HERITAGE INSTITUTE OF TECHNOLOGY

Implementation of Genetic Algorithm for function minimization.

Authors:

UPASANA DUTTA (1551114)

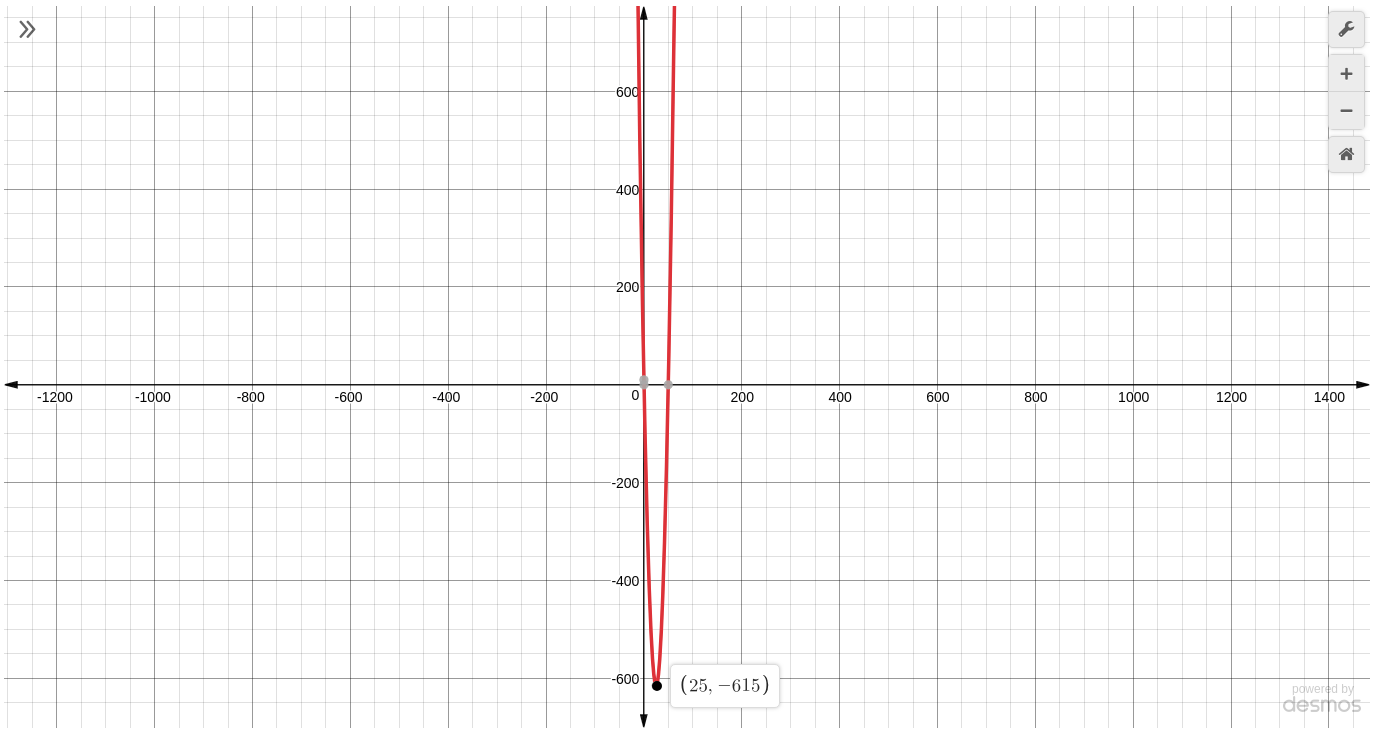
ARIJIT DUTTA (1551131)

Objective -

The objective of this project is to implement a Genetic Algorithm using Python programming language to minimize a function with single variable.

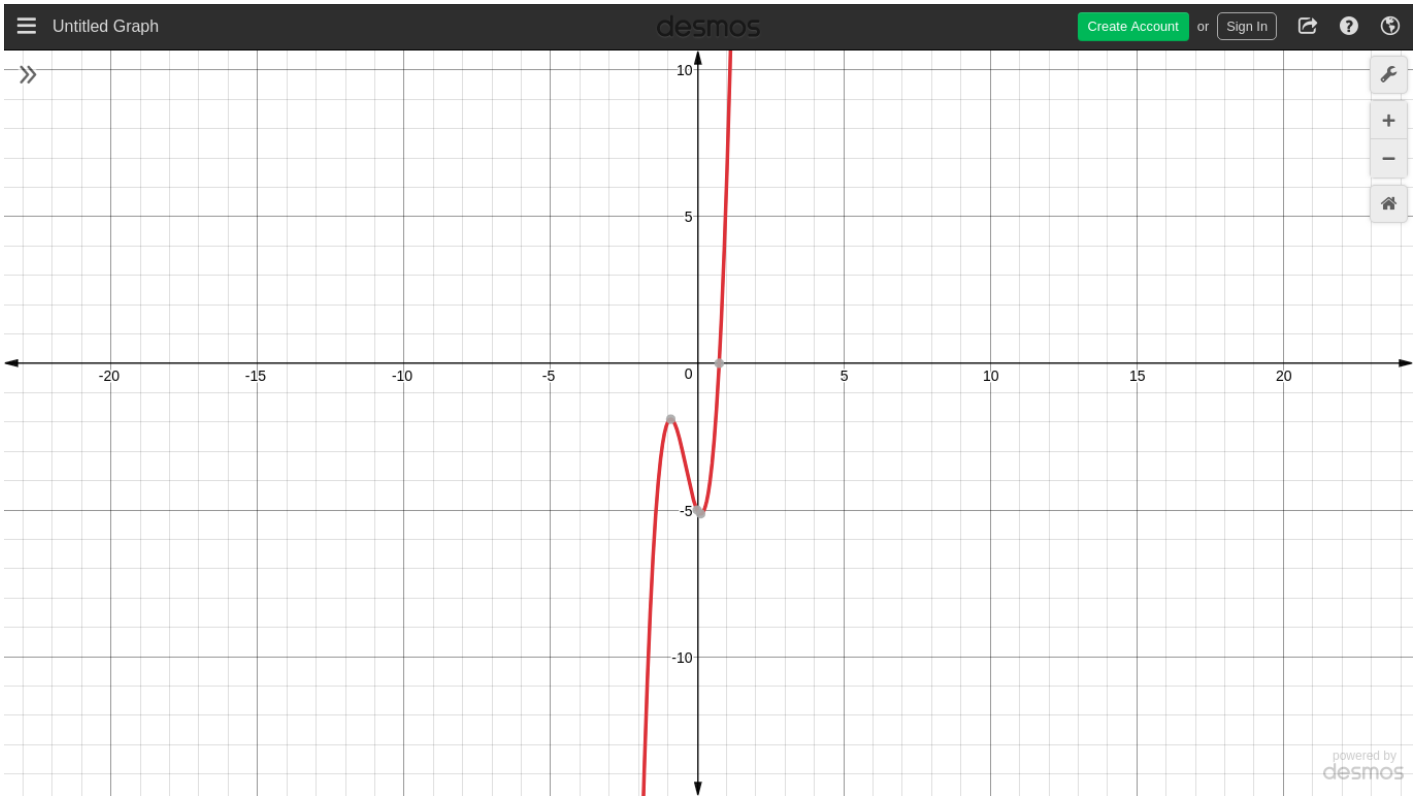
Functions used -

- $f(x) = x^2 - 50x + 10$



$f(x_{min}) = -615$ where $x_{min} = 25$

- $f(x) = 6x^3 + 7x^2 - 2x - 5$



$f(x_{min}) = -1.6889E+15$ where $x_{min} = -65536$

Introduction -

A genetic algorithm applies the biological principles of natural evolution in the artificial systems. It is an iterative procedure that includes a population of individuals, each of them represented by a finite string of symbols known as genes, which encode a possible solution in the space of a given problem.

- **Population** − It is a subset of all the possible (encoded) solutions to the given problem.
- **Chromosomes** − A chromosome is one such solution to the given problem.
- **Gene** − A gene is one element position of a chromosome.
- **Allele** − It is the value a gene takes for a particular chromosome.

- **Genotype** − Genotype is the population in the computation space. In the computation space, the solutions are represented in a way which can be easily understood and manipulated using a computing system.

- **Fitness Function** − A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output.

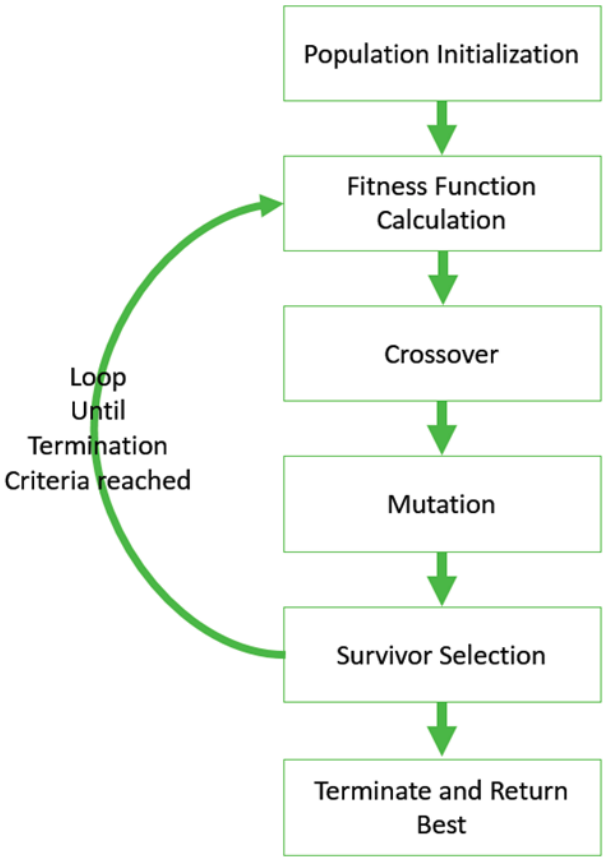


Figure 1. Block schema of a simple genetic algorithm

Theory -

The usage of GA requires the definition of six fundamental steps:

- Solution representation.
- Creation of the initial population.
- Evaluation function.
- Termination criteria.
- Selection function.
- Genetic operators.

A. Solution Representation / Encoding -

Most commonly used encodings are -

- **Binary Encoding** - In this type of representation the genotype consists of bit strings, i.e strings of 0s and 1s.

- **Real Value Encoding** – The genes use continuous values rather than discrete values.

- **Permutation Encoding** – The genotype is represented by an order of some numbers.

We have used signed integers as our phenotype and binary encoding to generate our genotype. We have used 16 bits strings.

Example:	Chromosome A	1011001011001010
	Chromosome B	1111111000001100

B. Initialization -

GA should generate a population to start with. We have randomnly generated 100 signed integers within our range

- **Range of x** : - 65536 to + 65536

- **Population size** : 500

C. Fitness function -

The fitness function simply defined is a function which takes a candidate solution to the problem as input and produces as output how "fit" our how "good" the solution is with respect to the problem in consideration.

We have to find the value of **x** where the function **f(x)** is minimum. The fitness function in our case can be defined as the reciprocal of the value of **f(x)** for some particular **x** in the given range.

Fitness function=1 / f(x)

Smaller the value of **f(x)** greater is the value of **fitness function** and higher the chance for that corresponding **x** to get included in the population.

D. Termination Criteria -

Usually, we keep one of the following termination conditions −

- When there has been no improvement in the population for X iterations.
- When we reach an absolute number of generations.
- When the objective function value has reached a certain pre-defined value.

In our case we have limited the number of iterations to 2000. Increasing the number of iterations may help in bringing better chromosomes into the population and thus improving the result.

E. Selection Function -

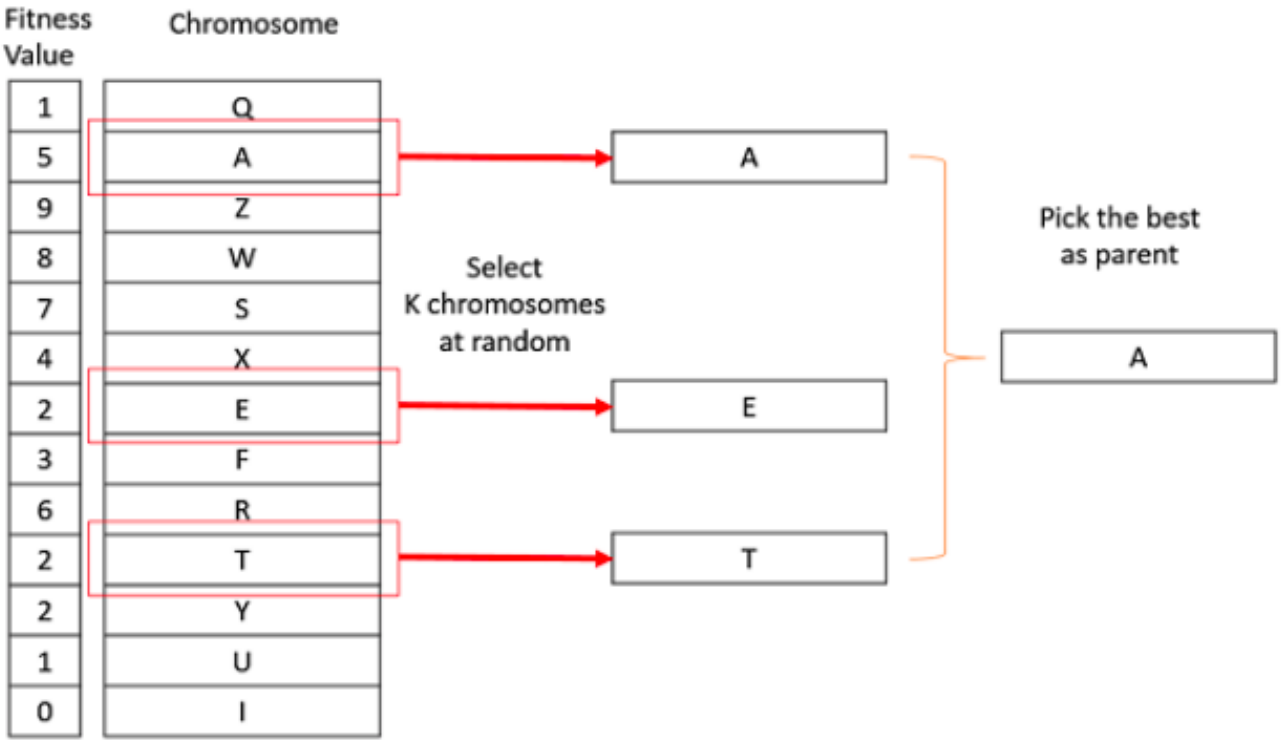
Parent Selection is the process of selecting parents which mate and recombine to create off-springs for the next generation.

Maintaining good diversity in the population is extremely crucial for the success of a GA. This taking up of the entire population by one extremely fit solution is known as premature convergence and is an undesirable condition in a GA.

Types of Parent Selection Methods -

- Roulette Wheel Selection.
- Tournament Selection.
- Rank Selection.
- Random Selection.

We have used **Tournament Selection** in our code as it is very popular in literature and it can work with negative fitness values.



In K-Way tournament selection, we select K individuals from the population at random and select the best out of these to become a parent. The same process is repeated for selecting the next parent.

We have taken K=2 in our code.

F. Genetic operators -

There are two basic types of operators :

- **Crossover** - In a crossover, more than one parent is selected and one or more off-springs are produced using the genetic material of the parents.

- **Mutation** - Mutation may be defined as a small random tweak in the chromosome, to get a new solution.

Crossover -

It is a process that recombines the genetic material of the parent individuals to create the offspring of the next generation.

We have used **Arithmetic Crossover** in our coe. Arithmetic crossover using the AND performs bitwise AND between the two parents and gives the offspring. An example of which is given below.

1111000010100101 + 1010010100001111 = 1010000000000101 **(AND)**

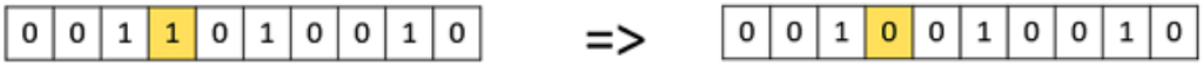
Mutation -

- Mutation is done to prevent the GA from falling into local optima.

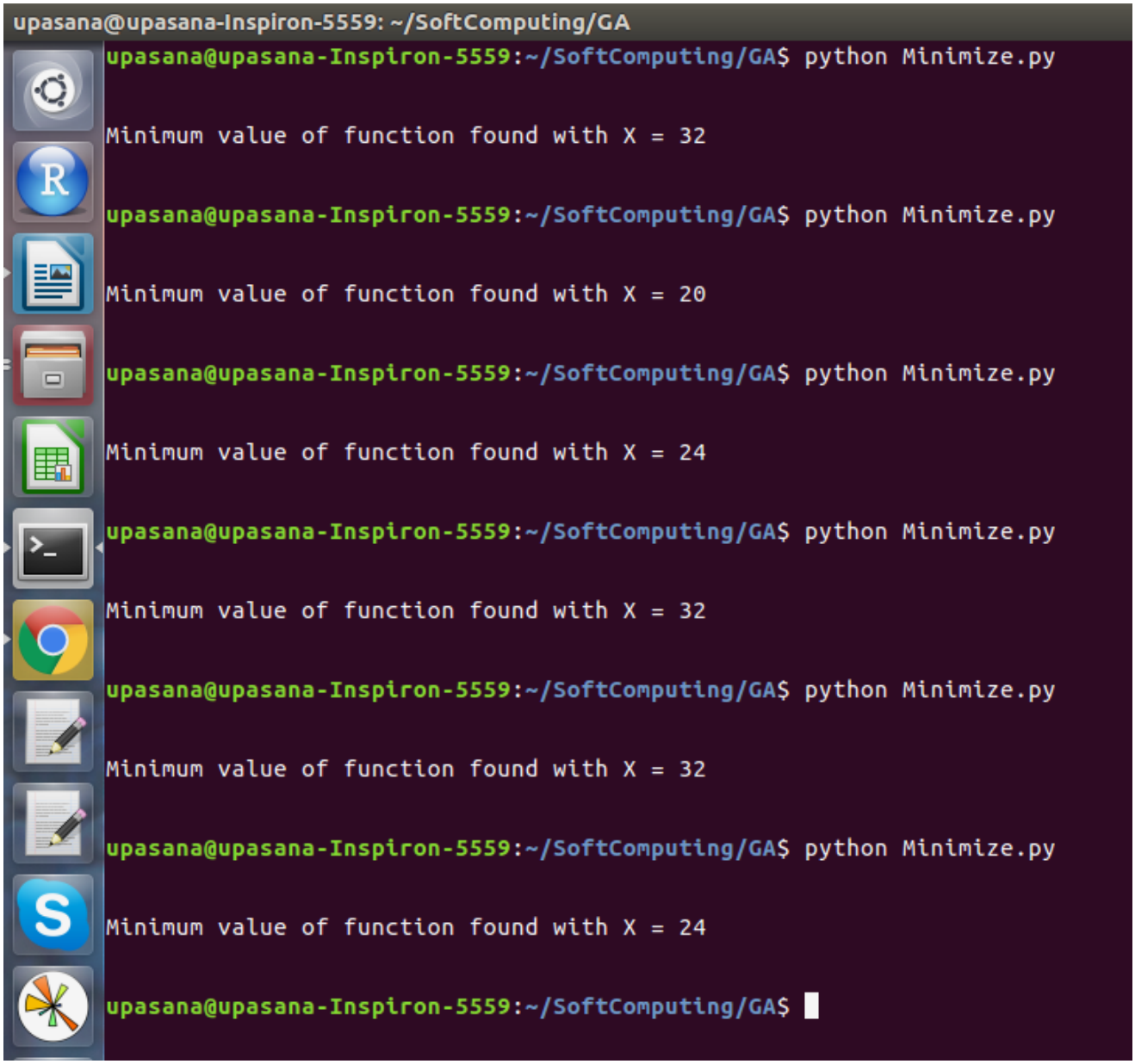
- In our code we have used a **mutation probability of 1%**

- It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability.

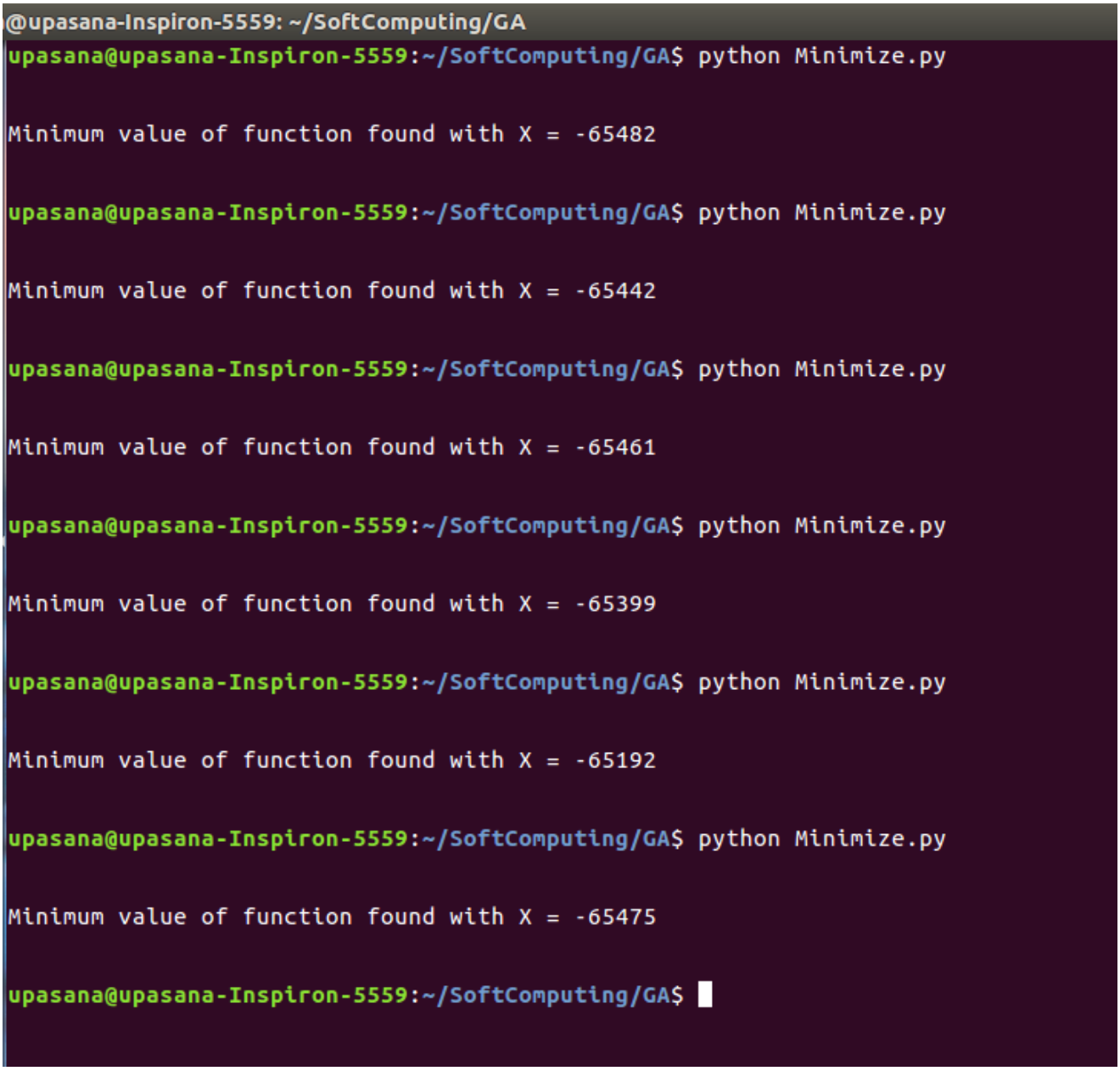
- We have used **Bit Flip Mutation** - In this bit flip mutation, we select one or more random bits and flip them.



Results -



$$f(x) = x^2 - 50x + 10$$



$$f(x) = 6x^3 + 7x^2 - 2x - 5$$

Observations -

- Increasing the number of iterations increases the accuracy of the algorithm.
- Genetic Algorithm does not gurantee the correct answer as it is subject to randomness and the best answer may not be found within limited number of iterations.