Namal Institute Mianwali

Computer Science Department

Artificial Intelligence

Fall 20

Document:	AI Project 01	**Date:**	21-10-2022
Prepared by:	Abubakar Bhatti (BSCS-2020-41) Attiq Ullah Shah (BSCS-2020-21)	**Prepared for:**	Dr. Junaid Akhtar (junaid.akhtar@namal.edu.pk)
Document Details:	This document contains the solution to our observation and explanation of the AI Evolutionary Algorithm in view of the four stages of the scientific process. It also contains the coded implementation of the Evolutionary Algorithm in python.		

Contents

1. Abstract 5

1.1 Overview 5

1.2 Natural Phenomenon 5

1.2.1 Evolution of Deer Mice 5

1.2.2 Evolution of Giraffe 6

1.2.3 Evolution of Finch's Beak 7

 Change in specie physics happens due to natural selection for the survival of the fittest for the easy availability of food to every one according to their respective island environment...... 7

2. Evolution 8

2.1 Evolutionary Theory 8

2.2 Evolutionary Algorithm 9

2.3 Features of Evolutionary Algorithm 9

2.3.1 Representation 10

2.3.2 Selection 10

2.3.3 Recombination / Crossover 10

2.3.4 Mutation 10

2.3.5 Fitness Function 10

2.3.6 Survival Decision 11

3. Coded Implementation of Algorithm 11

3.1 Technologies Used 11

3.2 Computation Model Used 11

Initialization 11

Fitness Score / Evaluation 12

Selection 12

Crossover and Mutation 13
Algorithm Termination 15
4. Experimentation 15
4.1 Hypothesis # 01 15
4.1.1 Experiment for Hypothesis # 01 15
4.2 Hypothesis # 02 16
4.1.2 Experiment for Hypothesis # 02 16
5. Limitations 17
6. Conclusion 17
7. Reference: 18

AI Project # 01:
Evolutionary Algorithm

Problem Statement:

Given a larger and a smaller template image, you have to write a complete Evolutionary Algorithm that takes the two images and solves the template matching problem, i.e. finds the coordinates or location of the smaller image within the larger image.

Objectives:

Mainly there are two objectives for solving the above problem;

- Firstly, we need to know that many linear search algorithms are available to solve the given problem, but they do not follow the philosophy of Artificial Intelligence. In the given problem, we are implementing one of the search algorithms of Artificial Intelligence named the **Evolutionary Algorithm**.

- The second objective is to understand the scientific process. We have to use the scientific process in view of Artificial Intelligence principles (Evolutionary Algorithm). We must follow the following stages of the scientific process;

    1. Observation
    2. Hypothesis
    3. Experiment
    4. Analysis
    5. Modification
    6. Final Results

1. Abstract

1.1 Overview

- We will use the template image to define a correlation function that maximizes the fitness value. This will be done after the algorithm's multiple iterations and will end up with some threshold value of fitness.
- This will follow the Natural Phenomenon principle, the fundamentals of the Evolutionary Algorithm.

1.2 Natural Phenomenon

- Anything occurring without any human input is called a natural phenomenon. The evolution of deer mice, army ants, Giraffes, and Finch birds are some examples of natural phenomena.

1.2.1 Evolution of Deer Mice

Deer mice were dark and light-colored in the early ages when they lived in the woods. It was hard to see the dark-colored mice in the woods, but light-colored mice were seen even from so far. It was observed that dark-colored mice remained in the woods, while light-colored mice started living in sandy areas. This is because of the natural evolution respective to the environment.

To justify it, let us take an example of 100 deer mice, where 50 are dark-colored and 50 are light-colored, living in woods. It was observed that the survival ratio of light-colored mice was 5 to 10 percent. In contrast, the dark-colored mice survived to almost 90 percent because it was apparent to see the light-colored mice by their predators in the particular environment. The survival ratio of the light-colored mouse in sandy areas was the same as dark-colored in woody areas. This was observed by the scientist how natural selection evolved for the survival of the fittest according to the respective environment.

1.2.2 Evolution of Giraffe

The example of the giraffe's evolution is evidence of Darwin's evolution theory. Biological change in the species is due to the change of their genes which varies according to environmental conditions. The gene change depends on environmental factors, including temperature, climate, food, light, and water. These changes are due to natural selection for the survival of the fittest. The giraffe's evolution is shown in Figure no 1 below;

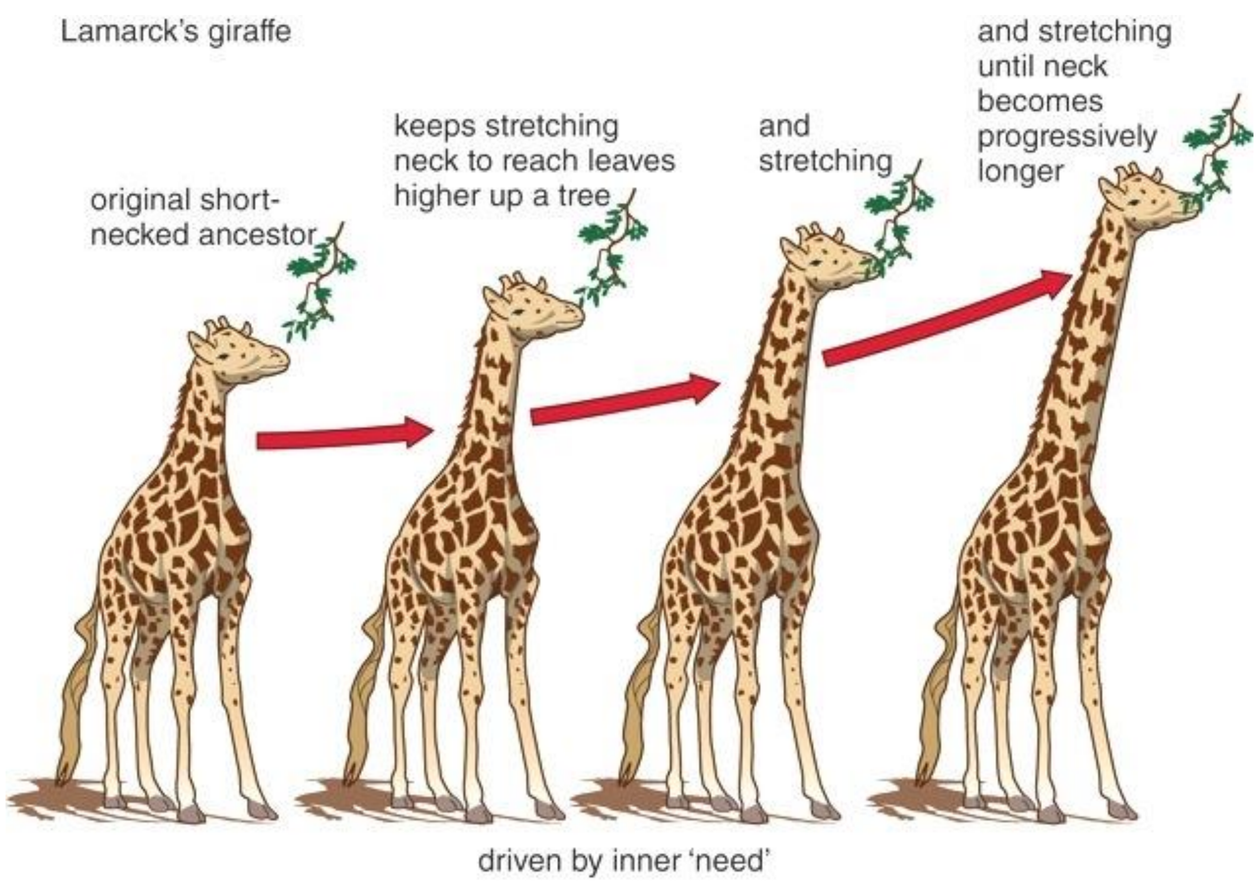


Figure No 01

1.2.3 Evolution of Finch's Beak

- One of the most famous examples of evolution is Darwin Finch's evolution which he observed during his voyage on different islands. He observed the change in the beak of finch birds from island to island. Different islands possess different environments, and this change in the environment leads to different features in the shapes of the species.
- Change in specie physics happens due to natural selection for the survival of the fittest for the easy availability of food to every one according to their respective island environment.
- This evolution applied to every living organism around us for the survival of the fittest, and so is Darwin's evolution theory.
- Finch evolution is shown in Figure no 02 below;

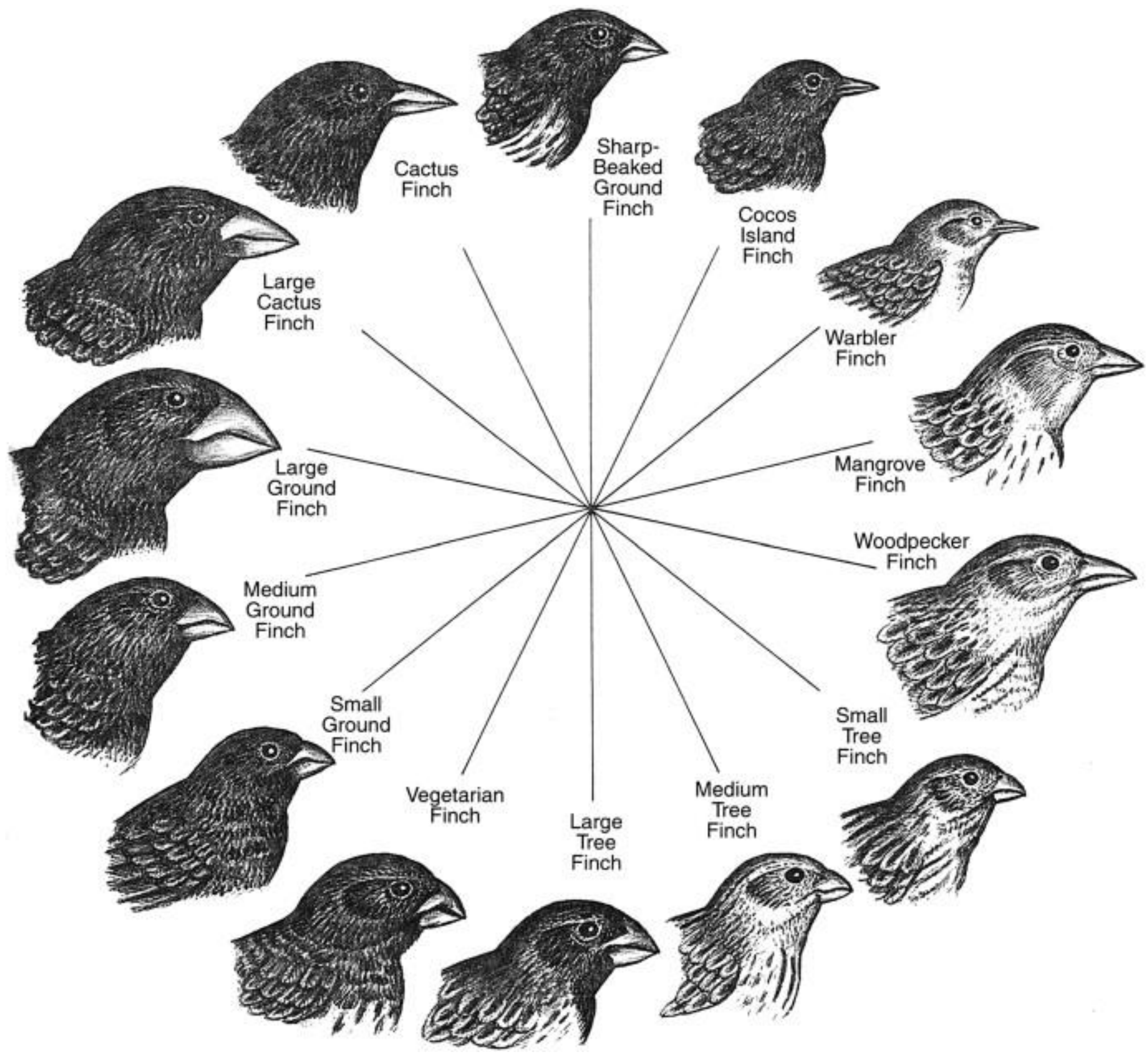


Figure no 02

2. Evolution

2.1 Evolutionary Theory

- Evolution is a process that causes the gradual development of something, and we believe every new generation will be the fittest from the earliest.
- Darwin's evolutionary theory states that evolution happens by natural selection [1.2.1, 1.2.2, 1.2.3].

- We used to consider ourselves at the world's central point in early age, but we accepted the natural reality over time when we searched in depth. The study shows that we are just a tiny part lying in one of the corners of a vast universe consisting of millions of galaxies and hundreds of solar systems where each system consists of many planets. We are not mature enough to understand Darwin's theory because it might conflict with some religious points. However, we cannot ignore the natural evolution process, which causes various changes in the environment due to mutation, leading to diversity.

2.2 Evolutionary Algorithm

- Darwin's Theory of Evolution is the fundamental idea behind the evolutionary algorithm.
- This algorithm uses a decision-making approach to solve the problems.
- This is an optimized search algorithm used to solve problems quickly.
- We used random x and y coordinates to select the fittest.
- The algorithm selects the fittest based on the correlation function value.
- The algorithm follows the method of natural selection for the survival of the fittest after passing through the process of mutation and crossover.
- As we use the random method for the selection of the fittest so we cannot guarantee 100 percent accuracy, but it is an approximately optimized form of decision.

2.3 Features of Evolutionary Algorithm

Following are the six features of the evolutionary algorithm;

1. Representation
2. Selection
3. Recombination / Crossover
4. Mutation

5. Fitness Function
6. Survival Decision

2.3.1 Representation

Representation defines the individual and stores the parameters in an optimized way. Representation can be of the following types;

- Binary representation
- Real Value Representation

Representations are the stated problem's requirements and parameters that may vary from situation to situation.

2.3.2 Selection

Selection chooses the recombination of the fittest for the next generation.

2.3.3 Recombination / Crossover

Crossover or recombination decides how the genes can be combined of the selected fittest parents. In binary representation, it can be the value of the bits in the crossover while in real value representation, it can be the values of the genes.

2.3.4 Mutation

The mutation changes the bit/gene of every individual.

2.3.5 Fitness Function

The fitness function tells about the scalability of each individual according to correlation value.

2.3.6 Survival Decision

Further survival decision tells about the survival of the fittest one.

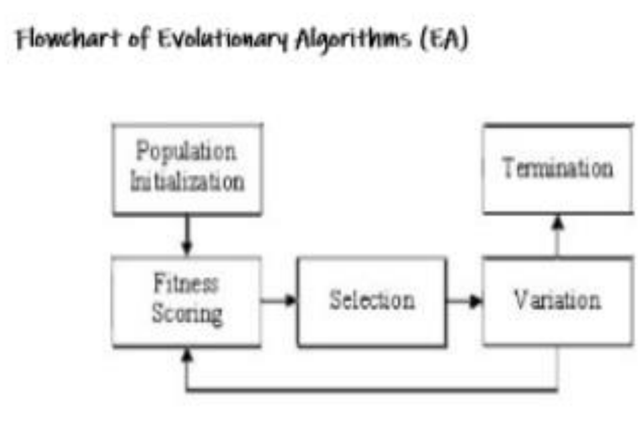
3. Coded Implementation of Algorithm

3.1 Technologies Used

We use python language for the coded implementation of the Evolutionary Algorithm. We use python libraries like **NumPy, Matplotlib, and Random**, which provide ease.

3.2 Computation Model Used

For the implementation of the algorithm, we followed all the features of the evolutionary algorithm and the evolutionary cycle flowchart provided by the instructor;



The coded implementation of the features of the evolutionary algorithm is given below;

Initialization

- Using randint () built-in python function, we generate random x and y coordinates, and these random x and y coordinates are our initial population.

```python
def Initialize_Population(Group_Row, Group_Col, Size):
    Current_Generation = []
    Count = 0
    while Count < Size:
        row = random.randint(Group_Row)
        col = random.randint(Group_Col)
        if row + boothi.shape[0] > Group_Row or col + boothi.shape[1] > Group_Col:
            Count = Count
        else:
            Current_Generation.append(tuple([row, col]))
            Count += 1
    return Current_Generation
```

Fitness Score / Evaluation

- In the fitness score, we create images from the points (x, y coordinates) according to the size of the boothi image (test image).
- Then we find the correlation of all images with respect to the test image.
- In the end, this function returns a dictionary containing the randomly generated points with their correlation value. Here key will be the point(x, y), and the value will be the correlation of that point with the test image.

```python
def Fitness_Evaluation(Current_Gen, Source_Img, Test_Img):
    Small_Img = []
    Corelation_Val = {}
    for i in range(len(Current_Gen)):
        Small_Img.append(Source_Img[Current_Gen[i][0]:Current_Gen[i][0] + boothi.shape[0], Current_Gen[i][1]:Current_Gen[i][1] + boothi.shape[1]]

    for i in range(len(Small_Img)):
        # Corelation = cv.matchTemplate(Small_Img[i], Test_Img, cv.TM_CCOEFF_NORMED)
        Corelation = correlation_coefficient(Small_Img[i], Test_Img)
        # Corelation_Val.append(Corelation[0][0])
        Corelation_Val.update({Current_Gen[i]:Corelation})

    return Corelation_Val
```

Selection

- The dictionary obtained from the fitness evaluation has been passed to selection.
- We sort the dictionary in descending order on the correlation value.

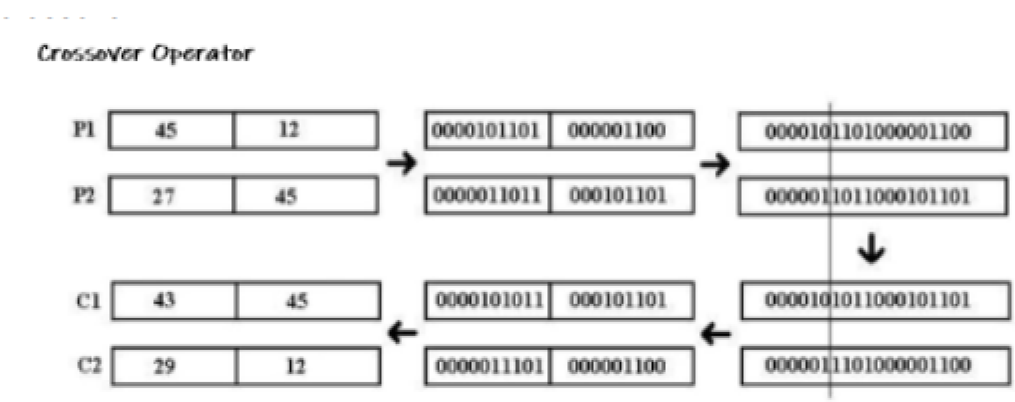
- Then we created a new dictionary where we stored the key value of correlation.
- This function will return the points with higher correlation values (ranked population) in the dictionary so that correlation value against each point can be accessed easily.

```
def Selection(Fitness_Values):
    Selected_Gen = {}
    for key, value in sorted(Fitness_Values.items(), key=lambda kv: kv[1], reverse=True):
        Selected_Gen.update({key : value})

    return Selected_Gen
```

Crossover and Mutation

- We passed the population with the higher correlation value to this function.
- Then it comes to the recombination stage. The recombination stage proceeds further through the binary conversion of the ranked population.
- Then we make the tuple of two different raked populations using their binary and then use the following binary cut technique given by the instructor.



- We convert the crossover binary into decimal and make a tuple containing the evolved population's value.

```python
def Mutation(Current_Gen_Dict):
    Current_Gen = []
    Mutated_Gen = []
    for key in Current_Gen_Dict.keys():
        Current_Gen.append(key)
    i = 0
    while i < len(Current_Gen)-1:
        j = i + 1
        Bin_Point1 = Binary_9_Bits(Current_Gen[i][0])
        Bin_Point2 = Binary_10_Bits(Current_Gen[i][1])

        Bin_Point3 = Binary_9_Bits(Current_Gen[j][0])
        Bin_Point4 = Binary_10_Bits(Current_Gen[j][1])
# print(Bin_Point1)
# print(Bin_Point2)
        L1 = Bin_Point1 + Bin_Point2
        L2 = Bin_Point3 + Bin_Point4

        random_point = random.randint(19)
# print(random_point)
        Slice1, Slice2 = L1[:random_point], L1[random_point:]
        Slice3, Slice4 = L2[:random_point], L2[random_point:]
# print(Slice1)
# print(Slice4)
        Cross_Over1, Cross_Over2 = Slice1 + Slice4, Slice2 + Slice3

# print(Cross_Over1)
        New_X1, New_Y1 = Cross_Over1[:9], Cross_Over1[9:]
        New_X2, New_Y2 = Cross_Over2[:9], Cross_Over2[9:]
```

```python
        num1 = ""
        num2 = ""
        num3 = ""
        num4 = ""
        for elem in New_X1:
            num1 = num1+str(elem)

        for elem in New_Y1:
            num2 = num2+str(elem)

        for elem in New_X2:
            num3 = num3+str(elem)

        for elem in New_Y1:
            num4 = num4+str(elem)

        # print(New_X1)
        # print(New_Y1)
        New_X1 = int(num1,2)
        New_Y1 = int(num2,2)
        New_X2 = int(num3,2)
        New_Y2 = int(num4,2)

        if New_X1 + 35 > 512 and New_Y1 + 29 > 1024:
            i = i
        else:
            Mutated_Gen.append(tuple([New_X1, New_Y1]))
            i += 2
        if New_X2 + 35 > 512 and New_Y2 + 29 > 1024:
            i = i
        else:
            Mutated_Gen.append(tuple([New_X2, New_Y2]))
```

Algorithm Termination

- The program needs to end at some point, and this point will be the highest value of correlation which we set as a threshold.
- We give a threshold of 0.85. When the algorithm reaches the given threshold, it will be terminated.
- The termination returns the value of the fittest one, which will be the final solution.

4. Experimentation

4.1 Hypothesis # 01

An increase in generation size offers a high chance of finding the threshold value for the test image.

4.1.1 Experiment for Hypothesis # 01

First Attempt

In first attempt, we keep the population size 100.

Second Attempt

In second attempt, we keep the population size 500.

Third Attempt

In third attempt, we keep the population size 3000.

Analysis

From the above attempt, we analyzed an equation for our hypothesis Generation size is directly proportional to fittest image. An increase in generation size offers a high chance of finding the threshold value for the test image. As we had fewer generation sizes at the start of the experiment, that gave us fewer fittest pictures, but when we gradually increased our generation size then, we saw a clear difference in the fittest images (increase in fitness).

Generation Size ∝ Fittest Image

4.2 Hypothesis # 02

If we increase the threshold value towards maximum and also increase in generation size, in this case, we think we will see some different results. Hopefully, we will receive the desired output (find boothi in the group image).

4.1.2 Experiment for Hypothesis # 02

First Attempt

In the first attempt, we set the threshold value at 7.

We set generation size at 700.

Second Attempt

In the second attempt, we set the threshold value at 8.

We set generation size at 850.

Third Attempt

In the third attempt, we set the threshold value at 9.

We set generation size at 1000.

Analysis

From the above experiment, we observe that if we increase the threshold and the generation size, the chances of finding boothi increase quite impressively. In the first attempt, the chances increase slightly when we set the threshold value at 7 and the generation size at 700. The result is quite impressive in the second attempt compared to the previous attempt. In the last attempt, when we set the threshold value at nine and the generation size to 1000, we saw an excellent result in which the chances of our success in finding boothi in the group image would increase to almost

80 to 90 percent, which is quite impressive.

5. Limitations

- As this is a computational algorithm, a lot of computation will be made. It consumes a lot of memory in our system.
- This algorithm is based on randomness. We will get the same correlation value for some points. For example, we can get the maximum fitter value for 100 generation points; sometimes, we get the same fitter point for 500 generation points.

6. Conclusion

We learned a lot in this project. We also use the concept of artificial intelligence for the first time in our life. Overall it was a good experience for both of us. However, we faced many problems during the implementation of the evolutionary algorithm, but with time and experience we learned how to initialize and solve the given problem. The whole experience of this project changed our way of thinking.
We solve this problem as computer scientists, which itself a good practice and an achievement for us to think that way and solve this exciting problem. As a computer scientist, our approach is quite different as compare to our previous problem solving techniques. This time we solve the problem in a structural way as the scientific way of solving a problem.
Our code can be improved more and more, and many bugs can be removed from this code.
Our code can also be concise and optimized. One thing which we missed in this project is the use of graphs to explain the behavior of our new evolved population. Nevertheless, we tried our best to explain everything in this report.s

7. Reference:

https://www.yourgenome.org/facts/what-is-evolution.

https://connect.collins.co.uk/repo1/Content/Live/Infuze/COL/GCSE_Science_Core_SB_OCR_Gateway/content/Page78.html

https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/darwins-finches