Group id: 1801cs08_1801cs11_1801cs53

Group Members:
1. Ammaar Ahmad - 1801cs08
2. Ayush Pandey- 1801cs11
3. Sumit Raj- 1801cs53

Subject: Artificial Intelligence Lab 1
Date: 02/09/2021

Note: all the python codes are written and tested on google colab.

e. Constraints to be checked:

1. h1= admissible, h2= admissible and h3 = not admissible.

2. h3 = h1*h2 - is not admissible as it gives an estimate higher than the true cost of optimal solution.

3. If blank tile is considered as another tile, then the heuristics h1 and h2 are no longer admissible.
If the blank tile is not considered as another tile and only a blank, then h1 and h2 are admissible.

4. If the search algorithm gets stuck into local optimum, then we can do the following things:

- **(i) If it is *hill-climbing* search, then we can try another variants of it, like, *random restart hill-climbing*. It will increase the probability of finding a global maximum.**
- **(ii) If it is *simulated annealing*, then the algorithm will come out of the local optimum if the temperature is lowered appropriately. It will move to a neighbour with higher energy with some probability if the temperature is high enough. Then, it may find a global maximum.**

Now, let's analyse the feedback of these two algorithms in 8-puzzle problem:

1. Hill Climbing:

a. **the algorithm stops at a local maximum (a plateau) after exploring around 7-14 states, with sideways movements allowed.**

b. **It cannot guarantee to find a solution when it exists.**

c. **There are 9! States in total, but only 8 discrete values for heuristic function h1. So, many states have the same value, and the hill climbing algorithm finds it difficult to navigate through.**

d. **It almost always took a constant amount time before getting stuck on a local maxima. The time complexity depends on how far it is from goal, how quickly it can make uphill moves and the number of sideways movements allowed.**

2. Simulated Annealing:

a. **the algorithm discovers around 4-6 states before getting stuck on a local maximum**
b. **It is less probable that it finds a path to the solution.**
c. **If the temperature is decreased slowly enough, then it will more chances of finding the solution, But the time complexity will increase.**
d. **it is incomplete as it cannot guarantee to find a solution when it exists**
e. **The time complexity depends on the temperature schedule and how quickly the cooling function decreases the temperature to 0.**
f. **if an exponential cooling function is chosen, then the algorithm will finish quickly, and when a linear cooling function is chosen, it will take longer.**