NAMAL INSTITUTE

Namal Institute Mianwali

Evolutionary Algorithm

Artificial Intelligence

Submitted by: Syed Dawood Shah

Roll No: BsCs-2019-43

Submitted to: Dr. Junaid Akhtar

Table of Contents

1. Acknowledgement

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 Evolutionary Algorithms., which also helped me in doing a lot of Research and I came to know about so many new things.
Any attempt at any level can not be satisfactorily completed without the support and guidance of my parents and friends. So, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame. I will extend my gratitude to my Teacher's Assistant Ali Raza Khan for his valuable suggestions.

I am overwhelmed 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.

2. Abstract

Evolutionary algorithm (EA) emerged as an important optimization and search technique in the last decade. Due to its flexible nature and robust behavior inherited from Evolutionary Computation, it becomes an efficient means of problem solving methods for widely used global optimization problems.

Here, I will throw some light on What an Evolutionary Algorithm is? How is it optimized by Natural phenomena? and some real life problems and applications of Evolutionary Algorithms.

3. Introduction

What's the first thing you do when you're asked to find a certain image in the large image? We typically look out for that image in the large image, and make our decision.

Our brain is able to analyze, in a matter of milliseconds, what kind of image we have to find. But what if it is a complex one? Obviously, we can still find it but it will take more time. That's where technology comes into play. Technology is for our comfort and time saving. So, here arises the question: **Can machines do that?**

The answer was an emphatic **'no'** till a few years back. But the rise and advancements in computer vision have changed the game. We are able to build computer vision models that can find objects, determine their shape, predict their position in a certain area, and many other things. You might have guessed it; that's the powerful technology behind face recognition like in face lock.

So, here we'll be dealing with the problem of template finding (matching) using Evolutionary Algorithms inspired by Darwin's theory of Evolution. It is a powerful algorithm that takes us to a whole new level of working with image data. We will take a look at the process of how Evolutionary Algorithms are inspired by Natural phenomena.

4. Key Words

Natural selection, Evolutionary Algorithm, Genetic Algorithm, Survival of the fittest.

5. Natural Reality

If we look at nature, we will find that there is a lot of diversity in nature. Everything that is present around us differs from each other in one way or the other. No matter how many times we check, diversity is always there. But if we look at this diversity, certain questions arise in our minds that **"Is this diversity which is always there around us is the separate act of God?"** Or **"Is this diversity a result of evolution of just a single organism?"**. These questions became the basis for the **Theory of Evolution.** To find answers to these questions **Charles Robert Darwin**, an English naturalist started with his ideas. In 1831, Charles Darwin received an astounding invitation to join the HMS Beagle as ship's naturalist for a trip around the world. For most of the next five years, the Beagle surveyed the coast of South America, leaving Darwin free to explore the continent and islands, including the Galapagos. It paves the path towards Darwin's **Theory of Evolution.**

6. Theory of Evolution

“A creationist when he visited the Galápagos Islands, Darwin grasped the significance of the unique wildlife he found there only after he returned to London” **(Frank J. Sulloway)**

On his travels, Darwin had collected finches from many of the Galapagos Islands (off the coast of Ecuador), which helped him to formulate his idea. Some of these finches had stout beaks for eating seeds, others were insect specialists. But Darwin realized that they were all descendants of a single ancestor. **Darwin’s revolutionary theory** was that new species arise naturally, by a process of evolution, rather than having been created forever immutable by God. The theory of evolution describes how organisms evolve over generations through the inheritance of physical or behavioral traits.

The theory starts with the premise that within a population, there is variation in traits. Individuals with traits that allow them to adapt to their environments will help them survive and have more offspring, which will inherit those traits. Individuals with less adaptive traits will less frequently survive to pass them on. Over time, the traits that allow species to survive and reproduce will become more frequent in the population and the population will change, or evolve. Through natural selection, Darwin suggested, a diverse life-form could arise from a common ancestor.

6.1 Key Points of Darwin’s Theory of Evolution

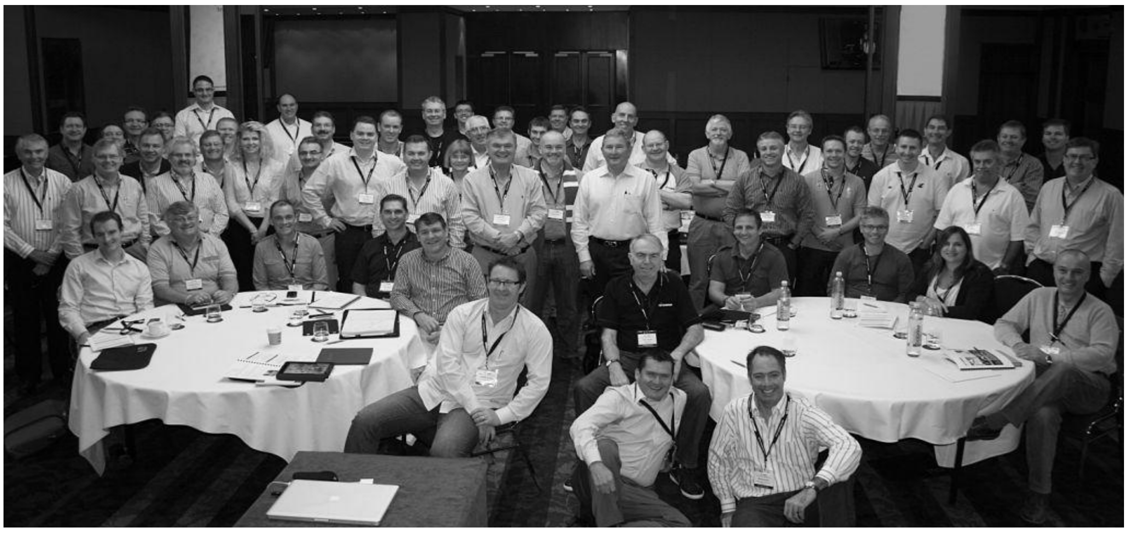
- Individuals of a species are not identical. (Population with varying fitness)
- Traits are passed from generation to generation. (Environment Selects (or prioritizes) the fittest)
- More offspring are born than can survive. (Fittest produce the next generation of population)
- Only the survivors of the competition for resources will reproduce.
- Population keep evolving slowly (back to step 1)

Based on these facts, an algorithm was designed which is known as **Evolutionary Algorithm.**

7. Computational Model

In the real world, we come across many problems that are really difficult to understand. So, for this we derive **computational models**. A computational model uses computer programs to simulate and study complex systems using an algorithmic or mechanistic approach. The system under study is often a complex nonlinear system for which simple, intuitive analytical solutions are not readily available. Rather than deriving a mathematical analytical solution to the problem, experimentation with the model is done by adjusting the parameters of the system in the computer, and studying the differences in the outcome of the experiments. Operation theories of the model can be derived/deduced from these computational experiments.

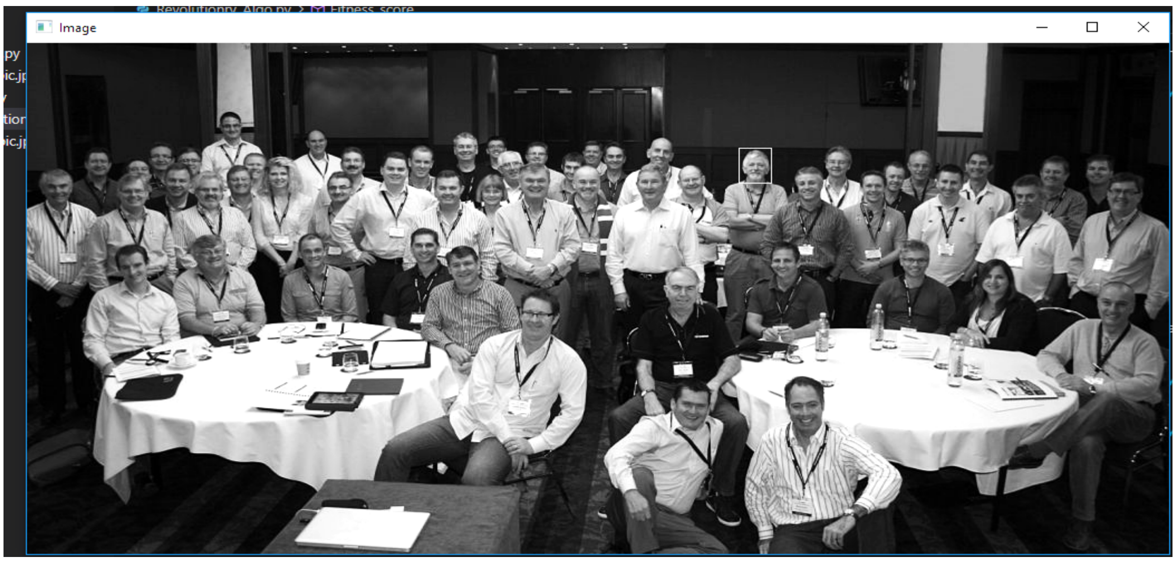
Recently, we came across the problem of locating a certain template in the large image. We call it image finding or locating technique. For example, let we have two images, a small one (template image) and a large image (main image).



(Main image)

(template image)

We want to locate the exact position (roughly speaking) of the template image in the main image like this;



We can do this quite easily by conventional methods using exhausted search technique.In exhausted search, we start searching from the start position and we keep on searching in a linear manner until we find our desired position or until our search exhausts. But that's not the ideal or best way to do this. We can get our desired output by doing so, but it is very expensive in terms of time complexity. So, to tackle this problem, we will design a computational model/ algorithm based on **Darwin's revolutionary theory** which is known as **Evolutionary Algorithm**.

7.1. Evolutionary Algorithm

An evolutionary algorithm (EA) is an algorithm that uses mechanisms inspired by **nature** and solves problems through processes that emulate the behaviors of living organisms. **EA is a component of both evolutionary computing and bio-inspired computing.**

The design of evolutionary algorithms can be divided into several components:

- Representation
- Parent selection
- Crossover operators
- Mutation operators

- Survival selection
- Termination conditions.

EAs are inspired by the concepts in **Darwinian Evolution**. In EAs, the solutions play the role of individual organisms in a population. The mix of potential solutions to a problem is populated randomly first. Then the population is tested for fitness to check how well and how quickly it solves a problem. Next, the fittest individuals are selected for reproduction. The cycle begins again as the fitness of the population is evaluated and the least fit individuals are eliminated.

As the mechanisms by which EAs work are inspired by evolution and living organisms, functions might include selection, reproduction (Cross Over), and mutation. The adaptive process of choosing the best available solutions to a problem where selection occurs according to fitness is analogous to **Darwin's survival of the fittest**. Algorithm solutions that work best among the available options reproduce; the least fit, being eliminated, do not. By testing fitness according to measured performance, optimization occurs over generations through such functions as mutation.

EAs are excellent at optimizing solutions. It is important to note though that while EAs optimize effectively, they don't necessarily find the optimal solution. Instead, EAs constantly find working solutions and measure performance against one another, which may or may not find the absolute best possible solution. It is because the whole algorithm is based on randomness. Here we will be using Genetic Algorithm.

7.1.1. Genetic Algorithm

➢ This is the most popular type of EA. One seeks the solution of a problem in the form of strings of numbers (traditionally binary, although the best representations are usually those that reflect something about the problem being solved), by applying operators such as recombination and mutation (sometimes one, sometimes both). This type of EA is often used in optimization problems.

Phases of Genetic Algorithms are;

i. Initialization of Population(Coding)

- Every gene represents a parameter (variables) in the solution. This collection of parameters that forms the solution is the chromosome. Therefore, the population is a collection of chromosomes.
- Order of genes on the chromosome matters.
- Chromosomes are often depicted in binary as 0's and 1's, but other encodings are also possible.

ii. Fitness Function

- We have to select the best ones to reproduce offspring out of the available chromosomes, so each chromosome is given a fitness value.
- The fitness score helps to select the individuals who will be used for reproduction.

iii. Selection

- This phase's main goal is to find the region where the chances of getting the best solution are more.

- Inspiration for this is from the survival of the fittest.

iv. Reproduction

Generation of offsprings happen in 2 ways:

- Crossover
- Mutation

a) Crossover

Crossover is the most vital stage in the genetic algorithm. During crossover, a random point is selected while mating a pair of parents to generate offspring. There are 3 major types of crossover. But we will be using Single Point Crossover in which a point on both parents' chromosomes is picked randomly and designated a 'crossover point'. Bits to the right of that point are exchanged between the two parent chromosomes.The new offspring are added to the population.

b) Mutation

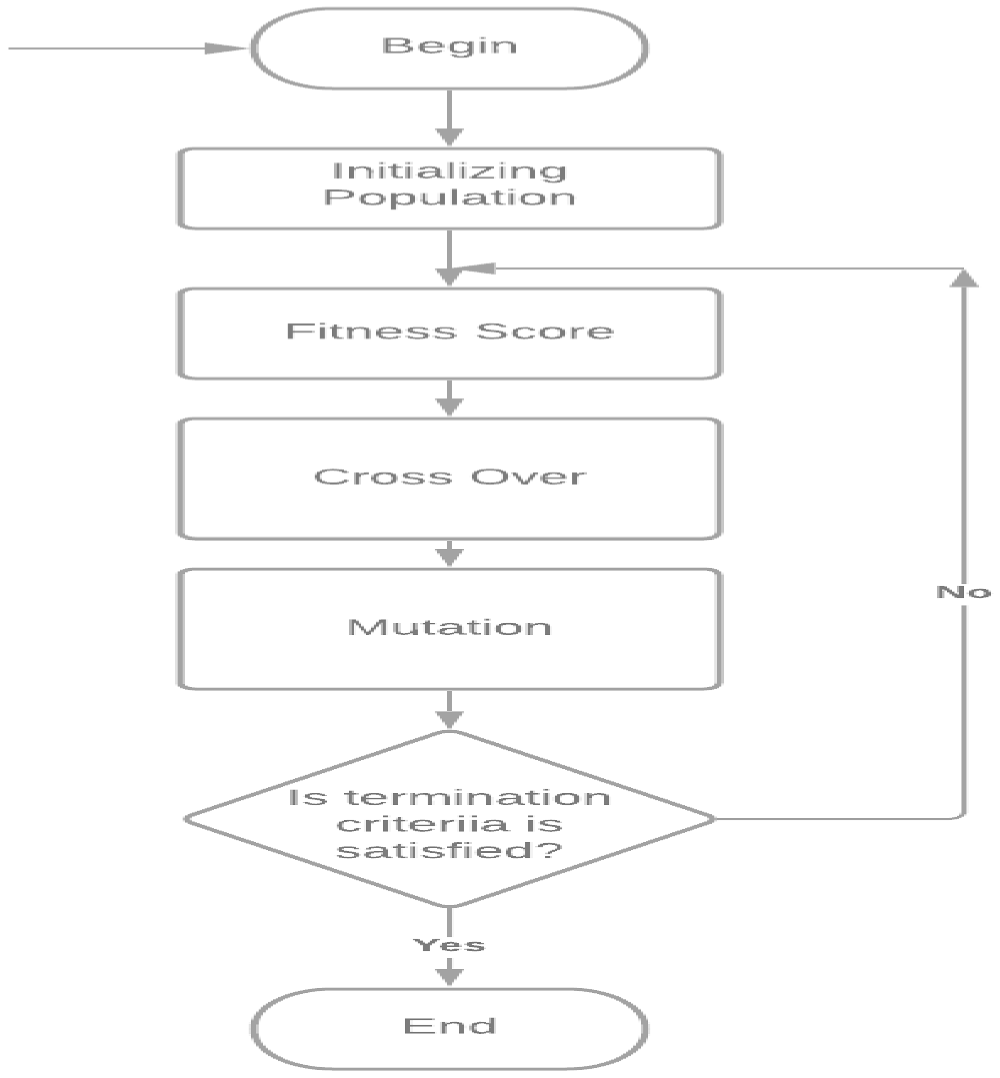
In a few new offspring formed, some of their genes can be subjected to a low random probability mutation. This indicates that some of the bits in the bit chromosome can be flipped. Mutation happens to take care of diversity among the population and stop premature convergence.

v. Convergence (when to stop)

Few rules which are followed which tell when to stop is as follows:

- When there is no improvement in the solution quality after completing a certain number of generations set beforehand.
- When a hard and fast range of generations and time is reached.
- Till an acceptable solution is obtained.

Below is the computational model of Designed for our problem;



7.2. Pseudo Code

Input = image, template

Output = best fit individual

```
Initialize population;
evaluate the fitness value of each individual;
while the optimal solution is not found and;
 the number of generations defined is not reached;
 select parents;
 apply genetic operators (CrossOver and Mutation) to the selected individuals;
 evaluate fitness values of new individuals;
 select individuals for the next generation;
end while;
return the best individual;
```

8. Application

I coded the GA using python. I used certain libraries to help me out.

Libraries are,

```python
from math import log10

from typing import Sized

from matplotlib import pyplot as plt

import numpy as np

import matplotlib

from numpy.lib.type_check import imag
from matplotlib.patches import Rectangle

import cv2

import random
```

I created a total of 13 functions out of which 6 were my main functions and the remaining 7 were helper functions.

8.1 Main Functions

i. Population_Initialising

input = null

output = list of population

In this function, I initialize a randomly generated population of 100 individuals. I gave the range according to the size of the main image so that no individual is generated out of bound

ii. Fitness_Score

Input = list of population

output = list of population

In this function, I first generated a matrix started by the individual of the population and calculated its fitness value. For calculating fitness value, I used a sub function called correlation_coefficient to make my function small.

iii. cross_over

input = sorted population list

output = next generation list

This function was crossing over parents to make a new generation. I used a single point crossover. Here, I generated a random index around which the genes(binary digits) were swapped from one parent to the other giving rise to the new off-spring.

iv. mutation

Input = next generation
Output = Population list
Here, I picked 2 individuals from the next generation randomly selected and mutated them. For mutating, I generated a random index to take a compliment of that bit.

v. display

Input = tuple(row, column)

Output = display rectangle in the main image

This function was just displaying a rectangle of the size of the template image against that particular tuple provided to it in the main image.

vi. Sorting

Input = Population

Output = ranked Population

This function was just sorting the individuals in the population list on the basis of their fitness score.

8.2. Experiments

i. Experiment 1

In this experiment, I used a population of 100 sizes. Here, I calculated the fitness score (correlation) using the following approach;

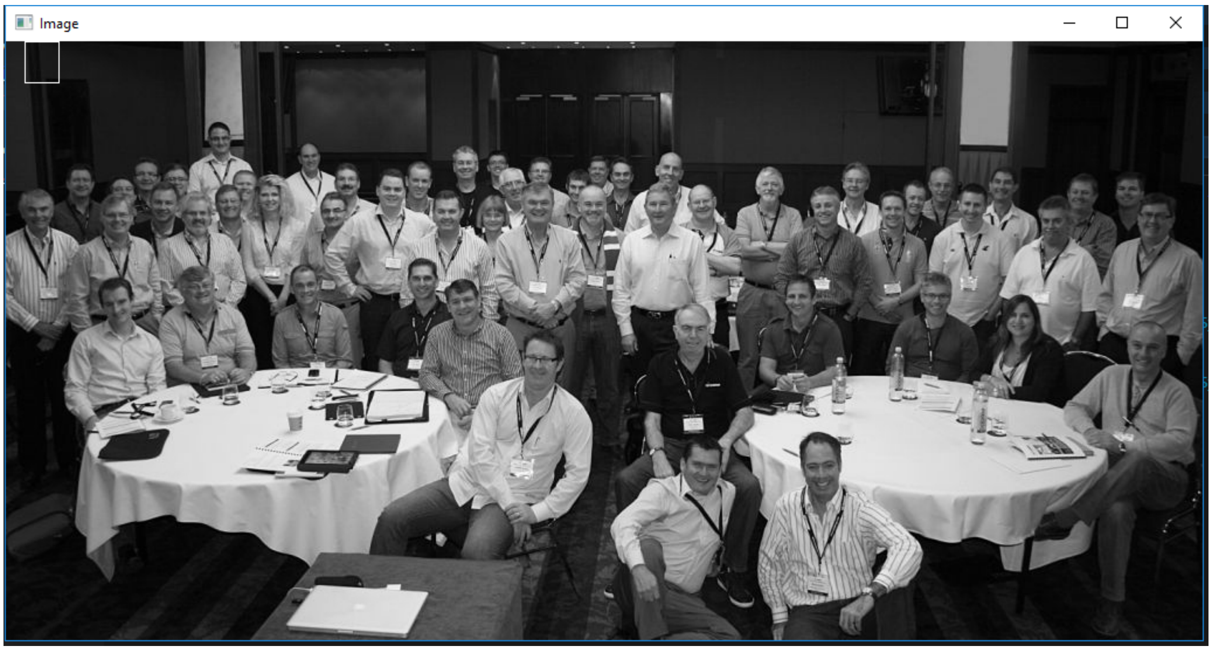
My **hypothesis** was,"If I count the total numbers of pixels matched between the two matrices, I will get a value between 0 and 1. The closer the value to the 1, the higher the goodness and vice versa. "

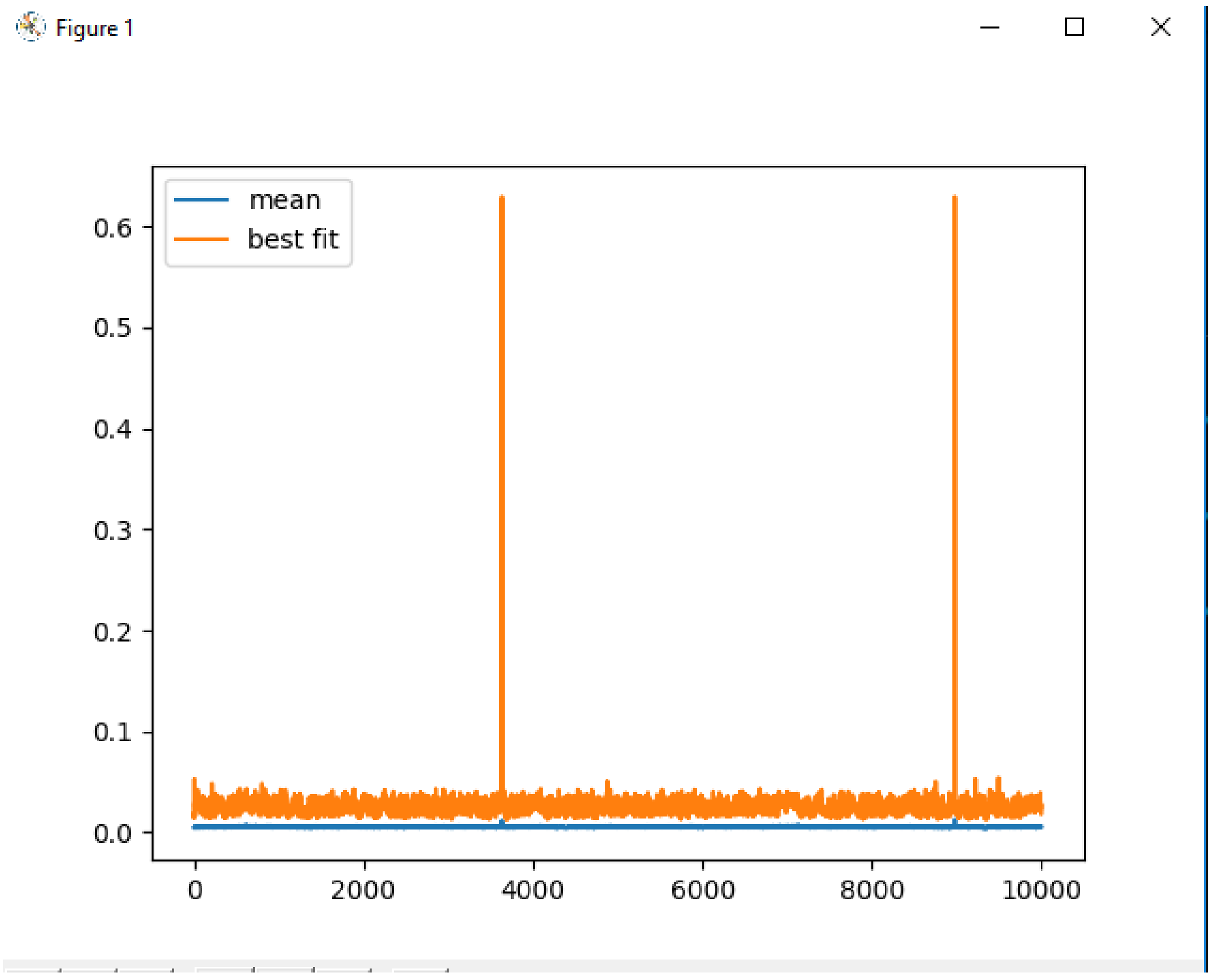
Setup:

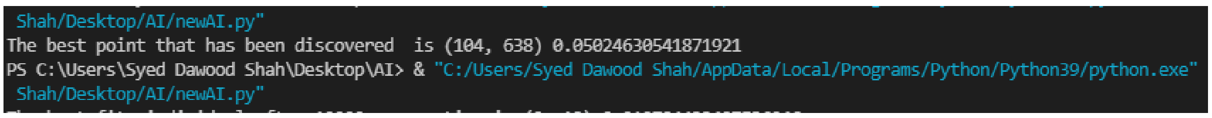
- In this experiment, I was comparing the two matrices of pixels (one extracted from the main image against some individual and the second one was of small image). And while comparing, if any of the pixels between two matrices is equal, I counted them. In the end, I divided the total number of pixels matched with the size of the matrix to get the fitness value.

Result:

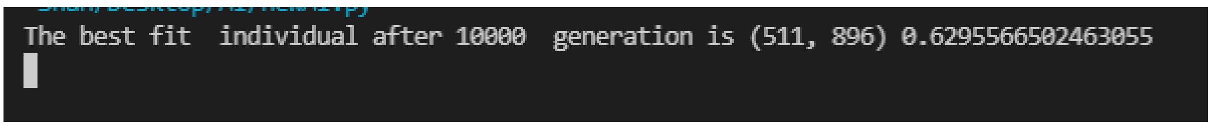
- I was unable to locate the best fit individual in the first attempt. I performed it multiple times and I got some unusual results.







```
 Shah/Desktop/AI/newAI.py"
The best point that has been discovered  is (104, 638) 0.05024630541871921
PS C:\Users\Syed Dawood Shah\Desktop\AI> & "C:/Users/Syed Dawood Shah/AppData/Local/Programs/Python/Python39/python.exe"
 Shah/Desktop/AI/newAI.py"
```



```
The best fit  individual after 10000  generation is (511, 896) 0.6295566502463055
```

Observations:

- The maximum fitness score was not getting more than 0.62.
- It calculated0.05 fitness score for the individual (104, 638) which is almost the best fit individual but I got a 0.62 fitness score for the individual (511, 896) which is very much away from being the best fit point.

- Crossing over all the individuals of the population at a fixed point (I used the 6th bit, after the 6th bit the remaining bits were switched between the two parents to generate new offspring).
- Mutating all the individuals of the population.

So, clearly there was some issue with my fitness evaluation function. So, in the next experiment I used a different approach for calculating fitness.

iii.Experiment 2

Setup:

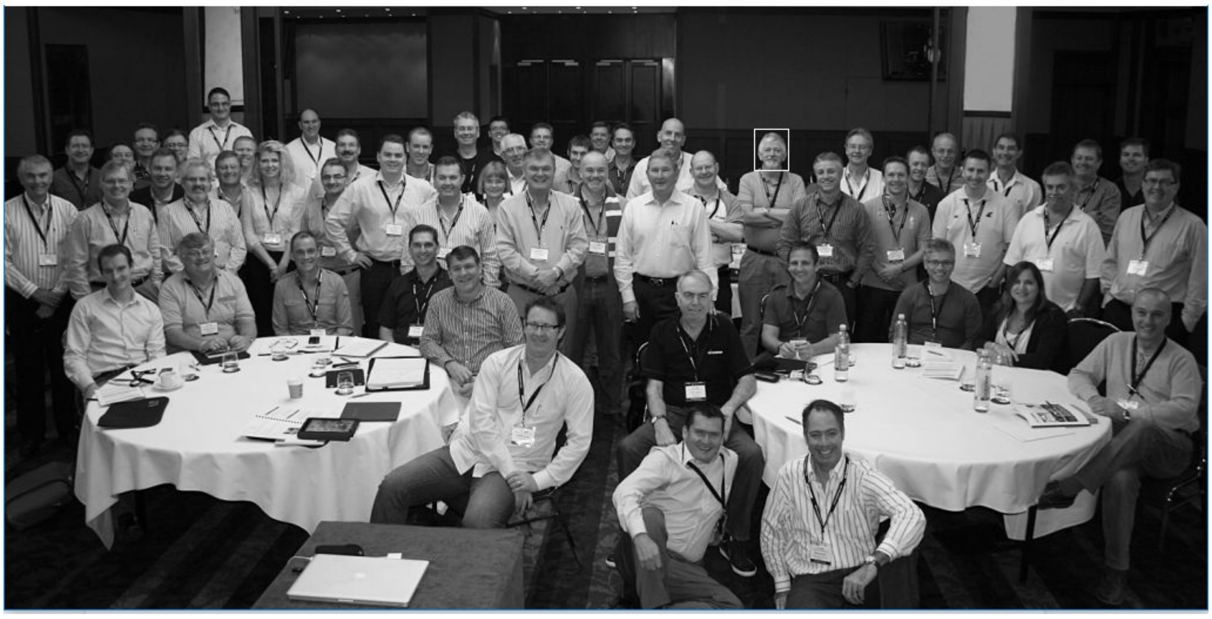
- Crossing over all the individuals of the population at a fixed point (I used the 6th bit, after the 6th bit the remaining bits were switched between the two parents to generate new offspring).
- Mutating all the individuals of the population.
- Calculating fitness with the help of the following formula

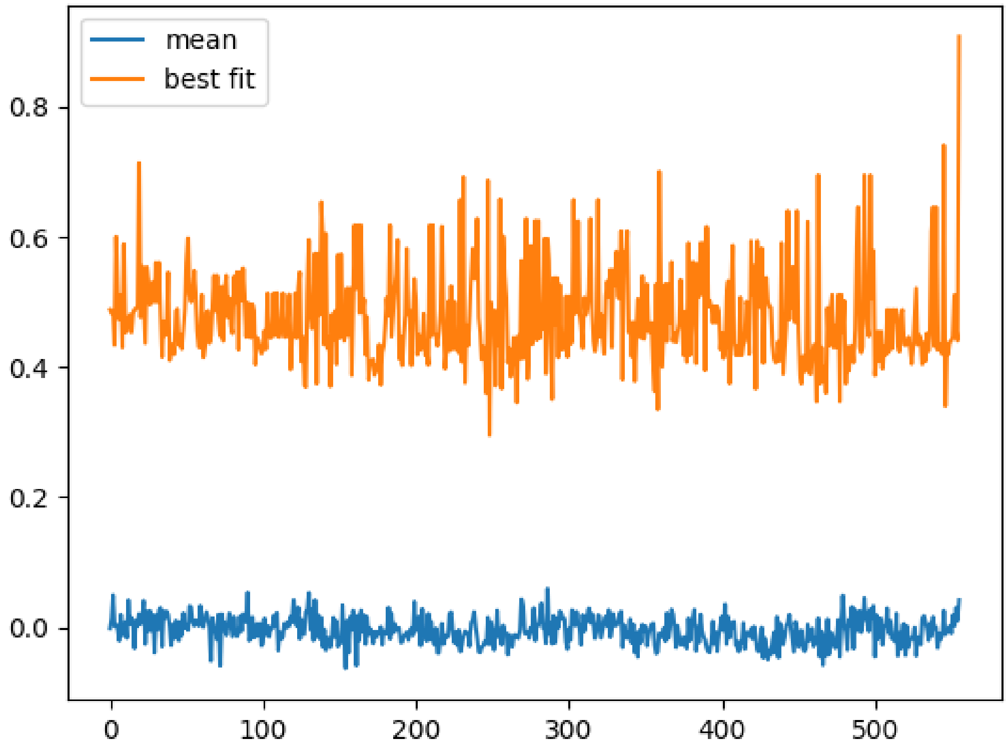
$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Hypothesis: “ By using the above stated setup, I might be able to get the best fit individual.”

Results:

- Luckily, I found the best fit individuals that find almost the exact location of the template image in the main image within 600 generations.





Observation:

- The best fit individual was (106, 639) with fitness score 0.92.

- The graph was showing non-convergent behaviour.

So, to deal with this non-convergent behaviour, I performed another experiment.

iv. Experiment 3

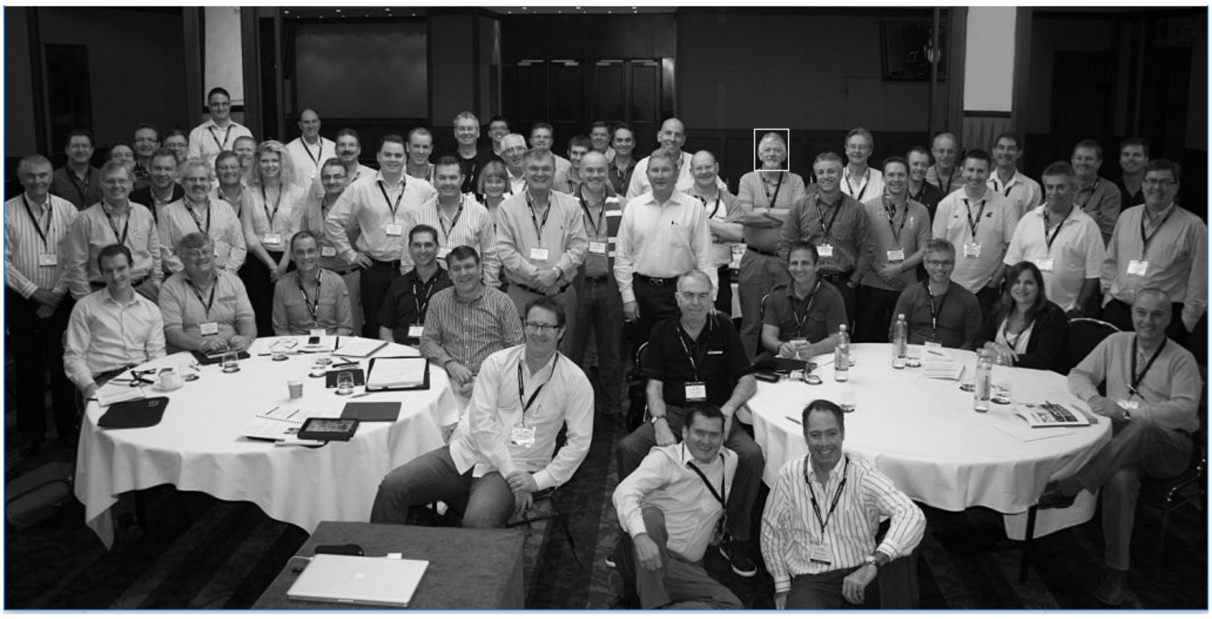
Hypothesis: " By introducing randomness in the crossover and mutating only some portion of individuals might help to avoid non-convergent behaviour."

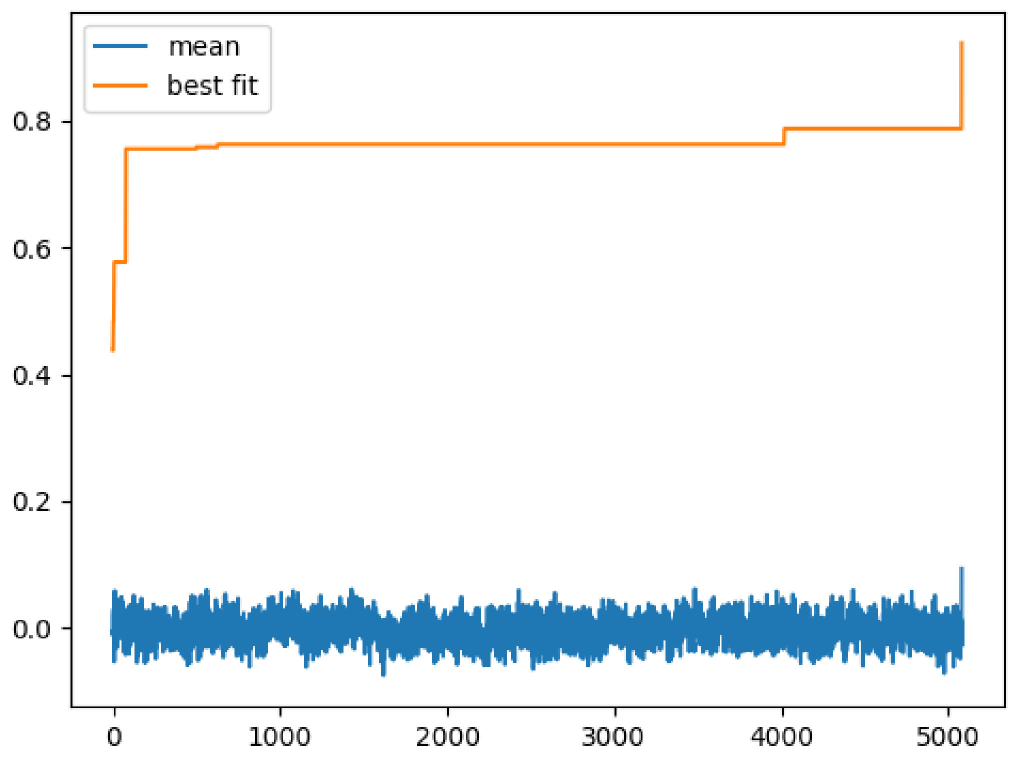
Setup:

- Crossing over all the individuals of the population at the randomly selected bit (I generated a number between 4-16 range and switched the remaining bits of the parents after that randomly selected bit)
- Mutating a small portion of the population (I mutated 3 randomly selected individuals).

Result:

- I was again able to locate the best fit point on the first attempt.



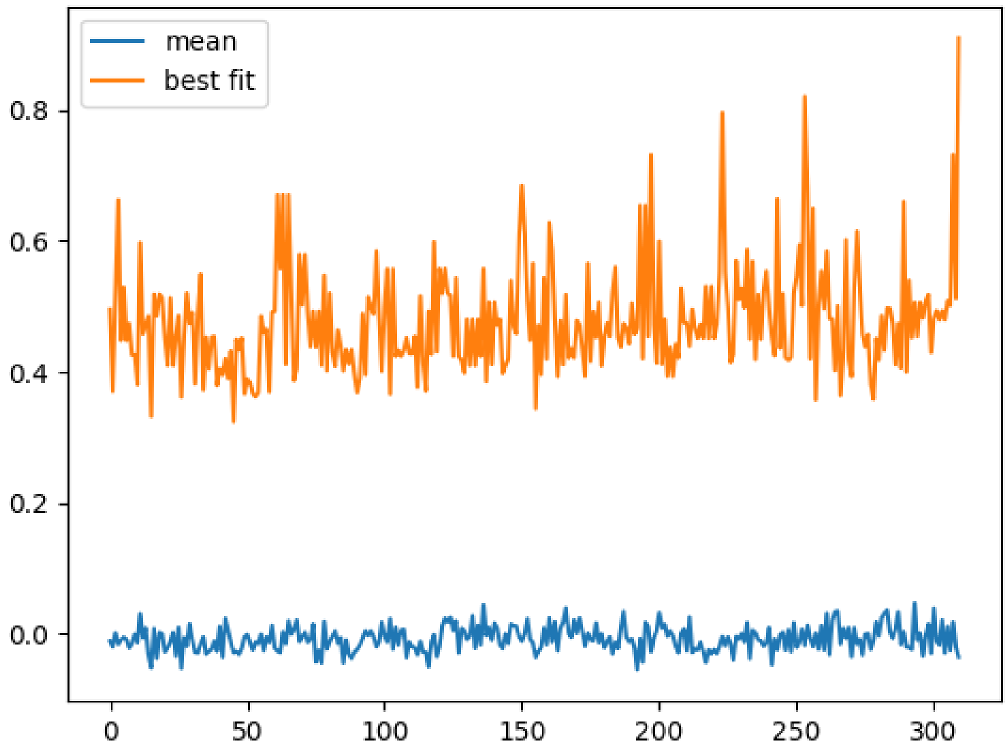


Observation:

- I note here that the non-convergent behaviour of the best fit score of each generation was controlled to some extent.
- But, I wasn't sure about the mean fit score of each generation that what's going on here.

8.3. Conclusion

- I repeated Experiment 4, but I was never able to get that graph again. Every single time I tried, I got the same results like this,



- I tried my best to figure it out, but I think that is the limitation oy my program. It will find the best individual but there will be non-convergent behaviour.

9. References

https://www.livescience.com/474-controversy-evolution-works.html

Key points of Darwin's theory of Evolution

Genetic Algorithm

Evolutionary Algorithm