BIRZEIT UNIVERSITY

Faculty of Engineering & Technology – Electrical & Computer Engineering Department

First Semester 2022 – 2023

Artificial Intelligence – ENCS3340

Class Distribution Project (Genetics Algorithm)

Prepared By:
Ahmaide AlAwawdah – 1190823

Ali Mohammad – 1190502

Khalid Addabe – 1190507

Instructor:
Mr. Aziz Qaroush

Section: 1

Date: December 2022 – January 2023

Abstract

The main objective of this project is to give the university students in the electrical and computer engineering department the widest choices while making their yearly semester schedule with the best possible solutions for courses sections times in the course browser.

The algorithm that is used in this project in order to optimize the students schedule choices is the genetics algorithm, that will hope to get good solutions when giving time slots for the department courses resulting in minimum conflicts, better results, and more flexibility.

❖ Table of Content

❖ Table of Figures

1. Problem Specification

In this project the problem is specified to be the timetabling for all the available department’s courses constrains that can result in many issues and conflicts.

These constrains can be either hard or soft constrains, as the hard constrains are the ones that are absolutely essential, where the soft ones are considered to be the ones that are not enforced yet still desirable

The available resource is simple the department’s course browser for the fall semester of the year 2022 – 2023, that can be accessed using the link below:
https://ritaj.birzeit.edu/hemis/courses?term=1221&bu=7261&lang=ar&mode=B

The department’s timetable contains elements such as: courses, sections, timeslots, instructors. As the input information are known as below:

- The courses can be either lecture classes or labs, and each lecture course or lab has its own sections with each one has it’s given instructor.
- Each type of course (lecture or lab) can have a time slot from its own set of time slots.
- Lecture Courses time slots contains two days that the lecture is given at these two days can be any of the three (Saturday, Monday, Wednesday) or the two days can be Tuesday and Thursday, and for the lecture time it is 1 hour and 15 minutes with six possible courses times in the pair of two days.
- Labs time slots contains one days that the lab is given at, a lab section can be on any day of the week, and its lecture time is 2 hours and 40 minutes.

2. Project Description

When using an algorithm to solve a problem the algorithm components need to be specified for that problem in order to get to a conclusion of how the algorithm will solve this problem.

For this project as the genetics algorithm will solve the problem of the course distribution over the available time slot, the genetics algorithm components need to be specified by how they are initialized for the department's courses to figure out how it can help in finding and optimizing the solution for this problem.

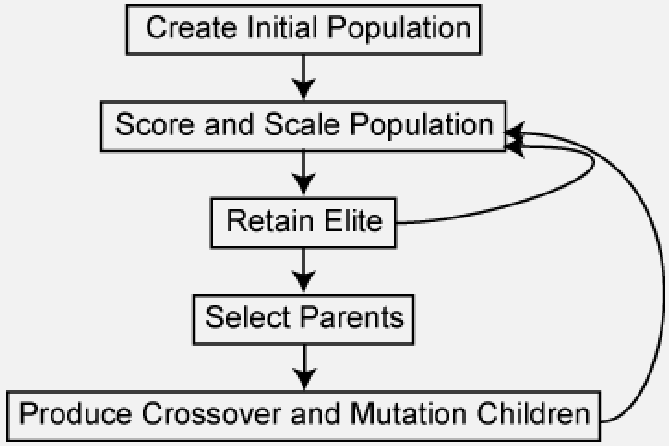

Figure 2-1: Genetics Algorithm Steps

3.1. The Chromosome Representation

To solve the timetabling problem using genetic algorithm, the schedule elements need to be represented as a chromosome. In our chromosome representation, every gene header represents a course's section. The value of gene is a sequence of Time Slots.

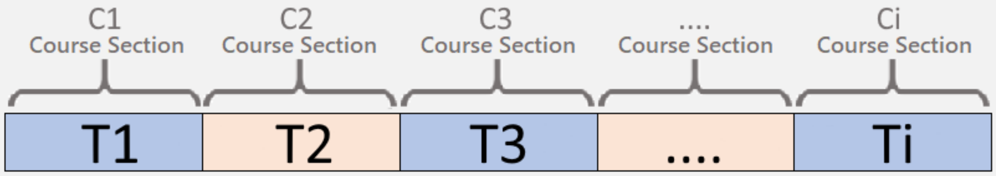

Figure 2-2: Chromosome Representation

Where Ti is the ID of time slot, and Ci is the course ID.

According to the representation in the chromosome, the length of the chromosome is given by the following formula:

$$Chromosome_{length} = Number\ of\ courses\ sections$$

3.2. The Population

An initial population of size 2000 is generated. This size is used to reach the highest fitness, so it is experimentally chosen. However, the size is big due to the number of chromosome's genes which needs more random solutions to cover an acceptable solution.

Each section is assigned randomly to one time slot. The population of this genetic algorithm grows till a maximum size of 3000. When the population size exceeds 3000, a number of chromosomes are removed according to criterion discussed later to maintain the size limit to 3000 chromosomes.

3.3. Parents Selection

Two chromosomes are selected every iteration to generate two new children (off-springs). The tournament algorithm is used to determine the parents. In this algorithm, a random sub-group is selected from the population and the best two chromosomes of it are chosen. The size of the sub-group here is 20% of the population size.

3.4. The Crossover

Coming up with new solutions can come up from existing solutions, by choosing two chromosomes (solutions) as the parents, two new chromosomes known as children can be generated by them as they'll have mixed genes that are bought from both parents, as if one child gets a certain gene from a parent that other child will get the gene from the second parent and so on (as both parents have the same chromosome length).

In this project three methods of crossover between the courses sections time slots are used (to determine the mixture of the taken genes from the parents), these three methods are:

1- **One Point Crossover:**

One point crossover is simply done by choosing an index in the chromosomes where child one will get the genes of the first parent before that index and genes of the second parent after the index, child 2 will have it the other way around, as shown in figure 1-5 below.

2- Two Point Crossover:

Similarly, to the first method this method works on flipping by indexes, yet this method uses two points indexes, as child one gets the genes of the first parent before the first index, after the first index and until the second index child one gets the genes from the second parent, and after the second index it goes back to taking the genes of the first parent, where child two takes from the opposite parent of child one.

3- Uniform Crossover:

This method of crossover goes on every gene and generates a random binary bit if the generated bit is 0, child one gets the gene from parent one and child two from parent two, if the generated bit is 1, child one gets the gene from parent two and child two from parent one.

The three types of crossover are all displayed in the figure below.

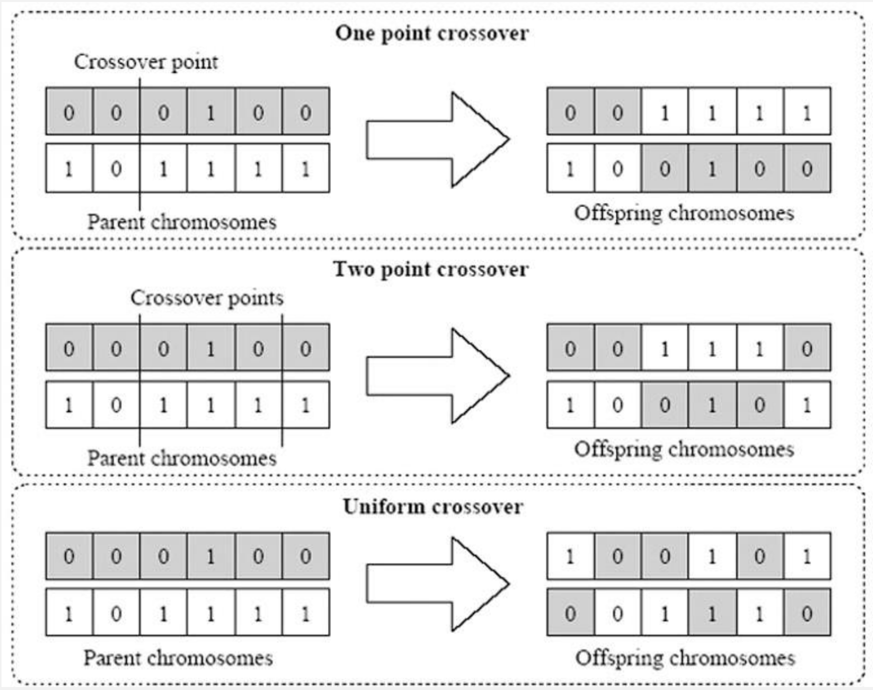

Figure 2-3: Types of Crossovers

3.5. The Mutation

The mutation is a process in the genetics algorithm that changes the value in a random gene of a specified chromosome in the population (random change in a time slot of a random course section) as mutation can be used to avoid getting stuck.

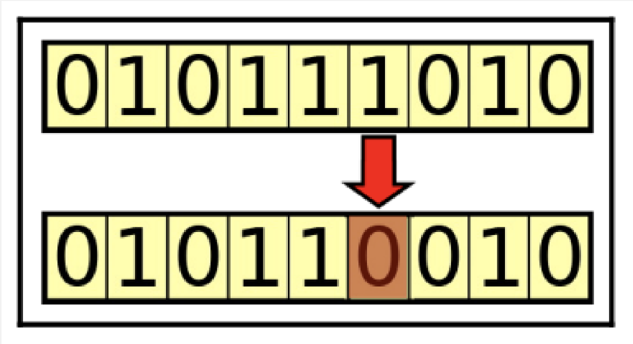

Figure 2-4: Mutation

3.6. Fitness Function

Fitness function evaluates how close a given solution is to the optimum solution of the desired problem. Each chromosome is given a score out of maximum fitness 100%. 60% of this fitness is given to the hard constraints (equally weighted -15% for each) and 40% to the soft constraints (distributed subjectively between them). Soft fitness is not computed till all hard constraints are satisfied. Otherwise, their fitness value has no meaning.

Hard Constrains

1. There are no conflicts in assigning courses for the same instructor. The mathematical measurement of this constraint is defined by the equation:

$$fitness = \left(1 - \frac{\#\ of\ conflicted\ sections}{\#\ of\ courses}\right) * 60\%$$

Soft Constrains

1. Minimizing the number of courses of the same level that have the same time. Let L be the number of levels, Xi the number of conflicted sections in level i, and Si number of sections in level i.

$$fitness = \left[\left(1 - \frac{\left\{\sum_{i=0}^{L} \frac{Xi}{Si}\right\}_{ENEE}}{L}\right) * 5\%\right] + \left[\left(1 - \frac{\left\{\sum_{i=0}^{L} \frac{Xi}{Si}\right\}_{ENCS}}{L}\right) * 5\%\right]$$

2. Minimizing the number of courses of the same level that have the same days. Let L be the number of levels, Xi the number of conflicted sections in level i, and Si number of sections in level i.

$$fitness = \left[\left(1 - \frac{\left\{\sum_{i=0}^{L} \frac{Xi}{Si}\right\}_{ENEE}}{L}\right) * 5\%\right] + \left[\left(1 - \frac{\left\{\sum_{i=0}^{L} \frac{Xi}{Si}\right\}_{ENCS}}{L}\right) * 5\%\right]$$

3. Minimizing the number of course sections that have the same days:

$$fitness = \left(1 - \frac{\#\ of\ sections\ that\ have\ the\ same\ days}{\#\ of\ all\ sections}\right) * 10\%$$

4. Minimizing the number of course sections that have the same time:

$$fitness = \left(1 - \frac{\#\ of\ sections\ that\ have\ the\ same\ days}{\#\ of\ all\ sections}\right) * 10\%$$

3.7. Iterations

An iteration is a cycle in the genetics algorithm, as the genetics algorithm has a number of maximum cycles (iterations) if it didn't get the best or asked fitness.

Each iteration it decides to do a crossover and gets two new chromosomes as children where the two best fitness available chromosomes in the population are chosen as parents, there is a percentage for doing the crossover in this project the percentage of a crossover is 60% for each iteration, and the type of crossover is chosen randomly for each iteration.

Each iteration can also decide to do a mutation of a specific chromosome in the population, the percentage of a mutation is 20% for each iteration.

At the end of an iteration if the population exceeded its maximum size 20% of it is removed, and then it starts the next iteration.

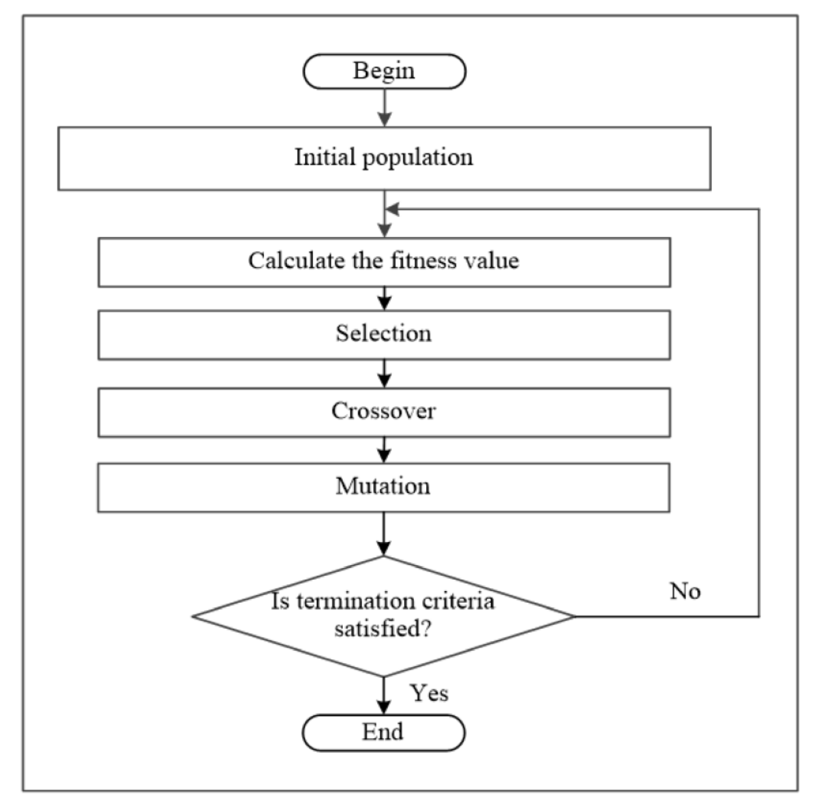

Figure 2-5: Iteration Process

3. Running The Program

3.1. Optimization targets

3.3.1. Depending on the number of conflicts in time and days for the same section

The aim of this optimization target is to distribute course sections to all available days and time which makes the students choose their courses freely since sections got different days and times.

However, the best fitness of this target is 97.2, it's reached by 40000 iterations.

The following is some of the optimization outputs:

ENCS2110: DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY

Section	Instructor	Day	Time
1	Jamal Othman Hosen Seyam	S	8:30-11:10
2	Ibrahim Ahmed Abdallah Nemer	T	8:30-11:10
3	Ismail Mohammad Mousa Khater	M	11:25-14:05
4	Adnan Hussein Mahmoud Yahya	M	14:15-16:55
5	Ismail Mohammad Mousa Khater	S	14:15-16:55
6	Adnan Hussein Mahmoud Yahya	T	11:25-14:05
7	Khader Shehadeh Nassar Mohammad	W	8:30-11:10
8	Bilal Jafar Hamed Karaki	R	8:30-11:10

Figure 3-1: Output 1

ENCS3130: LINUX LABORATORY

Section	Instructor	Day	Time
1	Mohammad Khaleel Issa Jubran	S	8:30-11:10
2	Mohammad Khaleel Issa Jubran	W	11:25-14:05
3	Khader Shehadeh Nassar Mohammad	T	11:25-14:05
4	Aziz Mohammed Ahmad Qaroush	M	11:25-14:05
5	Mohammad Khaleel Issa Jubran	R	11:25-14:05

Figure 3-2: Output 2

ENCS5322: NETWORK SECURITY PROTOCOLS

Section	Instructor	Day	Time
1	Ahmad Saleh Ibrahim Alsadeh	MW	14:15-15:30

ENCS5332: VLSI DESIGN

Section	Instructor	Day	Time
1	Khader Shehadeh Nassar Mohammad	SM	10:00-11:15

ENCS5341: MACHINE LEARNING AND DATA SCIENCE

Section	Instructor	Day	Time
1	Yazan Jamal Ibrahim Abu Farha	SW	15:40-16:55

ENCS5399: SP.TOP: VERIFICATION AND VALIDATION OF HARDWARE

Section	Instructor	Day	Time
1	Ayman Ali Mohammad Hroub	SW	8:30-9:45

Figure 3-3: Output 3

Hence, not all results are very satisfying since the optimization is done for each course. However, conflicts may appear between different courses.

The following is the graph of the genetic algorithm (Fitness vs. Iteration).

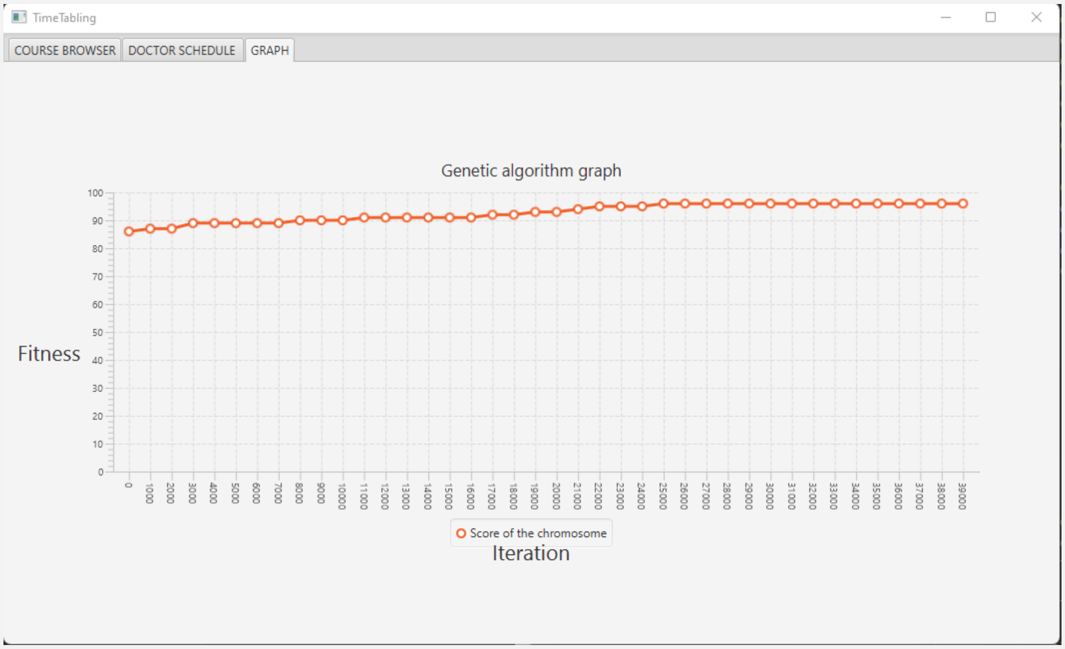

Figure 3-4: Fitness Graph 1

3.3.2. Depending on the number of conflicts in time and days for the same section, and the conflicts between the same level courses in days and time.

However, in this target the optimization is done for an advanced level which ensures that the student can register his year's courses without conflicts, but the fitness is hard and the optimization didn't give a satisfying result.

ENCS2110: DIGITAL ELECTRONICS AND COMPUTER ORGANIZATION LABORATORY

Section	Instructor	Day	Time
1	Jamal Othman Hosen Seyam	W	11:25-14:05
2	Ibrahim Ahmed Abdallah Nemer	T	14:15-16:55
3	Ismail Mohammad Mousa Khater	M	8:30-11:10
4	Adnan Hussein Mahmoud Yahya	S	14:15-16:55
5	Ismail Mohammad Mousa Khater	S	8:30-11:10
6	Adnan Hussein Mahmoud Yahya	M	14:15-16:55
7	Khader Shehadeh Nassar Mohammad	T	8:30-11:10
8	Bilal Jafar Hamed Karaki	R	8:30-11:10

Figure 3-5: Output 4

ENCS2340: DIGITAL SYSTEMS

Section	Instructor	Day	Time
1	Ismail Mohammad Mousa Khater	TR	15:40-16:55
2	Ismail Mohammad Mousa Khater	TR	11:25-12:40
3	Mohammed Sami Abdul Karim Hussein	TR	12:50-14:05
4	Bilal Jafar Hamed Karaki	SW	15:40-16:55
5	Anjad Jamal Rushdi Badran	SM	11:25-12:40
6	Ibrahim Ahmed Abdallah Nemer	MW	8:30-9:45
7	Ali Hasan Moath Abdo	SW	12:50-14:05
8	Anjad Jamal Rushdi Badran	MW	8:30-9:45

Figure 3-6: Output 5

ENCS3310: ADVANCED DIGITAL SYSTEMS DESIGN

Section	Instructor	Day	Time
1	Abdallatif S. A. Abuissa	TR	8:30-9:45
2	Abdallatif S. A. Abuissa	SW	12:50-14:05

ENCS3320: COMPUTER NETWORKS

Section	Instructor	Day	Time
1	Abdalkarim Ishaq Ibraheem Awad	TR	11:25-12:40
2	Ibrahim Ahmed Abdallah Nemer	SW	15:40-16:55
3	Abdalkarim Ishaq Ibraheem Awad	MW	8:30-9:45

Figure 3-7: Output 6

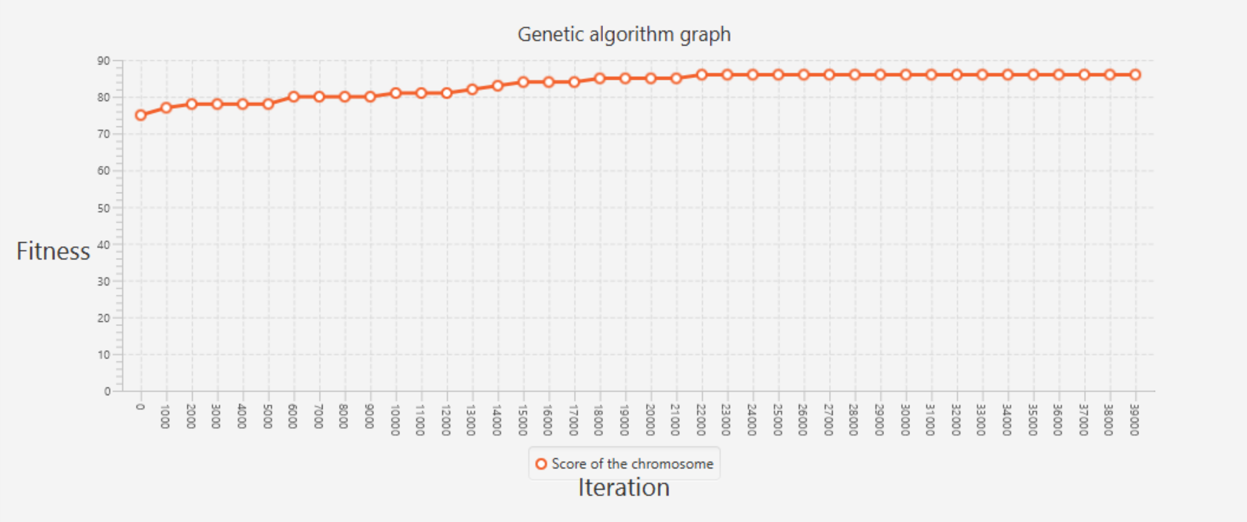

Figure 3-8: Fitness Graph 2

3.2. Performance

The performance of the program is measured by satisfaction of its conditions, the hard constraint (the instructors conflicts) is satisfied 100% in around 15000 iterations, after that as shown in figure 3-8 the fitness value starts to settle down as the most needed constrain is observed, as what remains by that point is to find the best solution out of the hard fitness satisfied solutions with the best fitness for the soft constrains until it reaches the maximum fitness or the max number of iterations (40000), by that time the fitness is settled to a value which is most likely around 90% as shown in figure 3-8.

4. Conclusion

In conclusion this project shows the effective performance and the importance of the genetics algorithm in the world of optimization, the algorithm resulted in a solution that satisfies the requirements and has excellent results that gives more than 90% of the maximum fitness.

The genetics algorithm is used in many applications today in order to optimize their outcomes, such as: medical science, economics, image processing, finding hardware bugs, and more, which shows how effective and useful this algorithm is.

5. References

[1]. Genetics Algorithm – mathWorks.com:

https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html

Accessed on January 14th 2023, at 8:51 PM

[2]. Fitness Function – jameskle.com:

https://jameskle.com/writes/genetic-algorithms

Accessed on January 14th 2022, at 9:01 PM

[3]. Population – kindsonthegenius.com:

https://www.kindsonthegenius.com/basics-of-genetic-algorithm-ga-explained-in-simple-terms/

Accessed on January 14th 2022, at 10:16 PM

[4]. Types of Crossovers – kindsonthegenius.com:

https://www.kindsonthegenius.com/basics-of-genetic-algorithm-ga-explained-in-simple-terms/

Accessed on January 14th 2022, at 10:40 PM

[5]. Mutation – analyticsvidhya.com:

https://www.analyticsvidhya.com/blog/2021/06/genetic-algorithms-and-its-use-cases-in-machine-learning/

Accessed on January 14th 2022, at 10:53 PM

[6]. Iterations – mdpi.com:

https://www.mdpi.com/2073-8994/12/11/1758

Accessed on January 14th 2022, at 11:05 PM