CS414-Artificial Intelligence

Lecture 6: Informed Search Algorithms

Waheed Noor

Computer Science and Information Technology,
University of Balochistan,
Quetta, Pakistan

Outline

Outline

Informed Search

- What we have seen so far are all those search algorithms that operate with out any domain knowledge, and try to solve the problem in brute-force fashion.
- Such methods are insufficient for most problems specially when the problem is large and complex.
- Now we will look into another type of search techniques called *informed search*.

Definition

Informed Search Incorporates a heuristic in the search that determines the quality of any state in the search space. In a graph search, this results in a strategy for node expansion. The heuristic may consider the problem knowledge to guide the search strategy.

Outline

Best First Search I

- Best-FS tries to minimize some estimated measure (evaluation function) of the cost of the solution.
- Generally, in Best-FS, state space is evaluated on the basis of an evaluation function $f(n)$ that incorporates an estimate of the measure of the cost of the path from the current state to the closest goal.
- Then the Best-FS may follow two strategies:

1. Greedy Search: Expands the node closest to the goal, i.e., $f(n) = h(n)$.
2. $A*$ Search: Expands the node on the least cost solution path, i.e., $f(n) = g(n) + h(n)$.

Best First Search II

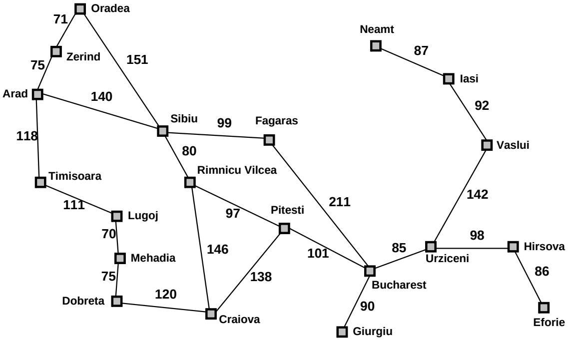
- where $h(n)$ is heuristic function and $g(n)$ is estimated path-cost from start node to current node n.
- The $h(n)$ can be an educated guess or estimate of the cheapest path from the state at node n to the goal.
- In Romania example, this may be a straight line between state at node n to goal.
- The $g(n)$ is the cost from the start node to node n.
- The algorithm is often implemented by maintaining two list, an open one and a closed one.
- Open list is a priority queue consist nodes yet to be visited sorted by their evaluation function.
- Closed list contains nodes that have already been evaluated.

Best First Search III

- Best-FS specializes to BFS when $f(n) = h(n)$ and to UCS when $f(n) = g(n)$.

An Example

Consider again the Romania example



Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Figure : Romania map with SLD

Greedy Fashion

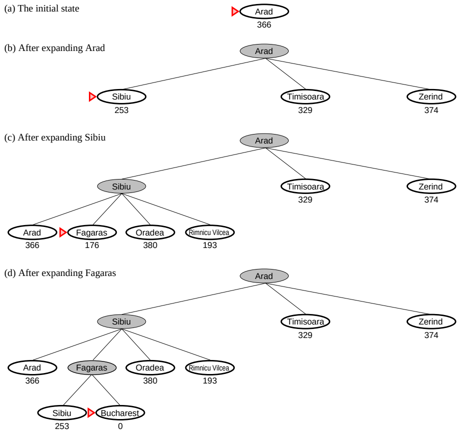
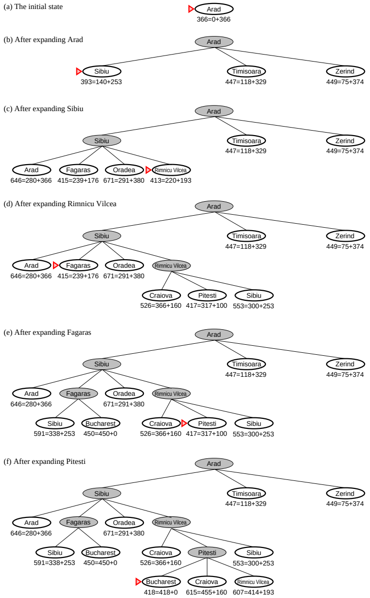


Figure : In the greedy fashion

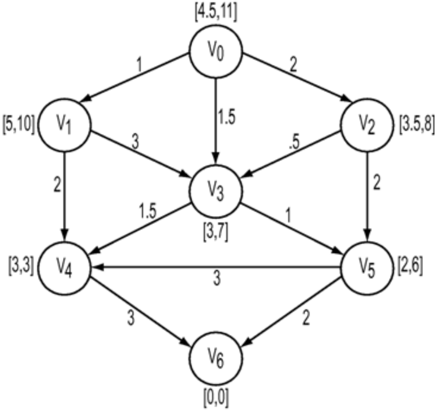
Outline

A* Search Example



Class Activity

Apply A* & Greedy Best-FS on following problem



A* Search

A* Search Rating

Complete:	Yes
Time Complexity:	$O(b^d)$
Space Complexity:	$O(b^d)$
Optimal:	Yes.

A* search algorithm have two flavors, the graph based and tree based.

Conditions for Optimality

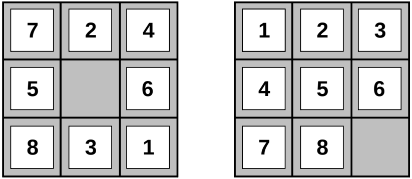
Admissibility and Consistency

Admissibility: That the heuristic is admissible i.e., that $h(n)$ will never overestimate the cost to reach the goal. For A*, $h(n)$ is admissible, since straight line is the shortest path between any two points.
Consistent: Consistency (also called monotonicity) describes that the cost along the path is increasing or decreasing. For A* graph version, the $h(n)$ needs to be the increasing function of n along the path to goal, i.e., $h(n) < c(n, a, n`) + h(n`)$.

- The A* tree search is optimal when $h(n)$ is admissible.
- The A* graph search is optimal when $h(n)$ is consistent.

Outline

Heuristic for 8-Puzzle



Start State **Goal State**

What possible heuristic $h(n)$ you can propose?

Two Possible Heuristic for 8-Puzzle

h_1

One possibility can be to count the number of misplaced tiles in the current state *n* than the goal state. For initial state given above $h_1(n) = 6$. Maximum possible $h_1(n) = 8$, since there are eight tiles. Is h_1 admissible?

h_2

Second possibility can be the sum of the distance of tiles in the current state *n* from their goal. As the tiles in this problem can not move diagonal, they can either move horizontal or vertical therefore, this distance will be the count of horizontal and vertical distance/steps. This is also called *Manhattan distance*. The Manhattan distance from 1 to 8 for above problem start state is:

$$h_2(n) = 4 + 0 + 3 + 3 + 1 + 0 + 2 + 1$$

Is h_2 admissible?

Outline

Reading

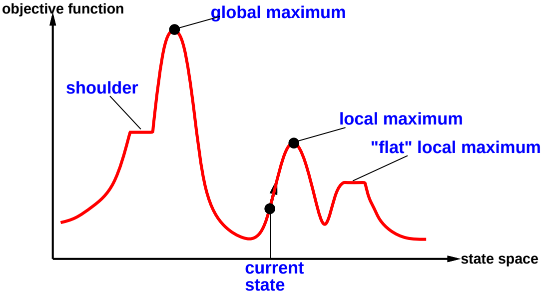
Read about *Recursive best-first search* (RBFS) and try on the Romania map example to develop your understanding.

Outline

Hill-Climbing Search

- An iterative algorithm that tries to find maximum (or minimum) value of an objective/evaluation function.
- In each iteration it moves towards the direction of increasing value.
- Unlike Best-First, it does not maintain a search tree and uses current state node and its immediate neighbors.
- Each time it follows a single node and hence its OPEN list contain only one node.
- It is also sometime called a greedy local search.

Hill-Climbing Search



Problems:

Since hill-climbing search considers immediate neighbor state that is its local, therefore it may end-up with a local optimal solution, i.e., local maximum(or minimum) rather than global optimum solution.

Solution: Random restart sometime solve the problem of local optimum.

Ridges and Plateau

- Ridges: Result in the a series of local optima and it is very difficult for greedy algorithm to overcome.
- Plateau: A plateau is the flat area of the state space with respect to the objective function. It can be a flat local optimum, for which no uphill exist or a *shoulder* where progress is possible.
- The hill-climbing may get lost in plateau, one possible solution can be to allow progress sideways, if plateau is shoulder.
- If plateau is flat local optima then it may result in an infinite loop, so we can then limit the number of sideway moves.

An Example

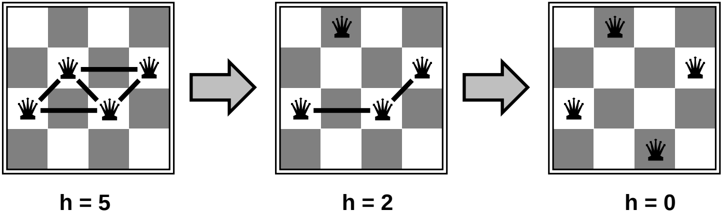


Figure : 4-Queen starting from state with heuristic cost estimate of $h(n) = 4$

- In this problem our heuristic can be the number of conflicts we can observe in the current state.
- Obviously then our goal is $h(n) = 0$.
- The hill-climbing algorithm will often (say 80% of the time for this problem) fails to find the solution.
- But when the algorithm solve the problem, it is very efficient even in very large state space.

Discussion

- The hill-climbing algorithm explained today is incomplete, since it will often lost in local optima.
- Multiple restarts (with random initial state in each restart) may solve the problem.
- If p is the probability of success of hill-climbing algorithm then we need at-least $1/p$ restarts.
- It is also not guaranteed to give optimal solution globally. However, local optimal is guaranteed.
- The time and space complexity both are linear, i.e., $O(d)$ and $O(b)$ respectively. Since the algorithm consider only one state node at a time.

Outline

Readings

You are supposed to read an alternative approach called *Simulated Annealing*, and discuss in a report the characteristics and comparison of hill climbing and simulated annealing algorithm. Your answer should contain at least 100 words.

References I

Stuart Russell and Peter Norvig.
Artificial Intelligence: A Modern Approach.
Pearson Education, Inc, USA, second edition, 2003.

George F. Luger.
Artificial Intelligence: Structures and Strategies for Complex Problem Solving.
Addison-Wesley Publishing Company, USA, 6th edition, 2008.