CS561 - ARTIFICIAL INTELLIGENCE LAB
ASSIGNMENT-2: Hill Climbing and Simulated Annealing

GroupID : 1801cs12_1801cs16_1801cs22
Date : 02/09/2021

1801CS12 : Bablu Kumar
1801CS16 : Mangesh Chandrawanshi
1801CS22 : Hrishabh Raj

Comparison between Hill climbing and Simulated annealing with respect to the time complexity (for near optimal solution) and no of steps

Hill Climbing will **quickly (in a few steps and less time both) find a local maximum/minimum** but the local optimum may not be the global optimum. Once Hill Climbing starts ascending or descending towards the goal there is **no way back**. Hill Climbing is **excellent when there is only a global maximum/minimum** or when the local optimum is quite similar to the local optima.

In the case of Simulated Annealing it updates the current state using a probability given by the actual temperature of the algorithm also if the state is worse than the current state. **At higher temperatures (hot temperature) the algorithm can accept worse solutions**. At lower temperatures (cold temperature) the algorithm accepts solutions which are closer or better to the current solution. This will make it **slower and require a large number of steps** (depending on the initial temperature and cooling function). The **advantage is that we will be able to reach the global optimum in more cases** than Hill Climbing.

Heuristics Notations :

h1 = No. of displaced tiles
h2 = Total Manhattan distance
h3 = h1*h2
h4 = No. of displaced tiles including blank tile
h5 = Total Manhattan distance including blank tile

What happens if we make a new heuristics h3 (n)= h1 (n) * h2 (n) ?

Suppose,

Start State		1 2 3		Goal State
1 2 3	-->	4 5 6	-->	1 2 3
4 5 6		7 0 8		4 0 6
7 8 0				7 5 8

For start state,
h1 = 2
h2 = 2
Then h3 = h1 * h2 = 4 but using two transitions we can get to Goal state i.e. actual cost(2) < h3(4), which implies that h3 is not admissible heuristic for 8-puzzle problem

What happens if you consider the blank tile as another tile?

Again if we take above example,

Start State		1 2 3		Goal State
1 2 3	-->	4 5 6	-->	1 2 3
4 5 6		7 0 8		4 0 6
7 8 0				7 5 8

For start state,
h4 = 3
h5 = 4
But using two transitions we can get to Goal state i.e. actual cost(2) < h4(3) && actual cost(2) < h5(4) , which implies that both h4 and h5 are not admissible heuristics for 8-puzzle problem

Check whether the heuristics are admissible.

So we can conclude that only h1 and h2 are admissible while other 3 (h3, h4, h5 are not admissible)

What if the search algorithm got stuck into Local optimum? Is there any way to get out of this?

Hill Climbing : There is no way to get out of Local Optimum in trivial implementation of Hill Climbing. Though there are techniques which can be introduced to get optimal solution in some cases like :

- **Stochastic hill-climbing** : Random selection among the next moves. Converges slowly than steepest ascent (best first search)

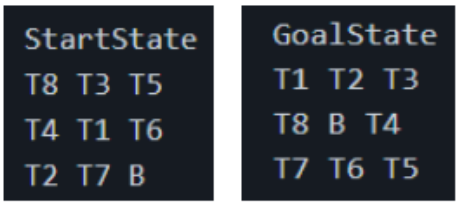
- **First Choice Hill Climbing** : Stochastic hill climbing by generating successors randomly until a better one is found. Good strategy when a state has many successors

- **Random restart Hill Climbing** : Conducts a series of hill-climbing searches from randomly generated initial states and stops when the goal is found. This works great if there are very few local optimas

Simulated Annealing : It is an upgrade over Hill Climbing though it can be very computation heavy if it's tasked with many iterations but it is capable of finding a global maximum and not stuck at local optimum.

Sample Execution for Hill Climbing : (Note : Output is diverted to test_output.txt)

Input States :




```
E:\github\CS571-AI-Lab\Assignment-2 Hill Climbing and Simulated Annealing\Hill Climbing>python main.py >> test_output.txt
```

Output : h1 = displace cost heuristic

```
Running HillClimbing on the puzzle tile configuration

displaceCost heuristics

No path available

Start State :
T8 T3 T5
T4 T1 T6
T2 T7 B

Goal State :
T1 T2 T3
T8 B T4
T7 T6 T5

Total number of states explored before termination : 11
--------------------------------------------------
```

Output : h2 = manhattan cost heuristic

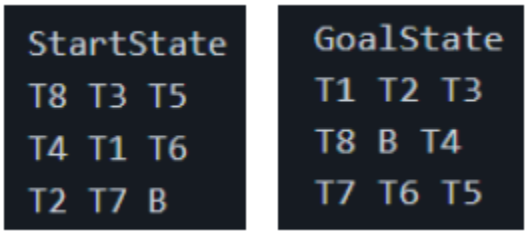
```
manhattanCost heuristics

Path exists

Start State :
T8 T3 T5
T4 T1 T6
T2 T7 B

Goal State :
T1 T2 T3
T8 B T4
T7 T6 T5

Total number of states explored : 14

Total number of states to optimal path :  15

Optimal Path :
T8 T3 T5
T4 T1 T6
T2 T7 B

T8 T3 T5
T4 T1 B
T2 T7 T6
```

……. (contains many intermediate states)

```
B T1 T3
T8 T2 T4
T7 T6 T5

T1 B T3
T8 T2 T4
T7 T6 T5

T1 T2 T3
T8 B T4
T7 T6 T5


Optimal Path Cost : 14

Time Taken For Execution : 0.000982522964477539 seconds
-----------------------------------------------------
```

Sample Execution for Simulated Annealing :
(Note : Output is written to test_output.txt)

Input States :




```
E:\github\CS571-AI-Lab\Assignment-2 Hill Climbing and Simulated Annealing\Simulate Annealing>python main.py

Running SimulatedAnnealing on the puzzle tile configuration

Enter choice for heuristic
0. Displced tiles Heuristic.
1. Manhattan distance Heuristic.
2. Displaced tile heuristic with blank tile cost included
3. Manhattan distance heuristic with blank tile cost included
4. Manhattan and displaced tile combined heuristic
: 1

Enter the max temperature : 100000

Enter Cooling Function Choice
1. maxTemperature*(0.9**iteration)
2. maxTemperature/iteration
: 2

E:\github\CS571-AI-Lab\Assignment-2 Hill Climbing and Simulated Annealing\Simulate Annealing>
```

```
manhattanCost heuristics

Path exists

Start State :
T8 T3 T5
T4 T1 T6
T2 T7 B

Goal State :
T1 T2 T3
T8 B T4
T7 T6 T5
```

```
Max temperature : 100000.0

Cooling Function : 2. maxTemperature/iteration

Total number of states explored : 105018

Total number of states to optimal path :  105019

Optimal Path :
T8 T3 T5
T4 T1 T6
T2 T7 B

T8 T3 T5
T4 T1 T6
T2 B T7
```

……. (contains many intermediate states)

```
 T1 T2 T3
 T8 T4 B
 T7 T6 T5

 T1 T2 T3
 T8 B T4
 T7 T6 T5


 Optimal Path Cost : 105018

 Time Taken For Execution : 5.77261757850647 seconds
 -----------------------------------------------------
```