ITCS 6150
Intelligent Systems

Lecture 5

Informed Searches

Informed Searches

We are informed (in some way) about future states and future paths

We use this information to make better decisions about which of many potential paths to pursue

A* Search

Combine two costs

- f(n) = g(n) + h(n)
  - g(n) = cost to get to n from the root
  - h(n) = cost to get to goal from n
    - ***admissible heurisitic***
    - h(n) is optimistic
    - f(n) never overestimates cost of a solution through n

Expand node with minimum f(n)

Repeated States and Graph-Search

Graph-Search always ignores all but the first occurrence of a state during search

- Lower cost path may be tossed
  - So, don't throw away subsequent occurrences
  - Or, ensure that the optimal path to any repeated state is always the first one followed
- Additional constraint on heurisitic, consistency

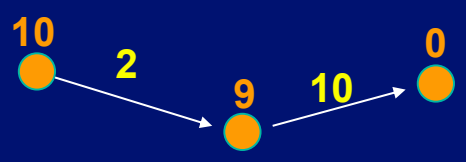
Consistent (monotonic) h(n)

Heuristic function must be monotonic

- for every node, *n*, and successor, *n'*, obtained with action *a*
  - estimated cost of reaching goal from n is no greater than cost of getting to n' plus estimated cost of reaching goal from n'
  - $h(n) \leq c(n, a, n') + h(n')$
- This implies f(n) along any path are non-decreasing

Examples of consistent h(n)

$h(n) \leq c(n, a, n') + h(n')$



- recall h(n) is admissible
  - The quickest you can get there from here is 10 minutes
    - It may take more than 10 minutes, but not fewer
- After taking an action and learning the cost
  - It took you two minutes to get here and you still have nine minutes to go
  - We cannot learn… it took you two minutes to get here and you have seven minutes to go

Proof of monotonicity of f(n)

If h(n) is consistent (monotonic)

then f(n) along any path is nondecreasing

- suppose n' is successor of n
  - $g(n') = g(n) + c(n, a, n')$ for some *a*
  - $f(n') = g(n') + h(n')$
    $= g(n) + c(n, a, n') + h(n')$
    $\geq g(n) + h(n) = f(n)$

monotonicity implies
$h(n) \leq c(n, a, n') + h(n')$

Contours

Because f(n) is nondecreasing

we can draw contours

- If we know C*
- We only need to explore contours less than C*

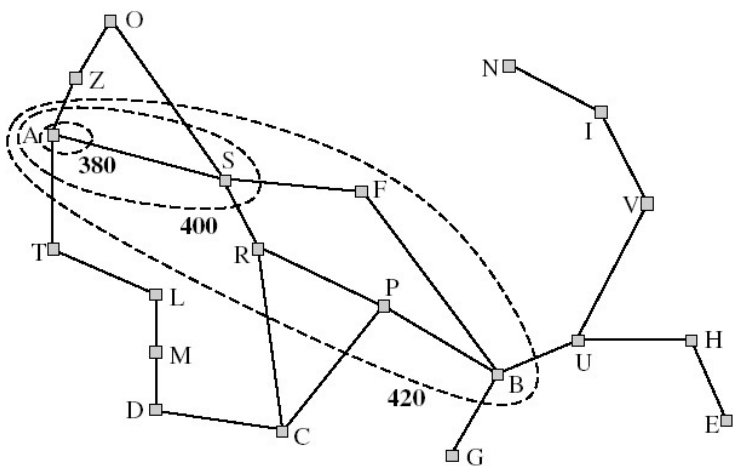


Figure 4.4 Map of Romania showing contours at $f = 380$, $f = 400$ and $f = 420$, with Arad as the start state. Nodes inside a given contour have f-costs less than or equal to the contour value.

Properties of A*

- A* expands all nodes n with $f(n) < C^*$
- A* expands some (at least one) of the nodes on the C^* contour before finding the goal
- A* expands no nodes with $f(n) > C^*$
  - these unexpanded nodes can be pruned

A* is Optimally Efficient

Compared to other algorithms that search from root

Compared to other algorithms using same heuristic

*No other optimal algorithm is guaranteed to expand fewer nodes than A**

Pros and Cons of A*

A* is optimal and optimally efficient

A* is still slow and bulky (space kills first)

- Number of nodes grows exponentially with the length to goal
  - This is actually a function of heuristic, but they all have errors
- A* must search all nodes within this goal contour
- Finding suboptimal goals is sometimes only feasible solution
- Sometimes, better heuristics are non-admissible

Memory-bounded Heuristic Search

Try to reduce memory needs

Take advantage of heuristic to improve performance

- Iterative-deepening A* (IDA*)
- Recursive best-first search (RBFS)
- SMA*

Iterative Deepening A*

Iterative Deepening

- Remember, as an uniformed search, this was a depth-first search where the max depth was iteratively increased
- As an informed search, we again perform depth-first search, but only nodes with f-cost less than or equal to smallest f-cost of nodes expanded at last iteration
- Don’t need to store ordered queue of best nodes

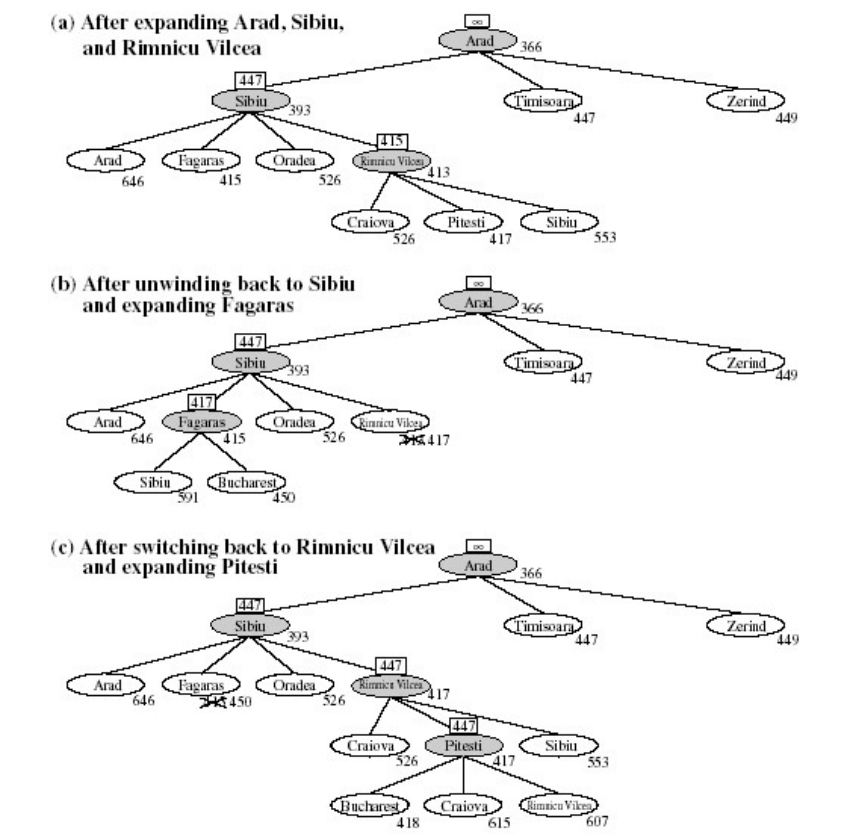
Recursive best-first search

Depth-first combined with best alternative

- Keep track of options along fringe
- As soon as current depth-first exploration becomes more expensive of best fringe option
  - back up to fringe, but update node costs along the way

Recursive best-first search

- box contains f-value of best alternative path available from any ancestor
- First, explore path to Pitesti
- Backtrack to Fagaras and update Fagaras
- Backtrack to Pitesti and update Pitesti



Meta-foo

What does meta mean in AI?

- Frequently it means step back a level from foo
- Metareasoning = reasoning about reasoning
- These informed search algorithms have pros and cons regarding how they choose to explore new levels
  - a metalevel learning algorithm may combine learn how to combine techniques and parameterize search

Heuristic Functions

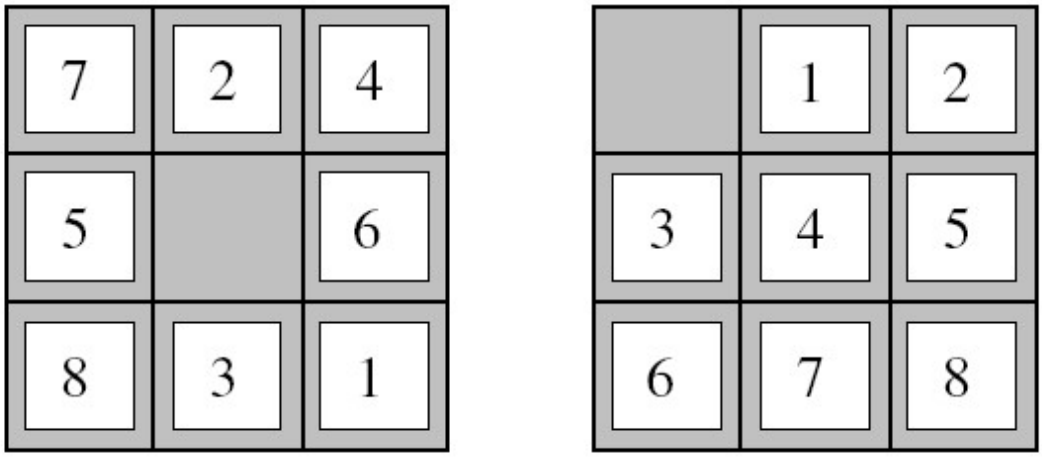
8-puzzle problem

Avg Depth=22

Branching = approx 3

3^{22} states

170,000 repeated



Start State Goal State

Figure 4.7 A typical instance of the 8-puzzle. The solution is 26 steps long.

Heuristics

The number of misplaced tiles

- Admissible because at least n moves required to solve n misplaced tiles

The distance from each tile to its goal position

- No diagonals, so use Manhattan Distance
  - As if walking around rectilinear city blocks
- also admissible