Foundations of Artificial Intelligence

13. State-Space Search: Heuristics

Malte Helmert

University of Basel

March 22, 2021

Foundations of Artificial Intelligence

March 22, 2021 — 13. State-Space Search: Heuristics

State-Space Search: Overview

Chapter overview: state-space search

- 5.–7. Foundations
- 8.–12. Basic Algorithms
- 13.–19. Heuristic Algorithms
  - 13. Heuristics
  - 14. Analysis of Heuristics
  - 15. Best-first Graph Search
  - 16. Greedy Best-first Search, A*, Weighted A*
  - 17. IDA*
  - 18. Properties of A*, Part I
  - 19. Properties of A*, Part II

13.1 Introduction

Informed Search Algorithms

- search algorithms considered so far: blind because they do not use any aspects of the problem to solve other than its formal definition (state space)
- problem: scalability
  ⇝ prohibitive time and space requirements already for seemingly simple problems
- idea: try to find (problem-specific) criteria to distinguish good and bad states
  ⇝ prefer good states

⇝ informed ("heuristic") search algorithms

13.2 Heuristics

Heuristics

Definition (heuristic)

Let $\mathcal{S}$ be a state space with states S.
A heuristic function or heuristic for $\mathcal{S}$ is a function

$$h : S \to \mathbb{R}_0^+ \cup \{\infty\},$$

mapping each state to a non-negative number (or ∞).

Heuristics: Intuition

idea: $h(s)$ estimates distance (= cost of cheapest path) from s to closest goal state

- heuristics can be arbitrary functions
- intuition: the closer h is to true goal distance, the more efficient the search using h

Heuristics are sometimes defined for search nodes instead of states, but this increased generality is rarely useful. (Why?)

Why "Heuristic"?

What does "heuristic" mean?

- heuristic: from ancient Greek ἑυρισκω (= I find)
  ⇝ compare: ἑυρηκα!
- popularized by George Pólya: How to Solve It (1945)
- in computer science often used for:
  rule of thumb, inexact algorithm
- in state-space search technical term for goal distance estimator

Representation of Heuristics

In our black box model, heuristics are an additional element of the state space interface:

State Spaces as Black Boxes (Extended)

- init()
- is_goal(s)
- succ(s)
- cost(a)
- h(s): heuristic value for state s
  result: non-negative integer or ∞

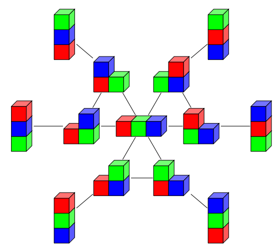
13.3 Examples

Example: Blocks World

possible heuristic:

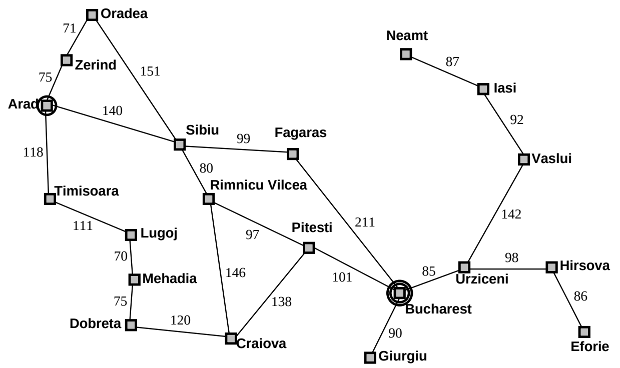
count blocks x that currently lie on y and must lie on $z \neq y$ in the goal (including case where y or z is the table)

How accurate is this heuristic?



Example: Route Planning in Romania

possible heuristic: straight-line distance to Bucharest



Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Example: Missionaries and Cannibals

Setting: Missionaries and Cannibals

- Six people must cross a river.
- Their rowing boat can carry one or two people across the river at a time (it is too small for three).
- Three people are missionaries, three are cannibals.
- Missionaries may never stay with a majority of cannibals.

possible heuristic: number of people on the wrong river bank

⇝ with our formulation of states as triples $\langle m, c, b\rangle$:
$h(\langle m, c, b\rangle) = m + c$

13.4 Summary

Summary

- heuristics estimate distance of a state to the goal
- can be used to focus search on promising states

⇝ soon: search algorithms that use heuristics