



BIRZEIT UNIVERSITY

Electrical and Computer Engineering Department

ENCS434 Artificial Intelligence

First Semester, 2021-2022

Programming Project 1

Prepared by:

Jihan Alshafi 1181641

Fatima Shrateh 1181550

Instructor: Dr. Aziz Qaroush

Section: 1

Date :12-12-2021

Project idea:

This programming project can be viewed as an application of searching search algorithm in real world problems

Genetic Algorithm :

Genetic algorithm is a part of evolutionary computing, it's rapidly growing area of artificial intelligence, it starts with a set of solutions we called population.

What is the population ?

The initial generation is populated with completely randomly generated populations , and uniformly distributed across the search space . The population which is a set of Solutions came from one population which has taken and used to form a new population, as there is a chance that the new population will be better than the old one, and so on.

Next step : Find the fitness

we need a function to calculate their value in a way that allows us to compare values of two different genomes. This function is called a *fitness function* and we can denote it as $f(x)$, It's usually always positive, and the larger the number the better the genome

Next step : Crossover

The goal of creating a new generation is to pass along the good attributes of the last generation, but create new variations in order to try and further improve the fitness. To do this, we perform a crossover operation. This done by taking two parent genomes which have the best (highest) fitness value and creates a number of child genomes (one or more), the more children have be made , the best solution we get.

There are many types of crossovers:

single-point crossover , two-point crossover, k-point crossover , uniform crossover and it could be a special crossover designed to satisfy a particular problem

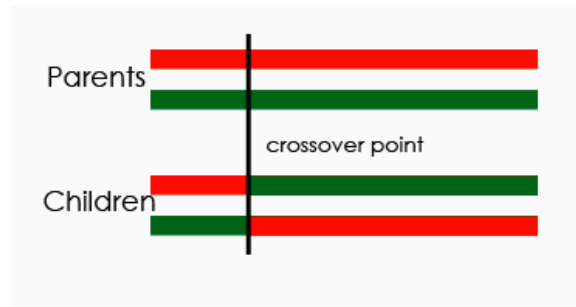


Figure 1 :Single point Crossover

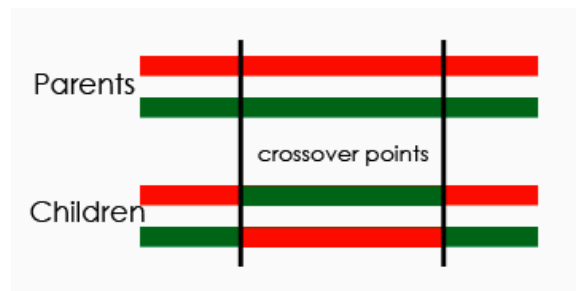


Figure 2:two point crossover

Next step : Mutation :

convergence still happens sometimes in the children that came by crossover two parents, because of the random nature of genetic algorithms. And it has to lower the probability that it will happen even more, for this we mutate genomes within a new generation with a certain probability .

Mutation can be done by replace each duplicate elements in the child , with a unique one that it isn't exist to get a new children with a unique elements and better than the one before.

Termination:

We keep building new generations until we reach one condition of these:

The best genome has satisfied the minimum criteria by the objective function, or if we reached a preset maximum number of generation or if The algorithm has exceeded maximum running time or if we got the best genome so that there will no be a better results than.

how we designed and implemented our program?

The used programming language: Python

In the beginning , Random function is generated 10 random lists , each list has a length of 36 , that this is the number of student groups , and we assume that the index of each element is represent the number of the group which has that project , each list of this contains a random number in the range 1 to 38 , that is 38 is the number of the available projects , that is each student is allowed to take one project only by giving us his 3 selections , the first has the most priority , it should be remind that it is NOT allowed that more than a group take the same project .

Next, our program reads the data from an excel file called “Students+selections.xlsx” , the reading technique is depending on read cell by cell , that is it start reading from cell C2 , that it’s the first value (the first selection for the first group) in that cell, it starts reading C2 , D2, E2,C3,D3,E3 and so on .

The range of the for loop is defined from C2 to the sheet’s maximum number or rows. The second for loop searches within predefined column names , “CDE” , but we separate the columns name and set a loop for each so that we can find the fitness directly after reading the cell , that C represent the first selection , and if the value of random list duplicate the one in the excel file, the fitness value will be the highest for this column , we assume for each duplicate in column C , 30 points will be given to the fitness value . We separate D column in a single loop , that is in each duplicate value from random list with excel file , 20 points will be given to the fitness value , in the last for E column , 10 points will be given to fitness value if there is a duplicate as we explain.

After reading the whole file, and giving the fitness point for each value of C , D and E , we find the total fitness value for this random list .

All this will be repeated for the 9 last lists , and find their fitness values to choose the best two parents from them (which has the highest fitness value) .

Why the highest fitness value ? The ideal case will happen when all the groups take the first choice that they've chosen , in this case each cell of C will get 30 points to the fitness value until the total fitness value = $30 * 36 = 1080$, and this case will never be happen when generating the random list , that is there are many groups have chosen the same first choice which is in the end , just a one group will take a project , Other group will not get it.

Next step is the crossover , we choose the single point crossover for our program, that after choosing the best 2 parents , that have the highest fitness value , we will take the first half values from the first parent with the second half of the second parent to get the first CHILD , and the first half of the second parent with the second half of the first parent to get the second CHILD , in this step , we got those 2 children with Maintaining the index values for each element in the child (the element index in his parent is the same index when this element is in the child)

It is possible that many elements in the final children will be duplicate in the same list , and this is not allowed (as we mentioned before) , because of this we have to make all the list has a unique elements , by replacing the duplicate element with another is not exist in it , and this operation called Mutation .

Now , after we have 2 children with unique elements , we have to find the fitness value for each, it has supposed to be that their fitness value is better than their parents .

In the end , those 2 children are a solution , but to improve them to a better solution , we make a crossover between them , to get the same steps again as before . If the fitness value of these final children is better , we are going to make crossover again , otherwise the 2 children we got will be the best solution.

The solution will be the child which had the highest fitness value !

The GUI :

This is the Graphic User Interface for our program, the user should enter the student excel file using the browse button, then choose the excel file he would to print the solution in by pressing Save As , and then click Run button to run the program .

- Note : we didn't connect the GUI with our code , but the Browse and Save As buttons is working as fig 4 below :

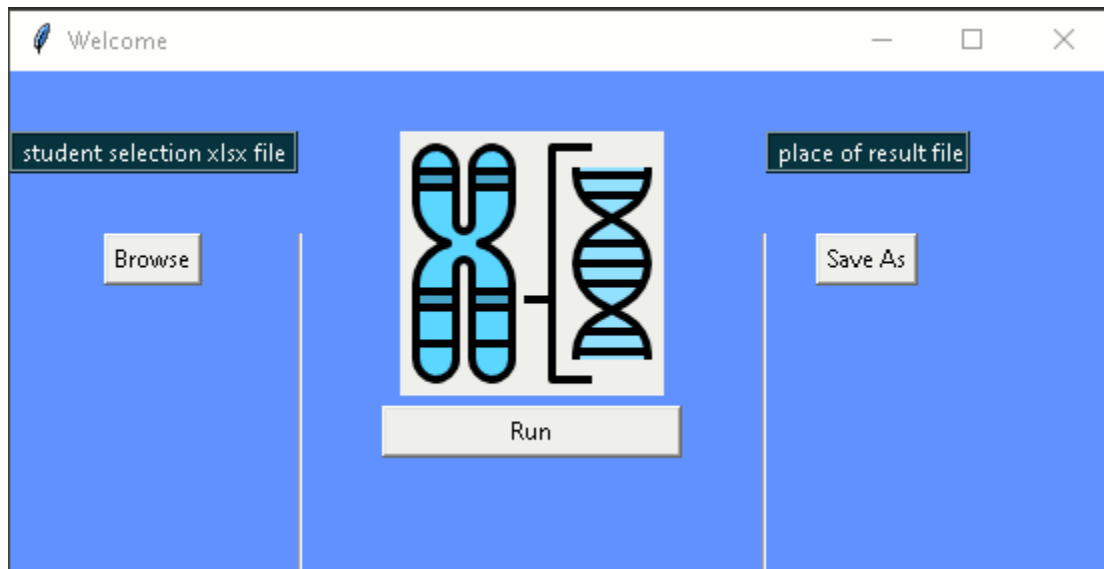


Figure 3:GUI

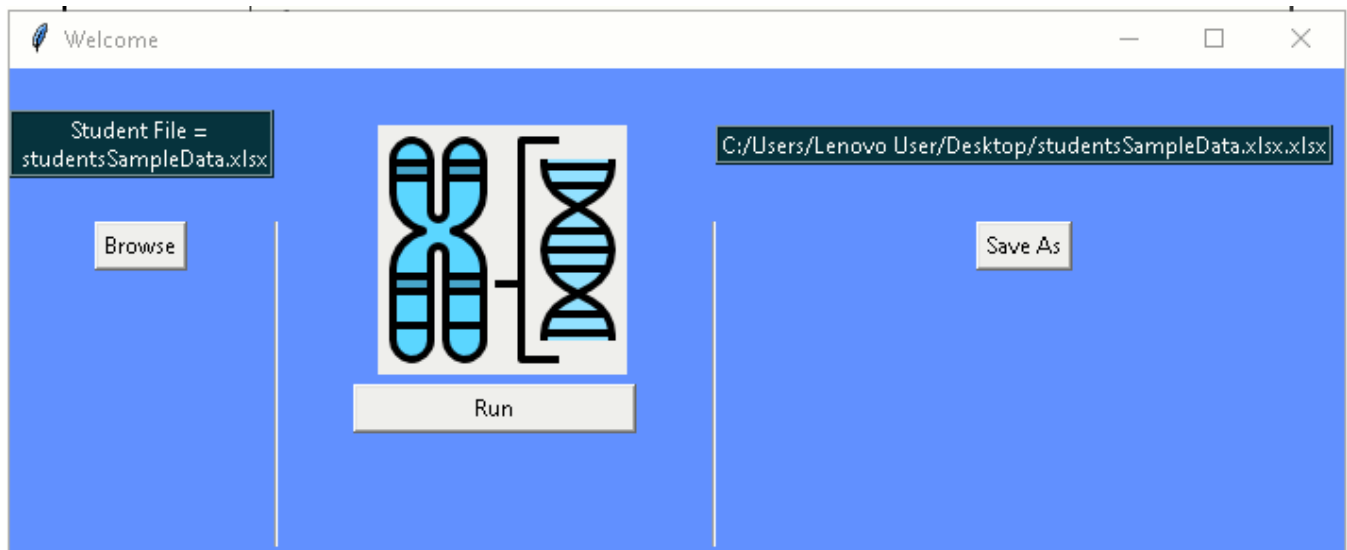


Figure 4:GUI take the input files

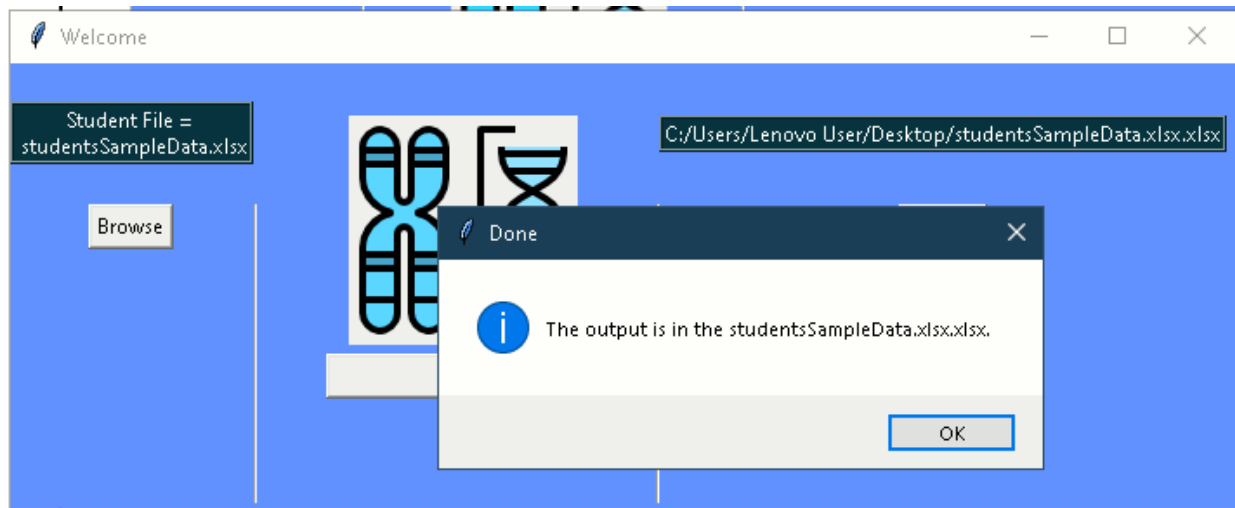


Figure 5:GUI shows a Done message after pressing RUN