INFORMED SEARCH AND EXPLORATION

Erika Fernanda Zapata Rendón
230151013

These types of searches are more precise, given that they have a prior knowledge, not only of the problem, but also of the environment in which it develops.

Inspired mainly by statistical physics (simulation) and evolutionary biology (genetic algorithms).

INFORMED SEARCH STRATEGIES (HEURISTICS)

They are search strategies that use the information that defines the problem and the cost of the current status to the objective. They require specific information about the problem.

You can mention search strategies like:

- First the best.
- A * (A star).
- Heuristic with bounded memory.
- Learn to look better.

The performance of heuristic search algorithms depends on the quality of the heuristic function. They have exploration problems when the agent does not have the slightest idea about the states and actions of their environment.

VORACIOUS SEARCH FIRST THE BEST

Or more specifically *seemingly the best search first*. This strategy tries to expand the node closest to the best objective, claiming that it will *probably* lead quickly to a solution.

It is characterized by the depth of following a path to the goal, but returning when it meets a dead end. It is not optimal and it is incomplete.

The node with the smallest value is expanded, estimating the path with the minimum cost, from the initial node to the target node. The structure behaves like a queue with priorities.

Evaluate the nodes using only the heuristic function, which is minimized because it refers to a cost:

$$f(n) = h(n)$$

SEARCH A *

Minimizes the total estimated cost of the solution, evaluating the combined nodes, the cost to reach the node and the cost of going to the target node. Trying first the node with the lowest value.

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

f (n) = Estimated cheapest cost of the solution through n.

g (n) = Cost of the path from the start node to node n.

h (n) = Estimated cost of the cheapest road from n to the target.

If the heuristic function meets the conditions, this strategy is both complete and optimal.

For the heuristic function to be admissible, the real cost must not be overestimated to reach the target node.

If this condition is not met, it becomes simply A, given that it is complete, but not optimal because it is not guaranteed that the result obtained will be the path with the lowest cost.

HEURISTIC SEARCH WITH BOUNDED MEMORY

Or searches with simplified memory.

This type of strategy seeks to expand the nodes in the same way as the search A *, to the point that the available memory is full. Having as objective not to enter more nodes to the tree, which causes that it eliminates the worse leaf. By forgetting the previous nodes, you lose the path by which to go, but the cost of that path is maintained.

LEARN TO LOOK BETTER

Unlike the previous fixed strategies, it works under the *meta-level state space*.

Each state captures the internal (computational) state of a program that searches in a state space at the object level. For example, the internal state of algorithm A * consists of the current search tree.

Each action in the metastate state space is a computation step that changes the internal state, that is, each compute step in A * expands a leaf node and adds its successors to the tree.

This strategy of meta-level states can learn from these experiences to avoid exploring unpromising subtrees.

The objective of learning is to minimize the total cost of solving the problem, offsetting the computational cost and the cost of the road.

HEURISTIC FUNCTIONS

A clear example of heuristics is the 8-puzzle, which aims to slide chips horizontally or vertically into empty space until a specific configuration is reached.

h1 = Number of pieces incorrectly placed. It is an admissible heuristic, because it is clear that any piece that is out of place must move at least once.

h2 = Sum of the distances of the pieces to their positions on the target. This is sometimes called distance in the city or distance from Manhattan. It is an admissible heuristic, because any movement that can be done is to move a piece one step closer to the target.

THE EFFECT OF HEURISTIC PRECISION ON PERFORMANCE

One way to characterize the quality of a heuristic is the b* effective branching factor. If the total number of nodes generated by A* for a particular problem is N, and the depth of the solution is d, then b* is the branch factor that a uniform tree of depth d should have to contain N 1 nodes

$$N + 1 = 1 + b^* + (b^*)^2 + \ldots + (b^*)^d$$

INVENT ADMISSIBLE HEURISTIC FUNCTIONS

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

The heuristic is admissible because the optimal solution in the original problem is, by definition, also a solution in the relaxed problem and therefore it must be at least as expensive as the optimal solution in the relaxed problem.

A problem with the generation of new heuristic functions is that it often fails to achieve a clearly better heuristic. When the best heuristic is not found, the best of all can be used by combining these functions.

$$h = \max. \{h1\ (n), \ldots .hn\ (n)\}$$

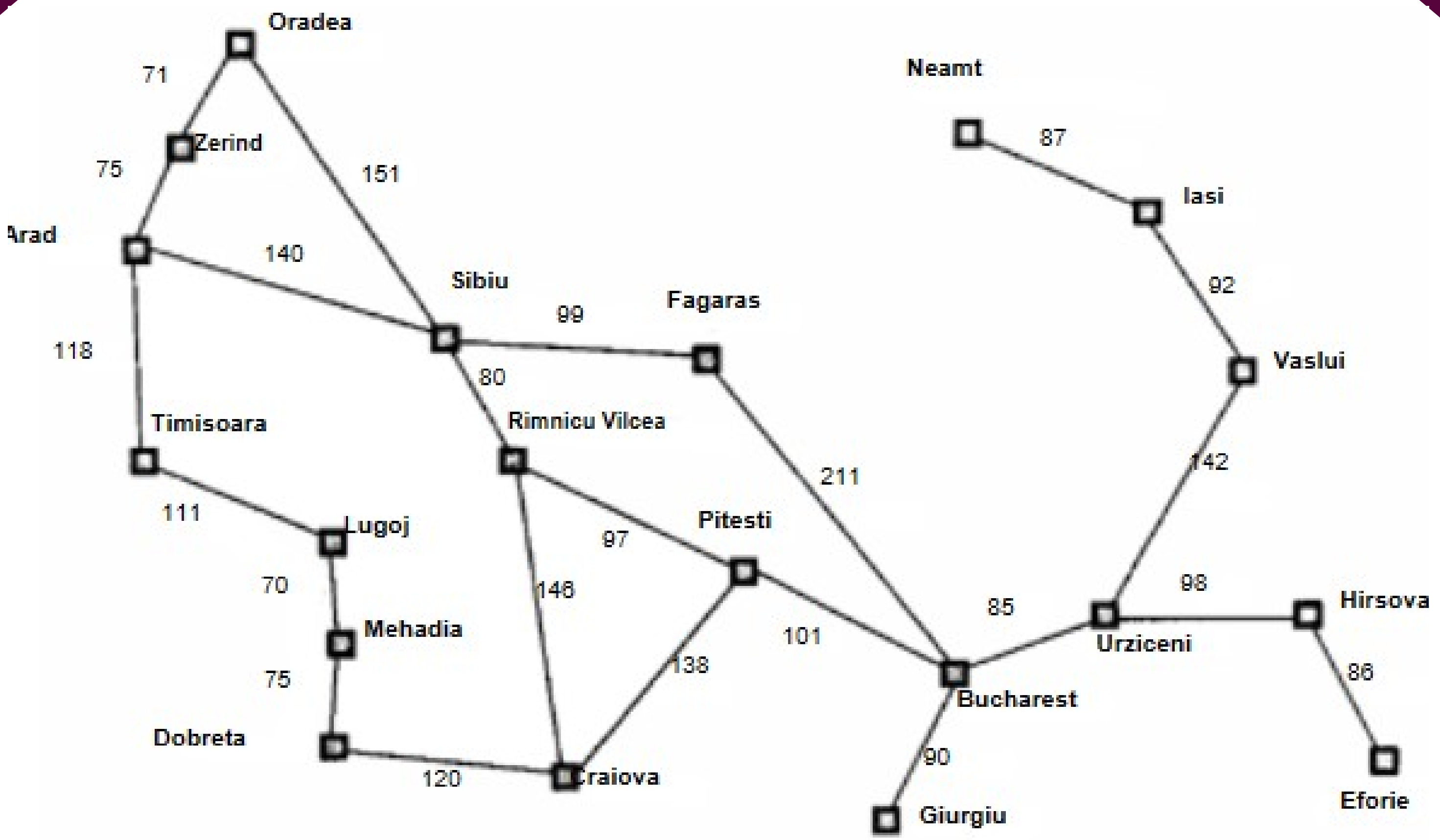
LOCAL SEARCH ALGORITHMS AND OPTIMIZATION PROBLEMS

SIMULATED QUEST SEARCH

It is a hybrid strategy which is the combination of a hill climbing algorithm that never makes downhill movements towards states with a lower value (or higher cost) being incomplete, stagnating in a local maximum and a purely random path, that is, moving to a successor chosen uniformly random from a set of successors, is complete, but extremely inefficient

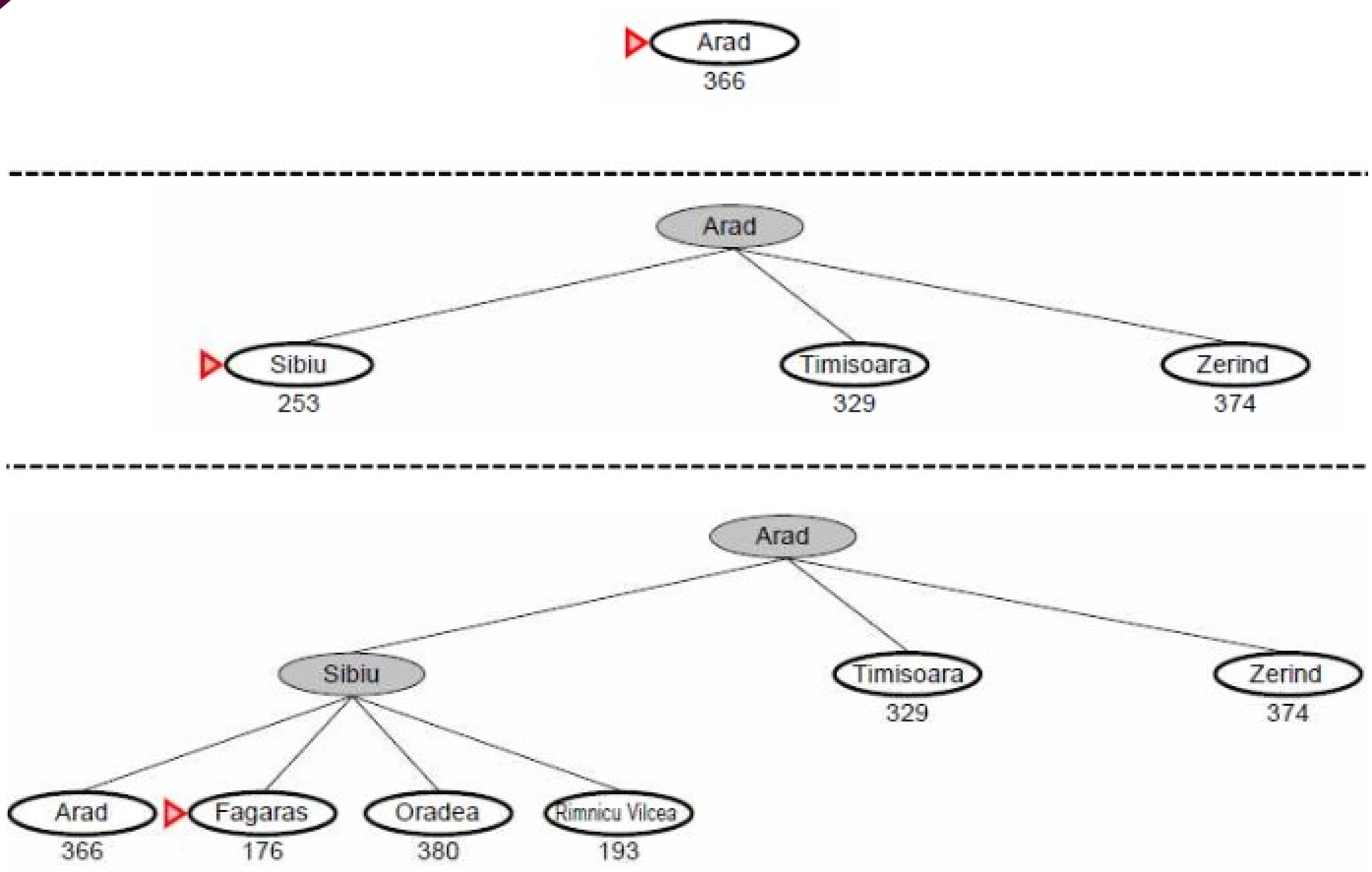
Example

Route in Romania using heuristic distance in a straight line

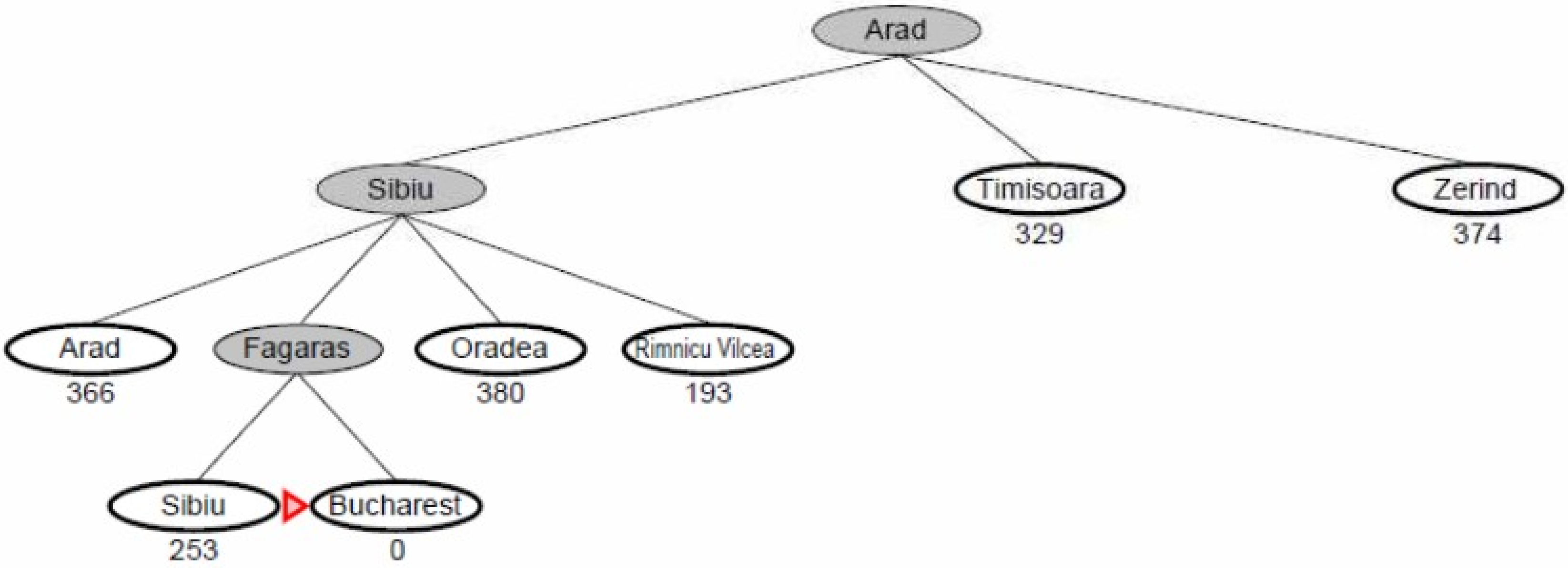

Oradea
71
Zerind
75
151
Arad
140
Sibiu
99
Fagaras
118
80
Rimnicu Vilcea
Timisoara
211
111
Lugoj
97
Pitesti
70
146
Mehadia
101
138
75
Dobreta
120
Craiova
Neamt
87
Iasi
92
Vaslui
142
85
98
Urziceni
Hirsova
86
Bucharest
90
Giurgiu
Eforie

Arad	366	Mehadia	241
Bucarest	0	Neamt	234
Craiova	160	Oradea	380
Dobreta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374

We will consider as function of evaluation (heuristic function) a *hDLR (n)* = Straight Line Distance from city n to Bucharest



Arad
366
Arad
Sibiu
253
Timisoara
329
Zerind
374
Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
176
Oradea
380
Rimnicu Vilcea
193


Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
Oradea
380
Rimnicu Vilcea
193
Sibiu
253
Bucharest
0

Results

Best way to get to Bucharest with the voracious search first the best:

Arad - Sibiu - Fagaras - Bucharest

Total cost: (140 + 99 + 211 + 0) = 450

However the best way is:

Arad – Sibiu – Rimmicu – Pitesti - Bucharest

Total cost: (140 + 80 + 97 + 101 + 0) = 418

BIBLIOGRAPHY

- Stuart Russell, Peter Norvig. *Inteligencia Artificial un Enfoque Moderno,* 2a edición.

- Jefferson Clíder Guilén Valenzuela. *Búsqueda informada y exploración*, Inteligencia Artificial II. Escuela superior politécnica agropecuaría de Manabí, 6 de Diciembre del 2015. [En Línea]

  https://es.slideshare.net/JeffoG92/bsqueda-informada-y-exploracin-55868885