

- **Table Contents:**

1. Problem Definition (Timetable problem)
2. The used algorithm in this problem (Genetic Algo.)
3. The input of the problem (Dataset) explanation and screenshots
4. Output explanation and screenshots.
5. URL of source code
6. References

- **Problem Definition:**

Scheduling faculty's timetable may encounter constraints (hard and soft) because of the difference compared to a school timetable as the requirements are very limited. We had to find a functional solution for any problems that have hard constraints and we have to optimize the performance of the solution by considering the soft constraints and finding a solution for it. In GA generate only one child solution with (selection, crossover, and mutation) for each generation and then the solution will be improved by local search. Finally, the worst population will be replaced with the new child individual.

- **Used Algorithm:**

A genetic algorithm is a powerful metaheuristic technique used to solve complex problems requiring huge search areas for possible solutions. GA is one of the evolutionary computations in AI so it's a category of evolutionary algorithms, it was developed by Professor John Holland in 1960. GA was inspired by the Darwinian theory of natural evolution which say that organisms multiply in geometric proportions leading to a struggle for survival because of the limited space and food and the stronger or the best will survive because they can make a species evolve due to those favorable variations so there is a small chance for the survival with injurious variations so the evolution is a process of natural selection.

GA often depends on adaptive systems to do good in different environments, in timetabling problems to increase the generality level we need a self-adaptive method & another example is the robot's complex behavior that needed to move in around the environment.

- **Input Explanation**

The predefined data contains:

- Instructors' IDs & names.
- Meetings IDs & dates
- Lecture room number & capacity of each room.
- Course numbers, IDs, instructors' names, and numbers of students who register for the course.
- departments and their courses

```
d: > Python > all.py > Database > __init__
13 class Database:
14     numofinstructors = [{"I1", "DR mohamed"}, {"I2", "Dr ahmed"}, {"I3", "Dr aya"}, {"I4", "Dr tabark"}]
15     meetings = [{"MT1", "MWF 08:30 - 09:30 AM"}, {"MT2", "MWF 09:30 - 10:30"},
16                 {"MT3", "TTH 08:30 - 10:00 A"}, {"MT4", "TTH 10:00 - 11:30"}]
17     numofrooms = [{"R1", 25}, {"R2", 45}, {"R3", 35}]
18     def __init__(self):
19         self._Instructors = []
20         self._Meetingtime = []
21         self._Rooms = []
22         for n in range(0, len(self.numofinstructors)):
23             self._Instructors.append(Instructor(self.numofinstructors[n][0], self.numofinstructors[n][1]))
24         for n in range(0, len(self.meetings)):
25             self._Meetingtime.append(meetingtime(self.meetings[n][0], self.meetings[n][1]))
26         for n in range(0, len(self.numofrooms)):
27             self._Rooms.append(Room(self.numofrooms[n][0], self.numofrooms[n][1]))
28         c3 = Course("C3", "Cs361", [self._Instructors[3]], 35)
29         c1 = Course("C1", "Cs352", [self._Instructors[0], self._Instructors[1], self._Instructors[2]], 35)
30         c2 = Course("C2", "is313", [self._Instructors[2], self._Instructors[3]], 30)
31         depart1 = Department("cs", [c1, c3])
32         depart2 = Department("is", [c2])
```

- **Output Explanation**

This Algorithm gets the timetable with fitness = 1 where the fitness formula is $(1 / 1.0 * \text{the number of conflicts} + 1)$ so the number of conflicts should be equal to 0 whereas this conflict occurred when the

number of students who registered for that course is greater than the capacity of the room where the course will be held, two courses are held in the same time at the same room or the same instructor is assigned to more than one course at the same time.

Generation 0						
Schedule_Num	Fitness	No_Of_Conflicts	Class(Dept,Course_Num,Room_Num,Instructor_ID,Meeting_ID)			
0	0.5	1	cs,Cs352,R3,I1,MT3 , cs,Cs361,R2,I4,MT4 , is,is313,R3,I4,MT4 , is,is313,R3,I4,MT4			
1	0.5	1	cs,Cs352,R1,I1,MT4 , cs,Cs361,R3,I4,MT4 , is,is313,R2,I3,MT3 , is,is313,R2,I3,MT3			
2	0.333	2	cs,Cs352,R1,I2,MT4 , cs,Cs361,R2,I4,MT4 , is,is313,R1,I3,MT2 , is,is313,R1,I3,MT2			
3	0.333	2	cs,Cs352,R3,I3,MT2 , cs,Cs361,R1,I4,MT1 , is,is313,R3,I4,MT1 , is,is313,R3,I4,MT1			
4	0.333	2	cs,Cs352,R3,I2,MT2 , cs,Cs361,R1,I4,MT4 , is,is313,R1,I3,MT3 , is,is313,R1,I3,MT3			
5	0.25	3	cs,Cs352,R1,I3,MT3 , cs,Cs361,R2,I4,MT2 , is,is313,R1,I4,MT2 , is,is313,R1,I4,MT2			
6	0.25	3	cs,Cs352,R1,I2,MT1 , cs,Cs361,R1,I4,MT4 , is,is313,R1,I3,MT2 , is,is313,R1,I3,MT2			
7	0.25	3	cs,Cs352,R1,I2,MT3 , cs,Cs361,R1,I4,MT1 , is,is313,R1,I3,MT2 , is,is313,R1,I3,MT2			
8	0.2	4	cs,Cs352,R1,I1,MT3 , cs,Cs361,R1,I4,MT3 , is,is313,R3,I4,MT3 , is,is313,R3,I4,MT3			
Class Num	Dept	Course(Number,Students_Num)	Room(Capacity)	Instructor(ID)	MeetingTime(MettingID)	
0	cs	Cs352(C1,35)	R3(35)	DR mohamed(I1)	TTH 08:30 - 10:00 A(MT3)	
1	cs	Cs361(C3,35)	R2(45)	Dr tabark(I4)	TTH 10:00 - 11:30(MT4)	
2	is	is313(C2,30)	R3(35)	Dr tabark(I4)	TTH 10:00 - 11:30(MT4)	

In this screenshot the first row there is one conflict between courses where the instructor with instructor ID4 is assigned to two courses (Cs361, Is313) at the same time MT4 so fitness will be equal $(1 / 1.0 * 1 + 1) = 0.5$, while at fourth row there is 2 conflicts that are an instructor with ID3 is assigned to two courses at the same time and the room (R1) capacity which is 25 is less than the number of students who register course (Cs361) which there number is 30.

If we compare between two rows the first-row fitness number (0.5) is greater than the fourth-row fitness number (0.3) so we take the best table within the first row.

Generation 1						
Schedule_Num	Fitness	No_Of_Conflicts	Class(Dept,Course_Num,Room_Num,Instructor_ID,Meeting_ID)			
0	1.0	0	cs,Cs352,R3,I2,MT3	cs,Cs361,R3,I4,MT4	is,is313,R2,I3,MT3	is,is313,R2,I3,MT3
1	0.5	1	cs,Cs352,R3,I1,MT3	cs,Cs361,R2,I4,MT4	is,is313,R3,I4,MT4	is,is313,R3,I4,MT4
2	0.5	1	cs,Cs352,R1,I1,MT4	cs,Cs361,R3,I4,MT4	is,is313,R2,I3,MT3	is,is313,R2,I3,MT3
3	0.5	1	cs,Cs352,R3,I1,MT3	cs,Cs361,R2,I4,MT4	is,is313,R1,I3,MT2	is,is313,R1,I3,MT2
4	0.5	1	cs,Cs352,R3,I1,MT3	cs,Cs361,R2,I4,MT4	is,is313,R3,I4,MT4	is,is313,R3,I4,MT4
5	0.5	1	cs,Cs352,R3,I1,MT3	cs,Cs361,R2,I4,MT4	is,is313,R2,I3,MT4	is,is313,R2,I3,MT4
6	0.333	2	cs,Cs352,R3,I1,MT3	cs,Cs361,R2,I4,MT4	is,is313,R1,I4,MT4	is,is313,R1,I4,MT4
7	0.333	2	cs,Cs352,R3,I2,MT2	cs,Cs361,R1,I4,MT4	is,is313,R1,I3,MT3	is,is313,R1,I3,MT3
8	0.25	3	cs,Cs352,R1,I1,MT4	cs,Cs361,R3,I4,MT4	is,is313,R3,I4,MT4	is,is313,R3,I4,MT4

Class Num	Dept	Course(Number,Students_Num)	Room(Capacity)	Instructor(ID)	MeetingTime(MettingID)
0	cs	Cs352(C1,35)	R3(35)	Dr ahmed (I2)	TTH 08:30 - 10:00 A(MT3)
1	cs	Cs361(C3,35)	R3(35)	Dr tabark(I4)	TTH 10:00 - 11:30(MT4)
2	is	is313(C2,30)	R2(45)	Dr aya(I3)	TTH 08:30 - 10:00 A(MT3)

The table at the first row is the final table whose fitness equals 1 and has no conflict.

References

- I. https://www.researchgate.net/publication/343430234_Automated_Timetable_Generation_using_Genetic_Algorithm HYPERLINK
- II. Wikipedia
- III. <https://www.geeksforgeeks.org/genetic-algorithms/> HYPERLINK