

domain-independent heuristics

A*, greedy-best first search, LRTA*, ...

all use an estimate of the number of actions needed to reach the goal

estimate = not exact value
only used as a guide



domain-dependent and domain-independent heuristics

examples:

domain-dependent

when searching in a planar grid, estimate is Eclidean distance

domain-independent

in a STRIPS problem, estimate is number of goal literals not yet true

[note] Some heuristics are for a specific domain. They may work well on that, but cannot be applied at all to other problems. On the converse, domain-independent heuristics are formulated on a general problem description language such as STRIPS or SAS+, which means that they can be applied to every problem that can be expressed in that language.

admissibility

in general:

good heuristics = close to the actual cost of reaching the goal

also relevant:

admissibility: $\text{heuristics} \leq \text{actual cost}$

why:

A* finds the optimal solution before the other if heuristics is admissible



specific heuristics

- pattern database
- merge and shrink
- structural patterns
- relaxation heuristics
- critical path
- disjunctive action landmarks

[note] These heuristics are now briefly described. They are detailed in a separate page each.

pattern database

remove some variables from the problem

simplifies problem, goal easier to reach \Rightarrow admissible heuristics

merge and shrink

start with a simplified version of the search space

refine it to make it more similar to the actual search space
while keeping size small

structural pattern

simplify the problem by making it fall into a tractable case

how: chop each action into a sequence of simpler actions



relaxation heuristics

remove negative effects

planning stil hard

simplify interactions between actions

[note] Just removing negative effects does not make the problem easy to solve. What complicates it is the interaction between actions: two variables may be made true by the same actions or by two separate actions. These heuristics take a simplistic view of these interactions.

disjunctive action landmarks

example: every plan contains either action c or action d

heuristics derived from costs of actions

critical path

cost of obtaining the goal = cost of obtaining its hardest part

the same for any other set of literals

symbols

$h^*(s)$

the actual cost of reaching the goal from state s
the "perfect" heuristics
hard to determine

$h^+(s)$

same, but in the problem without negative effects
used by some heuristics: delete relaxation, disjunctive action landmarks, critical path
still hard to determine, approximations are used



in details

- [pattern database](#)
- [merge and shrink](#)
- [structural patterns](#)
- [delete relaxation](#)
- [disjunctive action landmarks](#)
- [critical path](#)

properties:

admissibility, additivity, no-split-effects pattern database, cost partitioning, precomputing

others

[note] The page on the properties of heuristics is intended to be read after that on pattern database.