University of South Australia

COMP 2019

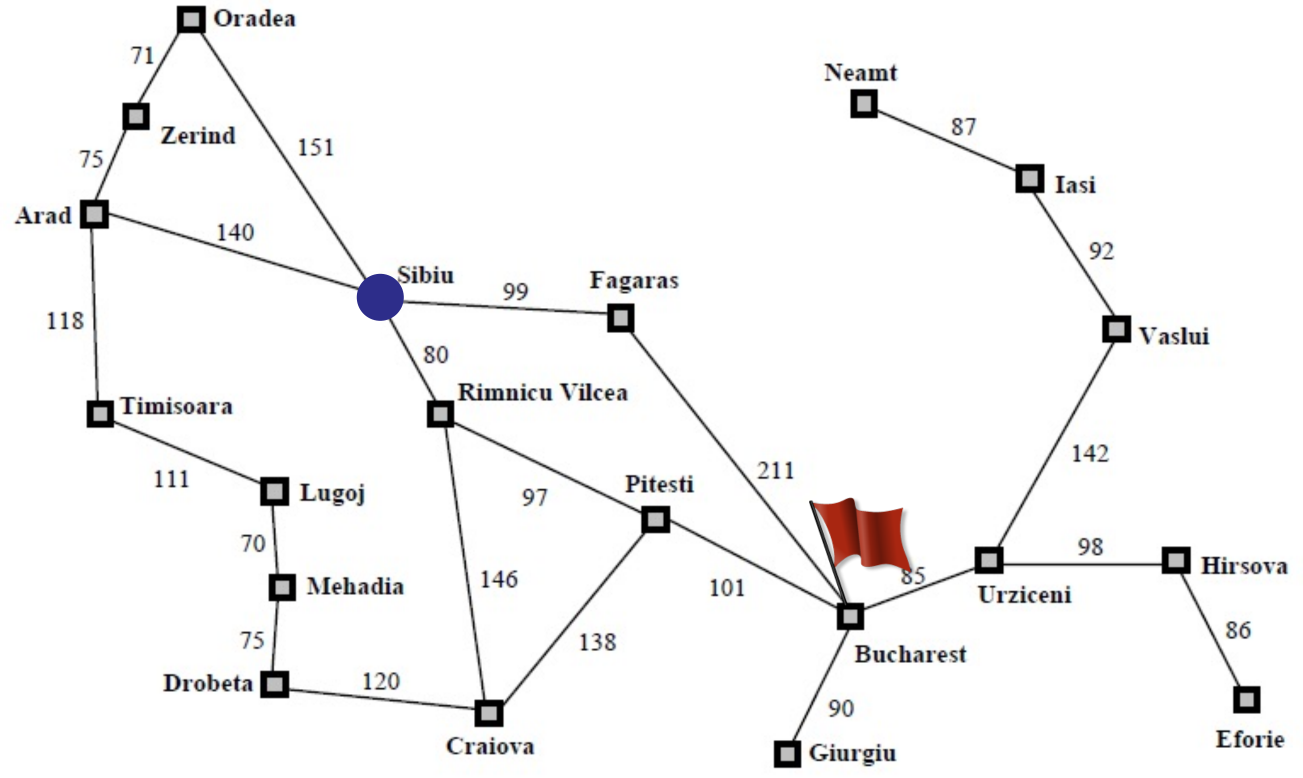
Week 3

Heuristic Search

Learning Objectives

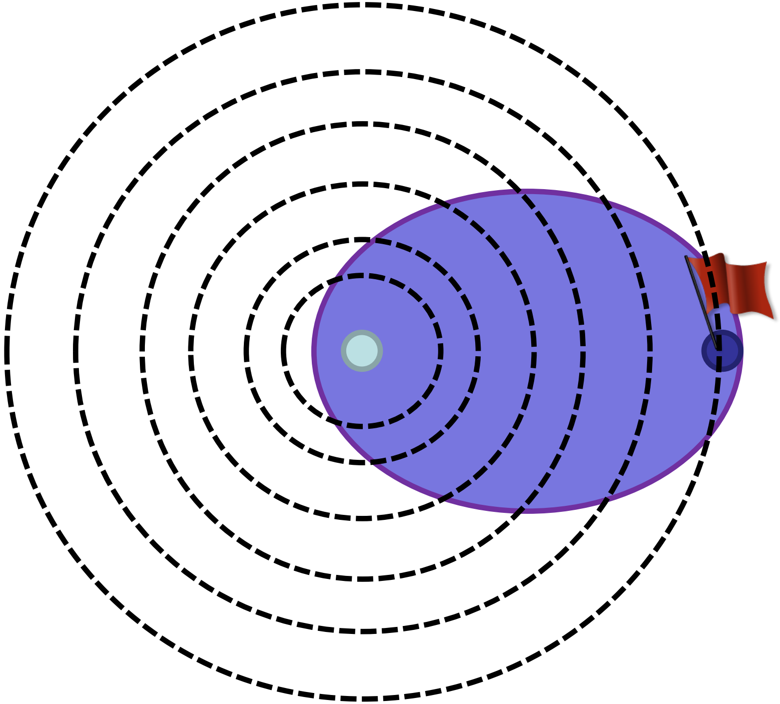
- Explain algorithms for solving *intractable* problems by searching (CO1)

Motivation


Oradea
71
Zerind
151
75
Arad
140
Sibiu
99
Fagaras
118
80
Rimnicu Vilcea
Timisoara
111
Lugoj
97
Pitesti
211
70
Mehadia
146
101
85
75
138
Bucharest
Drobeta
120
Craiova
90
Giurgiu
Neamt
87
Iasi
92
Vaslui
142
98
Urziceni
Hirsova
86
Eforie

Factors of Complexity in Searching

- Size of the state space
- Strategy for choosing actions
- Efficiency of the goal test



Idea behind A*

Total cost of path from S to G through state x:

$$f(x) = g(x) + h(x)$$

Heuristic Estimation Function

Consider states in order of increasing f-value.

$h(x) \geq 0$ for all states x
$h(g) = 0$ for all goal states g

Graph Search for A*

```
frontier = [ InitialState ]
explored = { }
loop:
        if frontier is empty then: return Fail
        path = Remove-Path(frontier)
        state = path.end
        add state to explored
        if IsGoal(state) then: return path
        for a in Actions(state):
                newpath = (path, Result(state,a))
                if newpath.end not in explored then:
                        update frontier with newpath
```

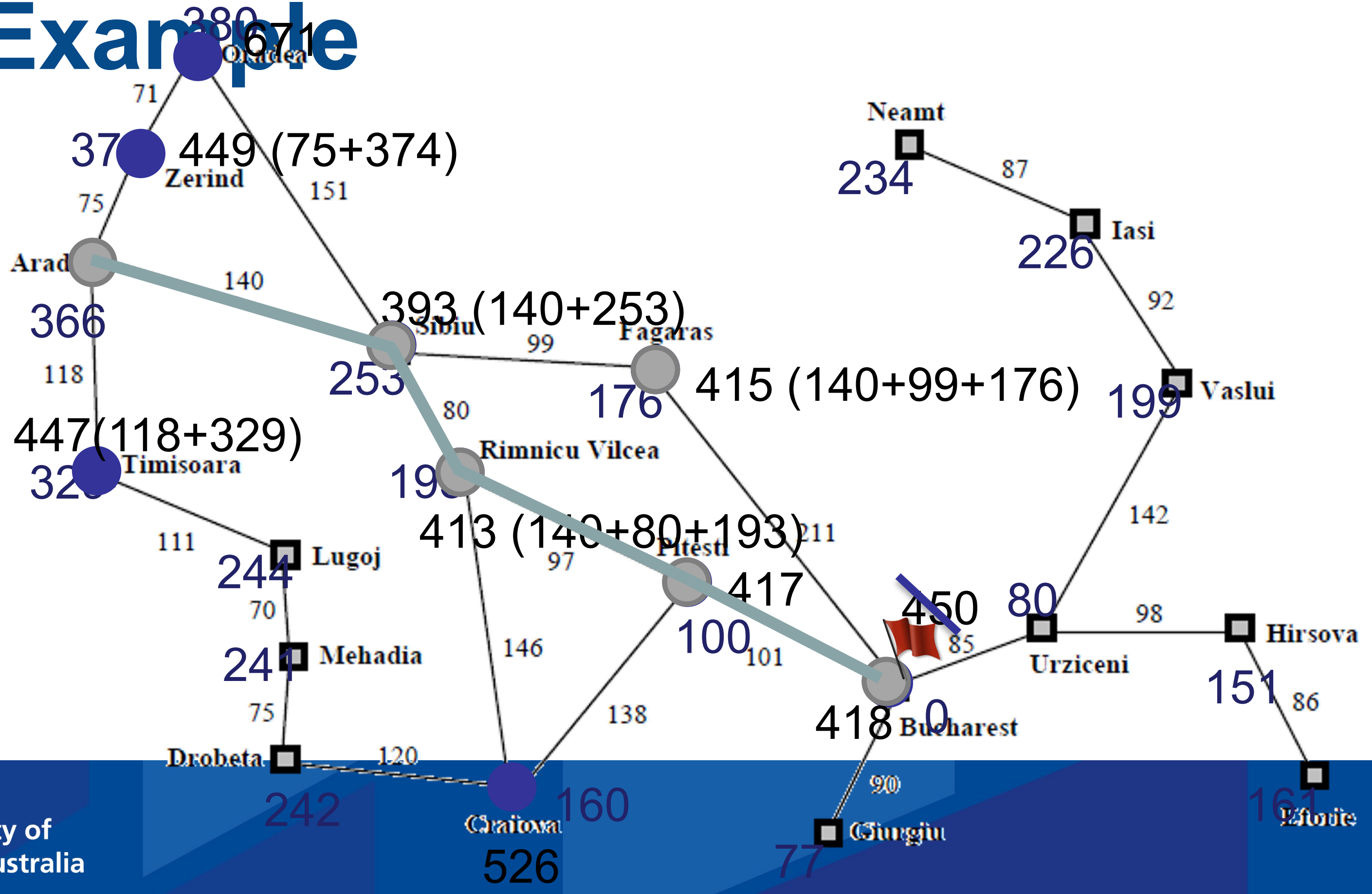
A* Algorithm

A* = Best-first search where cost is determined by f(x)

We remove the entry with lowest f value from the frontier.

We keep only the path to a state that has the least f value among the known paths to that state in the frontier

A* Example



University of South Australia

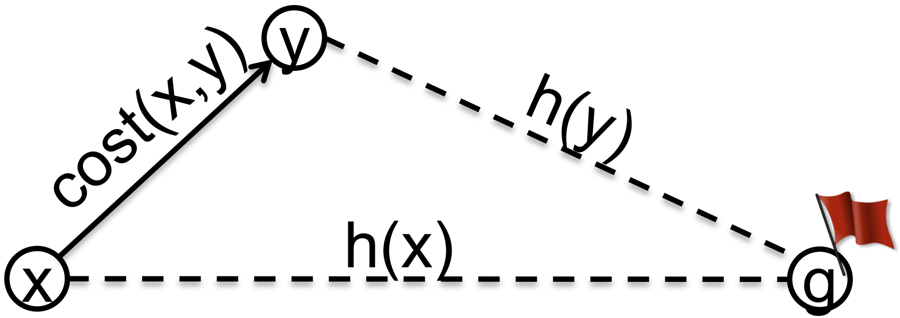
Monotonic ("Consistent") h

A heuristic function is monotonic iff

$$h(x) \leq h(y) + cost(x, y)$$

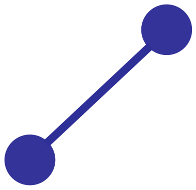
where y is a direct successor of x, and

$h(g) = 0$ for any goal state g.

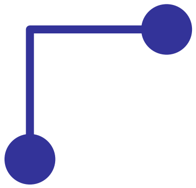


Common Heuristics (for 2D Spaces)

- Euclidean Distance
  - Straight line distance



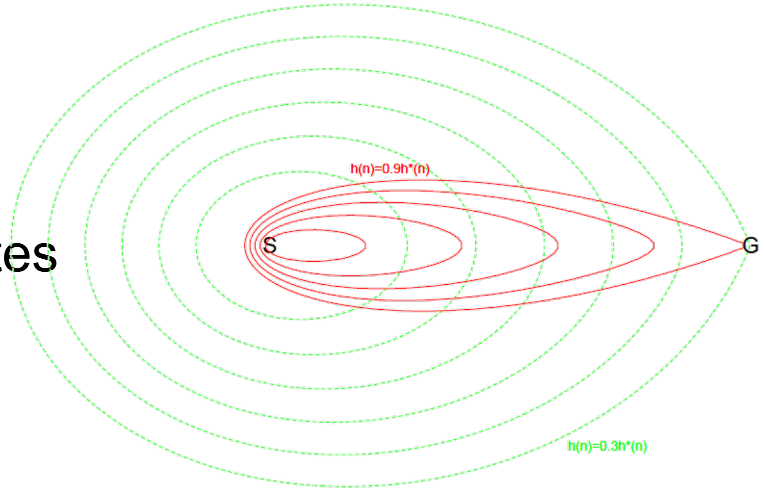
- Manhattan Distance
  - Sum horizontal and vertical distances.



Comparing Heuristic Functions

- Bad estimates can cause extra work.
- Heuristic function h_1 is better than h_2 if $h_1(x) > h_2(x)$ for all non-goal states x
- Better heuristics lead to fewer visited states

Red: $0.9\times$ true distance
Green: $0.3\times$ true distance



Other State Spaces

- Initial State
  - Permuted board
- Goal State
  - All tiles in order
- Actions
  - Move blank ←↑→↓
- Step cost: 1
  - Minimize number of moves

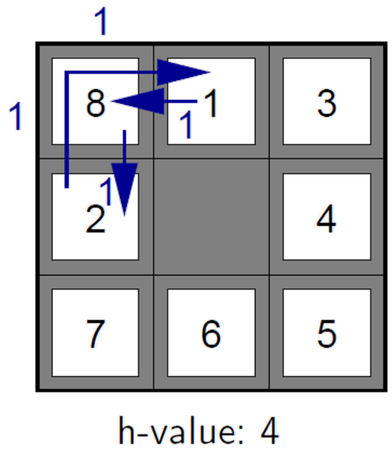
http://www.mypuzzles.xtreemhost.com/images/15%20Puzzle.jpg

Heuristics for 8/15 Puzzle

- Which heuristic to use?

- $h_1 =$ # misplaced tiles

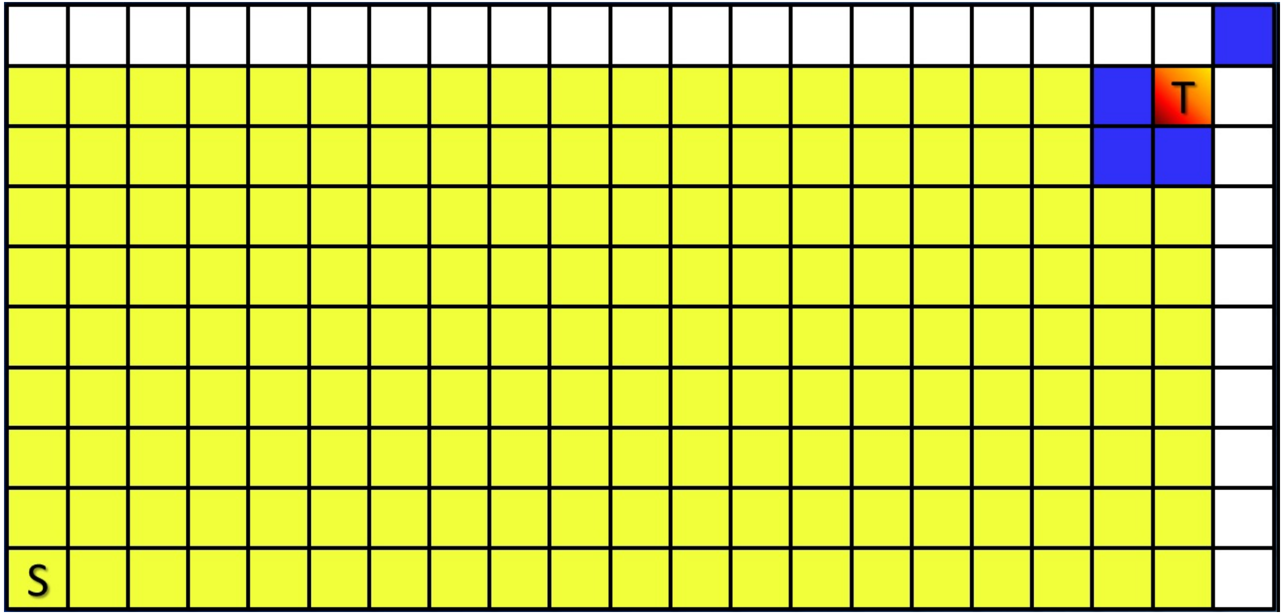
- $h_2 =$ Sum(distances of blocks)



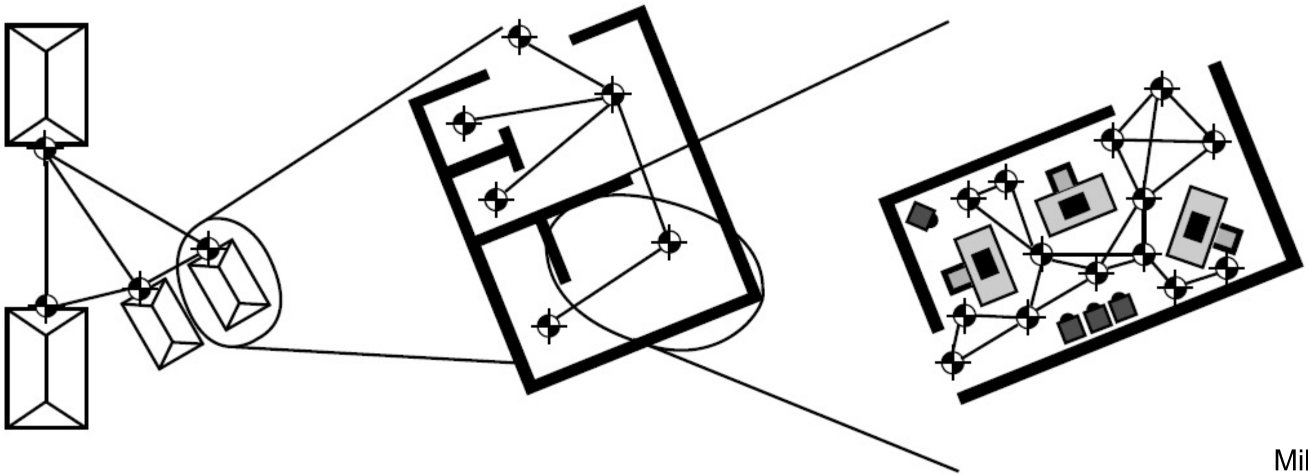
Finding Other Heuristics

- Relax the problem description
  - Abstract from constraints present in the full description.
  - Example: the Manhattan distance in the 15 Puzzle does not consider conflicts between tiles.
- Solve a sub-problem
  - Solve a sub-problem of the current state completely.
    Use the resulting costs as a lower bound of the current problem.
  - Assumption: sub-problems can be solved quickly.
  - Keep a database of small sub-problems and their costs.
  - Example (15-Puzzle): move tiles 1-3 into the correct position.
- Combine heuristic functions $f_1(x), \ldots, f_k(x)$ into $f(x) = \max(f_1(x), \ldots, f_k(x))$.
  - Typically, f_i are specialised for different situations.

Heuristics may not be enough



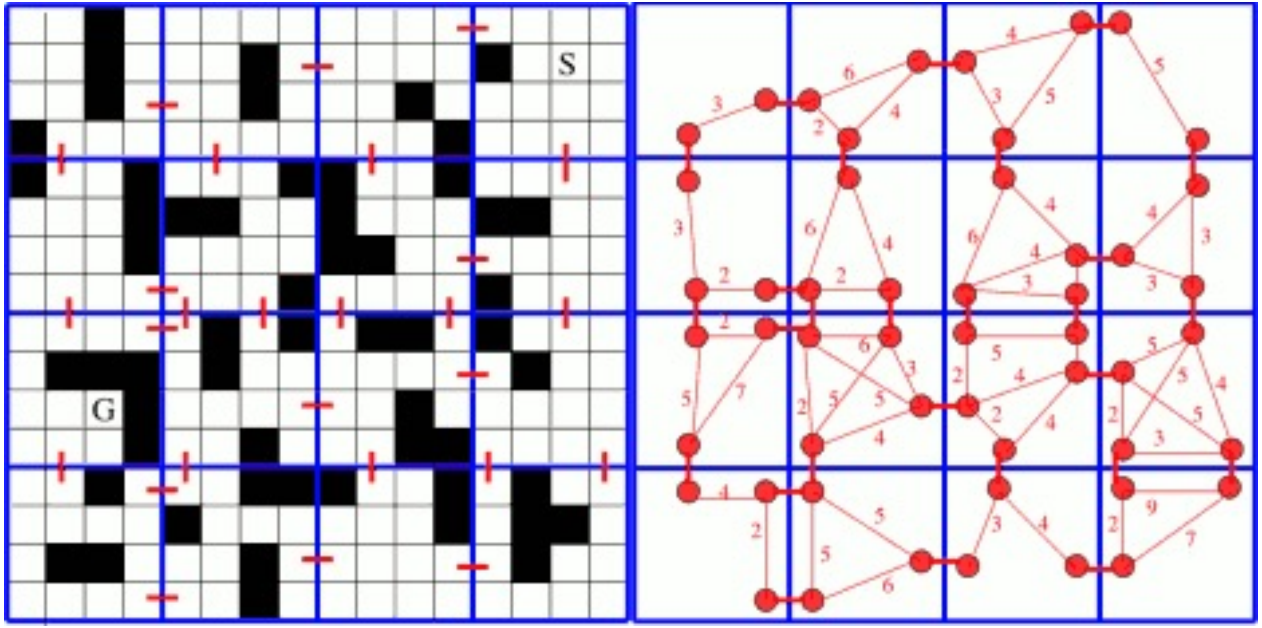
Hierarchical Pathfinding



Millington, Figure 4.37

Many approaches exist to divide a world into abstract locations.

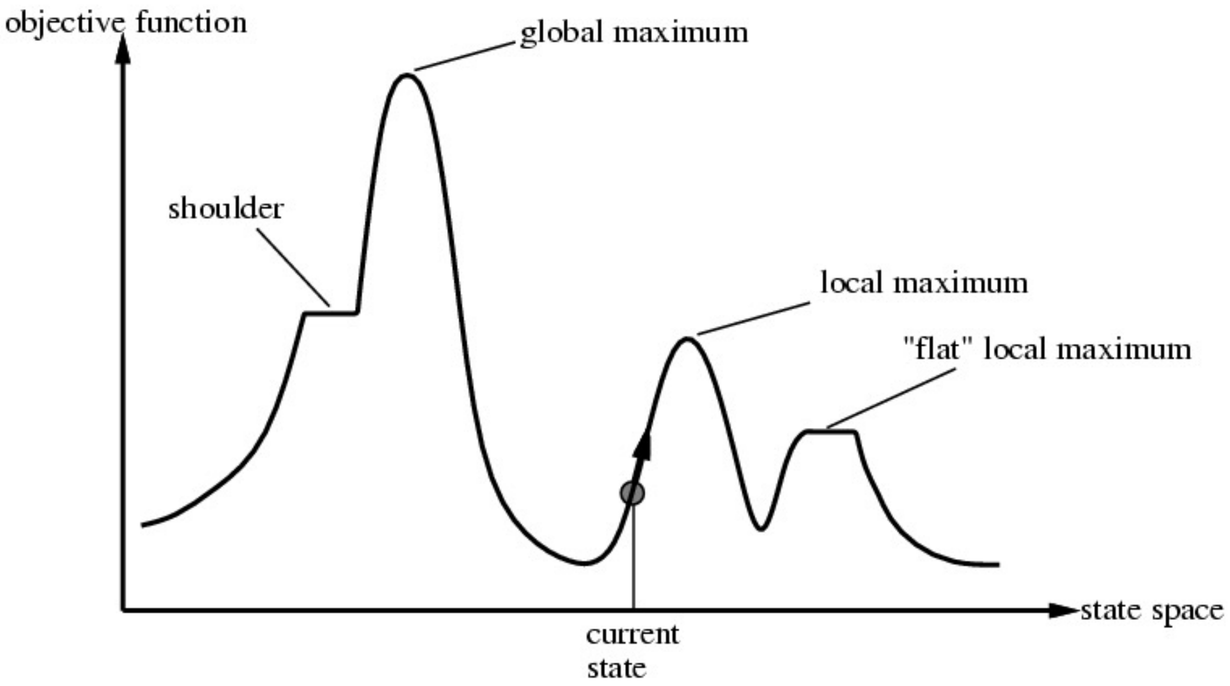
HPA*



Local Search & Optimisation

- Local Search: use no "global" information
  - Less memory
  - Find solutions in large (continuous) search spaces
- Search for solution x that optimises a given cost function cost(x)

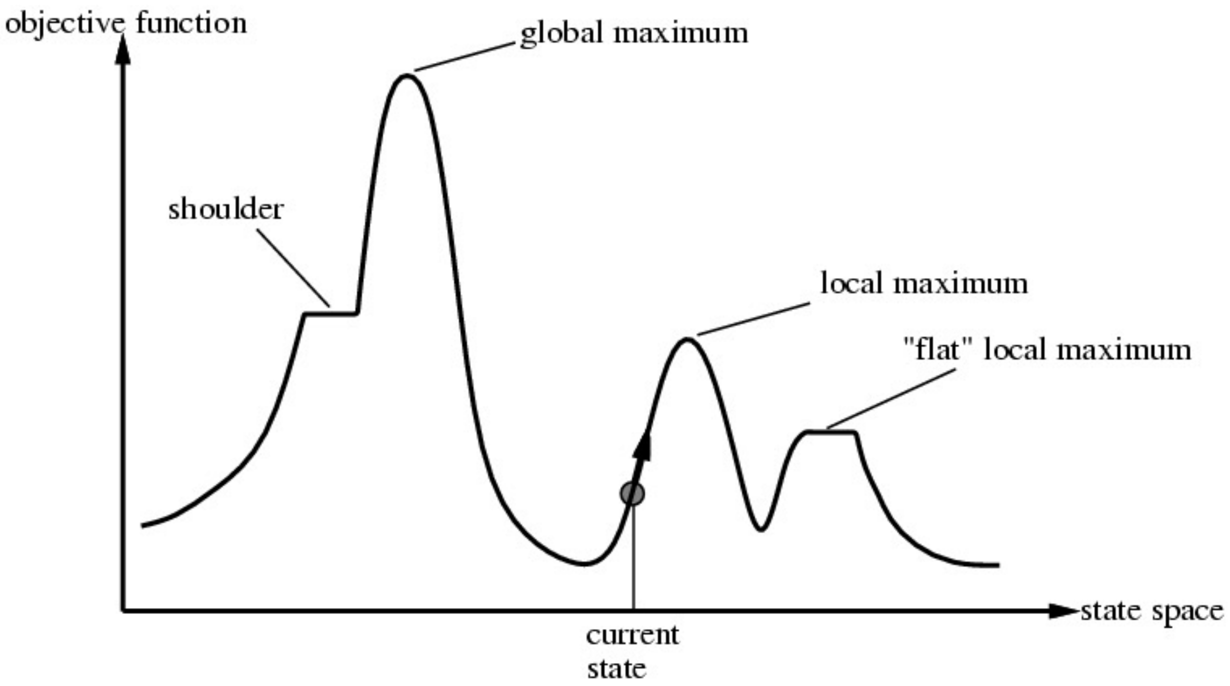
Hill Climbing



Local Search Methods

- Deterministic Methods
  - Step-by-step procedure
  - Example: Hill climbing
- Stochastic Methods
  - Iteratively improve solution
  - Rely on pseudo-random numbers to drive search
  - Example: Simulated annealing, Evolutionary Algorithms

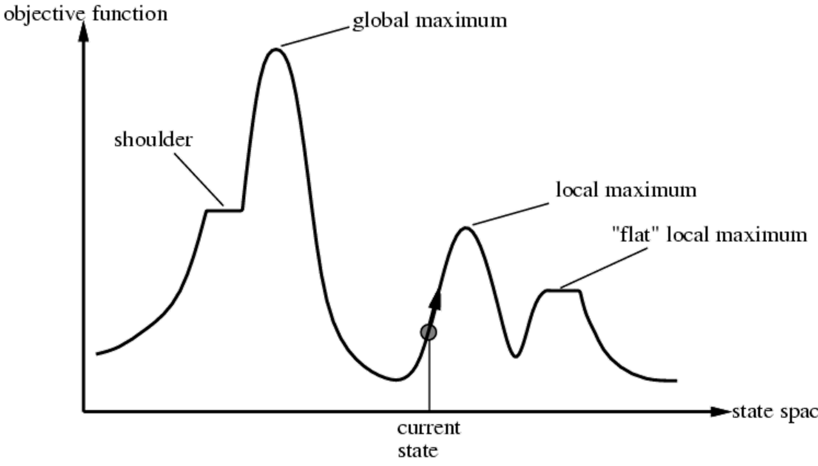
Hill Climbing



Hill Climbing Drawbacks

- Getting stuck in local maximum
- Slow progress on plateau
- Stray in plateau
- Step size?

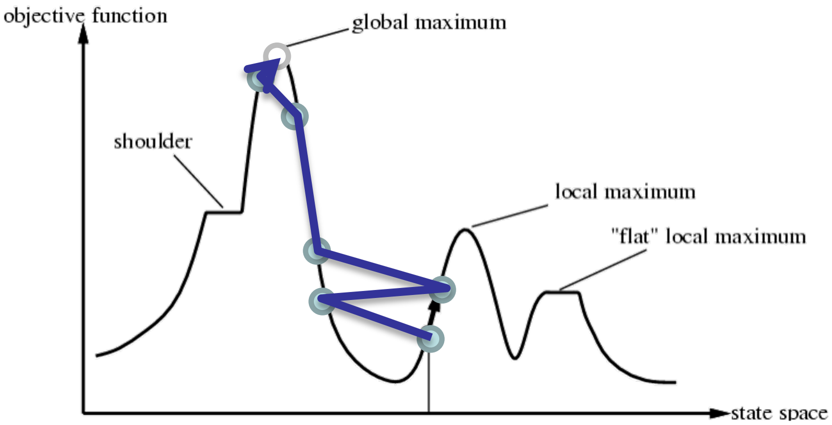
- Potential cures
  - Random restart
  - Simulated annealing
  - Tabu search, beam search
  - Evolutionary algorithms
  - …



Iterative Improvement Algorithms

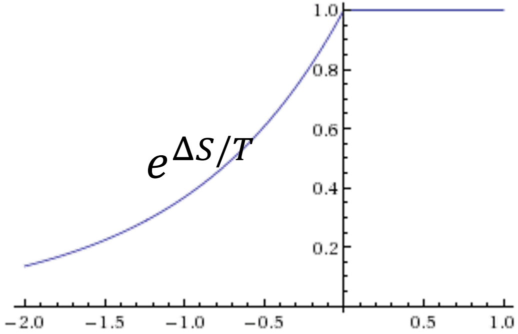
Loop:

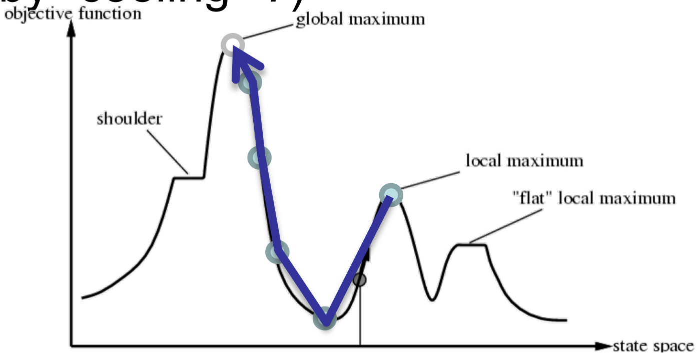
1. Find a solution s
2. Create a variation s’ from s
3. If s’ is better than s,
   make s’ the new s
4. Otherwise, make s’ the new s
   with some (small) probability



Simulated Annealing

- Escape local maxima by going downhill sometimes
  - Go uphill “most of times”
  - Keep a worse solution with small probability
  - Reduce probability over time (by “cooling” *T*)





Summary

- Heuristic search methods can find solutions to problems that are impossible to handle with uninformed search methods
- Heuristics help guide the search towards a solution
- Local search methods can find solutions for search problems that are too large for systematic methods

University of South Australia

Questions?