mathcal{P}} (w(p) \cdot [p \subseteq s])$$

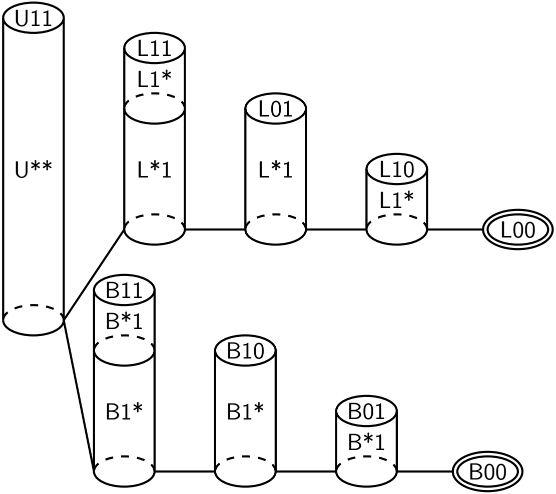
- $\mathcal{P}$ set of all possible partial assignments
- w weight function

Dimension of h^{pot} is maximal $|p|$ with $w(p) \neq 0$.

DDA Potential Heuristic

p	$w(p)$
U**	5
L1*	1
L*1	2
B1*	2
B*1	1

is DDA,
Dimension: 2



Correlation Complexity

Definition (Correlation Complexity)

The **correlation complexity** of a planning task Π is defined as the minimal dimension of all DDA potential heuristics for Π.

Correlation Complexity

Definition (Correlation Complexity)

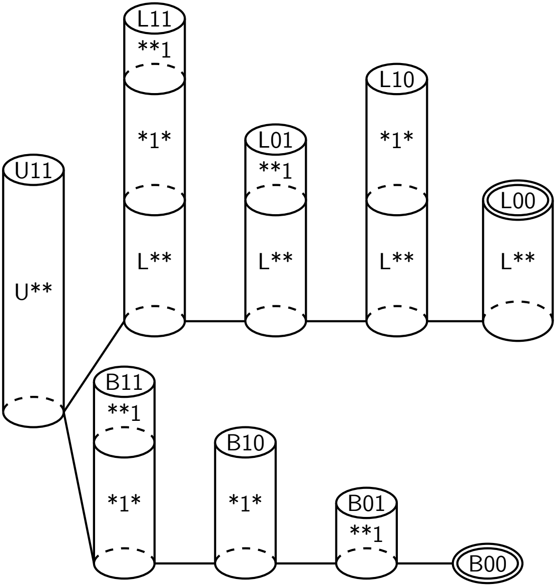
The **correlation complexity** of a planning task Π is defined as the minimal dimension of all DDA potential heuristics for Π.

Measures how 'hard' a planning task is.

DDA Potential Heuristic

p	$w(p)$
U**	4
L**	2
1	2
**1	1

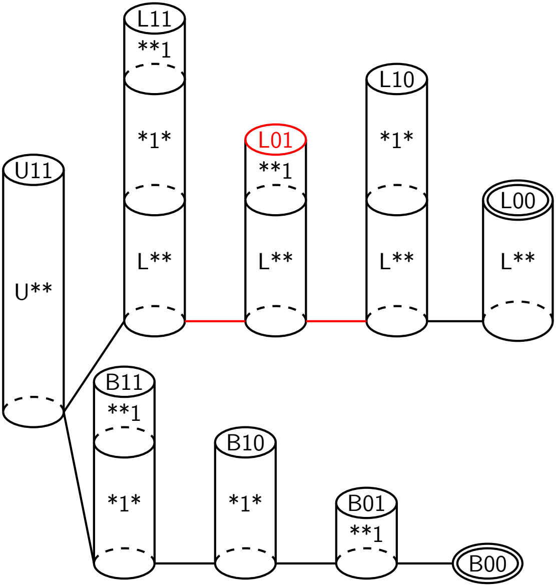
Dimension: 1
DDA?



DDA Potential Heuristic

p	$w(p)$
U**	4
L**	2
1	2
**1	1

Dimension: 1
DDA? No!



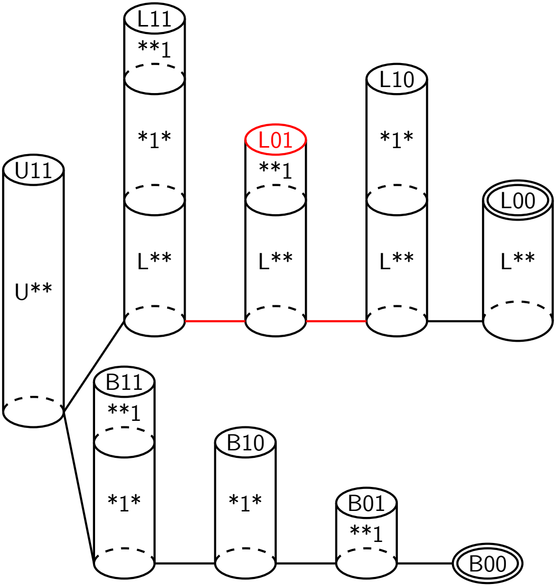
DDA Potential Heuristic

p	$w(p)$
U**	4
L**	2
1	2
**1	1

Dimension: 1

DDA? No!

But...



Practically Descending and Dead-end Avoiding

If `Simple Hill-climbing` is guaranteed to find a goal state, then the heuristic is **practically descending and dead-end avoiding** (PDDA).

Practically Descending and Dead-end Avoiding

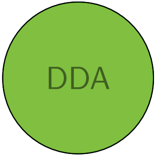
If `Simple Hill-climbing` is guaranteed to find a goal state, then the heuristic is **practically descending and dead-end avoiding** (PDDA).

- DDA $\Rightarrow$ `Simple Hill-climbing` finds goal
- `Simple Hill-climbing` finds goal $\Rightarrow$ PDDA

Practically Descending and Dead-end Avoiding

If `Simple Hill-climbing` is guaranteed to find a goal state, then the heuristic is **practically descending and dead-end avoiding** (PDDA).

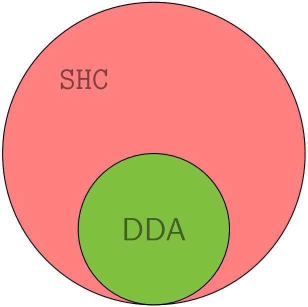
- DDA $\Rightarrow$ `Simple Hill-climbing` finds goal
- `Simple Hill-climbing` finds goal $\Rightarrow$ PDDA



Practically Descending and Dead-end Avoiding

If `Simple Hill-climbing` is guaranteed to find a goal state, then the heuristic is **practically descending and dead-end avoiding** (PDDA).

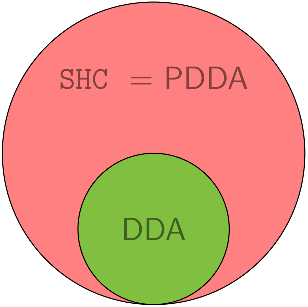
- DDA ⇒ `Simple Hill-climbing` finds goal
- `Simple Hill-climbing` finds goal ⇒ PDDA



Practically Descending and Dead-end Avoiding

If `Simple Hill-climbing` is guaranteed to find a goal state, then the heuristic is **practically descending and dead-end avoiding** (PDDA).

- DDA $\Rightarrow$ `Simple Hill-climbing` finds goal
- `Simple Hill-climbing` finds goal $\Rightarrow$ PDDA



Basel Measure

Definition (Basel Measure)

The **Basel measure** of a planning task Π is defined as the minimal dimension of all PDDA potential heuristics for Π.

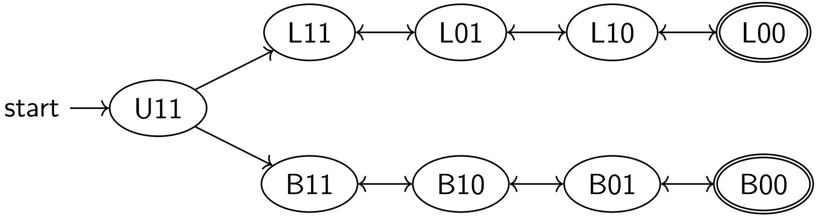
Basel Measure

Definition (Basel Measure)

The **Basel measure** of a planning task Π is defined as the minimal dimension of all PDDA potential heuristics for Π.

Theorem

Basel measure $\leq$ correlation complexity.



Correlation complexity: 2
Basel measure: 1

Novelty Width

- Based on a modification of `Breadth First Search`.
- Not states in the closed list but partial assignments of size k.
- If p is not in the closed list, then p is **novel**.
- Novelty width is the smallest k that guarantees to finds a plan.
- Measures how 'hard' a planning task is.

Novelty Width Algorithm

if $\gamma \in I$ **then**
  return I
$open := [I]$
$closed := \{p \mid p \subseteq I, |p| = k\}$
while open is not empty do
  $s :=$ pop first element of *open*
  foreach $s' \in$ **succ**(s) **do**
    if $\gamma \subseteq s'$ **then**
      return s'
    if $\exists p^* \subseteq s'$ **with** $|p^*| \leq k, p^* \notin$ **closed then**
      insert each $p \subseteq s'$ with $|p| = k$ in *closed*
      append s' to *open*
return fail

Novelty Width Algorithm

```
if γ ∈ I then
    return I
open := [I]
closed := {p | p ⊆ I, |p| = k}   ←
while open is not empty do
    s := pop first element of open
    foreach s' ∈ succ(s) do
        if γ ⊆ s' then
            return s'
        if ∃p* ⊆ s' with |p*| ≤ k, p* ∉ closed then
            insert each p ⊆ s' with |p| = k in closed
            append s' to open
return fail
```

Novelty Width Algorithm

if $\gamma \in I$ **then**
  return I
$open := [I]$
$closed := \{p \mid p \subseteq I, |p| = k\}$ ←
while open is not empty do
  $s :=$ pop first element of $open$
  foreach $s' \in$ **succ**(s) **do**
    if $\gamma \subseteq s'$ **then**
      return s'
    if $\exists p^* \subseteq s'$ **with** $|p^*| \leq k, p^* \notin$ **closed then** ←
      insert each $p \subseteq s'$ with $|p| = k$ in $closed$ ←
      append s' to $open$
return fail

Basel Measure vs. Novelty Width

Theorem

Basel measure $\leq$ novelty width $+1$

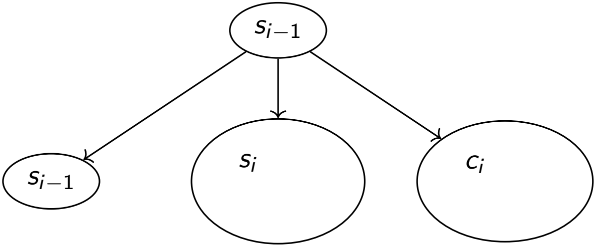
Basel Measure vs. Novelty Width

Proof sketch:

- states of plan found with novelty width algorithm: $s_0, s_1, \ldots, s_L$
- chose weights such that s_i is the only improving successor of s_{i-1}

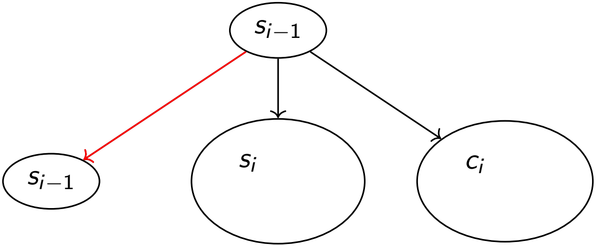
Basel Measure vs. Novelty Width

Part of the search tree:



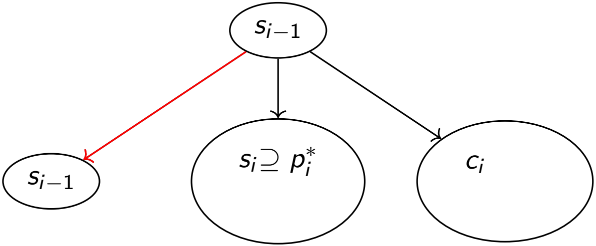
Basel Measure vs. Novelty Width

Part of the search tree:



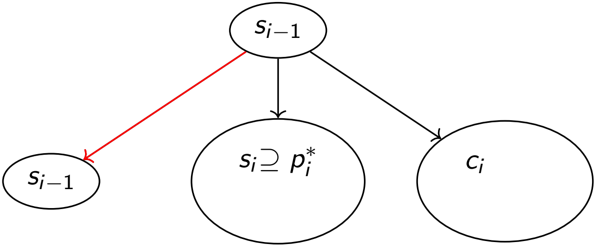
Basel Measure vs. Novelty Width

Part of the search tree:



Basel Measure vs. Novelty Width

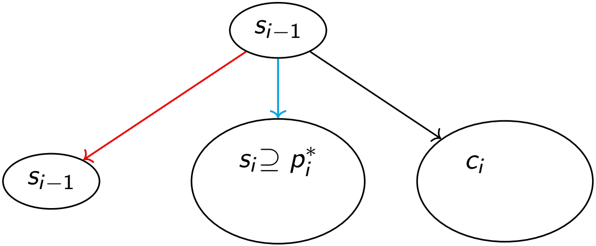
Part of the search tree:



$$w(p_i^*) := -\Omega^{2i}$$

Basel Measure vs. Novelty Width

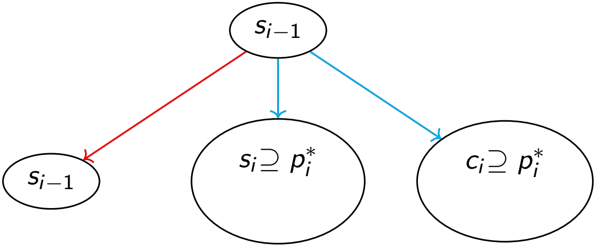
Part of the search tree:



$$w(p_i^*) := -\Omega^{2i}$$

Basel Measure vs. Novelty Width

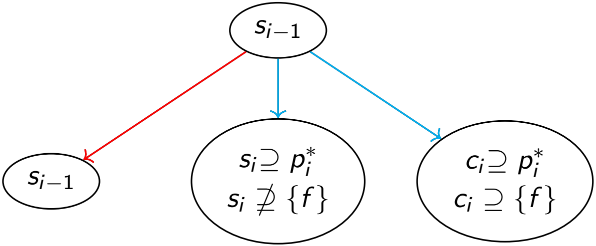
Part of the search tree:



$$w(p_i^*) := -\Omega^{2i}$$

Basel Measure vs. Novelty Width

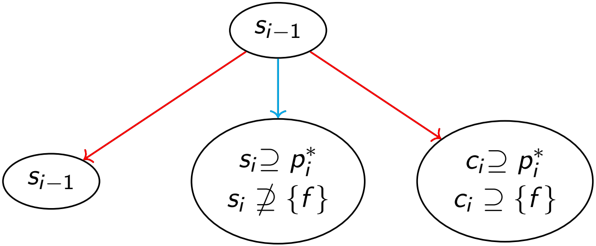
Part of the search tree:



$$w(p_i^*) := -\Omega^{2i}$$

Basel Measure vs. Novelty Width

Part of the search tree:



$w(p_i^*) := -\Omega^{2i}$ $\quad w(p_i^* \cup \{f\}) := +\Omega^{2i+1}$

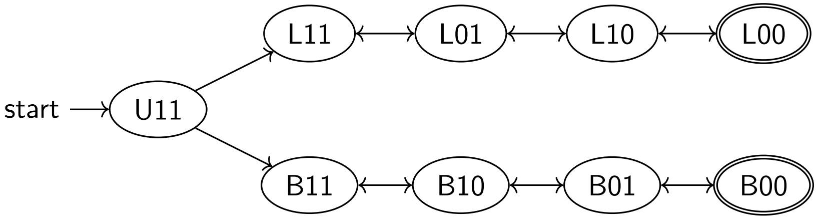
Basel Measure vs. Novelty Width

`Simple Hill-climbing` follows the plan found by the novelty width algorithm.
The heuristic is PDDA.

- $|p_i^*| =$ novelty width
- $|p_i^* \cup \{f\}| =$ novelty width $+1$

Basel measure is at most novelty width $+1$.

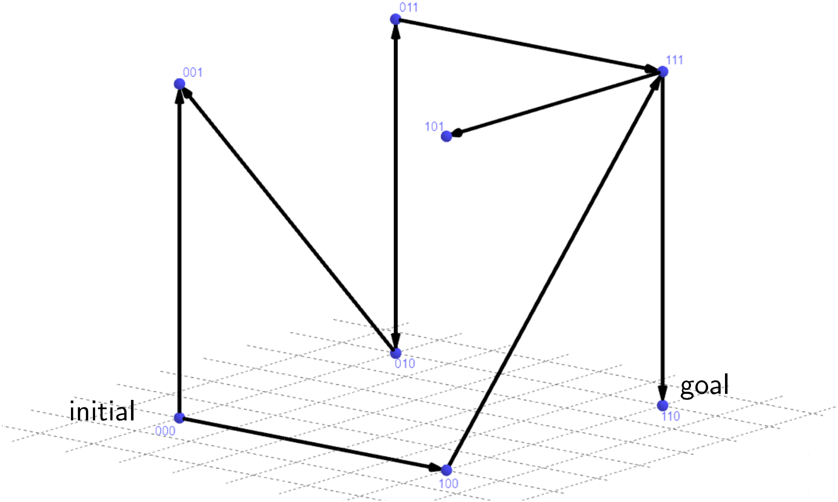
Example



Correlation complexity: 2. Why not 1?

State Space in 3D-Space

Treat state variables as dimensions.



Linear Algebra

Definition (Vectorization)

Let $\Pi = \langle V, I, O, \gamma \rangle$ a planning task with only $\{0, 1\}$ domains. The vector $\overrightarrow{t_{s,s'}} \in \mathbb{R}^{|V|}$ is the **vectorization** from the state s to the state s' where

$$\overrightarrow{t_{s,s'}}[i] := s'(v_i) - s(v_i)$$

for each $i \in \{1, \ldots, |V|\}$.

Linear Algebra

Definition (Vectorization)

Let $\Pi = \langle V, I, O, \gamma \rangle$ a planning task with only $\{0, 1\}$ domains. The vector $\overrightarrow{t_{s,s'}} \in \mathbb{R}^{|V|}$ is the **vectorization** from the state s to the state s' where

$$\overrightarrow{t_{s,s'}}[i] := s'(v_i) - s(v_i)$$

for each $i \in \{1, \ldots, |V|\}$.

Assume: $w(\{v \mapsto 0\}) = 0$ for each $v \in V$.
For 1-dimensional potential heuristics:

$$h^{pot}(s') - h^{pot}(s) = \sum_{v_i \in V} w(\{v_i \mapsto 1\}) \cdot \overrightarrow{t_{s,s'}}[i]$$

Linear Algebra

Definition (Vectorization)

Let $\Pi = \langle V, I, O, \gamma \rangle$ a planning task with only $\{0, 1\}$ domains. The vector $\overrightarrow{t_{s,s'}} \in \mathbb{R}^{|V|}$ is the **vectorization** from the state s to the state s' where

$$\overrightarrow{t_{s,s'}}[i] := s'(v_i) - s(v_i)$$

for each $i \in \{1, \ldots, |V|\}$.

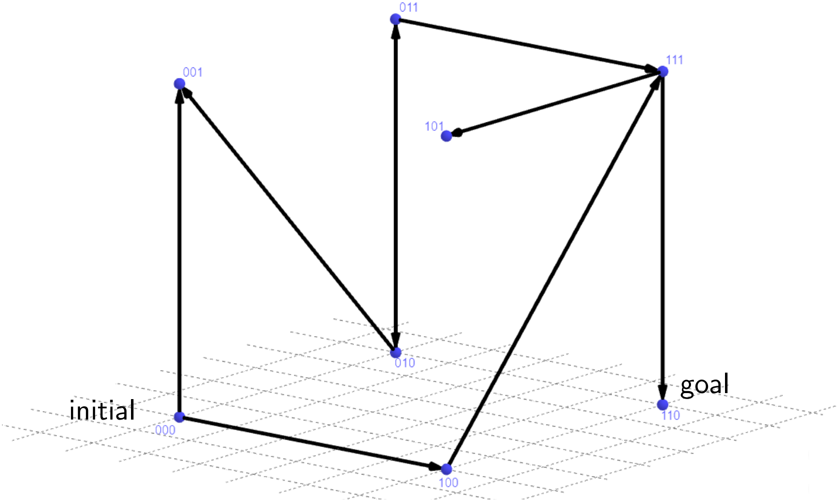
Assume: $w(\{v \mapsto 0\}) = 0$ for each $v \in V$.
For 1-dimensional potential heuristics:

$$h^{pot}(s') - h^{pot}(s) = \sum_{v_i \in V} w(\{v_i \mapsto 1\}) \cdot \overrightarrow{t_{s,s'}}[i]$$

Weight function w corresponds to a linear mapping.

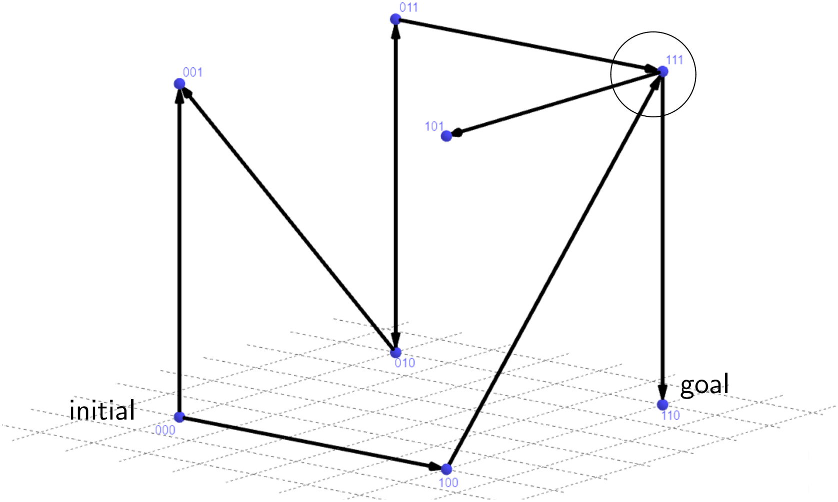
State Space in 3D-Space

Treat state variables as dimensions.



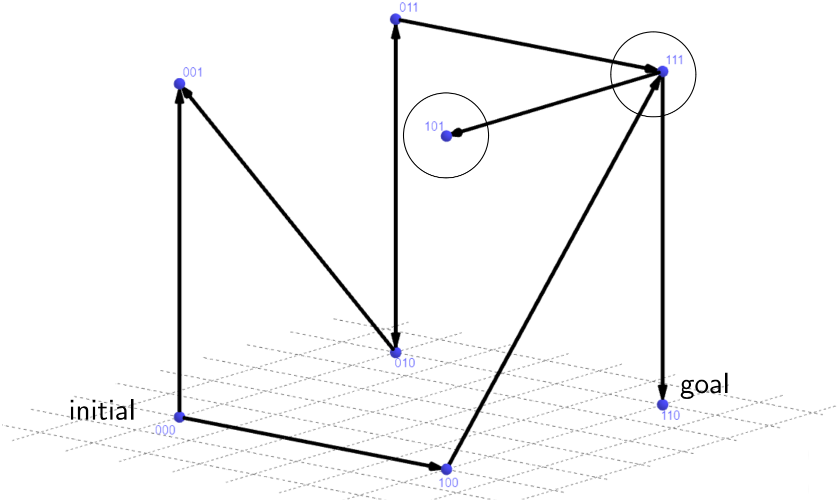
State Space in 3D-Space

Treat state variables as dimensions.



State Space in 3D-Space

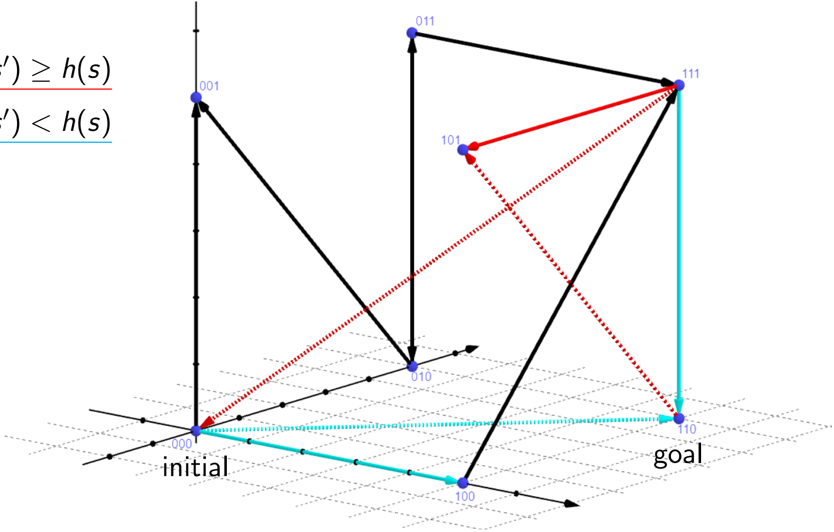
Treat state variables as dimensions.



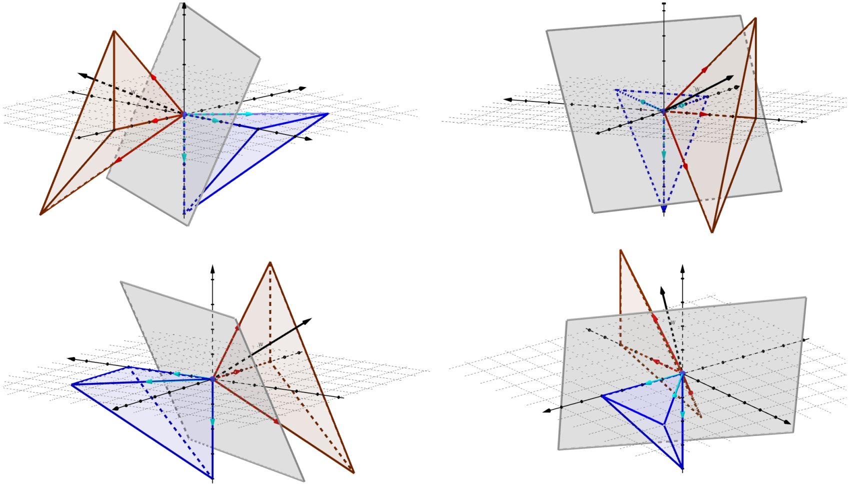
Planning Task in 3D-Space

$h(s') \geq h(s)$

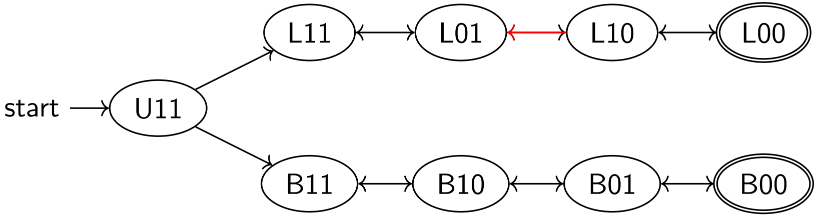
$h(s') < h(s)$



Separating Hyperplane

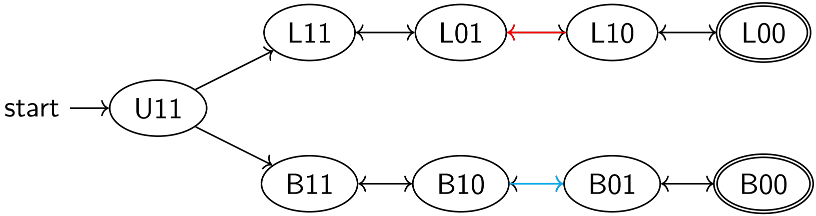


Example



For each DDA heuristic:
$h(L01) \geq h(L10) \Rightarrow \overrightarrow{t_{L10,L01}}$

Example

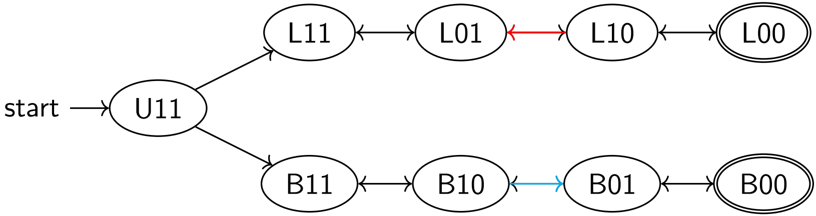


For each DDA heuristic:

$h(L01) \geq h(L10) \Rightarrow \overrightarrow{t_{L10,L01}}$

$h(B01) < h(B10) \Rightarrow \overrightarrow{t_{B10,B01}}$

Example



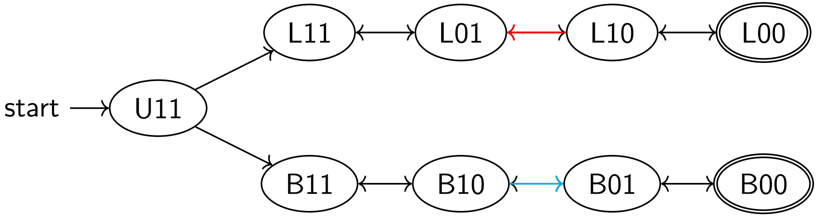
For each DDA heuristic:

$h(L01) \geq h(L10) \Rightarrow \overrightarrow{t_{L10,L01}}$

$h(B01) < h(B10) \Rightarrow \overrightarrow{t_{B10,B01}}$

$\overrightarrow{t_{L10,L01}} = \overrightarrow{t_{B10,B01}} \neq \overrightarrow{0}$

Example



For each DDA heuristic:

$h(L01) \geq h(L10) \Rightarrow \overrightarrow{t_{L10,L01}}$

$h(B01) < h(B10) \Rightarrow \overrightarrow{t_{B10,B01}}$

$\overrightarrow{t_{L10,L01}} = \overrightarrow{t_{B10,B01}} \neq \overrightarrow{0} \Rightarrow$ no separating hyperplane exists $\Rightarrow h$ is at least 2-dimensional $\Rightarrow$ correlation complexity is at least 2.

Linear Algebra

Detects correlation complexity of at least 2 on more tasks than other approaches in literature.

Find Tasks with Basel Measure 1

p	$w(p)$
U**	5
1	2
**1	1

Constraints:

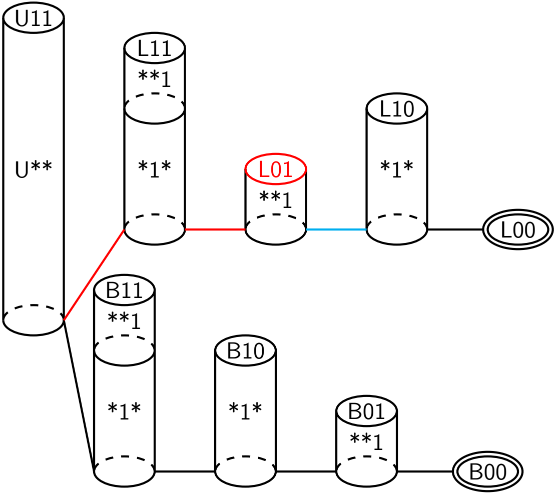
$h(U11) \leq h(L11)$

or

$h(L11) \leq h(L01)$

or

$h(L01) > h(L10)$



Find Tasks with Basel Measure 1

- Mixed Integer Program to refine h.
- Refine until h is PDDA $\Rightarrow$ Basel measure $= 1$.
- or solution space is empty $\Rightarrow$ Basel measure ≥ 2.

Results

task	Basel measure	task	Basel measure
gripper:		visitall-opt11-strips:	
prob01.pddl	≥ 2	problem02-full.pddl	1
prob02.pddl	≥ 2	problem02-half.pddl	1
prob03.pddl	≥ 2	problem03-full.pddl	1
prob04.pddl	≥ 2	problem03-half.pddl	≥ 2
movie:		pegsol-08-strips:	
prob01.pddl	1	p01.pddl	1
prob02.pddl	1	p02.pddl	≥ 2
prob03.pddl	1		
prob04.pddl	1		

Results

task	Basel measure	task	Basel measure
`gripper:`		`visitall-opt11-strips:`	
`prob01.pddl`	≥ 2	`problem02-full.pddl`	1
`prob02.pddl`	≥ 2	`problem02-half.pddl`	1
`prob03.pddl`	≥ 2	`problem03-full.pddl`	1
`prob04.pddl`	≥ 2	`problem03-half.pddl`	≥ 2
`movie:`		`pegsol-08-strips:`	
`prob01.pddl`	1	`p01.pddl`	1
`prob02.pddl`	1	`p02.pddl`	≥ 2
`prob03.pddl`	1		
`prob04.pddl`	1		

Conclusion

- Basel measure $\leq$ correlation complexity.
- Basel measure $\leq$ novelty width $+1$.
- We can use linear algebra to detect a correlation complexity of at least 2.
- Some IPC tasks have Basel measure of 1.
- In practice translation can change the Basel measure.