Informed search algorithms

Outline

Informed Search: use domain-specific hints about the location of goals, beyond the definition of the problem itself.

- Heuristic Search Methods
- Best-first search algorithms
  - Greedy best-first search
  - A^* search
- Admissible Heuristic
- Relaxed Problems

Heuristic Search Methods

node to be expanded

- Use an evaluation function *f(n)* for each node
  - Evaluation function $f(n) = f(h(n))$ (heuristic)

    h(n)= estimate of cost from *n* to *goal*

    h(n)=0 if n is a goal node
  - Order the nodes in fringe in decreasing order of desirability.
  - Expand most desirable unexpanded node. (lowest ← distance)
- Focus on states estimated to be most desirable (i.e. states with lowest or highest heuristic values).
- The performance highly depends on how good is its h(n).
- A good heuristic function can often find good (though possibly non-optimal) solutions in less than exponential time.

Additional thoughts on H

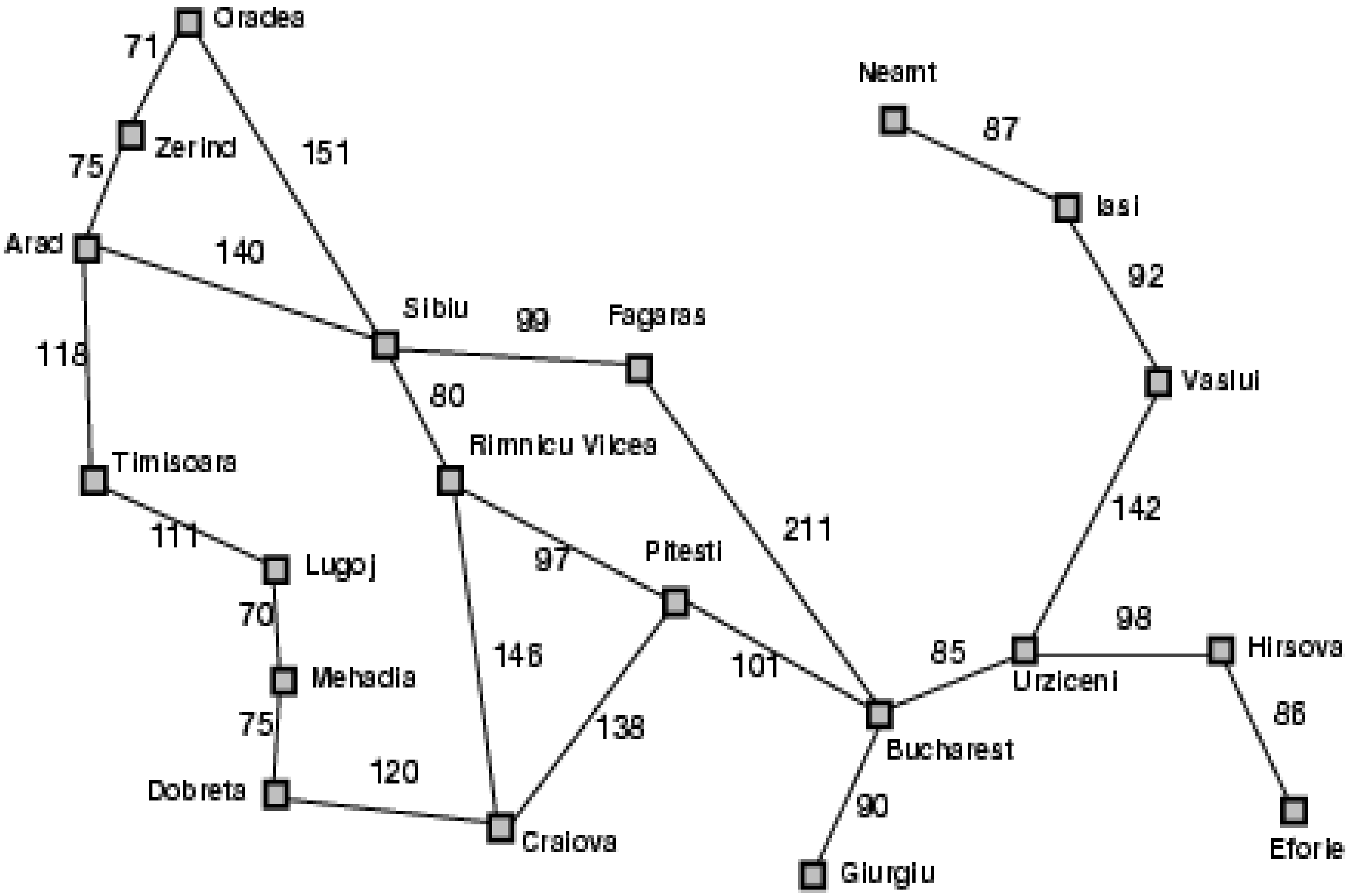
- People often are satisfied with a good, though not optimal, solutions. (could you think of a ex.?)
- Although heuristic search may not be better than uninformed search in worst cases, worst cases rarely arise in the real world.
- Trying to understand why a heuristic works, or why it does not work, often leads to a deeper understanding of the problem.
- There is a trade-off between the time spent in computing heuristic values and the time spent in search.

Greedy best-first search

- Idea: expands the node that appears to be closest to goal.
- Evaluation function $f(n)=h(n)$
- e.g.,

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

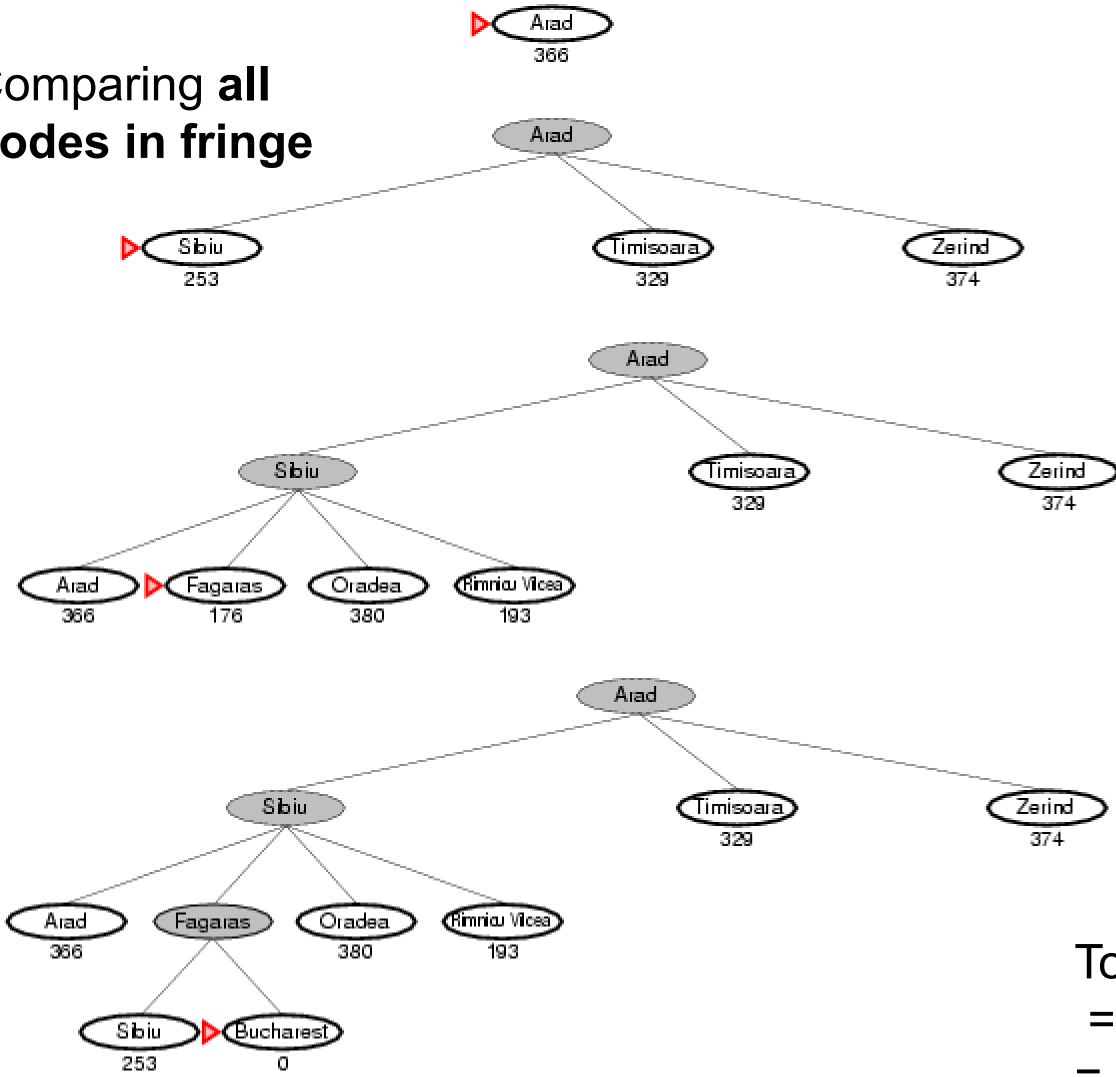
Romania with step costs in km



Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Greedy best-first search example

Comparing **all nodes in fringe**



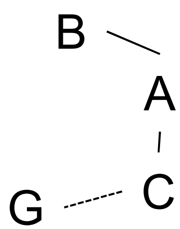
Greedy	Optimal
Arad	Arad
↓	↓
Sibiu	Sibiu
↓	↓
Fagaras	Rimnicu Vilcea
↓	↓
Bucharest	Pitesti
	↓
	Bucharest
Total cost = 140+99+211 = 450	Total cost = 140+80+97+101 = 418

Properties of greedy best-first search

- Complete?
- Time?
- Space?
- Optimal?

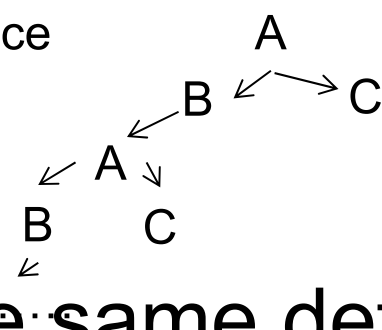
Properties of greedy best-first search

- Complete? Yes – in a finite tree. No – can get stuck in loops in infinite trees. e.g.,



Straight-line distance from n to G

A	15
B	10
C	20



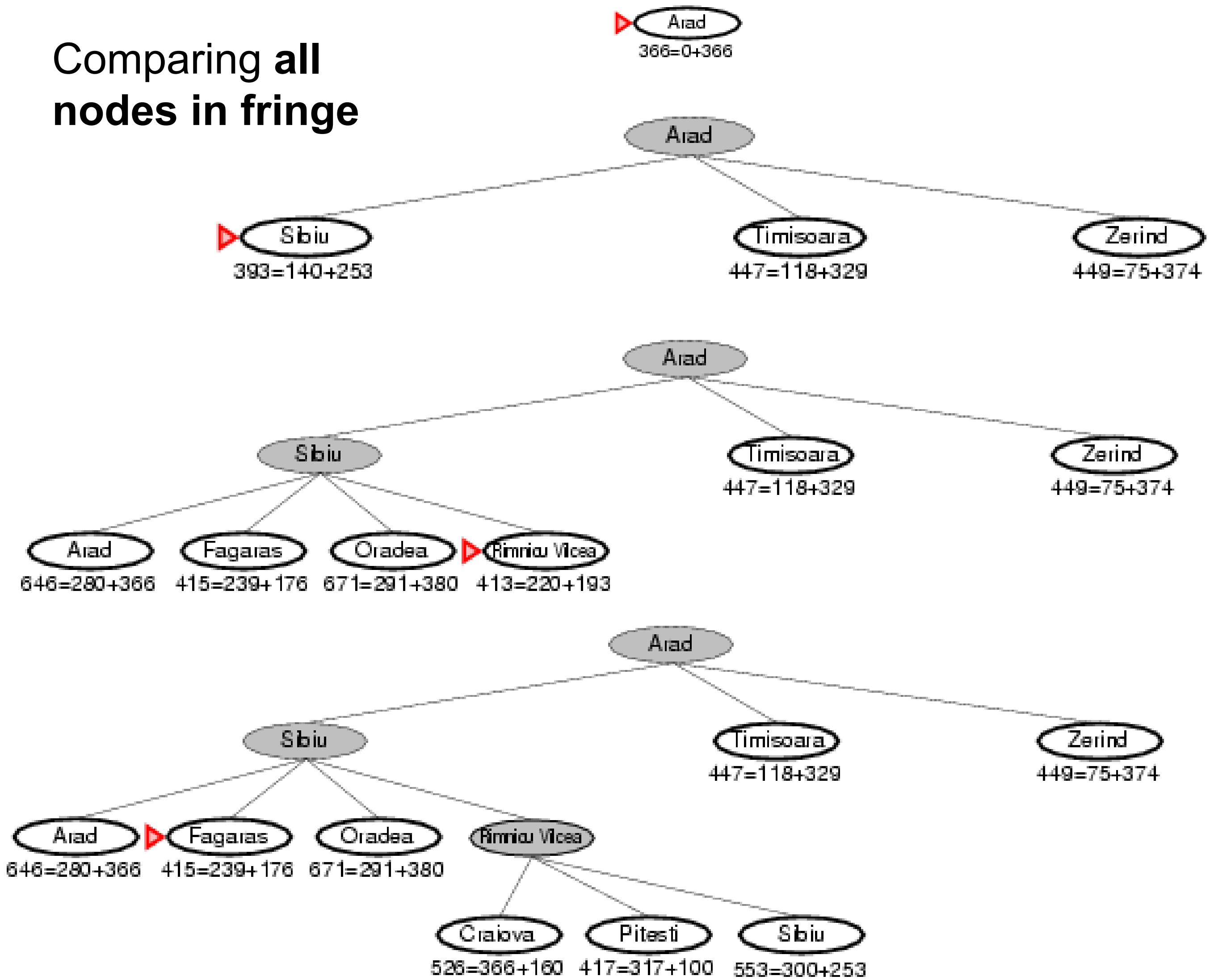
- Time? $O(b^m)$, (suffer the same defects as DFS) but a good heuristic can give dramatic improvement ($O(d)$)
- Space? $O(b^m)$
- Optimal? No

A^* search

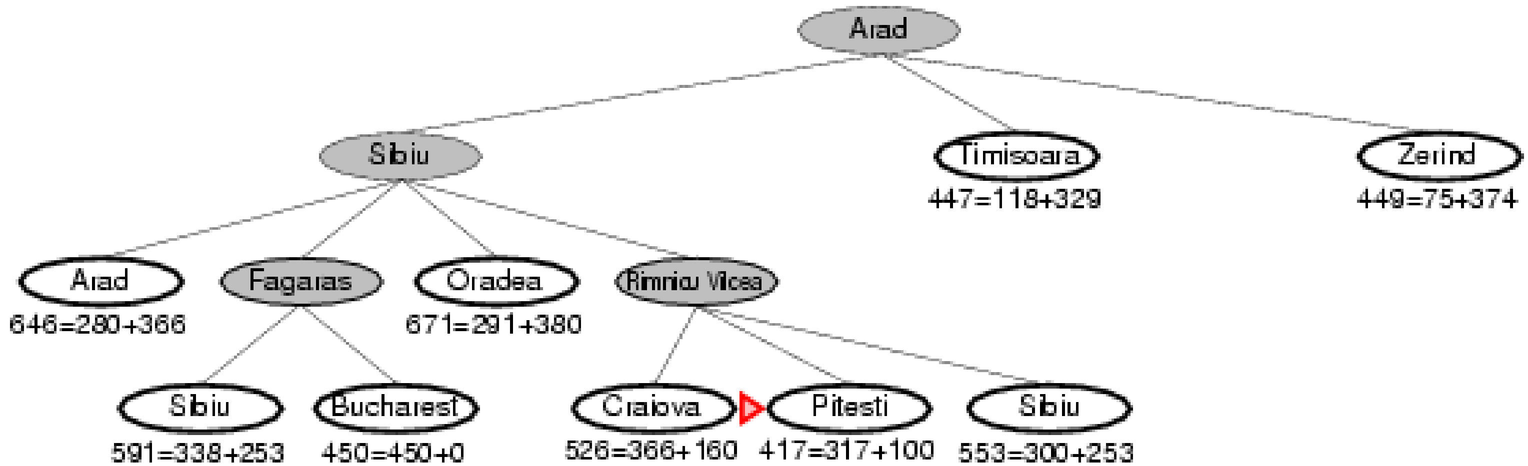
- Ideas
  - Minimizing the **total** estimated solution cost.
  - Avoid expanding paths that are already expensive
- Evaluation function $f(n) = g(n) + h(n)$
  - $g(n)$ = cost to reach n from root
  - $h(n)$ = cost from n to goal
  - $f(n)$ = estimated total cost of path through n to goal

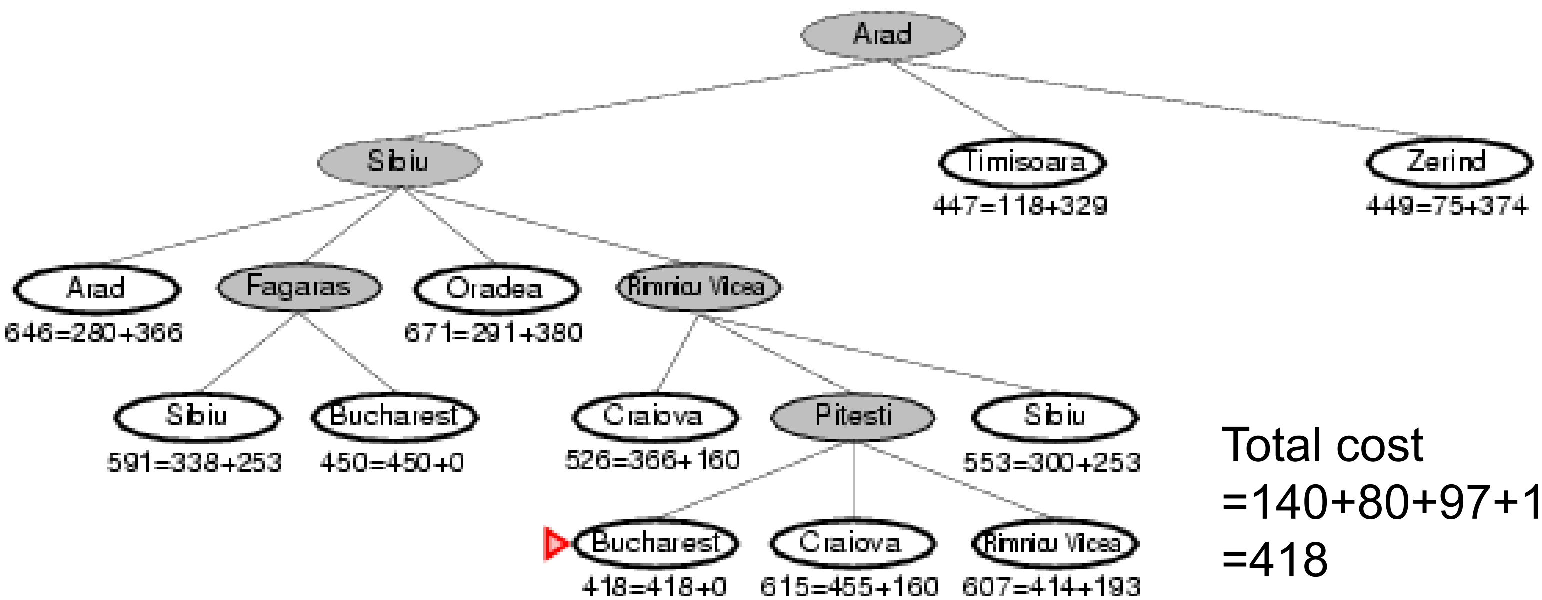
A* search example

Comparing **all nodes in fringe**



A* search example





Total cost
=140+80+97+101
=418

- BFS, DFS, Uniform-cost search are special cases of A*
  - BFS: f(n)=depth(n)
  - DFS: f(n)=-depth(n)
  - UCS: f(n)=g(n).

Properties of A*

- Complete?
- Time?
- Space?
- Optimal?

Properties of A*

- <u>Complete?</u> Yes (unless there are infinitely many nodes with $f(n) \leq f(G)$).
- <u>Time?</u> Exponential. A* expands all nodes with $f(n)<C^*$.
- <u>Space?</u> Keeps all nodes in memory.
- <u>Optimal?</u> Depends on quality of h. Yes, if h is *admissible* (`TREE-SEARCH`) or *consistent* (`GRAPH-SEARCH`).

Admissible heuristics

Theorem: If $h(n)$ is admissible, A* using TREE-SEARCH is optimal.

- A heuristic $h(n)$ is admissible iff

$$\forall n \;\; h(n) \leq h^*(n),$$

  where $h^*(n)$ is the true cost to reach the goal state from n.

- An admissible heuristic never overestimates the cost to reach the goal, i.e., it is optimistic.
- Question: $h_{SLD}(n)$ *is admissible?*

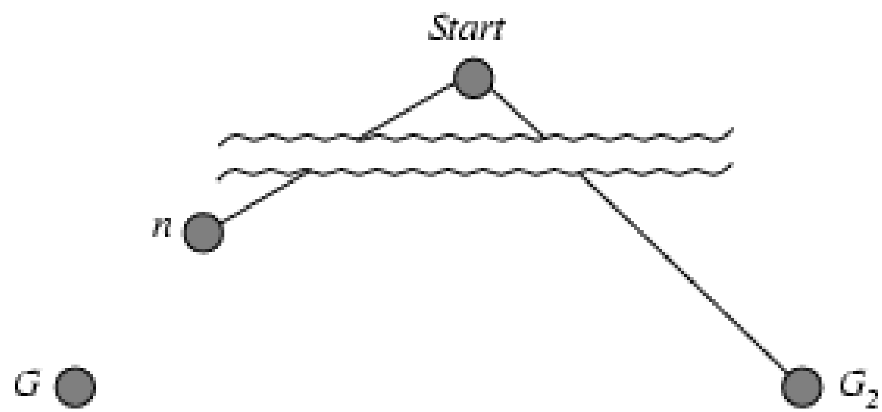
Admissible heuristics

- Question: $h_{SLD}(n)$ *is admissible?* ***Yes****, because the straight-line distance* never overestimates the actual road distance.

Optimality of A^* (proof)

- Suppose some suboptimal goal G_2 has been generated and is in the fringe. Let *n* be an unexpanded node in the fringe such that *n* is on a shortest path to an optimal goal *G*.

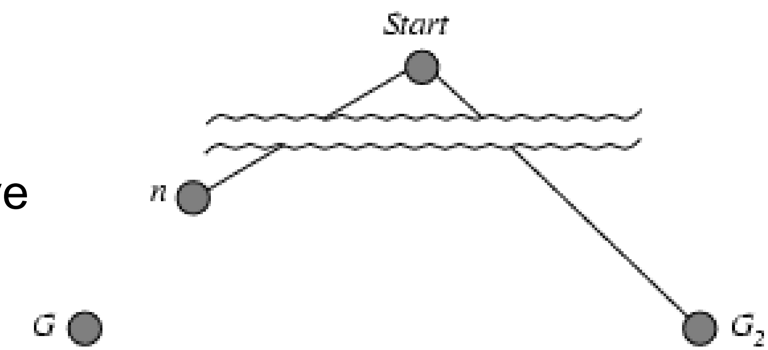
Idea: in this case, we will expand n not G_2,
i.e. we need to prove
f(n)<f(G2).



Optimality of A* (proof)

- Suppose some suboptimal goal G_2 has been generated and is in the fringe. Let *n* be an unexpanded node in the fringe such that *n* is on a shortest path to an optimal goal *G*.

In this case, we will expand n not G_2,
i.e. we need to prove
f(n)<f(G2)



- f(n) = g(n)+h(n)
- h(n) ≤ h*(n), h is admissible

$\Rightarrow$ g(n)+h(n) ≤g(n)+ h*(n) $\Rightarrow$ f(n) ≤f(G)

- $f(G_2) = g(G_2)$ since $h(G_2) = 0$
- f(G) = g(G) since h(G) = 0
- $g(G) < g(G_2)$ since G_2 is suboptimal

$\Rightarrow$ $f(G) < f(G_2)$

$\Rightarrow$ $f(n) < f(G_2)$

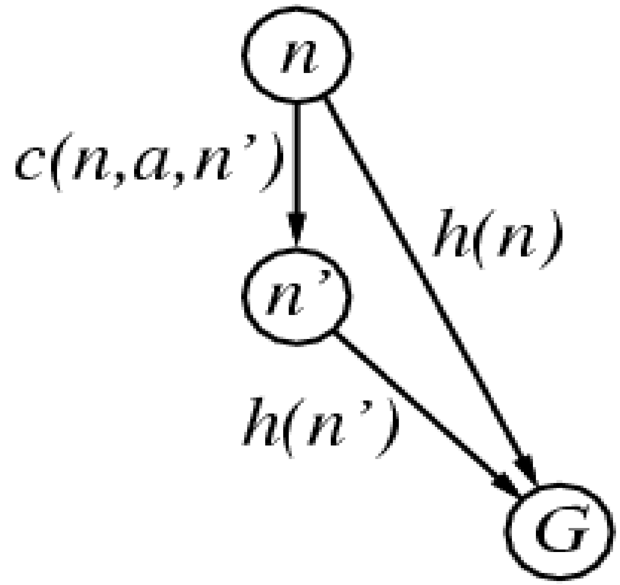
So A* will never select G_2 for expansion. So A* is optimal.

Consistent heuristics

Theorem: If $h(n)$ is consistent, A* using GRAPH-SEARCH is optimal.

- A heuristic is consistent if for every node n, every successor n' of n generated by any action a,

  $h(n) \le c(n,a,n') + h(n')$ (triangle inequality)



- If h is consistent, $f(n)$ is non-decreasing along any path

$$
\begin{aligned}
f(n') &= g(n') + h(n') \\
&= g(n) + c(n,a,n') + h(n') \\
&\ge g(n) + h(n) \\
&= f(n)
\end{aligned}
$$

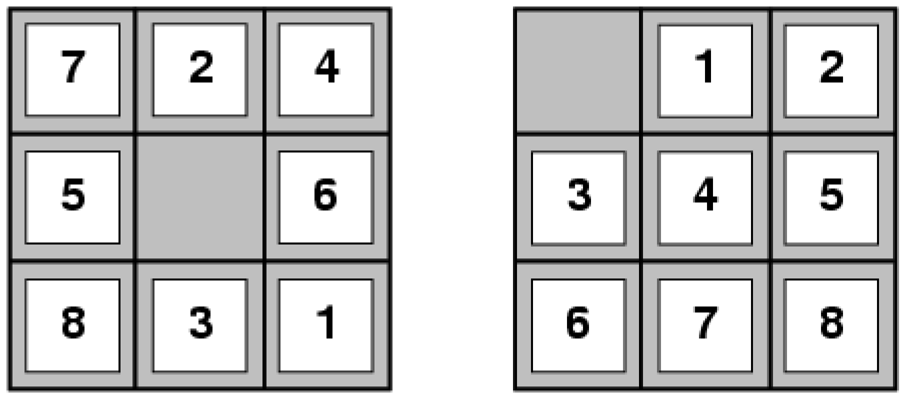
So the first goal node selected for expansion must be optimal, since all later nodes will be at least as expensive.

Every consistent heuristic is admissible.

$h_{SLD}(n)$ *is consistent.*

Example of Admissible heuristics

The 8-puzzle problem



Start State Goal State

Example of Admissible Heuristics

Complexity of the problem in general

Average b: 3

Average d (step): 22

State space: $b^d = 3^{22} = 3 \times 10^{10}$
including repeated states

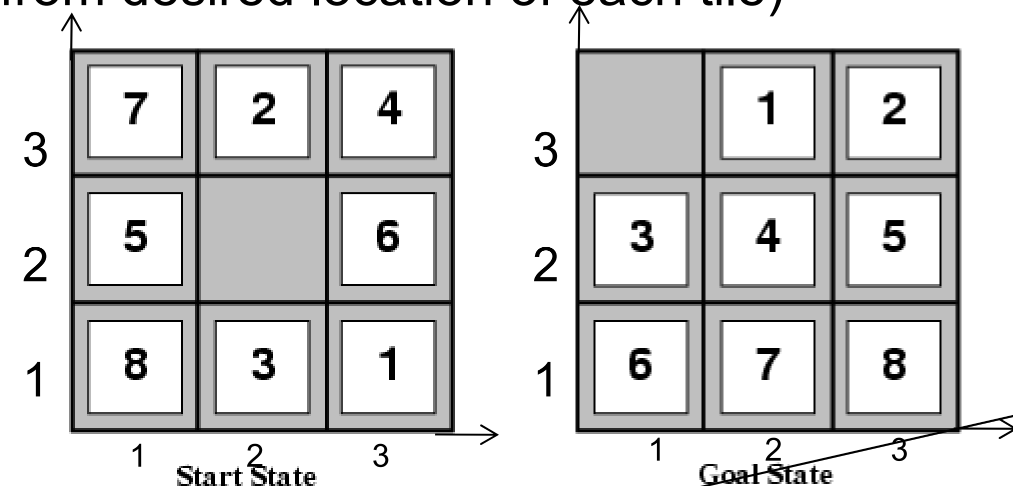
State space: 170,000 without repeated states

E.g., for the 8-puzzle:

- $h_1(n)$ = number of misplaced tiles
- $h_2(n)$ = total Manhattan distance

(i.e., # of squares from desired location of each tile)

$|x_s - x_g| + |y_s - y_g|$



(3, 1) vs (2, 3)

|3-2| + |1-3| = 3

Start state

- $h_1(S) = 8$
- $h_2(S) = 3+1+2+2+2+3+3+2 = 18$

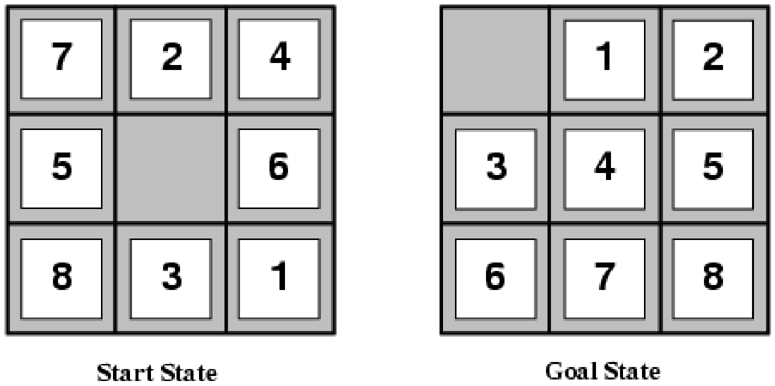
1 2 3 4 5 6 7 8

Relaxed problems

- A problem with fewer restrictions on the actions is called a relaxed problem.
- *The cost of an optimal solution to a relaxed problem is an admissible heuristic for the original problem.*

Original Problem		Relaxed Problem
C_o^*	$\geq$	C_r^*

An 8-puzzle example



- A tile can move A->B if
  - A is horizontally/vertically adjacent B, and
  - B is blank.
- Relaxed problem:
  - (1) A tile can move A->B.
  - (2) A tile can move A->B, if A is adjacent B.
  - (3) A tile can move A->B, if B is blank.
- If the rules of the 8-puzzle are relaxed so that a tile can move anywhere (1), then $h_1(n)$ gives the shortest solution.
- If the rules are relaxed so that a tile can move to any adjacent square (2), then $h_2(n)$ gives the shortest solution.