

# Advanced School in Artificial Intelligence

## Strategie di ricerca

Strategie informate

Marco Alberti

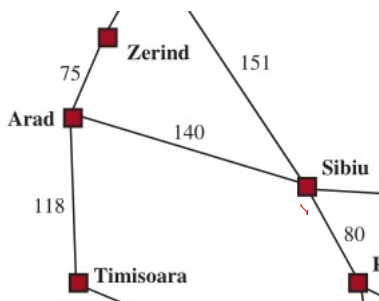
[marco.alberti@unife.it](mailto:marco.alberti@unife.it)

*Progetto di alta formazione in ambito tecnologico economico e culturale per una regione della conoscenza europea e attrattiva approvato e cofinanziato dalla Regione Emilia-Romagna con deliberazione di Giunta regionale n. 1625/2021*



**Università  
degli Studi  
di Ferrara**

## Strategie informate




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

Figure 3.16 Values of  $h_{SLD}$ —straight-line distances to Bucharest.

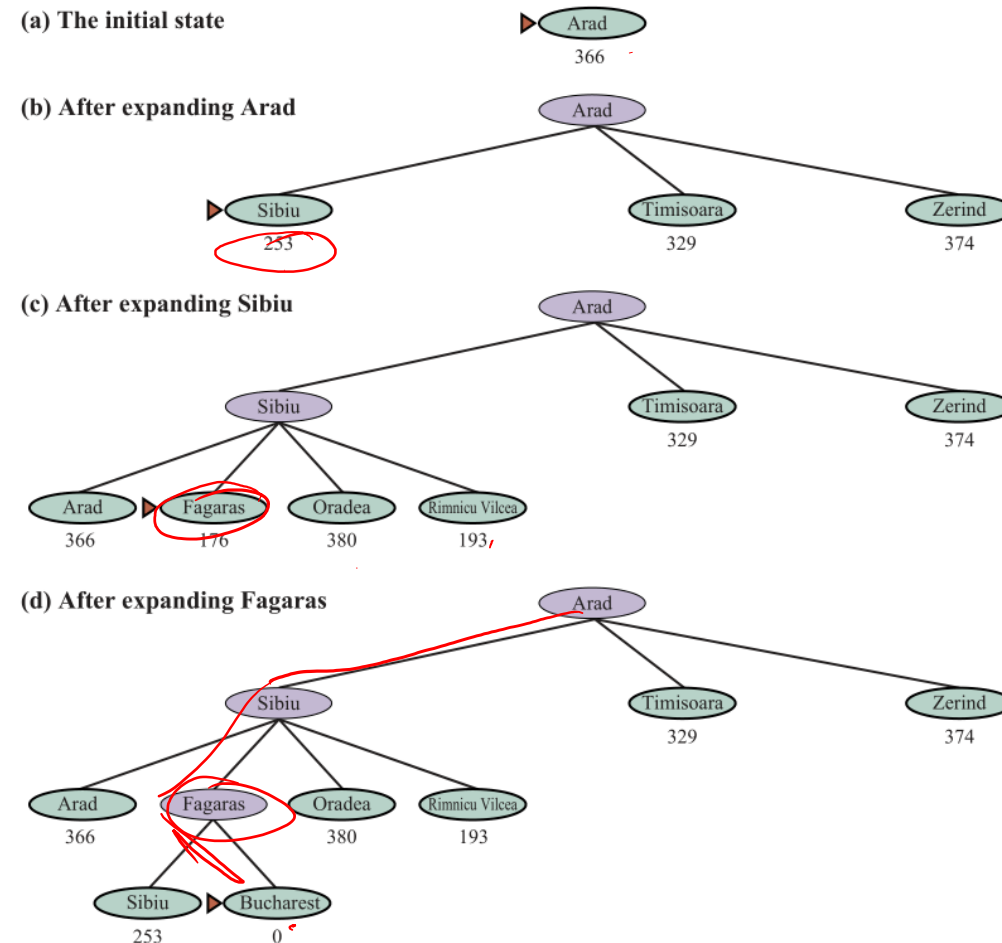
- Distanze da Bucarest in linea d'aria delle città raggiungibili da quella iniziale (Arad): Sibiu 253, Timisoara 329, Zerind 374
- Forse è meglio visitare prima il nodo di Sibiu (anche se non c'è garanzia che a minor distanza in linea d'aria corrisponda minor distanza stradale)
- Le strategie viste finora, nello scegliere un nodo, non considerano una stima del costo per raggiungere la soluzione dallo stato del nodo: per questo sono dette non informate
- Le strategie **informate** selezionano il nodo  $n$  da espandere considerando una stima (euristica)  $h(n)$  di tale costo



## Greedy best first

- I nodi dal costo stimato minore sono presumibilmente i più vicini alla soluzione
- Inserisce i nodi nella frontiera in ordine crescente di  $h(n)$  
- $greedy\_best\_first(problem, h) = best\_first\_search(problem, h)$
- In Romania seleziona il percorso, Arad-Sibiu-Fagaras-Bucarest: a ogni passo la città più vicina a Bucarest in linea d'aria fra quelle corrispondenti a nodi aperti
- Tuttavia non trova necessariamente il percorso di costo minimo (come in questo caso), in quanto non considera il costo del percorso dal nodo iniziale a quello da espandere

## Esempio greedy best first search



**Figure 3.17** Stages in a greedy best-first tree-like search for Bucharest with the straight-line distance heuristic  $h_{SLD}$ . Nodes are labeled with their  $h$ -values.

## A\*

- Per ovviare all'inconveniente del greedy best first search si possono ordinare i nodi in base alla somma dell'euristica  $h(n)$  e del costo  $g(n)$  del percorso fra il nodo iniziale e  $n$ .
- Questo è l'algoritmo A\*, corrispondente al best first search con

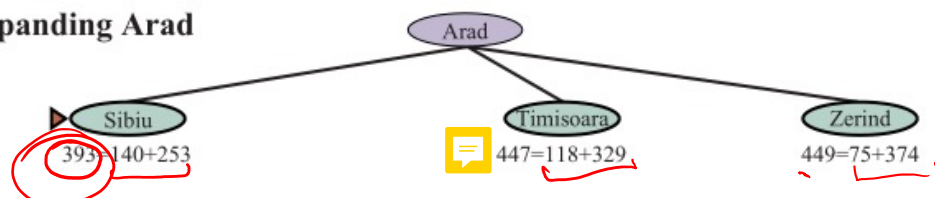
$$f(n) = \underbrace{g(n)} + \underbrace{h(n)}$$

# Advanced School in Artificial Intelligence

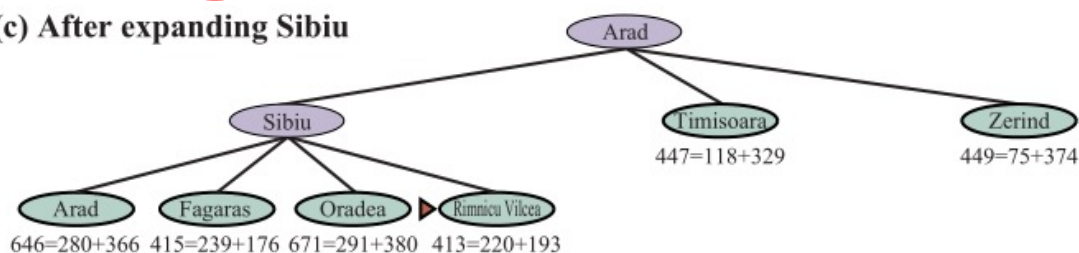
(a) The initial state



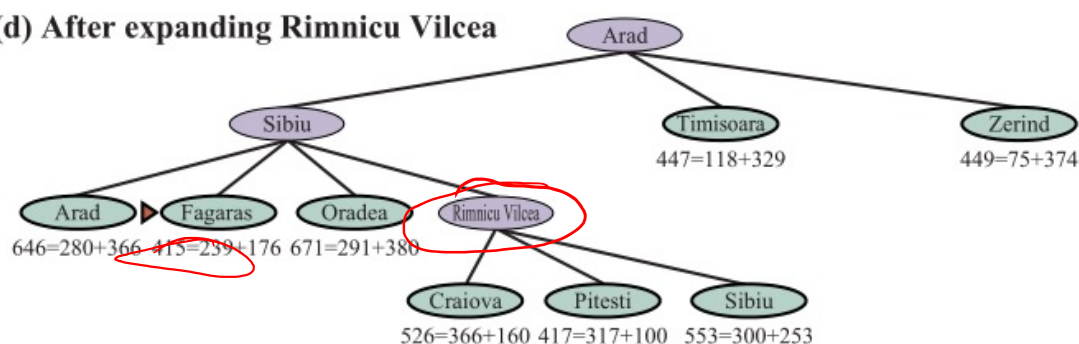
(b) After expanding Arad



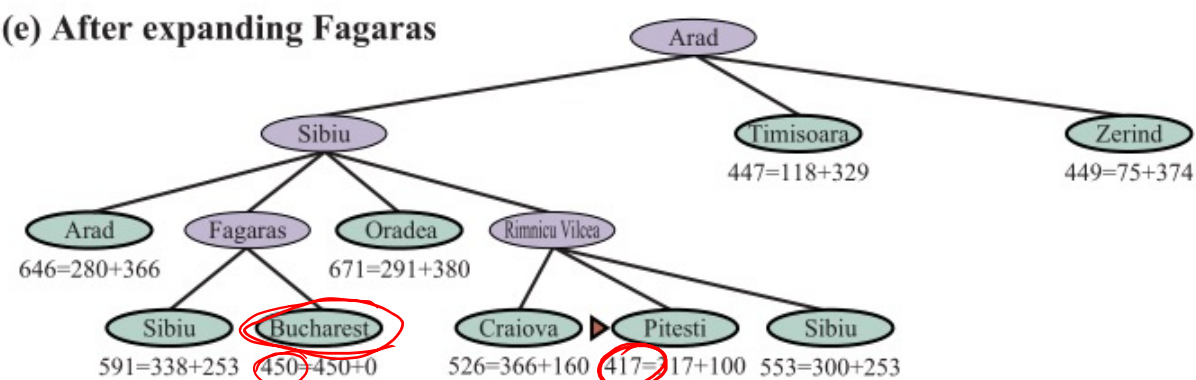
(c) After expanding Sibiu



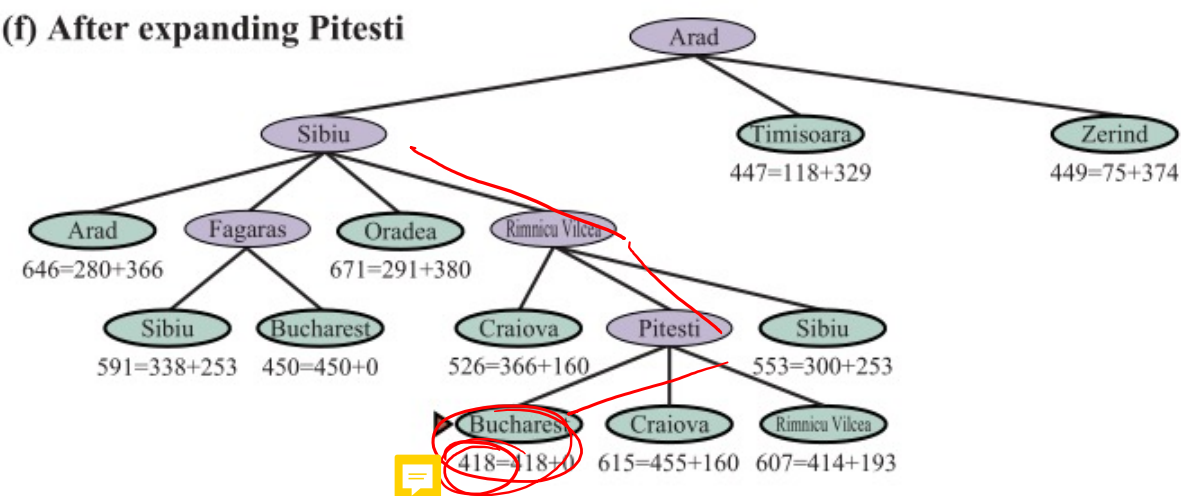
(d) After expanding Rimnicu Vilcea



(e) After expanding Fagaras

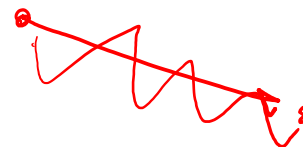


(f) After expanding Pitesti



## Proprietà di A\*

- Completezza: sì (se il costo di ogni transizione è  $\geq \epsilon > 0$ )
- Ottimalità: dipende dall'euristica.  $h(n)$
- Un'euristica è **ammissibile** se non sovrastima il costo:  $h(n) \leq h^*(n)$ , dove  $h^*(n)$  è il costo del cammino di costo minimo da  $n$  alla soluzione
- L'algoritmo A\* con euristica ammissibile è ottimale: l'ammissibilità garantisce che le soluzioni migliori saranno considerate.



## Difficoltà nella soluzione di problemi con ricerca



- Qual è la formulazione migliore?
  - rappresentazione di stati e azioni
  - giusto livello di astrazione
- Incertezza
  - Un'azione può portare a uno di più stati (non determinismo)
  - Stato iniziale non noto
  - Modelli probabilistici

