Project report

For

Solving a Faculty's Timetable Scheduling Problem

Using Genetic Algorithms

Maryam Sherif Fouad (Team leader)

ID: 20208221

Nada Muhammed Nofal

ID: 20208252

Farah Tarek Muhammad

ID: 20208177

Antonious Awad Ibrahim

ID: 20208051

Kareem Ragab Abd Elaziz

ID: 20208179

Level: 3

Department: Medical informatics

1-introduction & overview

1.1 **Project idea:** Solving a Faculty's Timetable Scheduling Problem using Genetic Algorithms

1.2 **Project overview:** A very famous scenario where Genetic Algorithms can be used is the process of making timetables or timetable scheduling. Consider you are trying to come up with a weekly timetable for classes in a college for a batch/class. We must arrange classes and come up with a timetable so that there are no clashes between classes. Here, the task is to search for the optimum timetable schedule. A possible definition for the problem is: Given a set of lecturers, a set of courses on individual topics and a Course Requirements matrix with integer elements representing the number of hours a lecturer teaches a course during each week, the problem is to allocate times to these hours so that a student may take as many suitable combinations of courses as possible. Or, simply to create a practical timetable for a whole faculty in which courses offered by different departments may be combined in various ways to suit individual students.

1.3 **Similar applications**:

- PeñaLara: (web/desktop application): **GHC** School Timetable Software allows you to set the strict conditions and preferences you need to achieve your optimal timetable, as well as to complete, display or transfer it.

GHC consists of 3 clearly different parts:

1- **Planner**

From this component, all the options needed to create the timetable are configured: availability of teachers and their preferences, classrooms, characteristics of the class units, groups of students, etc.

2- **Engine**

The engine provides, if available, a solution very quickly. It also helps you to analyze those conditions that are impossible to satisfy. The GHC engine finds the solution according to the criteria you set.

3- **Editor**

Each solution generated by the engine is displayed in the GHC editor through different views. The editor warns you if any conflicts appear when making modifications. It supports a variety of formats for presenting or exporting the timetable to other applications.

The **GHC** editor is a powerful tool that allows you to finally add, adjust and modify class units in a drag-and-drop assisted way. You will also be able to add meetings, on-call hours or other teacher activities.

Your **GHC** subscription includes additional tools for exchanging configuration data and timetable data with other applications. **GHC WEB APP**, by using user profiles, allows collecting preferences by department, having a teacher absence manager and coordinating the day-to-day running of timetables.

1.4 literature review:

- **Solving timetable scheduling problem using genetic algorithms**

B. Sigl, M. Golub, V. Mornar

Published 16 June 2003

Computer Science

Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.

A genetic algorithm for solving a timetable scheduling problem is described. The algorithm was tested on small and large instances of the problem. Algorithm performance was significantly enhanced with modification of basic genetic

operators. Intelligent operators restrain the creation of new conflicts in the individual and improve the overall algorithm 's behavior.

- **Solving Timetabling Problem Using Genetic and Heuristic Algorithms**

N. D. Thanh

Published 2007

Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)

In this paper, we propose a hybrid algorithm that combines genetic and heuristic approach. By using this method, solving timetabling problem is converted to finding the optimal arrangement of elements on a 2D matrix. This algorithm was implemented and tested with the synthetic and real data of Nong lam University of HCM City, Vietnam. The experimental results reveal the usability and potential of the proposed algorithm in solving timetabling problems.

- **Optimize Timetabling Problem Using Improved Genetic Algorithm**

- Wang Yun, W. Kun, Wang Xiang Yun
- Published 1 December 2008
- Computer Science
- 2008 IEEE International Symposium on Knowledge Acquisition and Modeling Workshop

The arrange timetable algorithms in common need to further improve. Aiming at the problem, classes divide into groups and elitist strategy with dissimilarity chromosome methods based on genetic algorithms (GA) was proposed. Compared with standard genetic algorithms by simulation, its efficiency was demonstrated. The experimental results show that it illustrates a good prospect of application and extension

- **Interactive timetabling system using genetic algorithms**

- H. Kanoh, Y. Sakamoto
- Published 10 October 2004
- Computer Science
- 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)

This paper discusses a new solution to university course timetabling problems. Problems that belong to the NP-hard class are very difficult

to solve using conventional optimization techniques. Our solution methodology is based on genetic algorithms which use an installed knowledge base. The knowledge here is a set of candidate partial solutions of the final solution. The proposed method is to use both a knowledge base and constraints to solve the problems efficiently. The timetables obtained can satisfy teachers' personal requests and present the advantages of past timetables. Experiments using timetables of University of Tsukuba showed that this approach is an effective solution method. The proposed method includes general techniques concerning the use of domain specific knowledge that can be applied to a variety of large-scale real-life combinatorial optimization problems.

- **An Application of Genetic Algorithm for University Course Timetabling Problem**

- R. Sanjay, S. Rajan
- Published 6 August 2016
- Computer Science
- International Journal of Applied Information Systems

Timetabling problem is a process of assigning given set of events and resources to the limited space and time under hard constraints which are rigidly enforced and soft constraints which are satisfied as nearly as possible. As a kind of timetabling problems, University course timetabling is a very important administrative activity for a wide

variety of institutes. Genetic algorithm is an advanced heuristics method which is very effective in many areas. It is frequently deployed meta-heuristics algorithm to solve difficult combinatorial optimization problems. In this paper, genetic algorithm is used to solve university course timetabling problem. At first, a model of problem to be solved is defined. Then, the genetic representation is determined and a fitness function is established according to the constraints. Finally, a case of university course timetabling from real-world is discussed and solved. It is demonstrated that the method proposed in this paper is feasible and efficient.

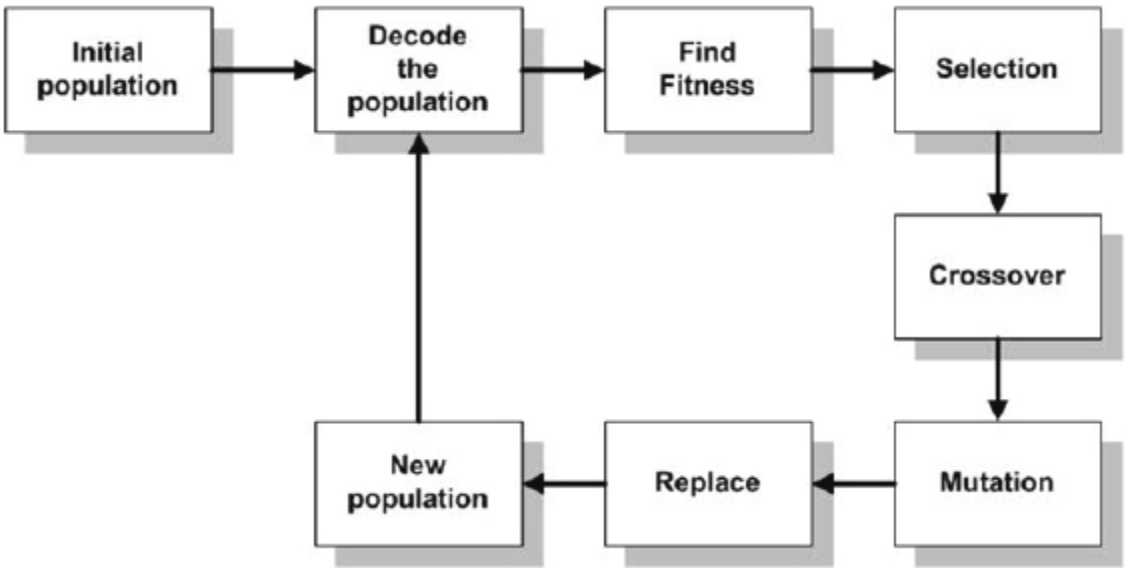
2-Applied algorithms:

2.1 Genetic algorithm: Genetic Algorithms (GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.

Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next

generation. In simple words, they simulate "survival of the fittest" among individual of consecutive generation for solving a problem. Each generation consist of a population of individuals and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

Block diagram



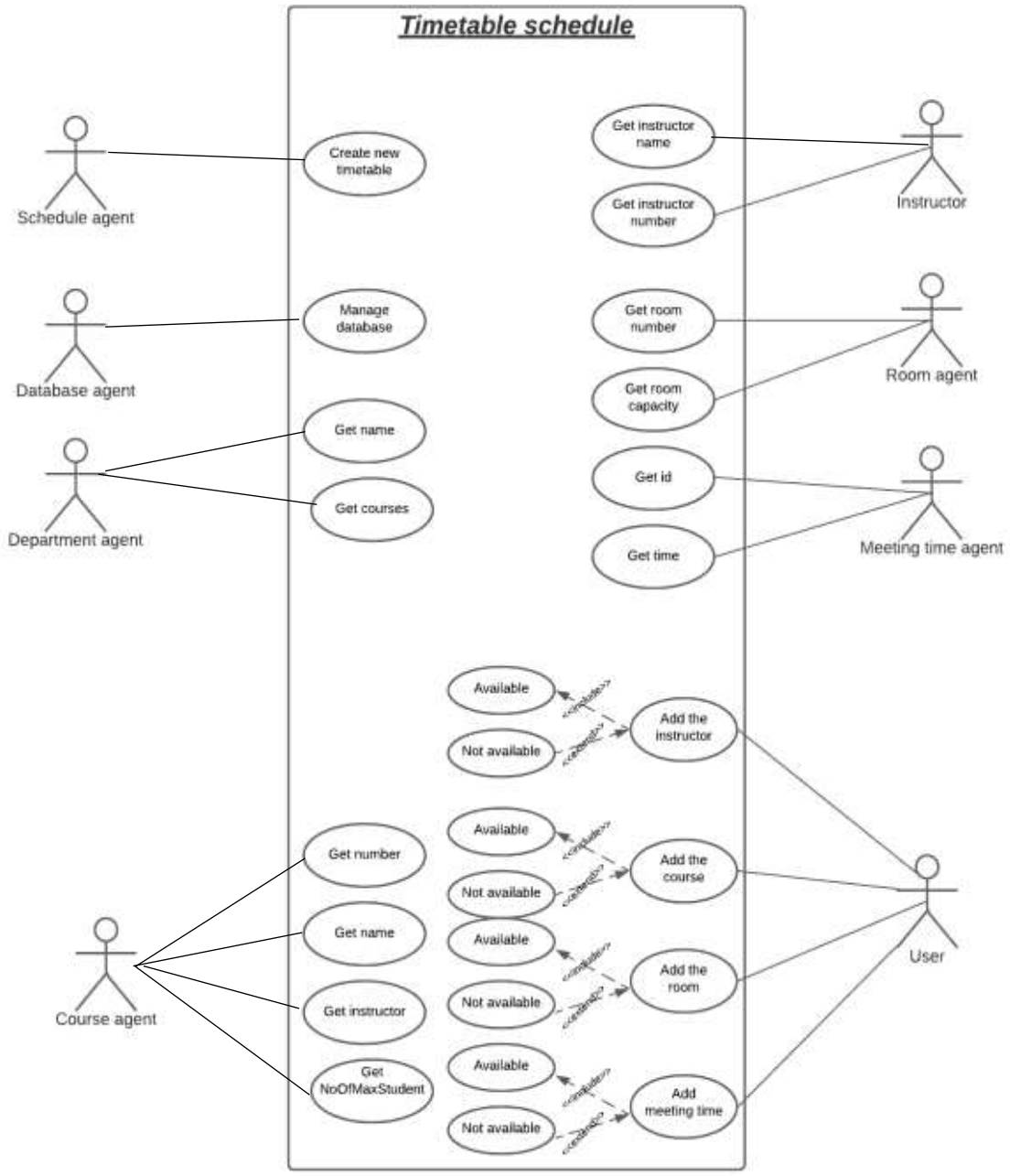
- Selection type: tournament selection
- Crossover type: uniform crossover (0.1)
- Mutation type: uniform mutation (0.01)

Tournament Selection (GA):

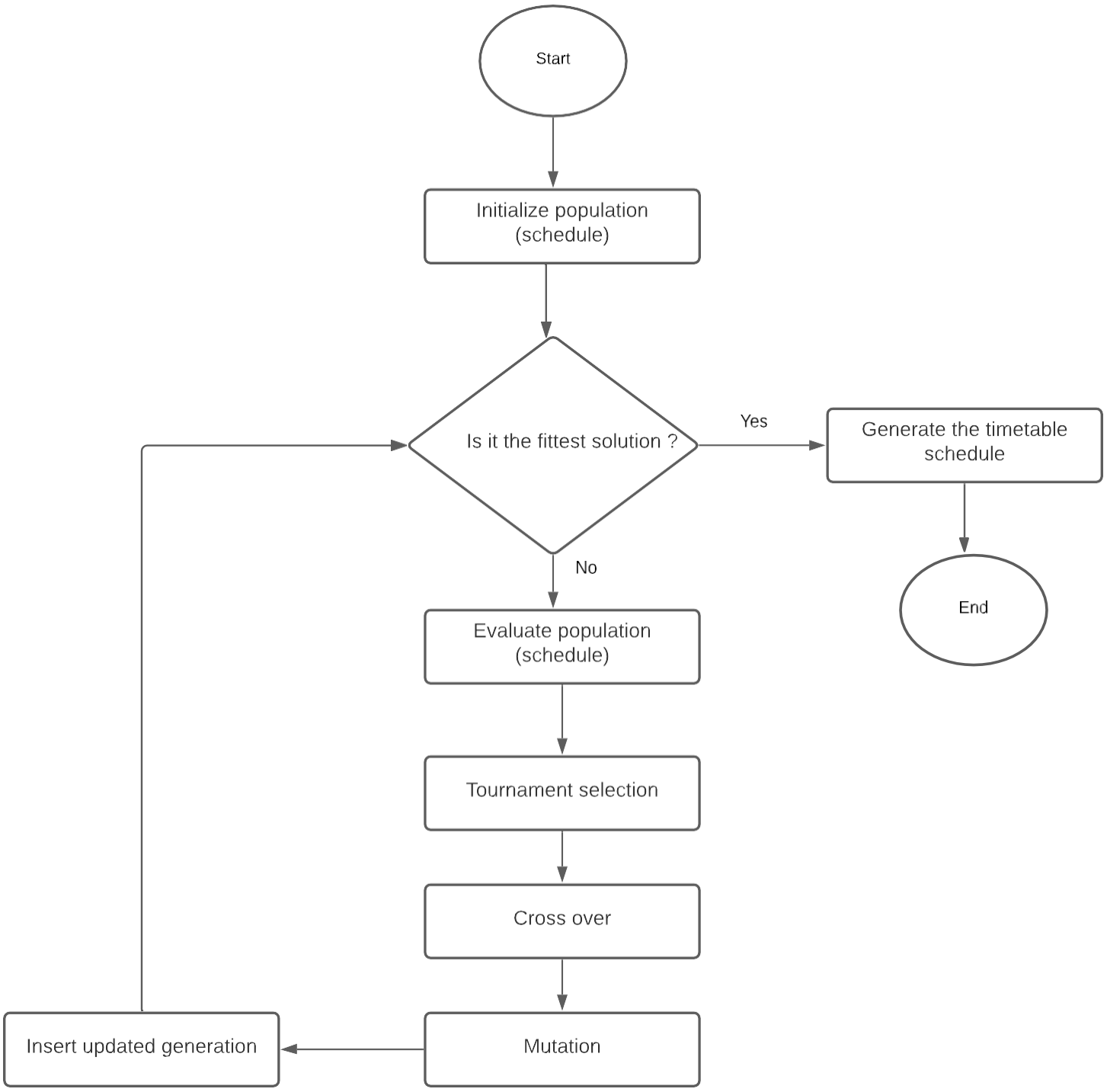
Tournament Selection is a Selection Strategy used for selecting the fittest candidates from the current generation in a

Genetic Algorithm. These selected candidates are then passed on to the next generation. In a K-way tournament selection, we select k-individuals and run a tournament among them. Only the fittest candidate amongst those selected candidates is chosen and is passed on to the next generation. In this way many such tournaments take place and we have our final selection of candidates who move on to the next generation. It also has a parameter called the selection pressure which is a probabilistic measure of a candidate's likelihood of participation in a tournament. If the tournament size is larger, weak candidates have a smaller chance of getting selected as it has to compete with a stronger candidate. The selection pressure parameter determines the rate of convergence of the GA. More the selection pressure more will be the Convergence rate. GAs are able to identify optimal or near-optimal solutions over a wide range of selection pressures. Tournament Selection also works for negative fitness values.

3-Main functionalities (use case diagram)

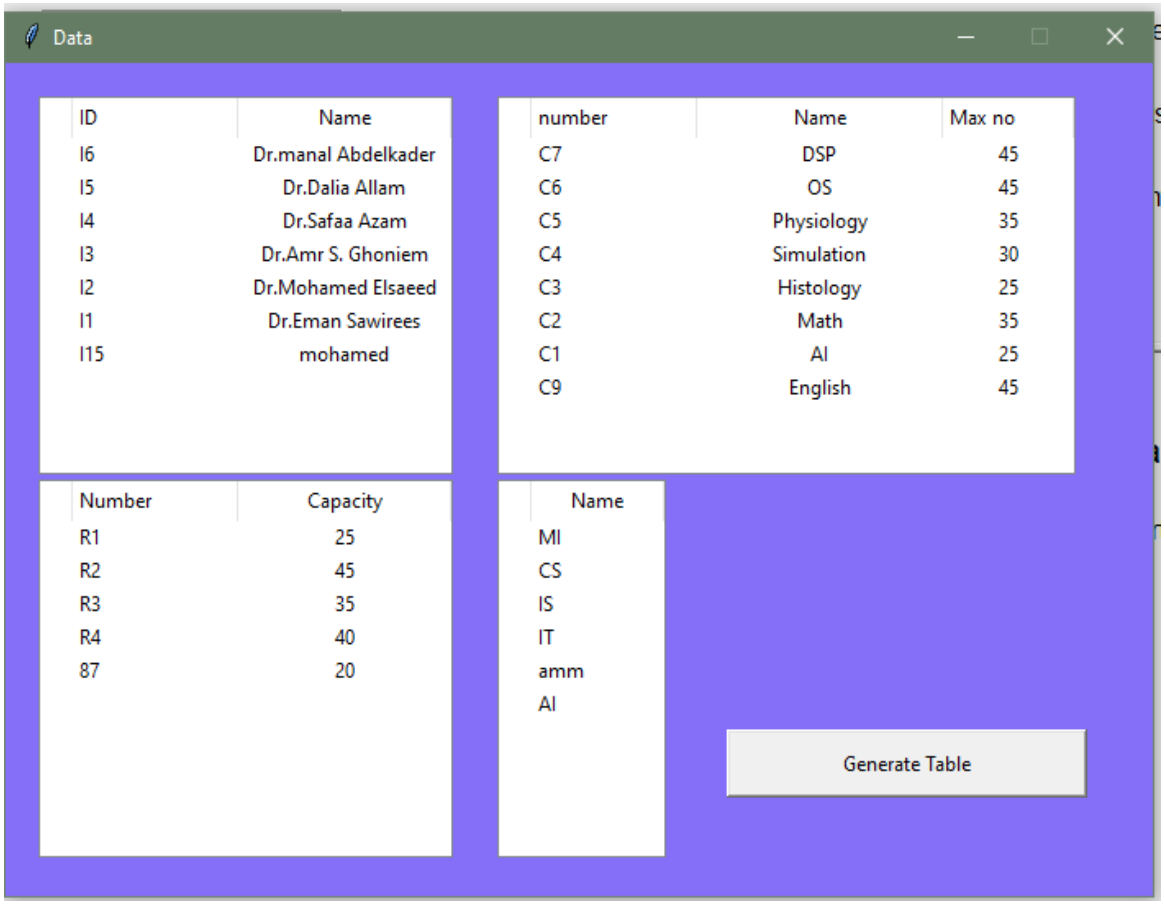

Timetable schedule
Schedule agent
Create new timetable
Database agent
Manage database
Department agent
Get name
Get courses
Get instructor name
Get instructor number
Instructor
Get room number
Get room capacity
Room agent
Get id
Get time
Meeting time agent
Available
Not available
Add the instructor
Available
Not available
Add the course
Available
Not available
Add the room
Available
Not available
Add meeting time
Course agent
Get number
Get name
Get instructor
Get NoOfMaxStudent
User

4-Flowchart

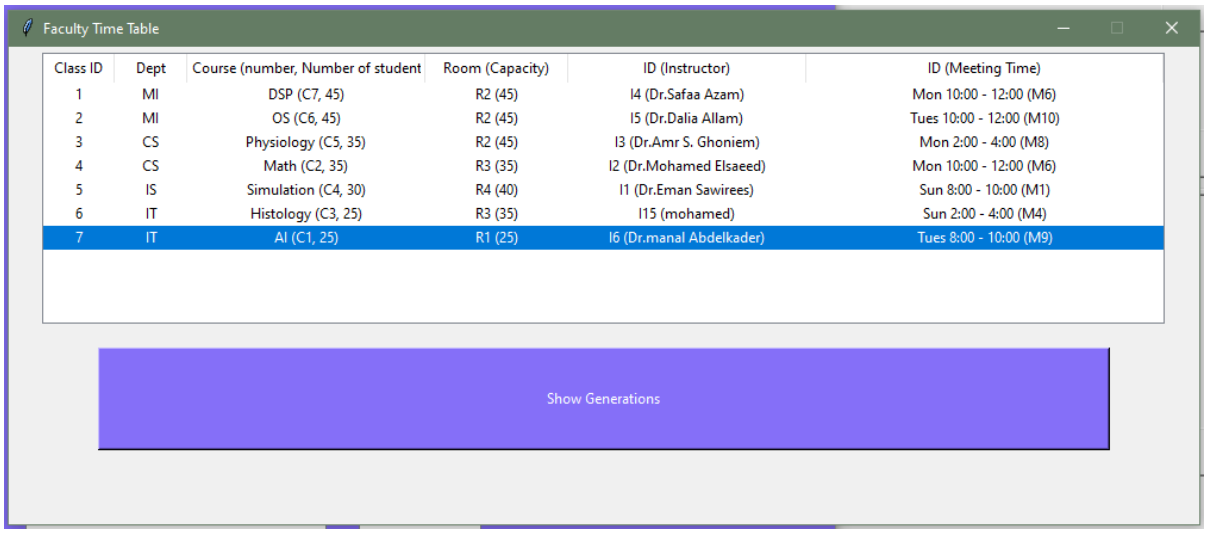

Start
Initialize population (schedule)
Is it the fittest solution ?
Yes
Generate the timetable schedule
End
No
Evaluate population (schedule)
Tournament selection
Cross over
Mutation
Insert updated generation

5-Experiments & Results

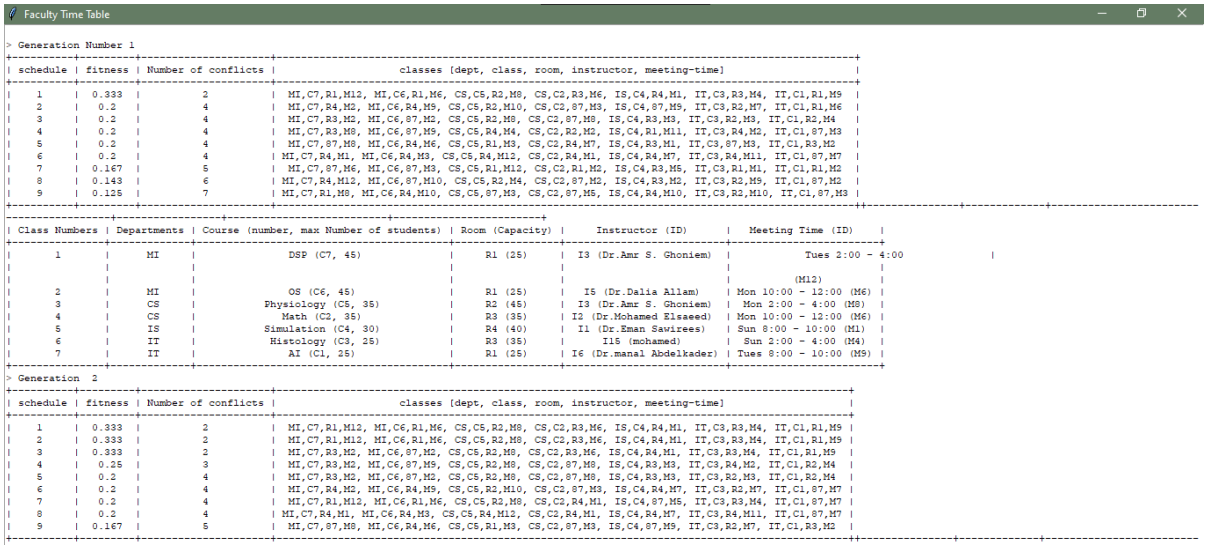
- Running the project was a bit tedious and challenging since we encountered an error in the genetic algorithm class and had to fix it after resolving the error and finishing the gui we finally had an output.



- After pressing generate table we finally achieved the output we desired which in our project was a zero conflict timetable.



- After pressing show generations we could see the number of generations that had to be created in order to achieve our zero conflict timetable and solve the problem.



- When we ran the project it took about 117 generations to achieve our goal.

> Generation 61

schedule	fitness	Number of conflicts	classes [dept, class, room, instructor, meeting-time]
1	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
2	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
3	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
4	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
5	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
6	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
7	0.5	1	MI,C7,R3,M8, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
8	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
9	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9

Class Numbers	Departments	Course (number, max Number of students)	Room (Capacity)	Instructor (ID)	Meeting Time (ID)
1	MI	DSP (C7, 45)	R1 (25)	I3 (Dr.Amr S. Ghoniem)	Tues 2:00 - 4:00 (M12)
2	MI	OS (C6, 45)	R2 (45)	I5 (Dr.Dalia Allam)	Tues 10:00 - 12:00 (M10)
3	CS	Physiology (C5, 35)	R2 (45)	I3 (Dr.Amr S. Ghoniem)	Mon 2:00 - 4:00 (M8)
4	CS	Math (C2, 35)	R3 (35)	I2 (Dr.Mohamed Elsaeed)	Mon 10:00 - 12:00 (M6)
5	IS	Simulation (C4, 30)	R4 (40)	I1 (Dr.Eman Sawirees)	Sun 8:00 - 10:00 (M1)
6	IT	Histology (C3, 25)	R3 (35)	I15 (mohamed)	Sun 2:00 - 4:00 (M4)
7	IT	AI (C1, 25)	R1 (25)	I6 (Dr.manal Abdelkader)	Tues 8:00 - 10:00 (M9)

> Generation 62

schedule	fitness	Number of conflicts	classes [dept, class, room, instructor, meeting-time]
1	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
2	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
3	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
4	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
5	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
6	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
7	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R4,M11
8	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
9	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9

> Generation 117

schedule	fitness	Number of conflicts	classes [dept, class, room, instructor, meeting-time]
1	1.0	0	MI,C7,R2,M6, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
2	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
3	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
4	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R2,M12
5	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
6	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
7	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
8	0.5	1	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,R2,M8, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9
9	0.333	2	MI,C7,R1,M12, MI,C6,R2,M10, CS,C5,87,M11, CS,C2,R3,M6, IS,C4,R4,M1, IT,C3,R3,M4, IT,C1,R1,M9

Class Numbers	Departments	Course (number, max Number of students)	Room (Capacity)	Instructor (ID)	Meeting Time (ID)
1	MI	DSP (C7, 45)	R2 (45)	I4 (Dr.Safaa Azam)	Mon 10:00 - 12:00 (M6)
2	MI	OS (C6, 45)	R2 (45)	I5 (Dr.Dalia Allam)	Tues 10:00 - 12:00 (M10)
3	CS	Physiology (C5, 35)	R2 (45)	I3 (Dr.Amr S. Ghoniem)	Mon 2:00 - 4:00 (M8)
4	CS	Math (C2, 35)	R3 (35)	I2 (Dr.Mohamed Elsaeed)	Mon 10:00 - 12:00 (M6)
5	IS	Simulation (C4, 30)	R4 (40)	I1 (Dr.Eman Sawirees)	Sun 8:00 - 10:00 (M1)
6	IT	Histology (C3, 25)	R3 (35)	I15 (mohamed)	Sun 2:00 - 4:00 (M4)
7	IT	AI (C1, 25)	R1 (25)	I6 (Dr.manal Abdelkader)	Tues 8:00 - 10:00 (M9)

6-Analysis, Discussion, and Future Work

6.1 Advantages:

As genetic algorithms have multiple offspring so they are able to look for solutions in many directions at a given time, so Solution space is wider as genetic algorithm doesn't guarantee an optimal solution but the fittest solution of the schedule.

Genetic algorithm is its capability to handle several parameters simultaneously. Usually many of the real-world

problems can only be described in terms of multiple objectives. As such, they cannot be expressed in the form of a single value for minimization and maximization purposes. A parameter can only be improved at the expense of another. Due to parallelism however, a GA can generate many solutions with one individual optimizing one parameter and another optimizing another. This allows one to select a desired solution for use.

6.2 Disadvantages:

The problem of identifying fitness function, choosing the parameters as the size of the population, mutation rate, cross over rate, the selection method and its strength, In addition to getting a good fitness function, it is also necessary to pay attention when selecting such factors as the population, crossover and mutation rates as less solution space created by a small population leads to inaccurate solution.

6.3 Why did the algorithm behave in such a way?

Because the schedule has many generations but genetic algorithm search only for the fittest solution.
We may try to use Mat lab toolbox as it is very common and easy way to run genetic algorithm.

7-development platform:

7.1 Tools

- Visual studio code
- PyCharm

- SQLite Studio

7.2 programming languages

- python
- sql

7.3 libraries

- **sqlite3**
- **random**
- **pretty table**
- **os**
- **tkinter**