Applications of the Simple Genetic Algorithm

Neekesh Panchal, Harshul Mehta, Adam Serrao

Physics and Computer Science Department, Wilfrid Laurier University, Fall 2022 Term

Objectives

The primary goal of this project was to successfully create and implement a simple genetic algorithm towards efficiently optimizing the benchmark OFs provided. The three benchmark OFs included are listed as:

- De Jong Sphere Function
- Rosenbrock's Valley
- Himmelblau's Function

Introduction

A Simple Genetic Algorithm follows the steps of selection, crossover, and mutation, to simulate Darwin's Theory of Evolution. This algorithm denotes the fitness of each member of a population and allocates a specific probability which depicts whether or not the member of the population will retain a influence on the genetic code of the next generation.

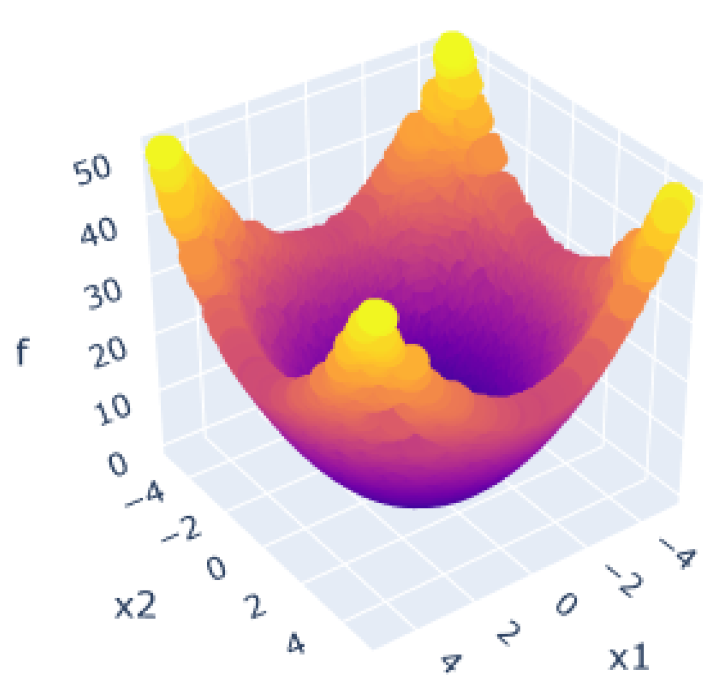

Figure 1:Three Dimensional Rendering of the De Jong Sphere Function (Natural Selection)

Procedure

We implemented three chronological phases of the Simple Genetic Algorithm:

- (1) Selection (or Reproduction)
- (2) Crossover
- (3) Mutation

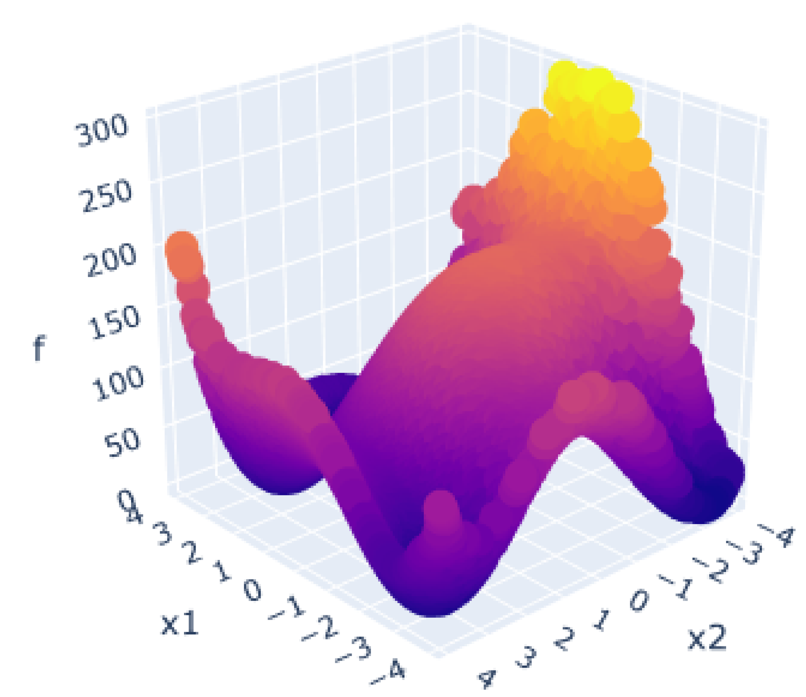

Figure 2:Three Dimensional Rendering of Himmelblau's Function (Natural Selection)

Phase Descriptions

- The **Selection** step consists of an input of the previous population of organisms and outputs a novel population that has been genetically influenced by the previous generation through a biased roulette.
- The **Crossover** phase consists of an input of the previous step's output. The output of this step is the result of combinations based on chance of the various organisms in the current population (CR=0.45). This step can also be referred to as the "breeding" step when considering semantics.
- The **Mutation** step is the final part of this three step process. The input of this step is the previous steps output. The output of this step is the current population with minimal alterations aside from specific organisms retaining variation in their genetic code. This is achieved through iterating through the population and picking organisms to be mutated based on a declared mutation rate (MR=0.045).

Optimizations

We applied two methods of optimization in order to efficiently reach optimal solutions:

- **Natural Selection**: Assigns a probability in respect to the fitness of each organism.
- **Elitism**: Assigns a probability in respect to the fitness of each organism *except* the highest fitness member of the population. The highest fitness member instead is guaranteed to genetically influence member(s) of the next generation.

Objective Function Formulae

De Jong Sphere Function

$$f(x) = \sum_{i=1}^{n} x_i^2$$

Himmelblau's Function

$$f(x) = f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$

Rosenbrock's Valley

$$f(x) = \sum n - 1^{i=1}[100(xi + 1 - x_i^2)^2 + (1 - x_i)^2]$$

Results

Treatments	DNA	Fitness
De Jong Sphere	001001001001000000000000	0.0015999...
Himmelblau's	111000000101101111110110	0.0061543...
Rosenbrock's	110010100010110011101011	0.0018550...

Table 1:Simple Genetic Algorithm Outputs

Conclusion

The exploration of this topic can be considered a success as we were able to provide evidence of our implementation reaching optimal points within each one of the benchmark objective functions provided. The three dimensional visualizations of each function depict the minimal point in each instance, implying that our simple genetic algorithm was able to perform efficiently within the confines of this demonstration.

Benchmark Objective Functions

Benchmark Objective Functions applied in our demonstration:

- De Jong Sphere Function (Easy)
- Himmelblau's Function (Medium)
- Rosenbrock's Valley (Hard)

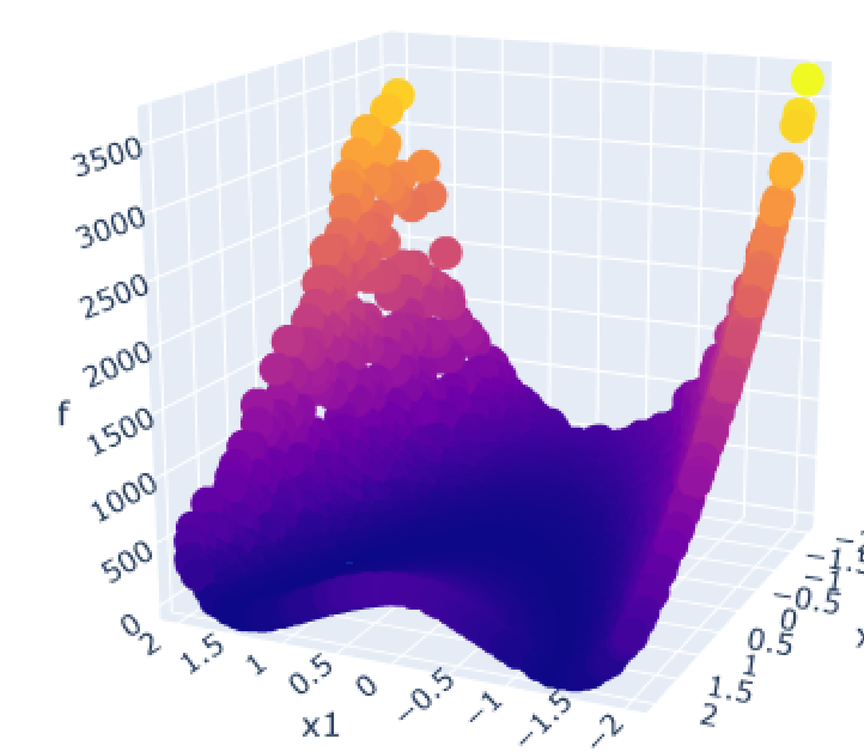

Figure 3:Three Dimensional Rendering of Rosenbrock's Valley (Natural Selection)

Acknowledgements

This Term Project is prepared for the CP468 - Artifical Intelligence course instructed by Dr. Ilias S. Kotsireas.