Informed Search Methods

CZ3005: Artificial Intelligence

Shen Zhiqi

NANYANG TECHNOLOGICAL UNIVERSITY

Outline

- Greedy search
- A * search
- Heuristic functions

General Search

Uninformed search strategies

- Systematic generation of new states($\longrightarrow$ Goal Test)
- Inefficient (exponential space and time complexity)

Informed search strategies

- Use problem-specific knowledge
  - To decide the order of node expansion
- Best First Search: expand the most desirable unexpanded node
  - Use an evaluation function to estimate the “desirability" of each node

Evaluation function

- Path-cost function $g(n)$
  - Cost from initial state to current state (search-node) n
  - No information on the cost toward the goal
- Need to estimate cost to the closest goal

- "Heuristic" function $h(n)$
  - Estimated cost of the cheapest path from n to a goal state $h(n)$
    - Exact cost cannot be determined
  - depends only on the state at that node
  - $h(n)$ is not larger than the real cost (admissible)

Greedy Search

Expands the node that appears to be closest to goal

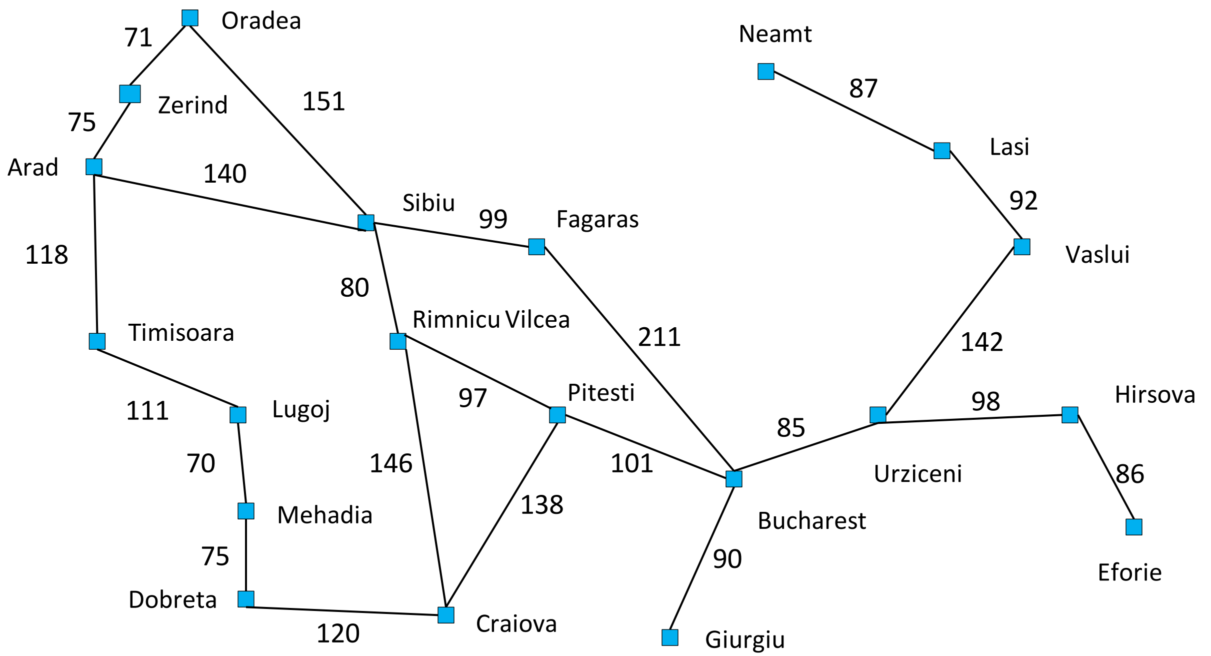
- Evaluation function $h(n)$:estimate of cost from n to $goal$
- Function Greedy-Search(problem) returns solution
  - Return Best-First-Search(problem, h) // $h(goal) = 0$

Question: How to estimation the cost from n to goal?

Answer: Recall that we want to use problem-specific knowledge

Example: Route-finding from Arad to Bucharest

$h(n)$ = straight-line distance from n to Bucharest

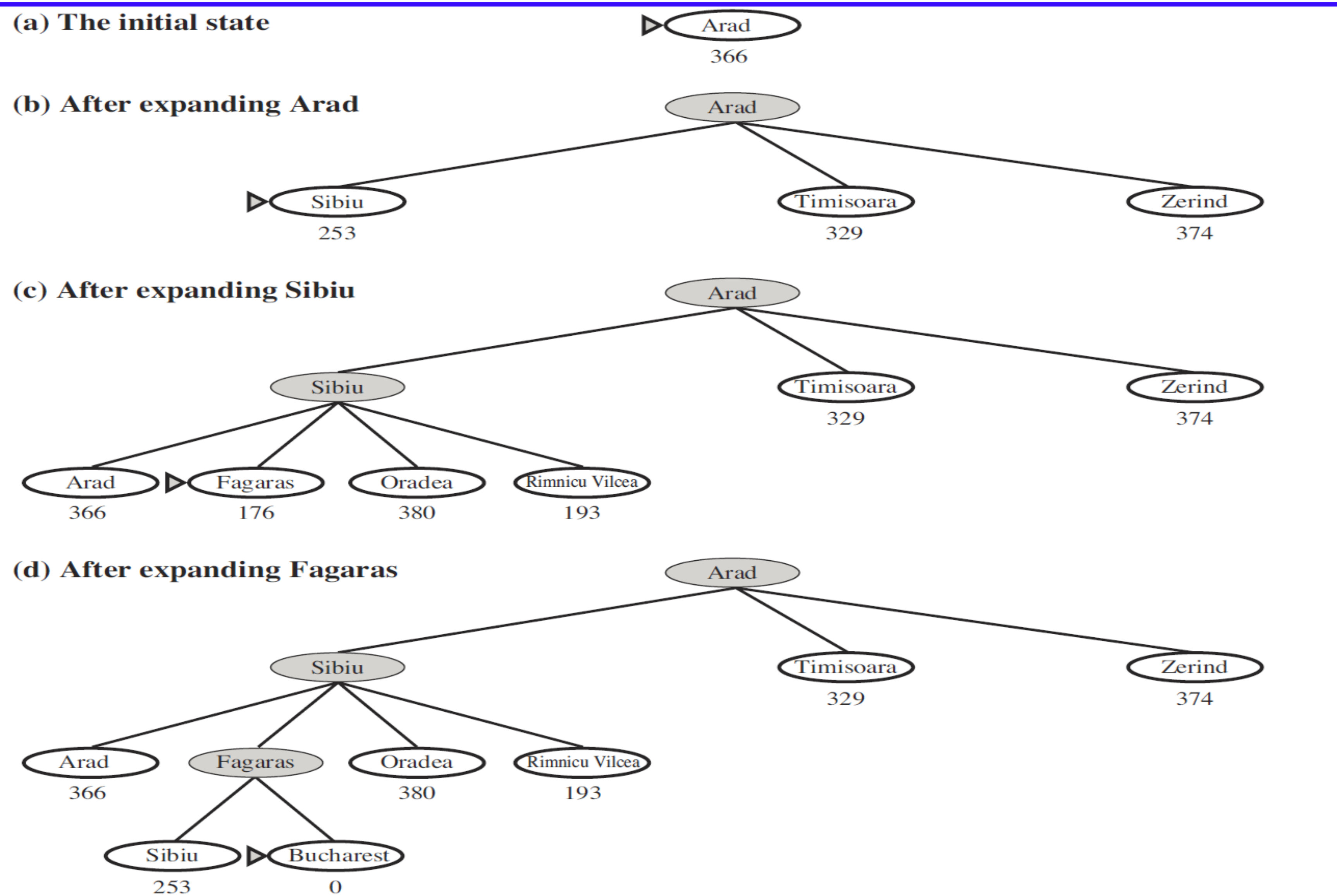


Straight-line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Efoire	161
Fagaras	176
Giurgiu	77
Hirsova	151
Lasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

- useful but potentially fallible (heuristic)
- heuristic functions are problem-specific

Example...

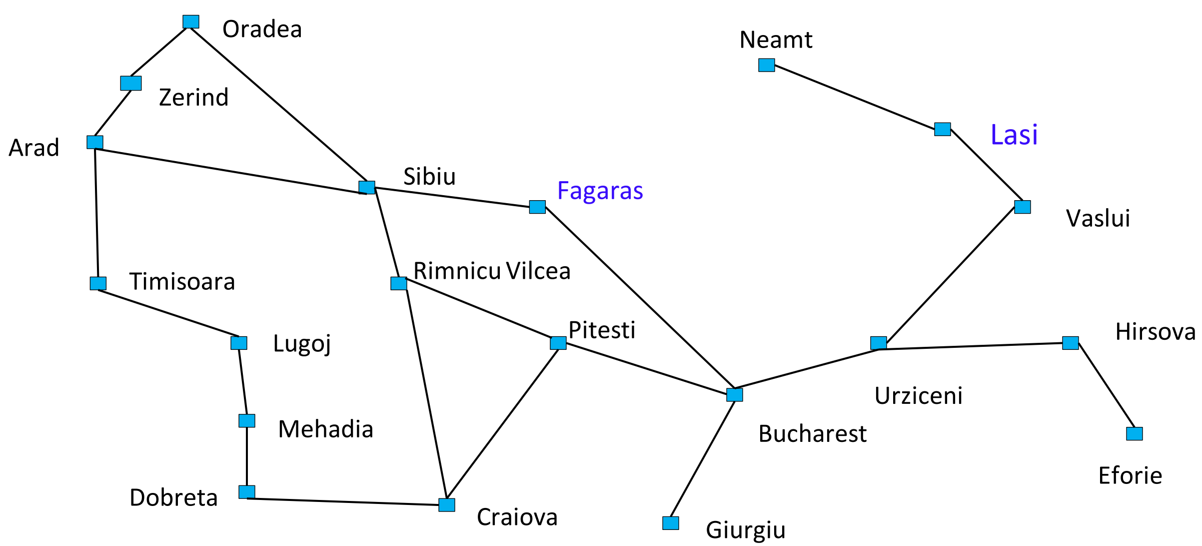


(a) The initial state
Arad
366
(b) After expanding Arad
Arad
Sibiu
253
Timisoara
329
Zerind
374
(c) After expanding Sibiu
Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
176
Oradea
380
Rimnicu Vilcea
193
(d) After expanding Fagaras
Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
Oradea
380
Rimnicu Vilcea
193
Sibiu
253
Bucharest
0

Complete?

Question: Is this approach complete?

Example: Find a path from Lasi to Fagaras



Answer: No

Greedy Search...

- m: maximum depth of the search space
- Time: $O(b^m)$
- Space: $O(b^m)$(keeps all nodes in memory)
- Optimal: No
- Complete: No

A * Search

- Uniform-cost search
  - *g(n):* cost to reach n (Past Experience)
  - optimal and complete, but can be very inefficient
- Greedy search
  - *h(n):* cost from n to goal (Future Prediction)
  - neither optimal nor complete, but cuts search space considerably

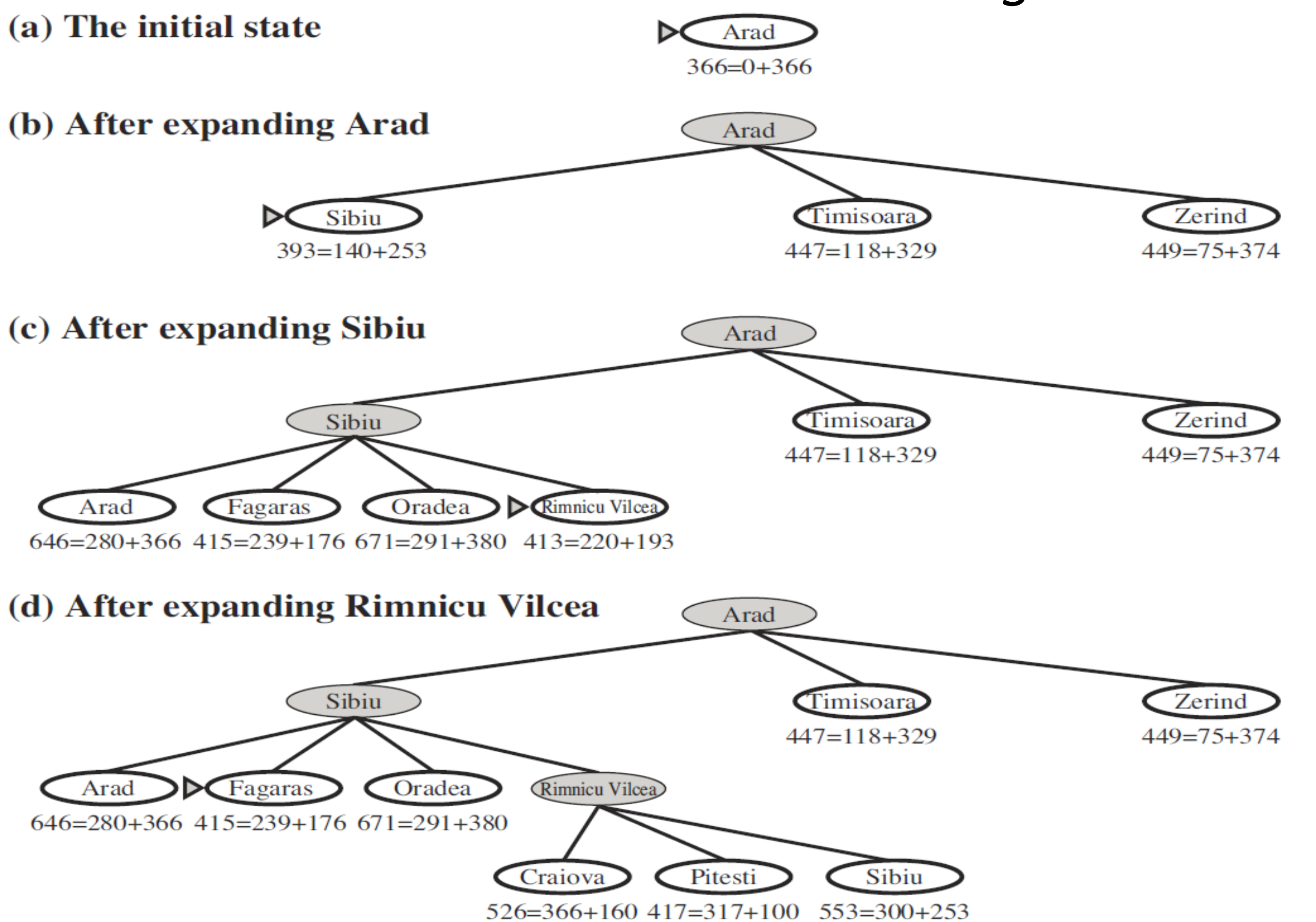
Idea: combine greedy search with uniform-cost search

Evaluation function: $f(n) = g(n) + h(n)$

- *f (n):* estimated total cost of path through *n* to goal (Whole Life)
- If g = 0 $\longrightarrow$ greedy search; If h = 0 $\longrightarrow$ uniform-cost search
- Function A* Search(problem) returns solution
  - Return Best-First-Search(problem, $g + h$)

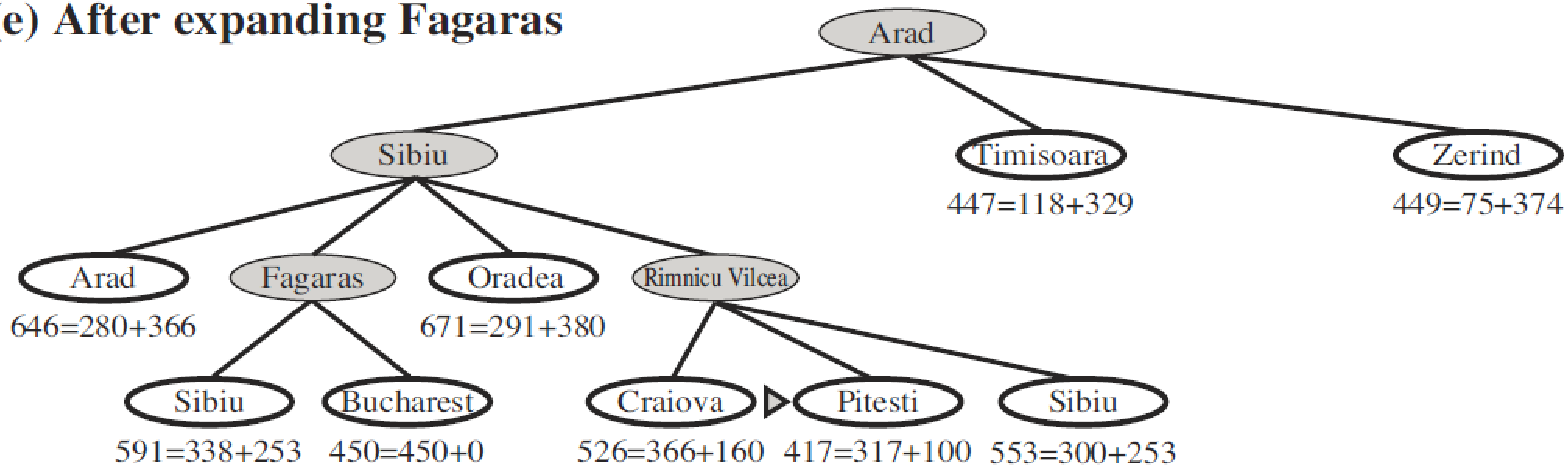
Example: Route-finding from Arad to Bucharest

Best-first-search with evaluation function *g + h*

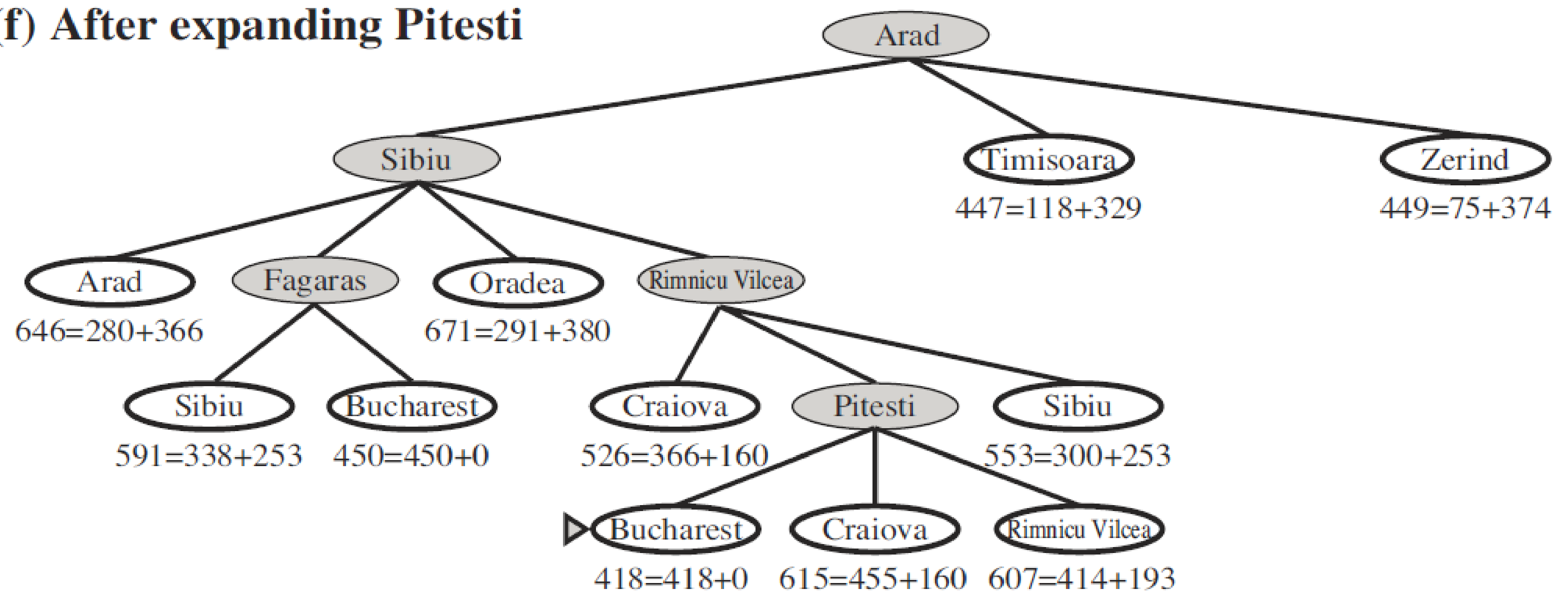


Example: Route-finding from Arad to Bucharest

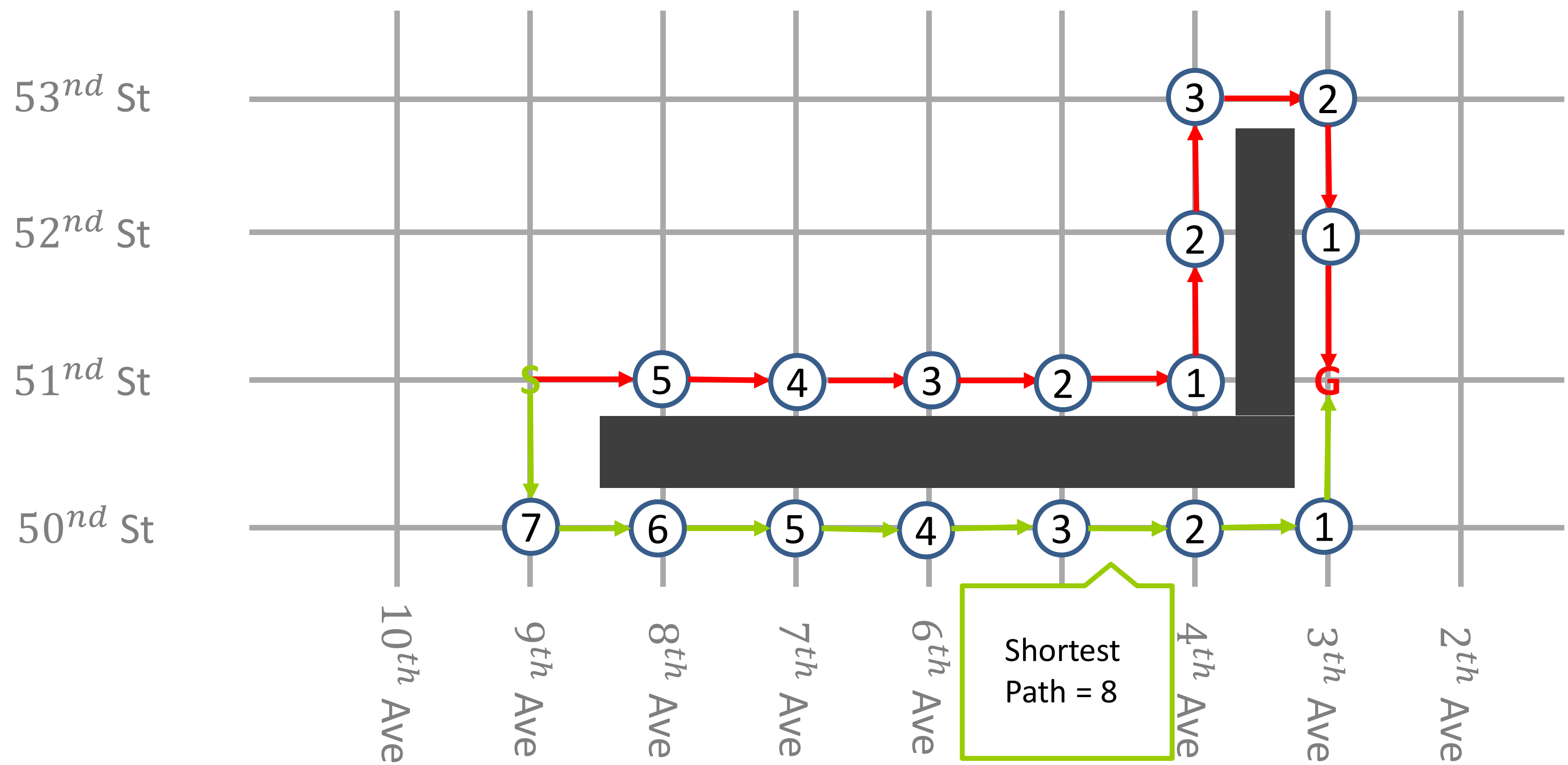
(e) After expanding Fagaras



(f) After expanding Pitesti

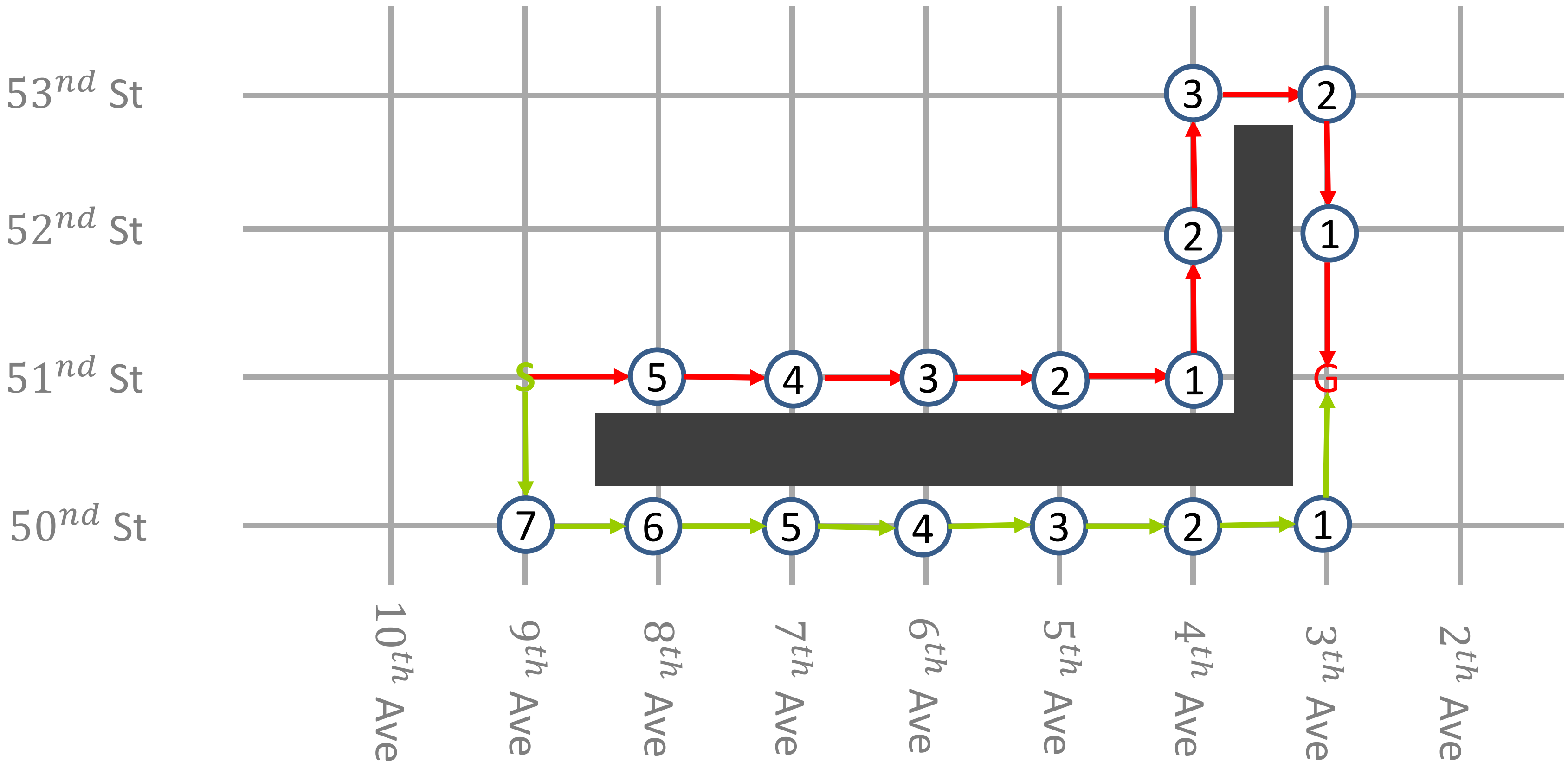


Example: Route-finding in Manhattan

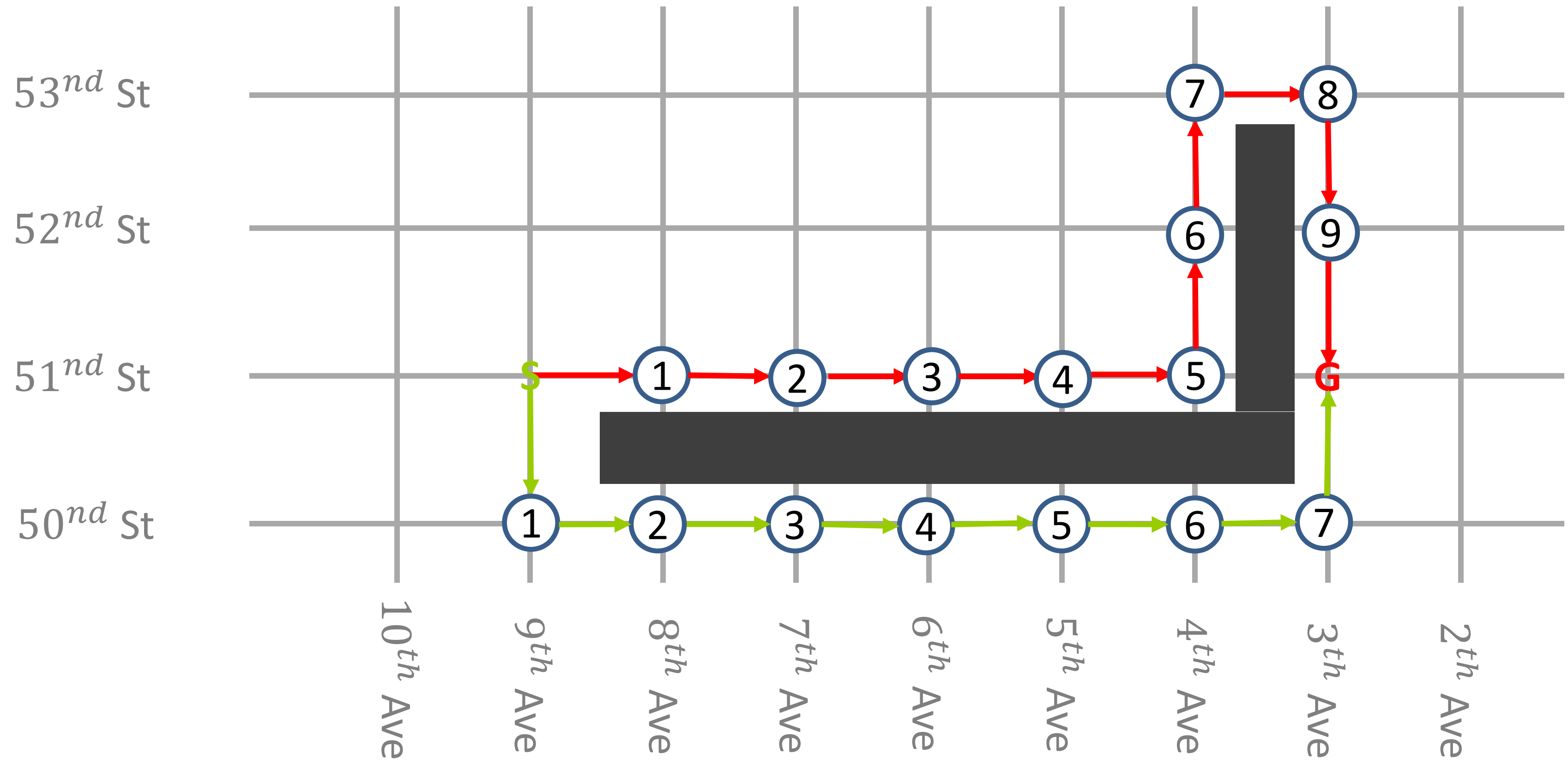


Manhattan
Distance Heuristic

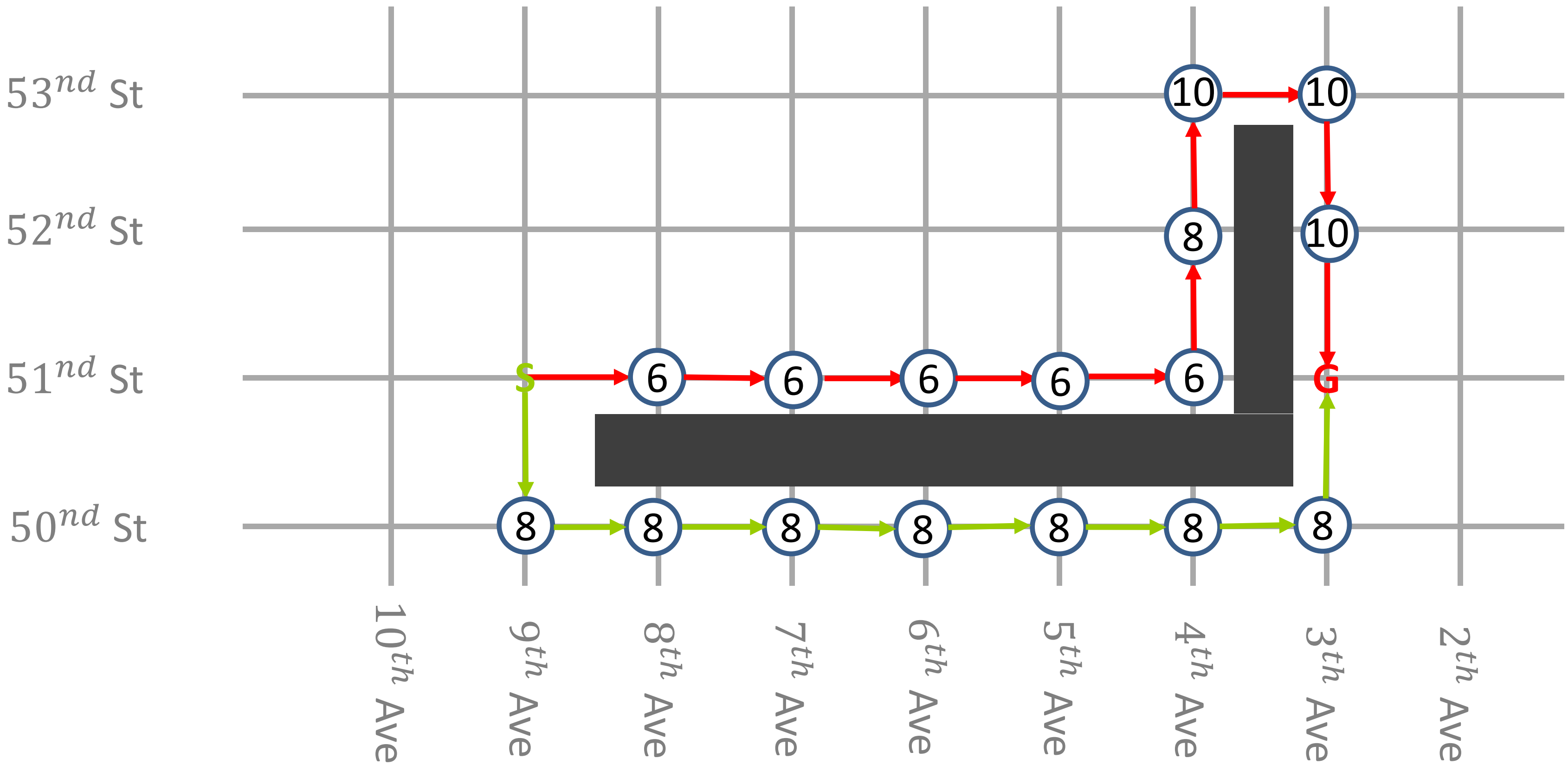
Example: Route-finding in Manhattan (Greedy)



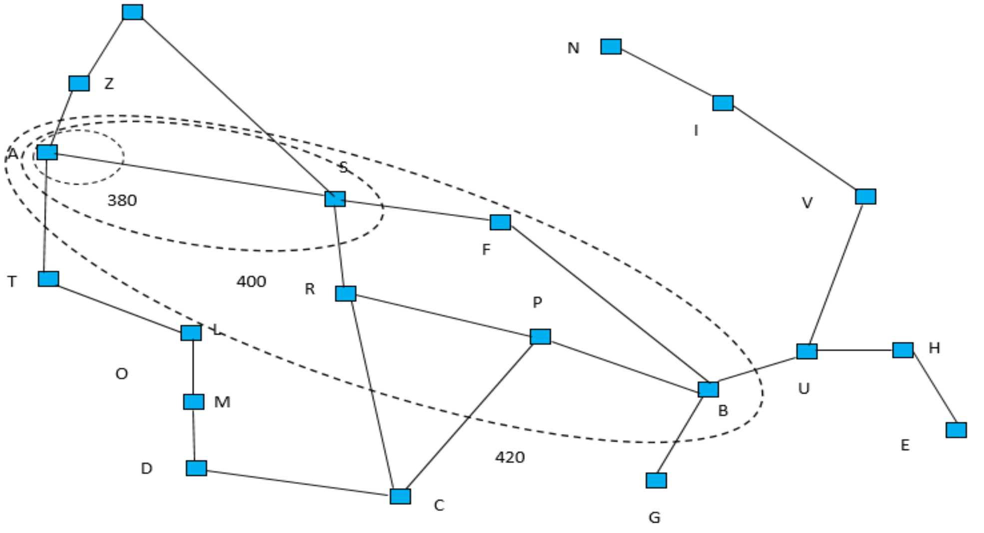
Example: Route-finding in Manhattan (UCS)



Example: Route-finding in Manhattan (A*)



Complexity of A*



Time

- exponential in length of solution

Space: (all generated nodes are kept in memory)

- exponential in length of solution

With a good heuristic, significant savings are still possible compared to uninformed search methods

Different h for 8-Puzzle

5	4	
6	1	8
7	3	2

Start state

1	2	3
8		4
7	6	5

Goal state

$h1(S)$= number of misplaced tiles

- $h1(S)$= *7*

$h2(n)$= sum of the horizontal and vertical distances (Manhattan distance)

- $h2(S)$ = 2+3+3+2+4+2+0+2 = 18

Dominance

Definition: h_2 dominates h_1 if both h_1, h_2 are admissible and $h_2(\mathrm{n}) \geq h_1(\mathrm{n})$ for all n

Example

Suppose there are A, B, C nodes:

$h_1(\mathrm{A}) = 2, h_1(\mathrm{B}) = 3, h_1(\mathrm{C}) = 1$

$h_2(\mathrm{A}) = 6, h_2(\mathrm{B}) = 3, h_2(\mathrm{C}) = 4$

h_2 dominates h_1

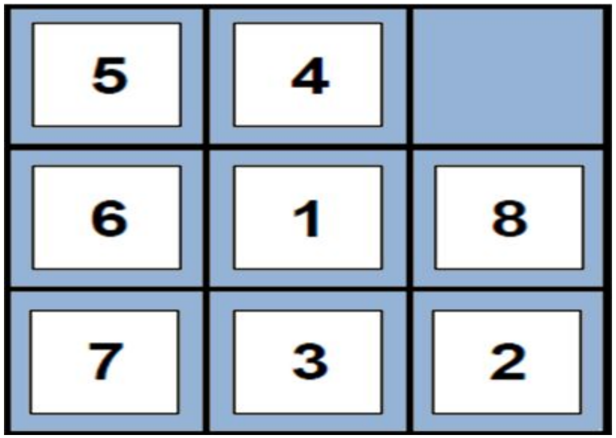
Dominance

Question: Which one is better? h_1 or h_2 ?

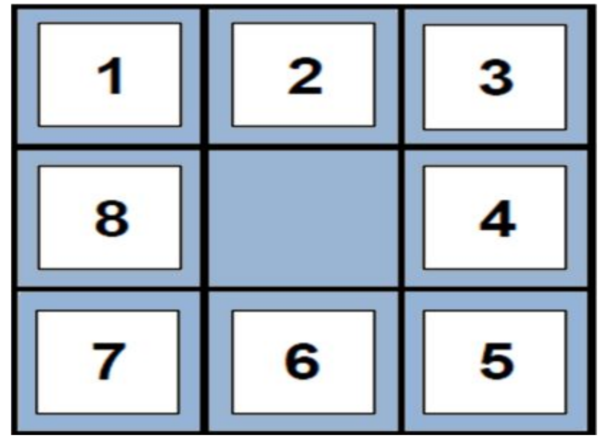
Answer:

- every node with $f(n) = h(n) + g(n) < f^*$ will be expanded,
- where f^* is the optimal cost of the search problem every node with $h(n) < f^* - g(n)$ (Max Improvement) will be expanded
- $h2(n) \geq h1(n)$
- Which one is better? $h2$!

Example: 8-Puzzle



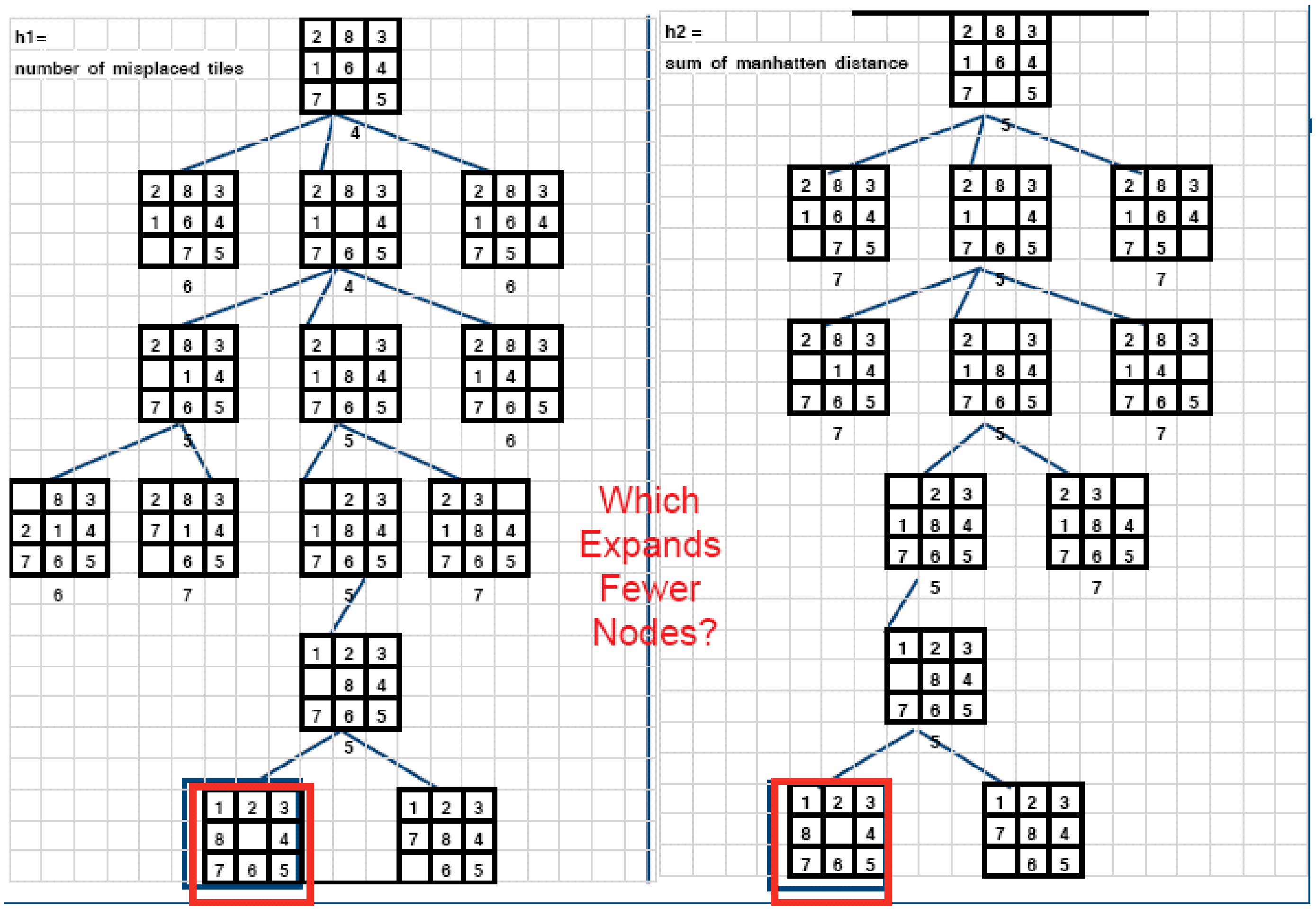
Start state



Goal state

- $h1(n)$ = number of misplaced tiles
- $h2(n)$= sum of the horizontal and vertical distances
- $h2$ is better

Example: 8-Puzzle...



Example: 8-Puzzle...

d	Search Cost		
	IDS	A* h_1	A* h_2
2	10	6	6
4	112	13	12
6	680	20	18
8	6384	39	25
10	47127	93	39
12	364404	227	73
14	3473941	539	113
16	-	1301	211
18	-	3056	363
20	-	7276	676
22	-	18094	1219
24	-	39135	1641

Inventing Heuristic Functions

The original 8-Puzzle is complicated

- How to design a heuristic?
- Simplify the rules of the original 8-Puzzle: a tile can move anywhere

Example ($h_1(n)$ = number of misplaced tiles)

$h_1(n)$ gives the shortest solution to this relaxed problem

Inventing Heuristic Functions…

- The original 8-Puzzle is complicated
- Simplify the rules of the original 8-Puzzle:
  - A tile can move to any adjacent square

Example ($\boldsymbol{h_1(n)}$ = sum of the horizontal and vertical distances)

$\boldsymbol{h_1(n)}$ gives the shortest solution to this relaxed problem

The cost of an exact solution to a relaxed version is a good heuristic for the original problem

Inventing Heuristic Functions…

Question: A set of heuristics h_1,…, h_m, none of them dominates. How to choose?

Answer: $h(n) = \max\big(h1(n), \ldots, hm(n)\big)$ dominates all the individual heuristics

Example

Suppose there are A, B, C nodes:
$h_1(A) = 2, h_1(B) = 3, h_1(C) = 1$
$h_2(A) = 6, h_2(B) = 2, h_2(C) = 4$
$h_3(A) = 1, h_3(B) = 4, h_3(C) = 2$
$h(A) = 6, h(B) = 4, h(C) = 4$
h dominates h_1, h_2 and h_3.