

MACHINE INTELLIGENCE AND EXPERT SYSTEMS

AUTUMN SEMESTER - 2021

COMPUTER ASSIGNMENT - Genetic Algorithm

Question : -

Question statement :

Program a complete genetic algorithm using PYTHON to maximize the number of 1's in a bit string of length 20. Thus our chromosomes will be binary strings of length 20, and the optimal chromosome that we are searching for is [11111111111111111111] .

Start with an initial population size of 200 chromosomes, select the best 100 according to the fitness score. From those 100 chromosomes make 50 pairs and do one-point crossover as well as uniform crossover for every pair to get a population of 200 again. Implant mutations in the resultant chromosomes with the given bit mutation rate.

Repeat the above procedure for the given number of iterations.

Given :

fitness function = sum of the bits in the chromosomes

Size of the population = 200 chromosomes

Bit mutation rate = 0.01

Number of iterations (generations) = 300

Location of crossover should be random for one-point crossover

Implementation details :

Make sure that your code has the following functions -

a) cal_fitness_score(string)

This takes a string as input and returns its corresponding fitness score

b) generate_chromosome()

This returns a randomly generated 20 bit string

c) `crossover(string1,string2)`

This takes 2 strings to be mated as inputs and perform uniform as well as one point crossover between them thus returning 4 child chromosomes

d) `mutate_chromosome(string)`

This takes a string as input and returns another string after mutating the input with the given bit mutation rate

Plot :

(your code must plot the following with clear labels)

Average fitness vs generation

Highest fitness vs generation

Report/print fittest chromosome for each generation

Report/print final fittest chromosome

Also briefly discuss the above results.