

Foundations of Artificial Intelligence

B12. State-Space Search: Greedy Best-first Search, A^* , Weighted A^*

Malte Helmert

University of Basel

March 23, 2026

Foundations of Artificial Intelligence

March 23, 2026 — B12. State-Space Search: Greedy Best-first Search, A*, Weighted A*

B12.1 Introduction

B12.2 Greedy Best-first Search

B12.3 A*

B12.4 Weighted A*

B12.5 Summary

State-Space Search: Overview

Chapter overview: state-space search

- ▶ B1–B3. Foundations
- ▶ B4–B8. Basic Algorithms
- ▶ B9–B15. Heuristic Algorithms
 - ▶ B9. Heuristics
 - ▶ B10. Analysis of Heuristics
 - ▶ B11. Best-first Graph Search
 - ▶ B12. Greedy Best-first Search, A^* , Weighted A^*
 - ▶ B13. IDA*
 - ▶ B14. Properties of A^* , Part I
 - ▶ B15. Properties of A^* , Part II

B12.1 Introduction

What Is It About?

In this chapter we study last chapter's algorithms in more detail:

- ▶ greedy best-first search
- ▶ A^*
- ▶ weighted A^*

B12.2 Greedy Best-first Search

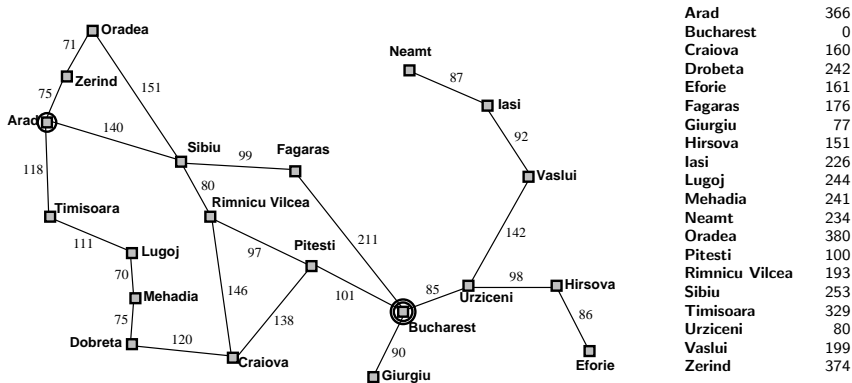
Greedy Best-first Search

Greedy Best-first Search

only consider the heuristic: $f(n) = h(n.state)$

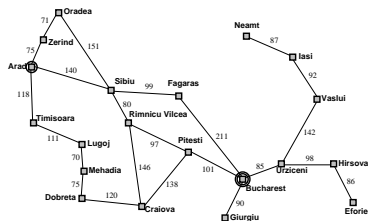
Note: usually *without reopening* (for reasons of efficiency)

Example: Greedy Best-first Search for Route Planning



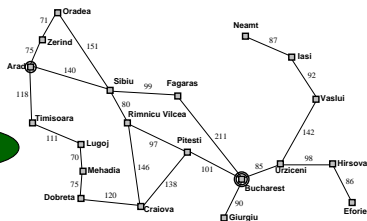
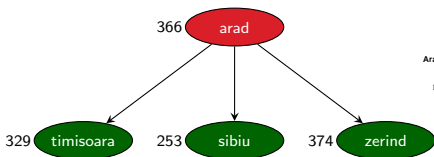
Example: Greedy Best-first Search for Route Planning

366


 arad


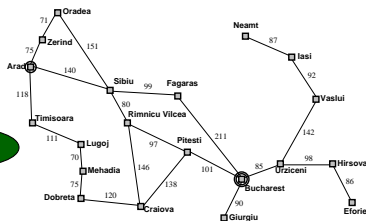
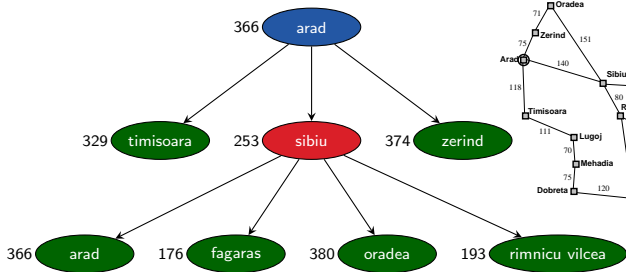
Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

Example: Greedy Best-first Search for Route Planning



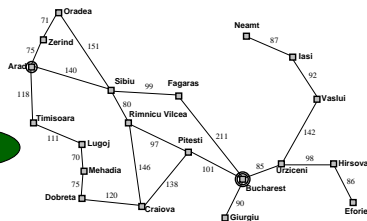
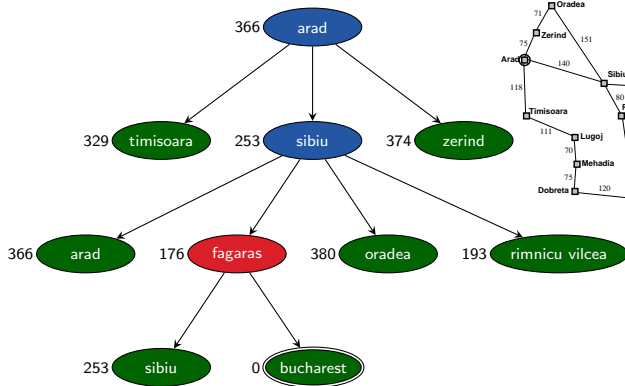
Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

Example: Greedy Best-first Search for Route Planning



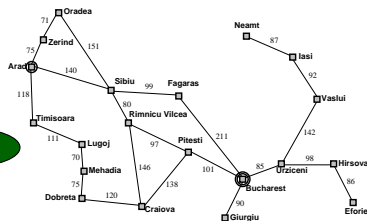
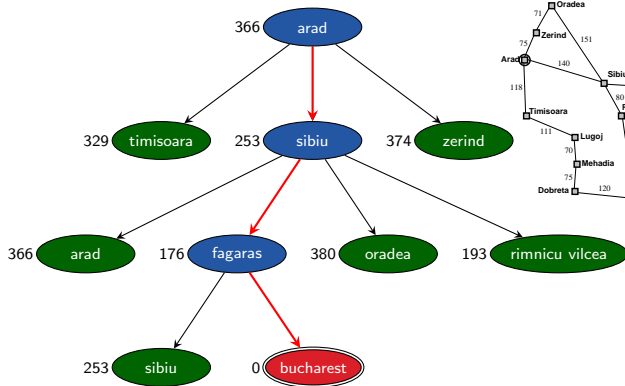
Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

Example: Greedy Best-first Search for Route Planning



Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

Example: Greedy Best-first Search for Route Planning



Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

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?**)

B12.3 A^*

A*

A*

combine greedy best-first search with uniform cost search:

$$f(n) = g(n) + h(n.state)$$

- ▶ **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



About 17.800 results (0,07 sec)

A formal basis for the heuristic determination of minimum cost paths

[PE Hart](#), [NJ Nilsson](#), [B Raphael](#) - IEEE transactions on Systems ..., 1968 - [ieeexplore.ieee.org](#)

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 ...

☆ Save Cite Cited by **19275** Related articles All 7 versions

Correction to" a formal basis for the heuristic determination of minimum cost paths"

[PE Hart](#), [NJ Nilsson](#), [B Raphael](#) - ACM SIGART Bulletin, 1972 - [dl.acm.org](#)

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 ...

☆ Save Cite Cited by 592 Related articles All 11 versions

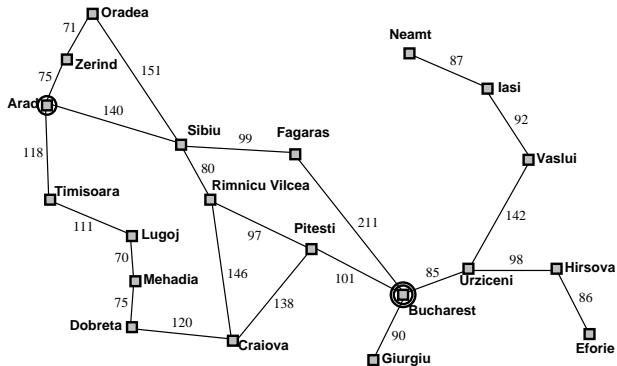
A method for computing heuristics in problem solving

[G Guida](#), [M Somalvico](#) - Information Sciences, 1979 - Elsevier

... more closely the **Hart-N&son-Raphael** algorithm [6], which ... $f_i(n)$ of the **Hart-NilssonRaphael** algorithm. The central idea is to ... **Hart-Nilsson-Raphael** algorithm that we are going to present. ...

☆ Save Cite Cited by 50 Related articles All 4 versions

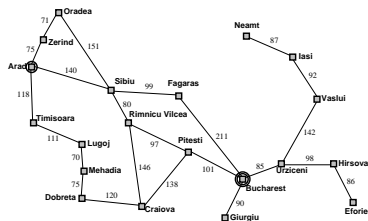
Example: A* for Route Planning



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

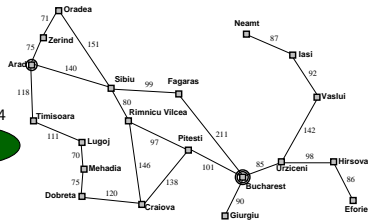
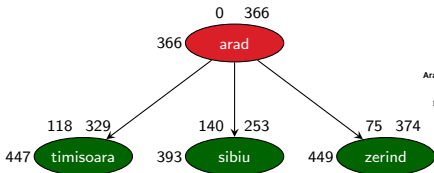
Example A* for Route Planning

0 366
366 **arad**



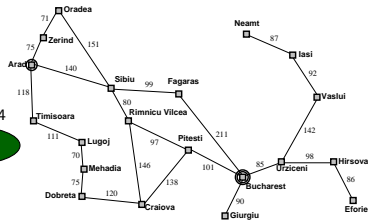
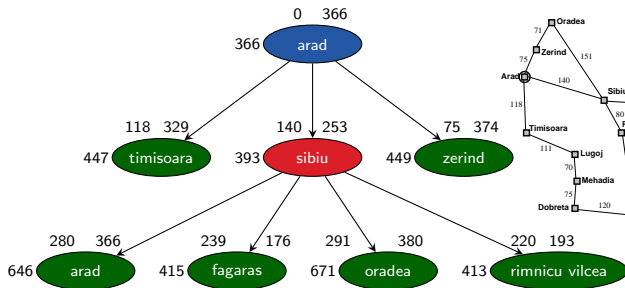
Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

Example A* for Route Planning



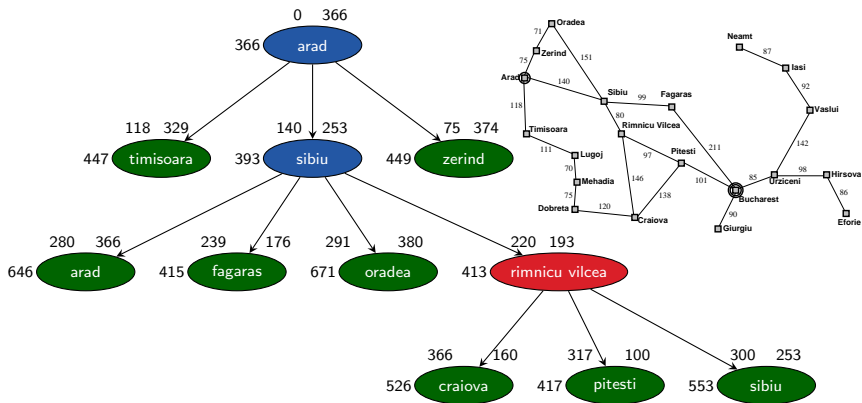
Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

Example A* for Route Planning



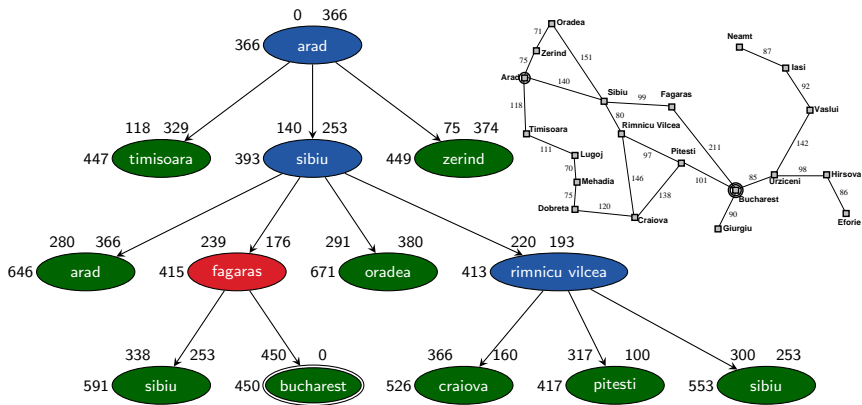
Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

Example A* for Route Planning



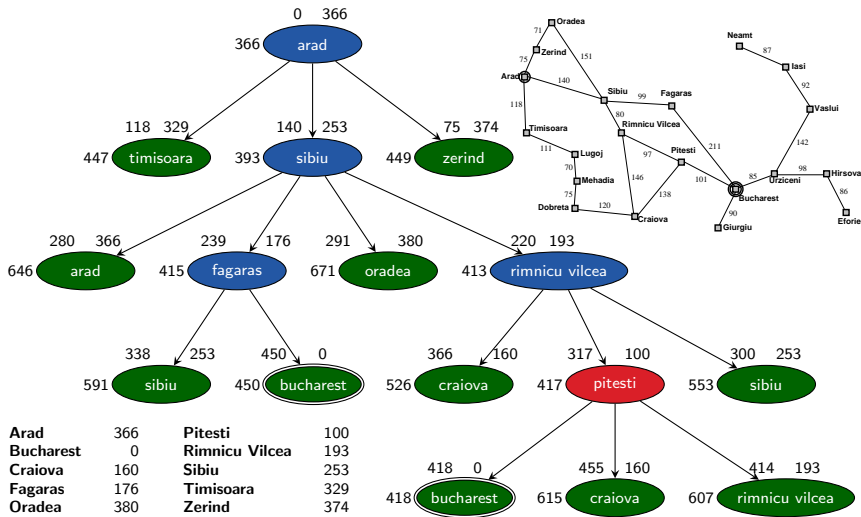
Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

Example A* for Route Planning

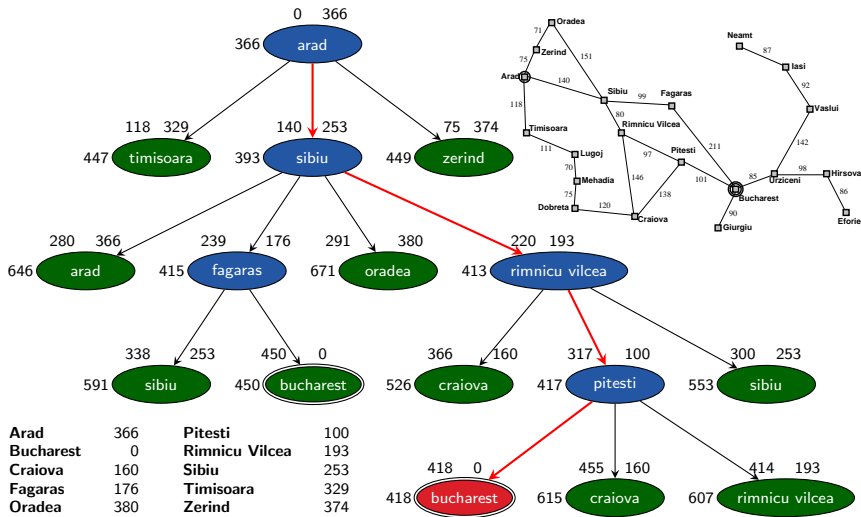


Arad	366	Pitesti	100
Bucharest	0	Rimnicu Vilcea	193
Craiova	160	Sibiu	253
Fagaras	176	Timisoara	329
Oradea	380	Zerind	374

Example A* for Route Planning



Example A* for Route Planning



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 B14 and B15

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

B12.4 Weighted A^*

Weighted A*

Weighted A*

A* with more heavily weighted heuristic:

$$f(n) = g(n) + w \cdot h(n.state),$$

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*
- ▶ $w \rightarrow \infty$: 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)

B12.5 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*