Evolutionary Genetic Algorithm for Template matching in python

Prepared for

Dr. Junaid Akhtar

NAMAL INSTITUTE MIANWALI

Prepared by

Muhammad Omair

BSCS2019-58

Computer science (3rd year)

November 29, 20--

Acknowledgment:

I am over helmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete. I would like to express my special thanks of gratitude to my teacher Dr. junaid Akhtar who gave me the golden opportunity to do this wonderful project on the topic GENETIC ALGORITHM, which also helped me in doing a lot of Research and I came to know about so many new things.

Table of contents:

Contents

Abstract:

A genetic algorithm is a search-based technique used in artificial intelligence to solve optimization problems. This method is useful because it solves tough issues that would otherwise take a very long time to solve. It's been employed in a variety of real-world applications, including datacenters, electronic circuit design, code-breaking, image processing, and artificial creativity.

This paper will walk the reader through the fundamentals of this algorithm and explain how it functions. It also describes how it has been used in template matching. There are many algorithms for template matching in the world but they are slow and inefficient. So in order to solve this problem, a fast and optimized template matching using genetic algorithm is proposed in this paper. We will also check the optimization of our algorithm by doing some experiments.

The Actual problem:

So the main problem was that we have given two images. One of them was large in size while the other one was small. The large image was named “groupGray” and the small image was named “boothiGray”. Our task was to find small image in the large image using genetic algorithm.

Introduction:

A genetic algorithm (GA) is a type of evolutionary algorithm that is inspired by the process of natural selection (EA). Genetic algorithms, which depend on biologically inspired operators, are often employed to develop high-quality solutions to optimization and search issues. (Wikipedia). I will discuss natural selection in a while.

So there was a scientist name Charles Darwin who formulated an evolutionary theory about how organisms evolve over generations through the inheritance of physical or behavioral traits.

What is evolution?

Evolution in biology is the change in a species' features over numerous generations that is based on the process of natural selection. The Darwin’s theory of evolution is built on the principle that all species are connected and change over time. Evolution is based on genetic diversity in a population, which changes an organism's physical features (phenotype). Some of these features may provide the individual with an edge over others, which they can then pass on to their children.

What is natural selection?

According to Charles Darwin's theory of evolution, evolution occurs through natural selection. Individuals within a species differ in their physical traits. This variance is due to genetic differences. Species who have features that are best adapted to their environment are more likely to live, locate food, avoid predators, and resist sickness. These species are much more likely to reproduce and pass on their genes to the next generation. On the other hand, species that are poorly suited to their environments have a lower chance of surviving and reproducing. As a result, their genes are less likely to transmit on to future generations. As a result, only those species that are most suited to their environment survive, and the species evolves over time.

Example of natural Selection: (Peppered moth)

The peppered moth was most typically a pale yellowish color with black markings until the industrial revolution in the mid-1700s. They could hide from predators on trees with pale-colored bark, such as birch trees, because of their coloring. The uncommon dark-colored peppered moths contrast well with the light bark of trees, making them more visible to predators.

On an oak tree, a pale peppered moth.

Image credit: Shutter Stock

The air in industrial districts became thick with soot as the Industrial Revolution grew to its peak. Trees and buildings became dark as a result of this. As a result, lighter moths became far more visible than darker moths, making them more vulnerable to bird attacks. The darker moths were now less likely to be eaten since they were concealed against the soot-stained trees. The darker moths became more frequent as the environment changed, whereas the pale ones became uncommon.

On a dark tree, a pale peppered moth

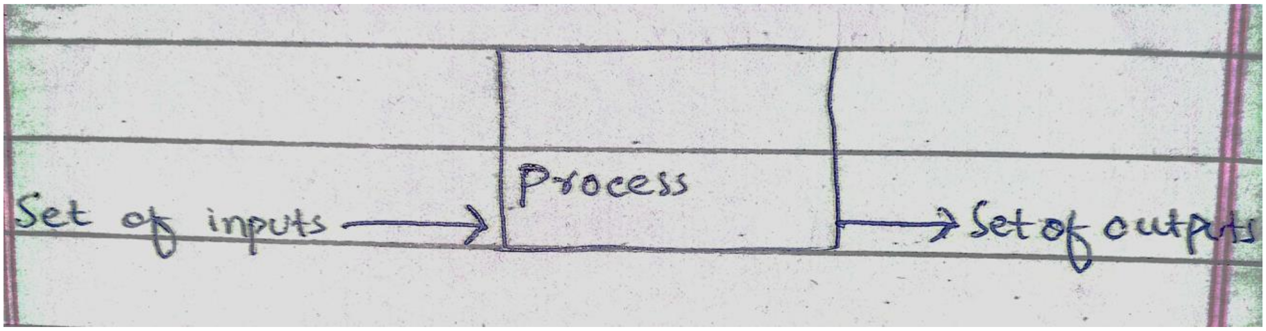
Image credit: Shutter Stock

Genetic algorithm:

Now that we discussed about Darwin's theory, we will see how genetic algorithms are inspired by this theory. So we all know that genetic Algorithm (GA) is a search-based optimization approach based on genetic and natural selection principles. It is commonly used to find optimum or near-optimal solutions to tough problems that would otherwise take a lifetime to solve. It is commonly used to tackle optimization issues, as well as Artificial intelligence. The question is what is optimization?

Optimization:

The process of making anything better is known as optimization. In any process, we have a set of inputs and a set of outputs, as illustrated in the picture below.



The term "optimization" refers to the process of determining the values of inputs in order to obtain the "optimal" output values. The term "best" varies depending on the task. The search space is the collection of all potential solutions or values that the inputs can take. In this search space, there is a point or combination of points that provides the best solution. The goal of optimization is to locate that single point or combination of points in the search space.

Terminologies:

Here are some of the important terminologies that are used in genetic algorithm.

- Initialize population
- Fitness Values
- Selection
- Crossover
- Mutation
- Termination criteria

I will discuss these one by one:

Initialize Population:

The genetic algorithm begins by creating an initialization of population. This initial population is made up of all possible solutions to the given problem. The usage of random binary strings is the most frequent approach for initialization. In our problem, I have used 100 initial population. Which means 100 random points will be selected for our solution.

Fitness Values:

The fitness function aids in determining the fitness of the entire population. It assigns a fitness score to each individual, which impacts the likelihood of being selected for reproduction. The greater the fitness score, the more likely it is that the individual will be picked for reproduction.

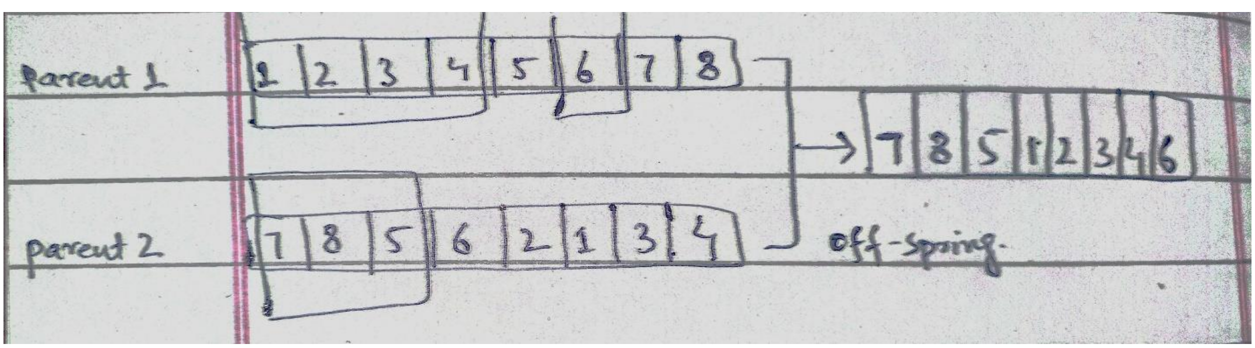
Selection:

Individuals are chosen for reproduction of offspring at this period. To improve reproduction, the selected individuals are sorted. These individuals walk on their genes to future generations.

The primary goal of this phase is to identify the region with the highest probability of producing the best solution to the problem (better than the previous generation). The fitness proportional selection strategy is utilized by the genetic algorithm to ensure that useful solutions are selected for recombination.

Crossover:

Mating between individuals is represented by crossover. The selection operator is used to choose two persons, and the crossover locations are determined at random. The genes at these crossover locations are then transferred, resulting in the creation of a completely new individual (offspring). For Example:



Mutation:

In a biological sense, do the offspring created share the same characteristics as their parents? My answer is No. Children's DNA alter during their development, causing them to vary from their parents. So this operator populates the new child population with new genetic information. This is accomplished by flipping certain genetic bits. Mutation overcomes the problem of local optimal solution and increases diversity.

Termination:

So the question that arises is that how will we get to know about our best solution. So for that we will define a termination criteria. So, in general, there are several termination circumstances, I will discuss some of them:

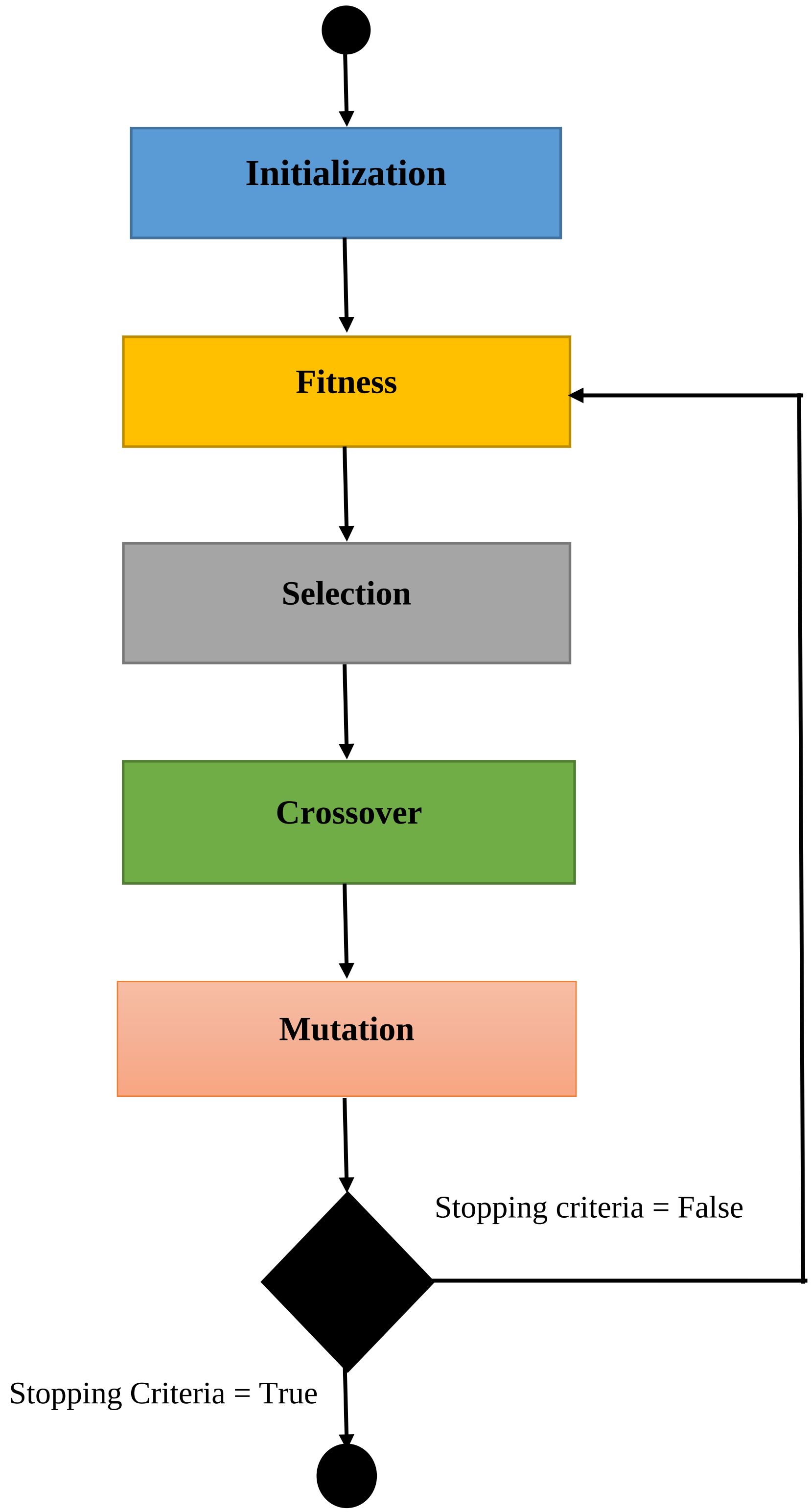
- For more than x iterations, there is no improvement in the population.
- For our algorithm, we have set an absolute number of generations.

Short summary:

The entire algorithm may be summed up as:

- Randomly initialize populations p
- Determine fitness of population
- Repeat until convergence:
    - Select parents from population
    - Crossover and generate new population
    - Perform mutation on new population
    - Calculate fitness for new population

Here is the Flow chart as well:


Initialization
Fitness
Selection
Crossover
Mutation
Stopping criteria = False
Stopping Criteria = True

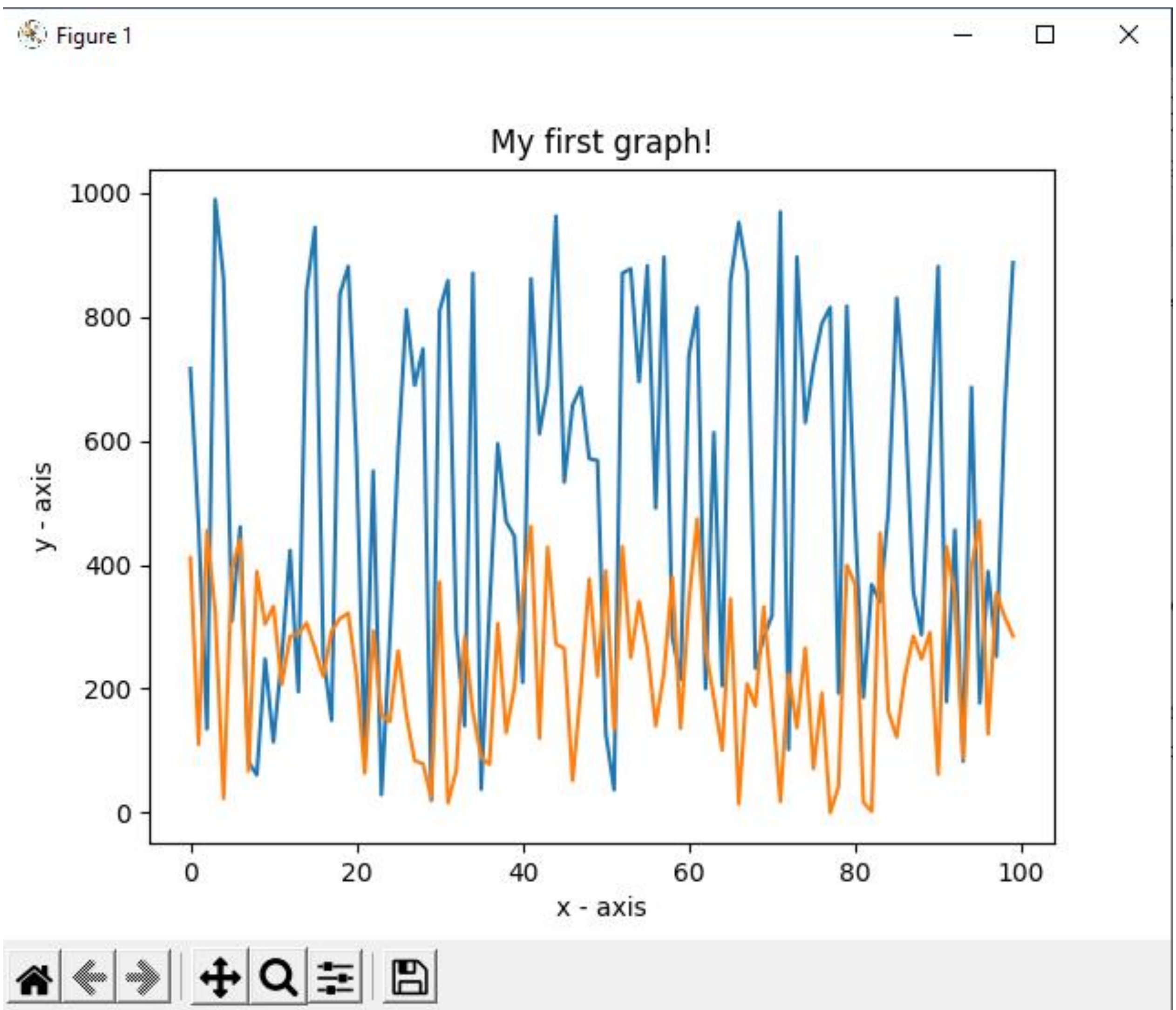
Analysis of Code:

Here I will doing analysis of my own code:

So everything has gone according to plan but my code is not working properly.

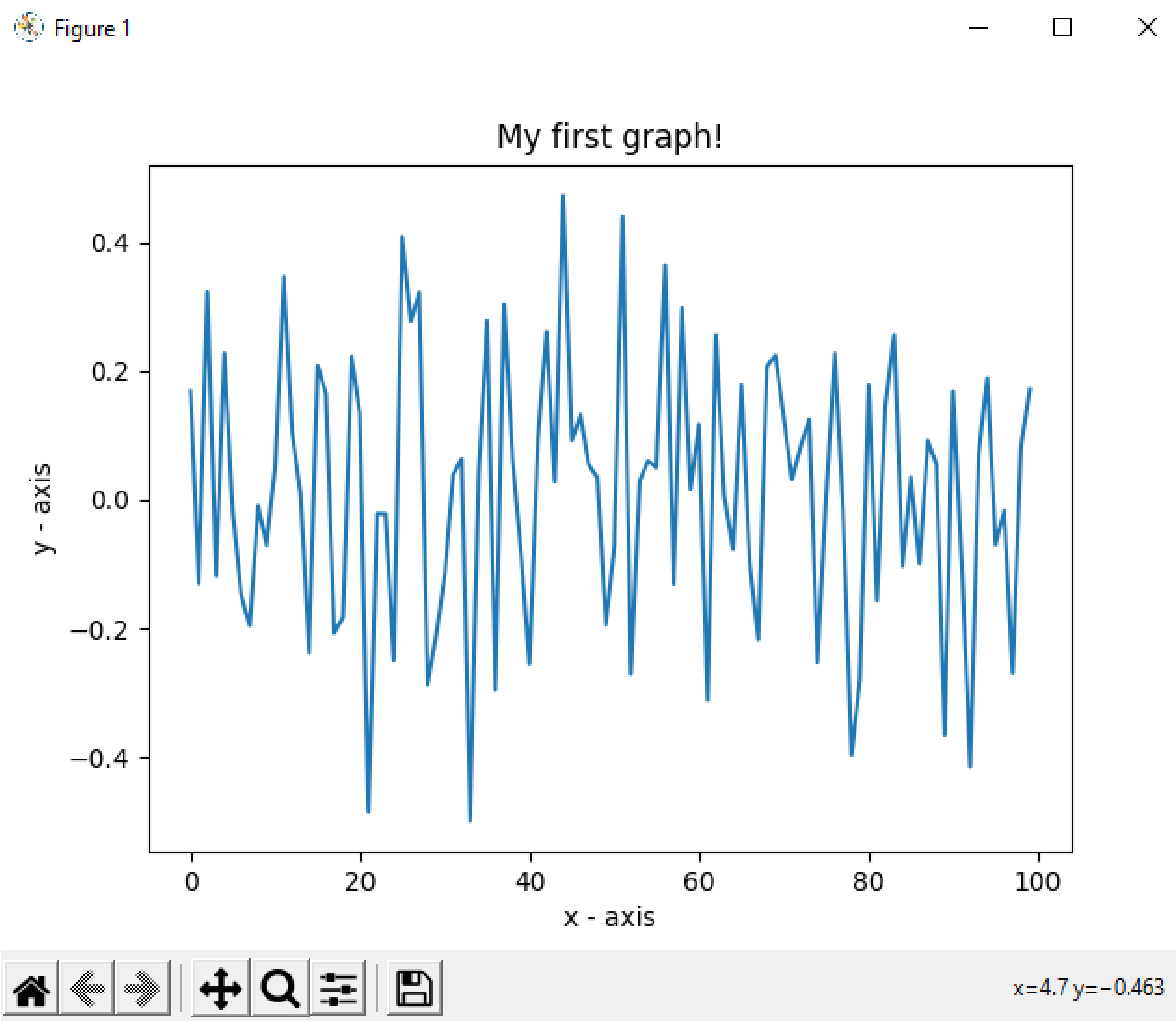
Initialization ():

The graph below shows the initialization of population. On x-axis there are total number of population while on Y-axis the actual random points of population are drawn.



Correlation and fitness Value ():

There are two functions made one is named is correlation and the other one is fitnessValue(). In fitness value, I have used the correlation function. I have also drawn a graph for fitnessValue(). On x-axis there are total number of population while on y-axis its their values.

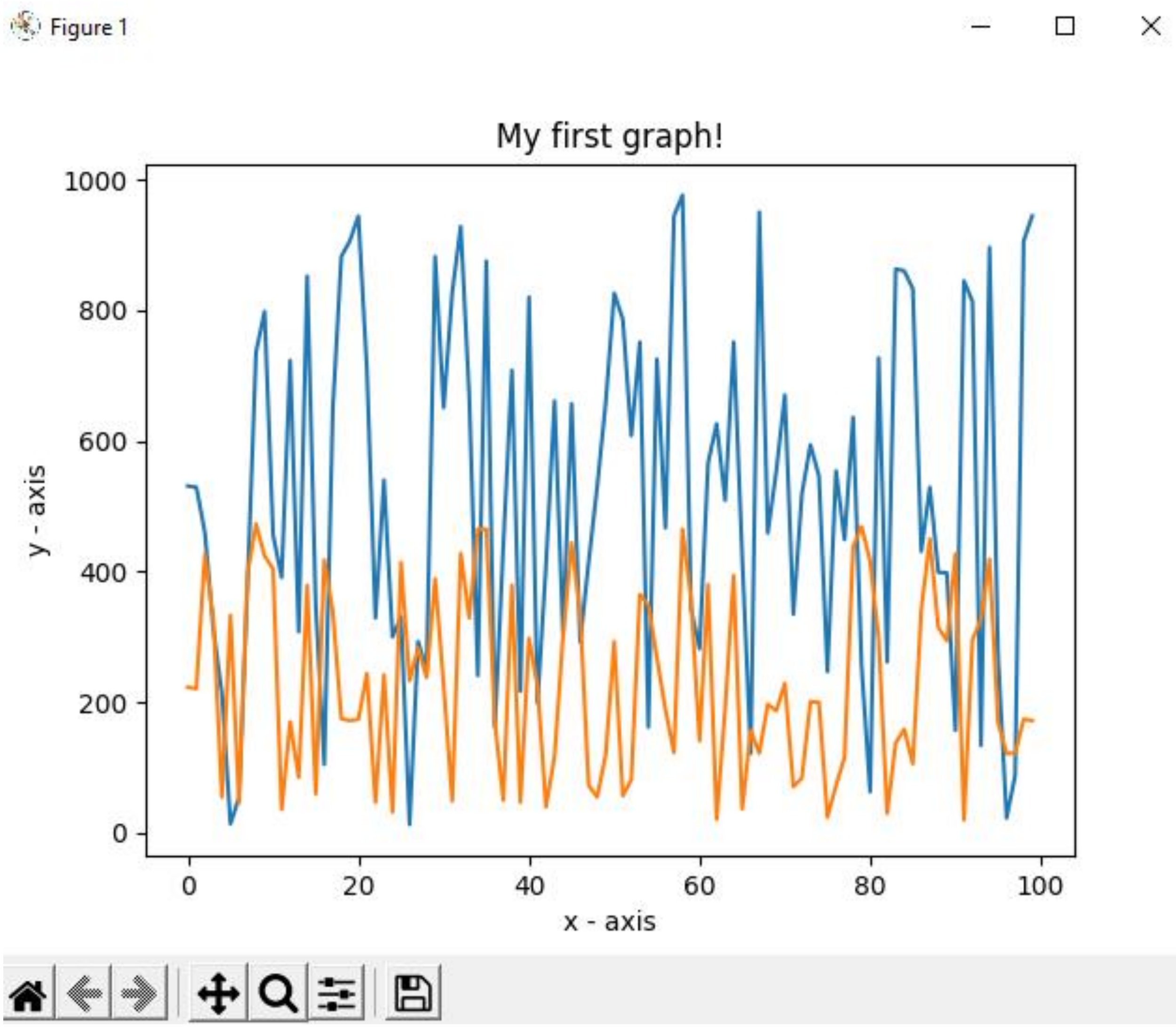


Selection():

There is a function selection which sort all the values of fitness and population.

CrossOver():

So previously we discussed about crossover. I will now show graph to clarify crossover. Following is the graph of population after crossover:



Mutation:

In last there is a function called mutation. We discussed about mutation previously. I think there is some mistake in my mutation function because when I draw its graph it is always empty. Below is the graph of mutation.

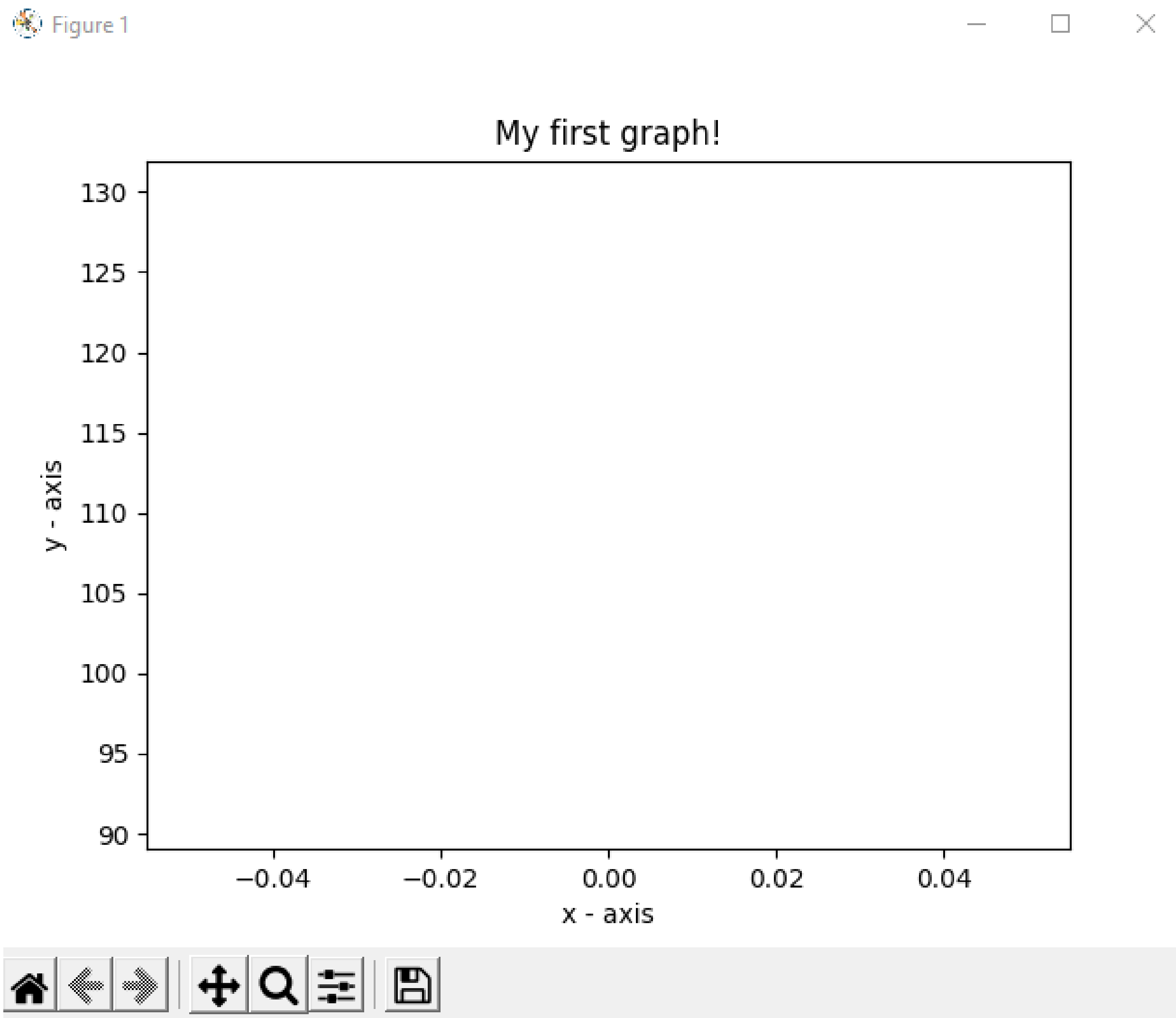


Figure 1
My first graph!
130
125
120
115
110
105
100
95
90
y - axis
−0.04
−0.02
0.00
0.02
0.04
x - axis