Classical Planning Heuristics

6. Heuristics for Classical Planning II

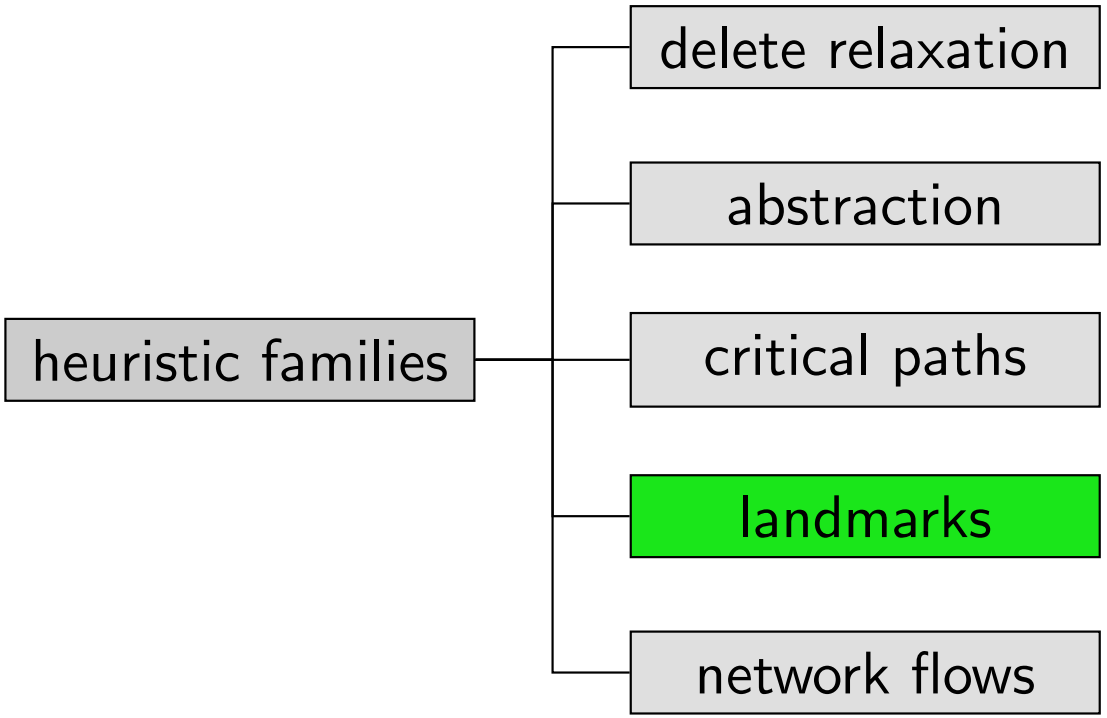
Gabriele Röger

ICAPS 2018 Summer School

June 22, 2018

Landmarks

Five Families of Heuristics



Planning Heuristics: Landmarks

Five classes of heuristics:

4. Landmarks

An action set A is a landmark if all plans include an action from A.

Compute a set of landmarks and use it to derive a cost estimate

Example: Landmarks in FreeCell

Landmarks in FreeCell:

- The set of actions that move the $\heartsuit Q$ to foundations.
- The set of actions that move the $\clubsuit 7$ away from the $\diamondsuit 8$.
- ...

Landmarks

Definition (Landmark)

A (disjunctive action) landmark is a set of operators L such that each plan must contain some element of L.

The cost of a landmark, $cost(L)$, is $\min_{o \in L} cost(o)$.

$\rightsquigarrow$ the cost of any landmark is a (crude) admissible heuristic

Example: Landmarks

Example (STRIPS task)

- Initially, only i is true
- In the goal, g should be true
- Four (delete-free) actions:
  - $pre(a_1) = \{i\}, add(a_1) = \{a, b\}, cost(a_1) = 3$
  - $pre(a_2) = \{i\}, add(a_2) = \{a, c\}, cost(a_2) = 4$
  - $pre(a_3) = \{i\}, add(a_3) = \{b, c\}, cost(a_3) = 5$
  - $pre(a_4) = \{a, b, c\}, add(a_4) = \{g\}, cost(a_4) = 0$

Example: Landmarks

Example (STRIPS task)

- Initially, only i is true
- In the goal, g should be true
- Four (delete-free) actions:
  - $pre(a_1) = \{i\}, add(a_1) = \{a, b\}, cost(a_1) = 3$
  - $pre(a_2) = \{i\}, add(a_2) = \{a, c\}, cost(a_2) = 4$
  - $pre(a_3) = \{i\}, add(a_3) = \{b, c\}, cost(a_3) = 5$
  - $pre(a_4) = \{a, b, c\}, add(a_4) = \{g\}, cost(a_4) = 0$

Some landmarks:

- $W = \{a_4\}$ (cost 0)
- $X = \{a_1, a_2\}$ (cost 3)
- $Y = \{a_1, a_3\}$ (cost 3)
- $Z = \{a_2, a_3\}$ (cost 4)
- but also: $\{a_1, a_2, a_3\}$ (cost 3), $\{a_1, a_2, a_4\}$ (cost 0), ...

Exploiting Landmarks

Assume we are given landmark set $\mathcal{L} = \{W, X, Y, Z\}$.

How do we exploit $\mathcal{L}$ for heuristics?

- sum of costs $0 + 3 + 3 + 4 = 10 \rightsquigarrow$ inadmissible!
- maximum of costs: $\max\{0, 3, 3, 4\} = 4 \rightsquigarrow$ weak
- best previous approach: optimal cost partitioning

Landmark Heuristics with Optimal Cost Partitioning

Optimal Cost Partitioning (Karpas & Domshlak, IJCAI 2009)

Idea: Derive a linear program (LP) from $\mathcal{L}$.

- one variable per landmark
- one constraint per operator

h^{L} value: objective value of the LP

Example: Optimal Cost Partitioning

Example

$cost(a_1) = 3, \quad cost(a_2) = 4, \quad cost(a_3) = 5, \quad cost(a_4) = 0$

$\mathcal{L} = \{W, X, Y, Z\}$
with $W = \{a_4\}, \quad X = \{a_1, a_2\}, \quad Y = \{a_1, a_3\}, \quad Z = \{a_2, a_3\}$

LP: maximize $w + x + y + z$ subject to

$$\begin{array}{rcrcrcl} & & x & + & y & \leq & 3 \\ & & x & + & z & \leq & 4 \\ & & & y & + & z & \leq & 5 \\ w & & & & & \leq & 0 \end{array}$$

and $w, x, y, z \geq 0$

Example: Optimal Cost Partitioning

Example

$cost(a_1) = 3, \quad cost(a_2) = 4, \quad cost(a_3) = 5, \quad cost(a_4) = 0$

$\mathcal{L} = \{W, X, Y, Z\}$
with $W = \{a_4\}, \quad X = \{a_1, a_2\}, \quad Y = \{a_1, a_3\}, \quad Z = \{a_2, a_3\}$

LP: maximize $w + x + y + z$ subject to

$$\begin{array}{cccccccc} & x & + & y & & & \leq & 3 \quad a_1 \\ & x & + & & & z & \leq & 4 \quad a_2 \\ & & & y & + & z & \leq & 5 \quad a_3 \\ w & & & & & & \leq & 0 \quad a_4 \\ W & X & & Y & & Z & & \end{array}$$

and $w, x, y, z \geq 0$

Example: Optimal Cost Partitioning

Example

$cost(a_1) = 3, \quad cost(a_2) = 4, \quad cost(a_3) = 5, \quad cost(a_4) = 0$

$\mathcal{L} = \{W, X, Y, Z\}$
with $W = \{a_4\}, \quad X = \{a_1, a_2\}, \quad Y = \{a_1, a_3\}, \quad Z = \{a_2, a_3\}$

LP: maximize $w + x + y + z$ subject to

$$\begin{array}{cccccccc}
 & x & + & y & & \leq & 3 & a_1 \\
 & x & + & & z & \leq & 4 & a_2 \\
 & & & y & + z & \leq & 5 & a_3 \\
w & & & & & \leq & 0 & a_4 \\
W & X & & Y & Z & & &
\end{array}$$

and $w, x, y, z \geq 0$

solution: $w = 0, \; x = 1, \; y = 2, \; z = 3 \quad \rightsquigarrow \quad h^{\mathrm{L}}(I) = 6$

Dual

Dual Perspective (Bonet & Helmert, ECAI 2010)

LP with same objective value but role of constraints and variables interchanged

- one variable per operator
- one constraint per landmark

Example: Dual

Example

$cost(a_1) = 3, \quad cost(a_2) = 4, \quad cost(a_3) = 5, \quad cost(a_4) = 0$

$\mathcal{L} = \{W, X, Y, Z\}$
with $W = \{a_4\}, \quad X = \{a_1, a_2\}, \quad Y = \{a_1, a_3\}, \quad Z = \{a_2, a_3\}$

minimize $cost(a_1)\,Y_{a_1} + cost(a_2)\,Y_{a_2} + cost(a_3)\,Y_{a_3} + cost(a_4)\,Y_{a_4}$
subject to

$$\begin{array}{rcrcrcl} & & & & Y_{a_4} & \geq & 1 \\ Y_{a_1} & + & Y_{a_2} & & & \geq & 1 \\ Y_{a_1} & + & & Y_{a_3} & & \geq & 1 \\ & & Y_{a_2} & + & Y_{a_3} & \geq & 1 \end{array}$$

and $Y_{a_1} \geq 0, Y_{a_2} \geq 0, Y_{a_3} \geq 0, Y_{a_4} \geq 0$

Example: Dual

Example

$cost(a_1) = 3,\ \ cost(a_2) = 4,\ \ cost(a_3) = 5,\ \ cost(a_4) = 0$

$\mathcal{L} = \{W, X, Y, Z\}$
with $W = \{a_4\},\ \ X = \{a_1, a_2\},\ \ Y = \{a_1, a_3\},\ \ Z = \{a_2, a_3\}$

minimize $cost(a_1)\,Y_{a_1} + cost(a_2)\,Y_{a_2} + cost(a_3)\,Y_{a_3} + cost(a_4)\,Y_{a_4}$
subject to

$$\begin{array}{rcrcrcll} & & & & & Y_{a_4} & \geq & 1 \quad W \\ Y_{a_1} & + & Y_{a_2} & & & & \geq & 1 \quad X \\ Y_{a_1} & + & & & Y_{a_3} & & \geq & 1 \quad Y \\ & & Y_{a_2} & + & Y_{a_3} & & \geq & 1 \quad Z \end{array}$$

and $Y_{a_1} \geq 0, Y_{a_2} \geq 0, Y_{a_3} \geq 0, Y_{a_4} \geq 0$

Example: Dual

Example

$cost(a_1) = 3, \quad cost(a_2) = 4, \quad cost(a_3) = 5, \quad cost(a_4) = 0$

$\mathcal{L} = \{W, X, Y, Z\}$
with $W = \{a_4\}, \quad X = \{a_1, a_2\}, \quad Y = \{a_1, a_3\}, \quad Z = \{a_2, a_3\}$

minimize $cost(a_1)\,Y_{a_1} + cost(a_2)\,Y_{a_2} + cost(a_3)\,Y_{a_3} + cost(a_4)\,Y_{a_4}$
subject to

$$\begin{array}{rcrcrcll} & & & & & Y_{a_4} & \geq 1 & W \\ Y_{a_1} & + & Y_{a_2} & & & & \geq 1 & X \\ Y_{a_1} & + & & & Y_{a_3} & & \geq 1 & Y \\ & & Y_{a_2} & + & Y_{a_3} & & \geq 1 & Z \end{array}$$

and $Y_{a_1} \geq 0, Y_{a_2} \geq 0, Y_{a_3} \geq 0, Y_{a_4} \geq 0$

solution: $Y_{a_4} = 1, \quad Y_{a_2} = Y_{a_3} = Y_{a_4} = 0.5 \quad \rightsquigarrow \quad h^{\mathsf{L}}(I) = 6$

Beyond Optimal Cost Partitioning

- $h^{\mathsf{L}}(I) = 6$ is a good estimate, but $h^{+}(I) = 7$!
- Can we do better with the same information?

Hitting Sets

Definition (Hitting Set)

Given: finite set A, subset family $\mathcal{F} \subseteq 2^A$, costs $c : A \to \mathbb{R}_0^+$

Hitting set:

- subset $H \subseteq A$ that "hits" all subsets in $\mathcal{F}$: $H \cap S \neq \emptyset$ for all $S \in \mathcal{F}$
- cost of H: $\sum_{a \in H} c(a)$

Minimum hitting set (MHS):

- minimizes cost
- classical NP-complete problem (Karp, 1972)

Example: Hitting Sets

Example

$A = \{a_1, a_2, a_3, a_4\}$

$\mathcal{F} = \{W, X, Y, Z\}$
with $W = \{a_4\}$, $X = \{a_1, a_2\}$, $Y = \{a_1, a_3\}$, $Z = \{a_2, a_3\}$

$c(a_1) = 3$, $c(a_2) = 4$, $c(a_3) = 5$, $c(a_4) = 0$

Minimum hitting set:

Example: Hitting Sets

Example

$A = \{a_1, a_2, a_3, a_4\}$

$\mathcal{F} = \{W, X, Y, Z\}$
with $W = \{a_4\}$, $X = \{a_1, a_2\}$, $Y = \{a_1, a_3\}$, $Z = \{a_2, a_3\}$

$c(a_1) = 3$, $c(a_2) = 4$, $c(a_3) = 5$, $c(a_4) = 0$

Minimum hitting set: $\{a_1, a_2, a_4\}$ with cost $3 + 4 + 0 = 7$

Hitting Sets for Landmarks

- can view landmark sets (with operator costs) as instances of minimum hitting set problem
- here, we got an admissible estimate that dominated h^{L}
- coincidence?

Hitting Set Heuristics

Let $\mathcal{L}$ be a set of landmarks.

Theorem (Hitting Set Heuristics are Admissible)

Let $h^{\text{MHS}}(I)$ be the minimum hitting set cost for $\langle A, \mathcal{L}, cost \rangle$.
Then:

1. $h^{\text{MHS}}(I) \leq h^{+}(I)$ *(hitting set heuristics are admissible)*
2. $h^{\text{MHS}}(I) \geq h^{\text{L}}(I)$ *(hitting sets dominate cost partitioning)*

Hitting Set Heuristics

Let $\mathcal{L}$ be a set of landmarks.

Theorem (Hitting Set Heuristics are Admissible)

Let $h^{\text{MHS}}(I)$ be the minimum hitting set cost for $\langle A, \mathcal{L}, cost \rangle$.
Then:

1. $h^{\text{MHS}}(I) \leq h^{+}(I)$ *(hitting set heuristics are admissible)*
2. $h^{\text{MHS}}(I) \geq h^{\text{L}}(I)$ *(hitting sets dominate cost partitioning)*

Proof sketch:

1. plans are hitting sets (by definition of landmarks)
2. dual of cost partitioning LP is LP relaxation of hitting set integer program

Landmark Heuristics in the Literature

Landmark heuristics in the literature (admissible in red):

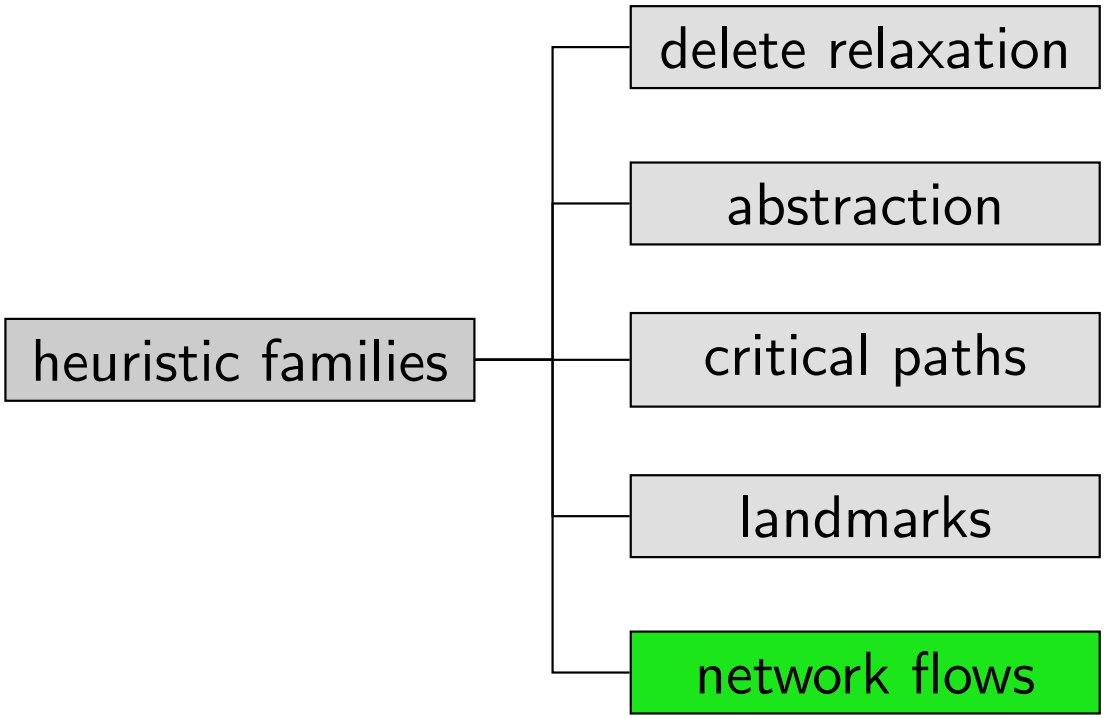
- LAMA heuristic (Richter et al., AAAI 2008)
- cost-partitioned landmarks (Karpas & Domshlak, IJCAI 2009)
- conjunctive landmarks (Keyder et al., ECAI 2010)
- landmark-cut heuristic (Helmert & Domshlak, ICAPS 2009)

in Fast Downward

`lmcount(...)`, `lmcut()`

Network Flows

Five Families of Heuristics



Planning Heuristics: Network Flows

Five classes of heuristics:

5. Network Flows

In every plan, the number of times each fact is produced vs. the number of times it is consumed must be "balanced".
Solve a linear program encoding this information.

Example: Flow Constraints in FreeCell

Flow constraints in FreeCell:

- Each card is moved into a free cell as often as it is moved out of a free cell.
- Each card is buried in a stack as often as it is uncovered.

(Give or take a bit to account for current state and goal.)

Example SAS$^+$ Task

One package, two trucks, two locations

- Variables describe positions of trucks and packages
- Package initially at location 1 and trucks at location 2
- Goal: package at location 2, truck 1 at location 1

Example Task: Observations

Consider some atoms of the example task:

- $\mathit{pos}\text{-}p = \mathit{loc}_1$ initially true and must be false in the goal
  - ▷ at location 1 the package must be loaded one time more often than unloaded.

Can we derive a heuristic from this kind of information?

Example Task: Observations

Consider some atoms of the example task:

- $pos\text{-}p = loc_1$ initially true and must be false in the goal
  - ▷ at location 1 the package must be loaded one time more often than unloaded.
- $pos\text{-}p = loc_2$ initially false and must be true in the goal
  - ▷ at location 2 the package must be unloaded one time more often than loaded.

Can we derive a heuristic from this kind of information?

Example Task: Observations

Consider some atoms of the example task:

- $\mathit{pos}\text{-}p = \mathit{loc}_1$ initially true and must be false in the goal
  - ▷ at location 1 the package must be loaded one time more often than unloaded.
- $\mathit{pos}\text{-}p = \mathit{loc}_2$ initially false and must be true in the goal
  - ▷ at location 2 the package must be unloaded one time more often than loaded.
- $\mathit{pos}\text{-}p = t_1$ initially false and must be false in the goal
  - ▷ same number of load and unload actions for truck 1.

Can we derive a heuristic from this kind of information?

Example: Flow Constraints

Let π be some arbitrary plan for the example task and let $\#o$ denote the number of occurrences of operator o in π.
Then the following holds:

- $\mathit{pos}\text{-}p = \mathit{loc}_1$ initially true and must be false in the goal
  - ▷ at location 1 the package must be loaded one time more often than unloaded.

  $\#\mathit{load}(t_1, \mathit{loc}_1) + \#\mathit{load}(t_2, \mathit{loc}_1) =$
  $1 + \#\mathit{unload}(t_1, \mathit{loc}_1) + \#\mathit{unload}(t_2, \mathit{loc}_1)$
- $\mathit{pos}\text{-}p = t_1$ initially false and must be false in the goal
  - ▷ same number of load and unload actions for truck 1.

  $\#\mathit{unload}(t_1, \mathit{loc}_1) + \#\mathit{unload}(t_1, \mathit{loc}_2) =$
  $\#\mathit{load}(t_1, \mathit{loc}_1) + \#\mathit{load}(t_1, \mathit{loc}_2)$

Classification of Actions

Actions that always produce atom $V = v$:

$AP_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v \text{ and } \mathit{pre}(a)[V] = v' \text{ with } v' \neq v\}$

Classification of Actions

Actions that always produce atom $V = v$:
$AP_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v \text{ and } \mathit{pre}(a)[V] = v' \text{ with } v' \neq v\}$

Actions that sometimes produce atom $V = v$:
$SP_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v \text{ and } \mathit{pre}(a)[V] \text{ is undefined}\}$

Classification of Actions

Actions that always produce atom $V = v$:
$AP_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v \text{ and } \mathit{pre}(a)[V] = v' \text{ with } v' \neq v\}$

Actions that sometimes produce atom $V = v$:
$SP_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v \text{ and } \mathit{pre}(a)[V] \text{ is undefined}\}$

Actions that always consume atom $V = v$:
$AC_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v' \text{ with } v \neq v' \text{ and } \mathit{pre}(a)[V] = v\}$

Classification of Actions

Actions that always produce atom $V = v$:
$AP_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v \text{ and } \mathit{pre}(a)[V] = v' \text{ with } v' \neq v\}$

Actions that sometimes produce atom $V = v$:
$SP_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v \text{ and } \mathit{pre}(a)[V] \text{ is undefined}\}$

Actions that always consume atom $V = v$:
$AC_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v' \text{ with } v \neq v' \text{ and } \mathit{pre}(a)[V] = v\}$

Actions that sometimes consume atom $V = v$:
$SC_{V=v} = \{a \in A \mid \mathit{eff}(a)[V] = v' \text{ with } v \neq v'$
$\text{and } \mathit{pre}(a)[V] \text{ is undefined}\}$

Flow Constraint for Atom $V = v$

$$\sum_{a \in AP_{V=v}} Y_a + \sum_{a \in SP_{V=v}} Y_a - \sum_{a \in AC_{V=v}} Y_a \geq LB$$

- Variable Y_a for each action a
- LB is lower bound on net change of atom $V = v$
  - 1 if atom is false but required to be true in goal
  - -1 if atom is true but not required to be true in the goal
  - 0 otherwise

Network Flow Heuristic

Definition (Flow Heuristic)

The flow heuristic $h^{\text{flow}}(s)$ is the objective value of the following LP or ∞ if the LP is infeasible:

$$\text{minimize} \quad \sum_{a \in A} \mathit{cost}(a) \cdot Y_a \quad \text{subject to}$$

$$\text{the flow constraints for all atoms } V = v \text{ and}$$

$$Y_a \geq 0 \quad \text{for all } a \in A$$

Side Note: Operator Counting Framework

- same variables and objective function as dual landmark LP
- there are some more heuristics with such a formulation
- can combine constraints in one LP
- dominates maximum of component heuristics
- related to a more general notion of cost partitioning

Network Flow Heuristics in the Literature

Network flow heuristics in the literature (admissible in red):

- flow heuristic (van den Briel et al., CP 2007)
- state equation heuristic (Bonet, IJCAI 2013)
- enhanced flow heuristics (Bonet & van den Briel, ICAPS 2014)
- relationship to abstraction and cost partitioning (Pommerening et al., ICAPS 2017)

in Fast Downward

```
operatorcounting([state_equation_constraints(),...])
```

What else happens. . .
. . . in heuristic planning?

Search Algorithms

Search algorithms mostly interesting for satisficing planning:

- escaping local minima and heuristic plateaus:
  - using multiple heuristics (e.g. Röger & Helmert, ICAPS 2010)
  - combining local and systematic search
    (e.g. Hoffmann & Nebel, JAIR 2001; Xie et al., AAAI 2014)
  - randomized systematic search (e.g. Valenzano et al., AAAI 2014)
  - random-walk algorithms (e.g. Nakhost & Müller, IJCAI 2013)
- search algorithms tailored to specific objectives:
  - anytime strategies for satisficing planning
    (Richter & Westphal, JAIR 2010)
  - many papers from Wheeler Ruml's research group
    (e.g., Thayer et al., ICAPS 2012)

Search Space Pruning

Search space pruning (mainly for optimal search):

- with symmetries (e.g., Pochter et al., IJCAI 2011)
- with redundant paths ("partial-order reduction"):
  (e.g. Wehrle et al., ICAPS 2013)

Invariant Synthesis

- invariant: property of all reachable states
  (e.g., Rintanen, ECAI 2008)
- example: mutually exclusive atoms
  - set of atoms of which at most one is true in each state
  - basis of translation from PDDL to SAS$^+$ (Helmert, AIJ 2009)
  - useful for strengthening heuristics, e.g., constrained PDBs
    (Haslum et al., AAAI 2005)

Preferred Operators

- heuristics often can identify promising actions
- for example helpful action: first action in a relaxed plan
  (Hoffmann & Nebel, JAIR 2001)
- try these actions first during search
- often highly positive impact on overall performance
  (Richter & Helmert, ICAPS 2009)

Portfolios

- Idea: solve tasks by running multiple (more or less) independent planning systems
- different strategies:
  - fixed schedule
  - select planners after task analysis

$\rightsquigarrow$ results of IPC 2014 "classic tracks"
`http://helios.hud.ac.uk/scommv/IPC-14/`

$\rightsquigarrow$ results of IPC 2014 "learning track"
`http://www.cs.colostate.edu/~ipc2014/`

Unsolvability

- concentrates on detecting/showing unsolvability
- tailoring heuristics for this purpose
  (e.g. Hoffmann et al., ECAI 2014)
- incremental learning (Steinmetz and Hoffmann AIJ 2017)
- proofs of unsolvability (Eriksson et al., ICAPS 2017; 2018)
- Unsolvability IPC 2016
  `https://unsolve-ipc.eng.unimelb.edu.au/`

The End

Thank you for your attention!