DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING

COLLEGE OF E&ME, NUST, RAWALPINDI

EC-350 Artificial Intelligence and Decision Support System

LAB MANUAL – 07

Course Instructor: Assoc Prof Dr Arsalan Shaukat

Lab Engineer: Kashaf Raheem

Student Name: ____

Degree/ Syndicate: ____

LAB # 7: GENETIC ALGORITHM

Lab Objective:

- To implement genetic algorithm in python

Hardware/Software required:

Hardware: Desktop/ Notebook Computer
Software Tool: Python 3.10.0

Lab Description:

Genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection. The basic structure of a GA is shown in Figure 1:

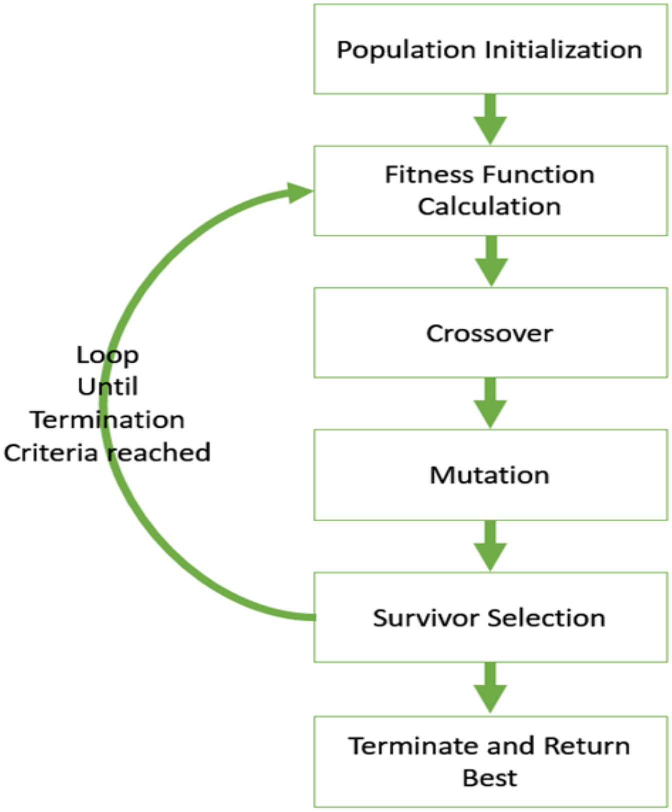

Population Initialization
Fitness Function Calculation
Crossover
Mutation
Survivor Selection
Terminate and Return Best
Loop Until Termination Criteria reached

Figure 1: Genetic Algorithm

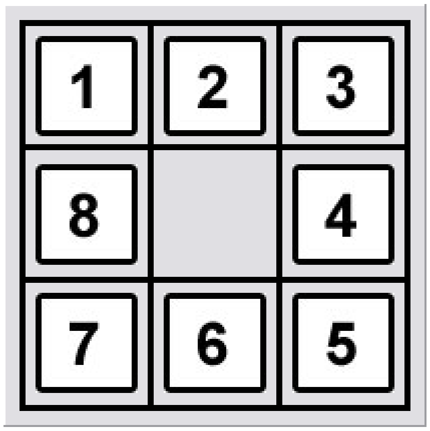
GA starts with initializing population (which may be generated at random or seeded by other heuristics), select parents from this population for mating. Apply crossover and mutation operators on the parents to generate new off-springs. And finally, these off-springs replace the existing individuals in the population and the process repeats. In this way, genetic algorithms try to mimic the human evolution to some extent.

Lab Tasks:

8-Slider Puzzle

An 8-slider puzzle is a combinational puzzle that challenges a player to slide pieces along certain routes to establish a certain end-configuration. The pieces to be moved may consist of simple shapes, or they may be imprinted with colors, patterns, sections of a larger picture, numbers, or letters. One of such example is shown below:

[0,0]	[0,1]	[0,2]
[1,0]	[1,1]	[1,2]
[2,0]	[2,1]	[2,2]



The optimizations through genetic algorithm can be used to solve 8-slider puzzle. The details about fitness, crossover and mutation are:

Chromosomes: 8 moves consisting of ‘Up’, ‘Down’, ‘Left’ and ‘Right’.

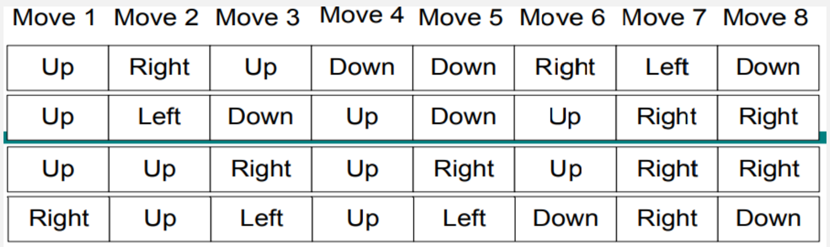
Fitness: Total number of correct moves for each chromosome.

Crossover: Crossover exchange moves between adjacent chromosomes after determining maximum fitness value.

Mutation: Change the first problematic move in chromosome with least fitness. E.g. if the move is ‘left’, then change is to ‘right’ and vice versa. If the move is ‘up’, then change it to ‘down’ and vice versa.

The pseudocode for GA is given below. Your job is to use GA to solve 8-slider puzzle.

Initial chromosomes:



Move 1	Move 2	Move 3	Move 4	Move 5	Move 6	Move 7	Move 8
Up	Right	Up	Down	Down	Right	Left	Down
Up	Left	Down	Up	Down	Up	Right	Right
Up	Up	Right	Up	Right	Up	Right	Right
Right	Up	Left	Up	Left	Down	Right	Down

```
GA():

    Initialize chromosomes ch1, ch2, ch3 and ch4 as shown above

    Initialize coordinates of blank space [1,1]

    Compute fitness for all chromosomes and store them in f1, f2, f3 and f4

    For I in range(8):

        Move=ch1[i]

        If move=up:

            If blank_space[0]>0:

                F1=f1+1

                Blank_space[0]=blank_spac[0]-1

        If move=down:

            If blank_space[0]<2:

                F1=f1+1

                Blank_space[0]=blank_space[0]+1

    while fitness of all chromosomes is less than 8

        if f1 != 8
            crossover ch1 with ch2  ch1[0:f1] + ch2[f1:8]
        end
        if f2 != 8
            crossover ch2 with ch1
        end
        if f3 != 8
            crossover ch3 with ch4
        end
        if f4 != 8
            crossover ch4 with ch3
        end
```

- Compute fitness for all chromosomes and store them in f1, f2, f3 and f4.
- Mutate chromosome with least fitness value.
- Re-compute fitness for all chromosomes and store them in f1, f2, f3 and f4 return all chromosome pairs.