Artificial Intelligence (CS571)

Assignment-4: Hill Climbing and Simulated Annealing

(Read all the instructions carefully & adhere to them.)

Date: 22-08-2019 **Deadline: 29-08-2019**

A. Hill Climbing:

A local search algorithm tries to find the optimal solution by exploring the states in the local region. Hill climbing is a local search technique which always looks for a better solution in its neighbourhood.

a. Implement the **Hill Climbing Search Algorithm** for solving the 8-puzzle problem. Your start state can be anything and the goal state will be {123;456;78B}, where B is blank tile.
b. **Input:** Input should be taken from an input file and processed as a matrix.
c. **Output:** All the following results should be stored in an output file:
    i. The success or failure message,
    ii. Heuristics chosen, Start state and Goal state,
    iii. (Sub)Optimal Path (on success),
    iv. The total number of states explored.
    v. The total amount of time taken.
d. **Heuristics to be checked:**
    i. $h_1(n)$= Number of displaced tiles.
    ii. $h_2(n)$= Total Manhattan distance.
e. **Constraints to be checked:**
    i. Check whether the heuristics are admissible.
    ii. What happens if we make a new heuristics $h_3(n)= 3h_1(n) - 2h_2(n)$.
    iii. What happens if you consider the blank tile as another tile.
    iv. What if the search algorithm got stuck into Local maximum? Is there any way to get out of this?
    v. What happens when all the neighbours of the current state have the same value? How to get out of this situation?

B. Simulated Annealing:

Simulated annealing (SA) is a generic probabilistic metaheuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global minimum of a given function in a large search space.

a. Implement the **Simulated Annealing Search Algorithm** for solving the 8-puzzle problem. Your start and Goal state can be anything desirable.
b. **Input:** Input should be taken from an input file and processed as a matrix. Other inputs are Temperature variable T, heuristic function, neighbourhood generating function, a probability function to decide state change, and a cooling function.
c. **Output:** All the following results should be stored in an output file:
    i. The success or failure message,
    ii. Heuristics chosen, Temperature chosen, cooling function chosen, Start state, and Goal state.
    iii. (Sub)Optimal Path (on success),
    iv. The total number of states explored.
    v. The total amount of time taken.
d. **Heuristics to be checked:**
    i. $h_1(n)$= Number of displaced tiles.
    ii. $h_2(n)$= Total Manhattan distance.
e. **Constraints to be checked:**
    i. Check whether the heuristics are admissible.
    ii. What happens if we make a new heuristics $h_3(n)= 2h_1(n) * h_2(n)$.
    iii. What happens if you consider the blank tile as another tile.
    iv. What if the search algorithm got stuck into Local optimum? Is there any way to get out of this?