Foundations of Artificial Intelligence
16. State-Space Search: Greedy BFS, A*, Weighted A*

Malte Helmert

Universität Basel

April 1, 2016

Foundations of Artificial Intelligence
April 1, 2016 — 16. State-Space Search: Greedy BFS, A*, Weighted A*

16.1 Introduction

16.2 Greedy Best-first Search

16.3 A*

16.4 Weighted A*

16.5 Summary

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

16.1 Introduction

What Is It About?

In this chapter we study last chapter's algorithms in more detail:
- greedy best-first search
- A*
- weighted A*

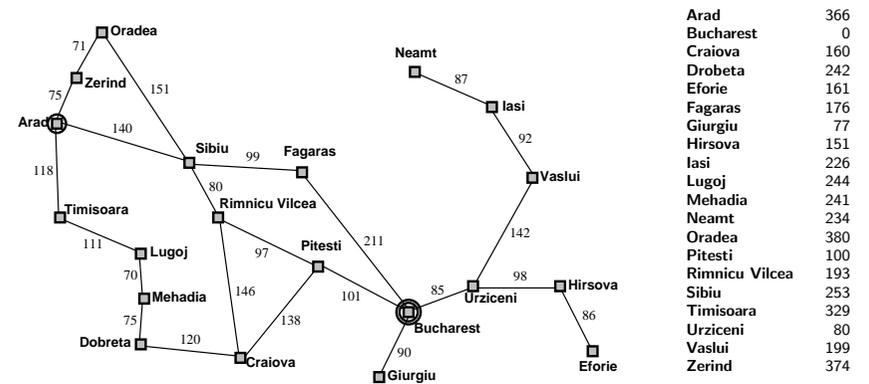# 16.2 Greedy Best-first Search
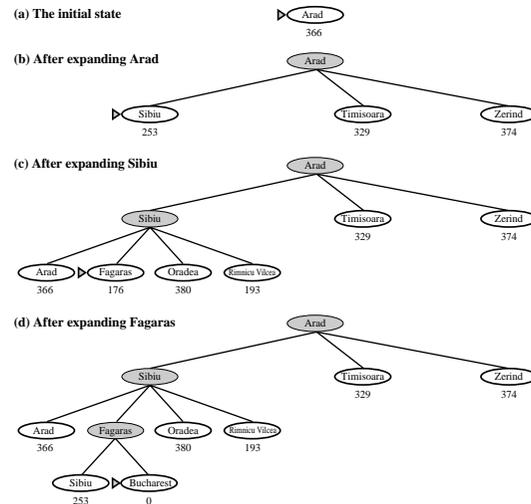

Greedy Best-first Search

Greedy Best-first Search

only consider the heuristic: $f(n) = h(n.\text{state})$

Note: usually without reopening (for reasons of efficiency)

Example: Greedy Best-first Search for Route Planning

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: Greedy Best-first Search for Route Planning

(a) The initial state

Arad
366

(b) After expanding Arad

Arad
Sibiu — Timisoara — Zerind
253   329   374

(c) After expanding Sibiu

Arad
Sibiu — Timisoara — Zerind
    329   374
Arad — Fagaras — Oradea — Rimnicu Vilcea
366   176   380   193

(d) After expanding Fagaras

Arad
Sibiu — Timisoara — Zerind
    329   374
Arad — Fagaras — Oradea — Rimnicu Vilcea
366   380   193
Sibiu — Bucharest
253   0

Greedy Best-first Search: Properties

- **complete** with **safe** heuristics
  (like all variants of best-first graph search)
- **suboptimal:** solutions can be **arbitrarily bad**
- often **very fast**: one of the fastest search algorithms in practice
- monotonic transformations of h (e.g. scaling, additive constants) do not affect behaviour (Why is this interesting?)

16.3 A*

A*

A*

combine greedy best-first search with uniform cost search:
$f(n) = g(n) + h(n.\text{state})$

- **trade-off** between path cost and proximity to goal
- $f(n)$ estimates overall cost of cheapest solution
  from initial state via n to the goal

A*: Citations

Google scholar — hart nilsson raphael — Search

My Citations

Scholar — Articles and patents — anytime — include citations — Create email alert

A formal basis for the heuristic determination of minimum cost paths
..., NJ Nilsson, B Raphael - Systems Science and ..., 1968 - ieeexplore.ieee.org
Abstract Although the problem of determining the minimum cost path through a graph arises
naturally in a number of interesting applications, there has been no underlying theory to
guide the development of efficient search procedures. Moreover, there is no adequate ...
Cited by 2591 - Related articles - All 13 versions

Correction to a formal basis for the heuristic determination of minimum cost paths
..., NJ Nilsson, B Raphael - ACM SIGART Bulletin, 1972 - dl.acm.org
Abstract Our paper on the use of heuristic information in graph searching defined a path-
finding algorithm, A*, and proved that it had two important properties. In the notation of the
paper, we proved that if the heuristic function ñ (n) is a lower bound on the true minimal ...
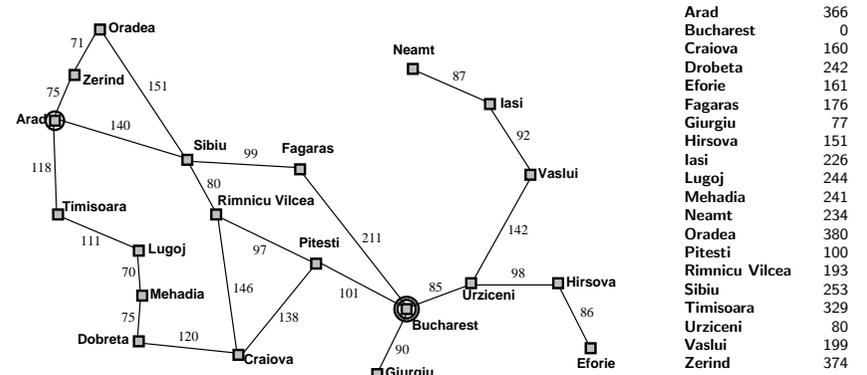Cited by 195 - Related articles - All 6 versions

[CITATION] Bi-directional search
I Pohl - 1970 - IBM TJ Watson Research Center
Cited by 337 - Related articles - Find in IDS Basel/Bern - All 2 versions

[CITATION] andB. Raphael. Aformalbasisforheuristicdetermination of minimum path cost
..., NJ Nilsson - IEEE Transaction on SSC, 1968
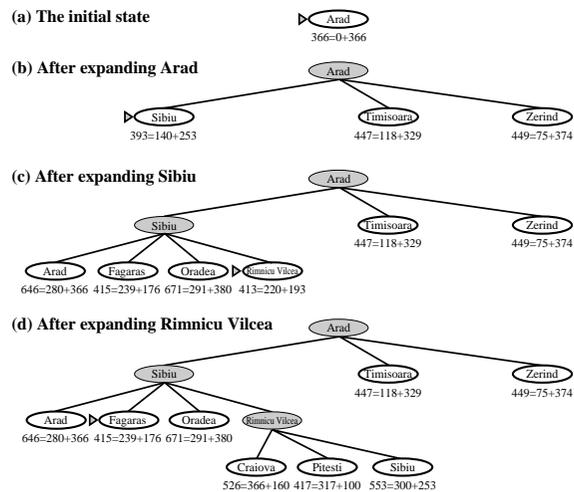Cited by 28 - Related articles

[CITATION] Research and applications—artificial intelligence
B Raphael, R Duda, RE Fikes, PE Hart, N Nilsson... - Final Report, SRI Project, 1971
Cited by 13 - Related articles

A mobile automaton: An application of artificial intelligence techniques
NJ Nilsson - 1969 - DTIC Document
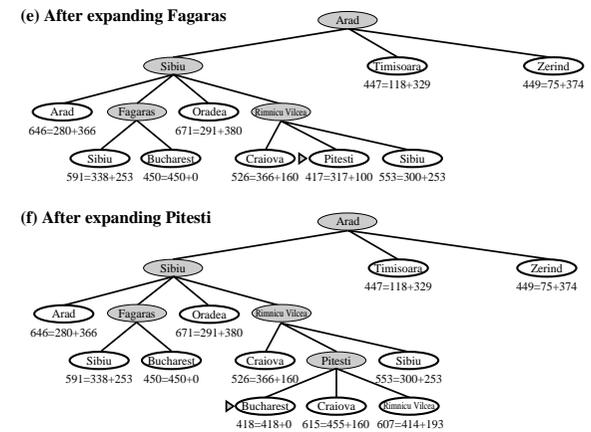A MOBILE AUTOMTON: AN APPLICATION OF ARTIFICIAL INTELLIGENCE TECHNIQUES Nils...

Example: A* for Route Planning

City	Value
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: A* for Route Planning

(a) The initial state
Arad 366=0+366

(b) After expanding Arad
Arad
Sibiu 393=140+253   Timisoara 447=118+329   Zerind 449=75+374

(c) After expanding Sibiu
Arad
Sibiu   Timisoara 447=118+329   Zerind 449=75+374
Arad 646=280+366   Fagaras 415=239+176   Oradea 671=291+380   Rimnicu Vilcea 413=220+193

(d) After expanding Rimnicu Vilcea
Arad
Sibiu   Timisoara 447=118+329   Zerind 449=75+374
Arad 646=280+366   Fagaras 415=239+176   Oradea 671=291+380   Rimnicu Vilcea
Craiova 526=366+160   Pitesti 417=317+100   Sibiu 553=300+253

Example: A* for Route Planning

(e) After expanding Fagaras
Arad
Sibiu   Timisoara 447=118+329   Zerind 449=75+374
Arad 646=280+366   Fagaras   Oradea 671=291+380   Rimnicu Vilcea
Sibiu 591=338+253   Bucharest 450=450+0   Craiova 526=366+160   Pitesti 417=317+100   Sibiu 553=300+253

(f) After expanding Pitesti
Arad
Sibiu   Timisoara 447=118+329   Zerind 449=75+374
Arad 646=280+366   Fagaras   Oradea 671=291+380   Rimnicu Vilcea
Sibiu 591=338+253   Bucharest 450=450+0   Craiova 526=366+160   Pitesti   Sibiu 553=300+253
Bucharest 418=418+0   Craiova 615=455+160   Rimnicu Vilcea 607=414+193

A*: Properties

- **complete** with **safe** heuristics
  (like all variants of best-first graph search)
- **with reopening: optimal** with **admissible** heuristics
- **without reopening: optimal** with heuristics
  that are **admissible** and **consistent**

⤳ proofs: Chapters 18 and 19

A*: Implementation Aspects

some practical remarks on implementing A*:

- **common bug:** reopening not implemented
  although heuristic is not consistent
- **common bug:** duplicate test "too early"
  (upon generation of search nodes)
- **common bug:** goal test "too early"
  (upon generation of search nodes)
- all these bugs lead to loss of optimality
  and can remain undetected for a long time

16.4 Weighted A*

Weighted A*

Weighted A*

A* with more heavily weighted heuristic:
$f(n) = g(n) + w \cdot h(n.\text{state})$,
where weight $w \in \mathbb{R}_0^+$ with $w \geq 1$ is a freely choosable parameter

Note: $w < 1$ is conceivable, but usually not a good idea
(Why not?)

Weighted A*: Properties

weight parameter controls "greediness" of search:

- $w = 0$: like uniform cost search
- $w = 1$: like A*
- $w \to \infty$: like greedy best-first search

with $w \geq 1$ properties analogous to A*:

- h admissible:
  found solution guaranteed to be at most w times
  as expensive as optimum when reopening is used
- h admissible and consistent:
  found solution guaranteed to be at most w times
  as expensive as optimum; no reopening needed

(without proof)

16.5 Summary

Summary

best-first graph search with evaluation function f:

- $f = h$: greedy best-first search
  suboptimal, often very fast
- $f = g + h$: A*
  optimal if h admissible and consistent
  or if h admissible and reopening is used
- $f = g + w \cdot h$: weighted A*
  for $w \geq 1$ suboptimality factor at most w
  under same conditions as for optimality of A*