Artificial Intelligence (CS571)

Department of CSE, IIT Patna

Assignment-2

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

Date: 21-Sept-2020

Instructions:

1. Markings will be based on the correctness and soundness of the outputs.
2. Marks will be deducted in case of plagiarism.
3. Proper indentation and appropriate comments (if necessary) are mandatory.
4. You should zip all the required files and name the zip file as:
roll_no_of_all_group_members .zip , eg. 1501cs11_1201cs03_1621cs05.zip.
5. Upload your assignment (the zip file) in the following link:

https://www.dropbox.com/request/VFHZ6mqNl8MD5Zrv0cfC

For any queries regarding this assignment contact:

kshitij mishra (kmishra.kings@gmail.com)

Questions

----

1. **Genetic Algorithm:**
   a. Implement the 8 puzzle problem using a genetic algorithm.
      Start state (Can take any random order of numbers with B denoting a blank)
      An Example:

5	B	8
4	2	1
7	3	6

Goal state (fixed):

1	2	3
4	5	6
7	8	B

b. At each step show the following

    i.   Initial population (assume to be 10)

    ii.   Selection (use **Roulette Wheel Selection**)

    iii.   Crossover (high probability value to be chosen, usually above 0.6)

    iv.   Mutation (low probability value to be chosen, usually below 0.2)

    v.   Fitness function: No. of misplaced tiles; Manhattan distance

c. Execute for a sufficient number of generations (or, iterations)

2. 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 **Simulated Annealing Search Algorithm** for solving the 8-puzzle problem. **Your start and Goal state should follow similar guidelines as given in Q.1.a**.

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. Total number of states explored.

        v. Total amount of time taken.

d. Objective functions to be checked:

        i. h1 (n)= Number of displaced titles.

        ii. h2 (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 h3 (n)= h1 (n) * h2 (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?