



Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.
CS482:Artificial Intelligence

Using Local Search Algorithms to Solve 8-Queens
Assignment(2)
Report

Name : Fatma Mohamed Abd El-Aty.

Id : 43.

1. Overall used algorithms :

a. Hill Climbing Algorithm :

- i. Start with initial state, calculate the cost_function and the best_neighbour(select one of the neighbours with minimum cost_function).
- ii. Then start again with this neighbour to get the best neighbour.
- iii. When stuck in shoulder, Restart again with random state.

b. K-Beam Search Algorithm :

- i. Start with initial state, calculate the cost_function and the best_neighbours(select K of the neighbours with minimum cost_function).
- ii. Then start again with one of the neighbours to get the best neighbours.
- iii. Stop when find state with cost_f = 0.

c. Genetic Algorithm :

- i. Start with 50 random states, calculate their cost_function and put them into a priority queue.
- ii. Select randomly 2 states and make the cross_over between them , then put the 4 states into the priority queue.
- iii. until find the top state with cost_function = 0.

d. Constraint Satisfaction Problem :

- i. From reference :

```
function MIN-CONFLICTS(csp, max_steps) returns a solution or failure
inputs: csp, a constraint satisfaction problem
         max_steps, the number of steps allowed before giving up

current  $\leftarrow$  an initial complete assignment for csp
for i = 1 to max_steps do
    if current is a solution for csp then return current
    var  $\leftarrow$  a randomly chosen conflicted variable from csp.VARIABLES
    value  $\leftarrow$  the value v for var that minimizes CONFLICTS(var, v, current, csp)
    set var = value in current
return failure
```

2. Data structures :

- a. Priority queues.
- b. 2D arrays.
- c. 1D arrays.
- d. state class has the following attributes:
 - i. 2D array \rightarrow the position of the queens.
 - ii. int \rightarrow the cost_function.
 - iii. int \rightarrow the depth(actual_cost).
- e. vectors<state>.

3. Sample run :

```
Opening the file
Hill_climbing :
time 314 ms
Cost : 5
no_expanded_nodes = 15942
# # # # Q # # #
# # # # # # # Q
# # # Q # # # #
Q # # # # # # #
# # # # # # Q #
# Q # # # # # #
# # # # # Q # #
# # Q # # # # #

Beam_Search :
time 3 ms
Best_K : 6
Cost : 45
no_expanded_nodes = 53
# # # Q # # # #
# # # # # # Q #
Q # # # # # # #
# # # # # # # Q
# # # # Q # # #
# Q # # # # # #
# # # # # Q # #
# # Q # # # # #

GA :
time 41 ms
Cost : 4
no_expanded_nodes = 20386
# # Q # # # # #
# # # # # Q # #
# # # Q # # # #
# Q # # # # # #
# # # # # # Q
# # # # Q # # #
# # # # # Q #
Q # # # # # #
```

4. How to run your code from terminal :

Go to the path of the project and type the following two commands :

```
$ g++ main.cpp -o main.out  
$ ./main.out
```