Roll No. : 41310
Name : Prem Vinod Bansod
Assignment No.: 02 (SCOA)

Problem Statement :

Implement genetic algorithm for benchmark function (eg. Square, Rosenbrock function etc) Initialize the population from the Standard Normal Distribution. Evaluate the fitness of all its individuals. Then you will do multiple generation of a genetic algorithm. A generation consists of applying selection, crossover, mutation, and replacement. Use:

- Tournament selection without replacement with tournament size s
- One point crossover with probability Pc
- bit-flip mutation with probability Pm
- use full replacement strategy

Objective :

- To familiarize with Mathematical foundations for Genetic algorithm, operator.
- To study the Applications of Genetic Algorithms.

Software and Hardware Requirement :

32/64 bit PC

Python

Theory :

Genetic Algorithm :

- Genetic algorithm is a search technique used in computing to find true or approximate solutions to approximate solutions to optimization & search problems.
- Genetic algorithms are inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved.
- Algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population.

- This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce.
- This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.

Outline of the Basic Genetic Algorithm

1. [Start] Generate random population of n chromosomes (suitable solutions for the problem)

2. [Fitness] Evaluate the fitness f(x) of each chromosome x in the population

3. [New population] Create a new population by repeating following steps until the new

population is complete

4. [Selection] Select two parent chromosomes from a population according to their fitness (the

better fitness, the bigger chance to be selected)

5. [Crossover] With a crossover probability cross over the parents to form a new offspring

(children). If no crossover was performed, offspring is an exact copy of parents.

6. [Mutation] With a mutation probability mutate new offspring at each locus (position in chromosome).

7. [Accepting] Place new offspring in a new population

8. [Replace] Use new generated population for a further run of algorithm

9. [Test] If the end condition is satisfied, stop, and return the best solution in current population

10. [Loop] Go to step 2

Conclusion :

The match word finding algorithm is implemented using the genetic algorithms which include all the genetic algorithm operators. Genetic algorithm includes the selection, crossover, mutation operators along with fitness function.