NAMAL INSTITUTE

Mianwali

Department of Computer Science

CS-340 Artificial Intelligence

Evolutionary Algorithms

Student's Name	**Hurmat Ilyas**
Roll Number	**BSCS2019-02**
Date	**31 Oct,2021**

Instructor: Dr. Junaid Akhtar

Namal Institute Mianwali

Acknowledgement

I would like to express my sincere gratitude to my respected teacher Dr.Junaid Akhtar for giving me an opportunity to do this project and TA Mr. Ali Raza Khan for helping me out in this project.

Namal Institute Mianwali

Abstract

Everything in this world evolves with time. Evolutionary algorithm derived from evolutionary theory is one of the best ways to solve problems in real time domain. Evolution simply means the change in characteristics of species over several generations using the process of natural selection. Evolutionary theory tells us that how the species are interlinked and the diversity among the individuals. Based on this theory is the algorithm, known as Genetic Algorithm or Evolutionary Algorithm, used for optimization and search problems by relying on biologically inspired operators such as selection, crossover and mutation. Using this algorithm, a search problem is solved in which a smaller image is being searched and matched from a bigger image. The functional model of science is used to solve this problem which consists of four major steps including Natural phenomenon, Theory, Computational model and Application. The outcome of the code based on Genetic Algorithm is a patch drawn on the small image in the large image which means that the image is found. The Fittest parents were selected and then crossed to populate the child that helped us find the X-Y coordinates where the smaller image lies in the bigger one.

Namal Institute Mianwali

TABLE OF CONTENTS

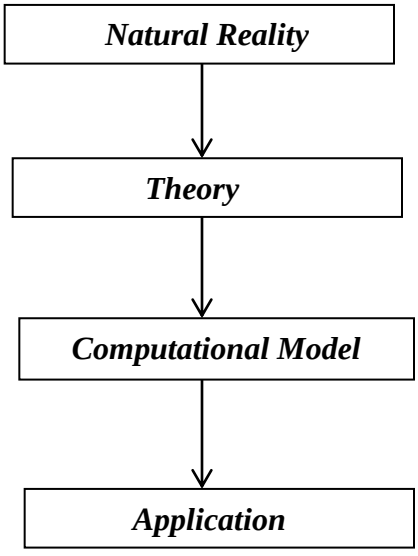
Namal Institute Mianwali

1-Problem Statement:

There is a smaller image of a man which we have to find in a group image using the evolutionary or genetic algorithm. By using functional model of science which consists of four steps i.e. Natural Reality, Theory, Computational model and Application this problem can be solved.

2-Methodology:

The scientific approach is used as mentioned above.

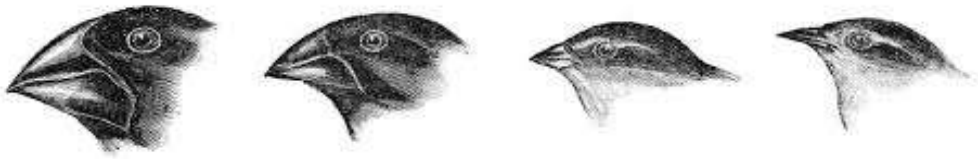


Namal Institute Mianwali

2.1- Natural Reality:

The scientific theories are based on some natural reality. Therefore, the theory of evolution is also based on the natural reality. The change in organisms, diversity and evolution are the basic concepts that made this theory. Every individual changes with time, many organisms evolve into other organisms, the species are different from each other and the species are interlinked with each other. Charles Darwin, a naturalist, is the founder of this theory of evolution. A natural phenomenon is "the species that best fits the environment can survive only" which means that only those species that are adjusted best to the environment are successful to survive and all the others will die. This is the part of a phenomenon known as natural selection which means that the organisms with more adaptive traits are able to survive and reproduce and others will not be able to reproduce or survive more. In contrast to artificial selection, Darwin preferred natural selection because the natural environment is selecting the fittest ones not the human beings.

The evolution by natural selection can be shown as:



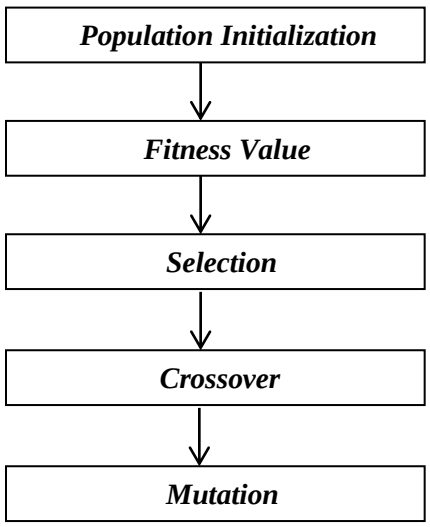
It shows that the beak shape of the birds has evolved because the change in the size and shape of the beak has enabled them to utilize food of different types for their survival. This evolution took place because of the change of environment.

The genetic algorithm is based on biological evolution. The evolutionary algorithms are totally based on the Darwin's theory of evolution. Summing up this whole thing as the change in species lead to the diversity and the fittest will survive.

2.2- Theory:

Theory is an idea or a concept which is experimented and tested based on the observations made and if they proved to be right, it becomes a theory. Same goes with the Charles Darwin who observed diversity, evolution and survival of the fittest and then tried to gather the evidences. His idea of evolution became Theory of evolution by natural selection.

The theoretical interpretation involves the following steps:



2.2.1-Population Initialization:

In the natural selection that took place in our environment, the first population is generated naturally. So, in the solution to this problem the population is initialized randomly to maintain the concept of natural selection that's why the population is not given rather it is initialized randomly.

2.2.2-Fitness Scoring:

In the code, the algorithm is optimizing the fitness value. It is the central point of the evolutionary algorithm. The speed of execution of the code is dependent on the fitness scoring. It determines that how much fit the solution is. The probability of selection of individual for reproduction depends on the fitness value.

2.2.3-Selection:

In Selection, the data is selected from the randomly generated population. The fittest individuals are selected for reproduction in order to populate generation that best fits in environment. Those with high fitness scoring have a higher probability to reproduce.

2.2.4-Crossover:

To populate the next generation, the crossover is used. The parents are crossed to produce new offspring.

2.2.5-Mutation:

Mutation is an operator which is basically flipping the bits.

2.3-Computational Model:

Using evolutionary Algorithm, the smaller image is being matched with the bigger one. Initially the population is initialized randomly. Then, the selection function is used which is choosing the individuals that can survive and best fit. It will be checked that either the image is matched or not. Then, the crossover and mutation are being done which are generating the offspring by crossing the parent's genes.

Then termination takes place which checks that if population does not produce offspring which are significantly different from the previous generation then the algorithm has given solution of this problem. The smaller image will be matched with the bigger template image. Two images are given, the bigger image is of 512 x 1024 and smaller image is 35 x 29.

The pictures are converted to 2D-Arrays and the matrix of smaller image is matched with the matrices of bigger picture. Using the function of correlation, the threshold is obtained. With the cropped image, the bigger image's selected area can be matched to get the location of small image in large one.

2.4-Application:

All the functions used to solve the template matching image are as follows:

- **Initialize Population:**

  The random population is generated and after that the fitness value is being checked based on the correlation function. In this, it is checked that individual is good or bad.

- **Selection:**

  After that, the selection is done and population is sorted rank-wise.

- **Crossover:**

  Two parents are selected for crossover to produce next generation. There chromosomes are swapped and new generation is populated.

- **Mutation:**

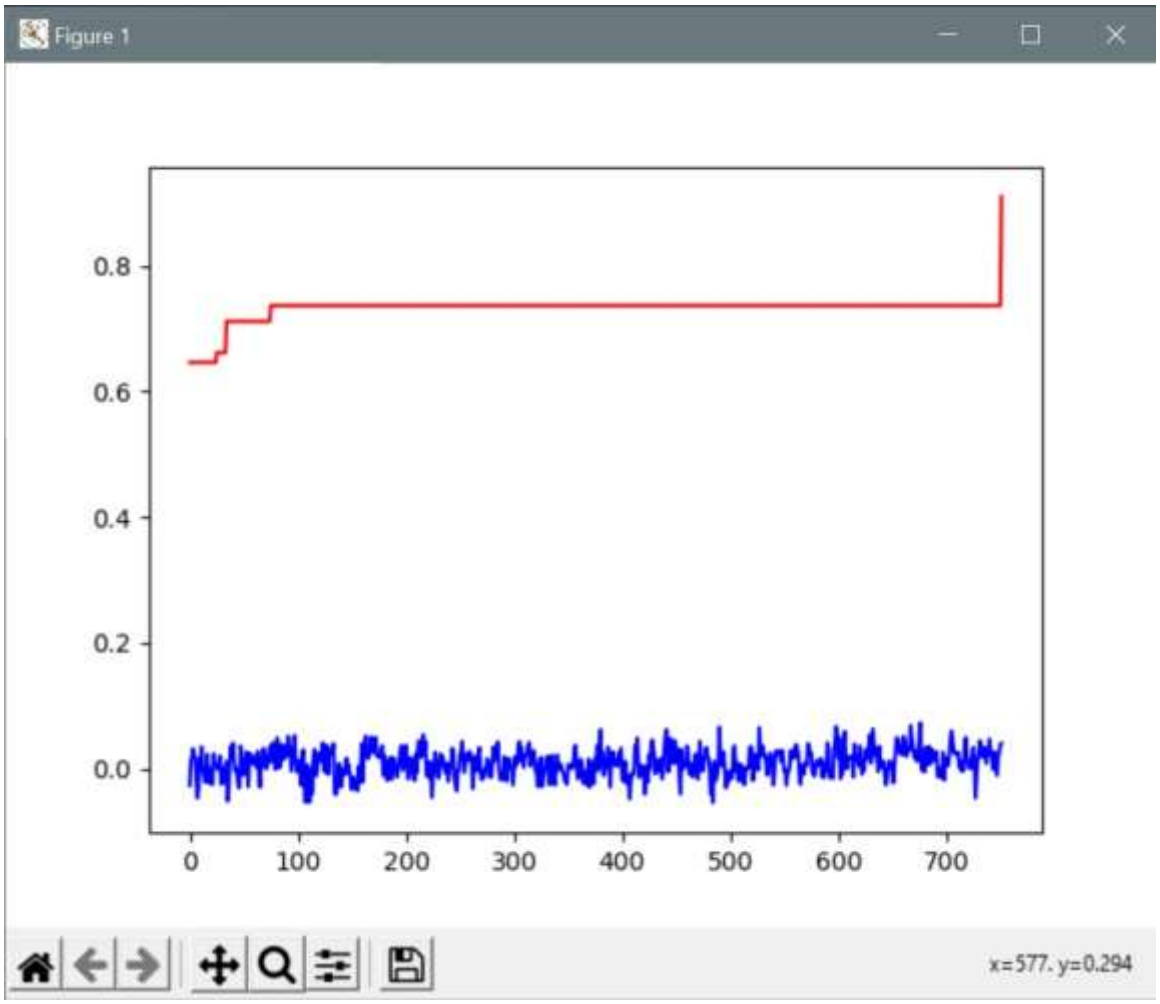
  A single bit is flipped in mutation.

3-Experiment:

Output:

Namal Institute Mianwali

Graph:



Code:

```
import matplotlib.image as img
import matplotlib.pyplot as plt
import random
import numpy as np
from matplotlib.patches import Rectangle
from numpy.lib.function_base import average

#Intialization of Population
def Population(row,column, size):
    population = []
    for i in range(size):
        population.append((random.randint(0, row-29), random.randint(0,
column-35)))
    return population
#Correlation Cofficient


def correlation_cofficient(A, B):
    return(np.mean(((A-A.mean())) * (B-B.mean())) /(A.std() * B.std()))




def fitness(image1, image2,pop):
    Fitness= []
    for i in pop:
        x = i[0]
        y = i[1]
        image3 = np.array([image1[y:y+35, x:x+29]])
        correlation= correlation_cofficient(image2, image3)
        Fitness.append(correlation)
    return Fitness

def selection(fitness,pop):
    rankedpopulation = []
    check = zip(fitness,pop)
    sort_check = sorted(check,reverse=True)
```

```
        for i in sort_check:
            rankedpopulation.append(i[1])

    return rankedpopulation

def crossover(rankedpopulation):
    for i in range(1,len(rankedpopulation)-1,2):
        coordinate1= rankedpopulation[i]
        coordinate2= rankedpopulation[i+1]
        x1coordinate=np.binary_repr(coordinate1[0],10)
        y1coordiate= np.binary_repr(coordinate1[1], 10)
        strngx= str(x1coordinate)
        strngy=str(y1coordiate)
        x2coordinate= np.binary_repr(coordinate2[0], 10)
        y2coordinate= np.binary_repr(coordinate2[1],10)
        strngx2= str(x2coordinate)
        strngy2= str(y2coordinate)
        coordinate1= list(strngx+strngy)
        coordinate2=list(strngx2+strngy2)
        k = random.randint(0, 19)
        for j in range(k, len(coordinate1)):
            coordinate1[j], coordinate2[j] = coordinate2[j],
coordinate1[j]
        coordinate1 = ''.join(coordinate1)
        coordinate2 = ''.join(coordinate2)
        rankedpopulation[i]=
(int(coordinate1[0:10],2),int(coordinate1[10:],2))
        rankedpopulation[i+1]=
(int(coordinate2[0:10],2),int(coordinate2[10:],2))
    return rankedpopulation



def mutation(crossover):
    nextgen= []
    for i in crossover:
        current = i
        while current[0] > 995:
            a= np.binary_repr(i[0],10)
            a1=str(a)
            a2=list(a1)
            k = random.randint(0,9)
            if a2[k]=='0':
                a2[k]='1'
```

```
                else:
                    a2[k]='0'

            a2 =''.join(a2)
            a2 = int(a2,2)
            current= (a2,current[1])
        while current[1] > 477:
            b =np.binary_repr(i[1],10)
            b1=str(b)
            b2=list(b1)
            k = random.randint(0,9)
            if b2[k]=='0':
                b2[k]='1'
            else:
                b2[k]='0'
            b2 =''.join(b2)
            b2 = int(b2,2)
            current= (current[0],b2)
        nextgen.append(current)
    return nextgen
# def termination(Fitness,population):
#     list1 = []
#     for i in range(len(Fitness)):
#         if Fitness[i]>0.9:
#             list1.append()



image1 = np.array(img.imread("groupGray.jpg"))
image2 = np.array(img.imread("boothiGray.jpg"))
pop = Population(1024,512,100)
fit_pop = []
max_fit = []
avg_fit = []
for i in range(1000):
    fittest= fitness(image1, image2,pop)
    max_fit.append(max(fittest))
    avg_fit.append(average(fittest))
    select= selection(fittest,pop)
    if max(fittest)>0.85:
        fit_pop.append(select[0])
        break
    cross= crossover(select)
```

```
    pop= mutation(cross)
plt.imshow(image1)
plt.gray()
ax = plt.gca()
for i in fit_pop:
    rect = Rectangle(i, 29, 35, linewidth =1
,edgecolor='r',facecolor='none')
    ax.add_patch(rect)



plt.show()
plt.figure(1)


plt.plot(max_fit,'r')
plt.plot(avg_fit,'b')
plt.show()
```