Name: Ahmad Farhan

Roll no. i211366

Section: A

Project Report

Problem: *Timetable Scheduling Using Genetic Algorithm*

Constraint Evaluation Functions

1. Classes only in Free Classrooms

Statement: "Classes can only be scheduled in free classrooms".

Steps:

1. Create a list of all lecture timeslots assigned to each room.
2. For each room, if its assigned timeslots list has duplicates, then there are clashes.
3. i.e., No. of all unique timeslots assigned not equals no. of all timeslots assigned.

2. Classroom Accommodation Capacity

Statement: "A classroom should be big enough to accommodate the section. There should be two categories of classrooms: classroom (60) and large hall (120)".

Steps:

1. Compare each section's strength against the capacity of the classroom assigned.
2. If Section Strength > Assigned Room Capacity, then that classifies as a clash

3. Professor Assigned 2 Lectures at Once

Statement: "A professor should not be assigned two different lectures at the same time".

Steps:

1. Create a list of all lecture timeslots assigned to each professor.
2. For each professor, if their assigned timeslots list has duplicates, then there are clashes. i.e., No. of unique timeslots assigned not equals no. of timeslots assigned.

4. Section Assigned 2 Classrooms at Once

Statement: "The same section cannot be assigned two different rooms at the same time"

Steps:

1. Create a list of all lecture timeslots assigned to each section.
2. For each section, if their assigned timeslots list has duplicates, then there are clashes. i.e., No. of unique timeslots assigned not equals no. of timeslots assigned.

5. Room Assigned more than 2 Sections at Once

Statement: “A room cannot be assigned for two different sections at the same time.”

Steps:

1. Create a list of all sections that have been assigned a lecture room-timeslot pair.
2. For each room-timeslot pair, if their assigned sections list has duplicates, then there are clashes. i.e., No. of unique sections assigned not equals no. of sections assigned.
3. Essentially, check if multiple sections have been assigned same room at same time

6. Professor Teaching more than 3 Courses

Statement: “No professor can teach more than 3 courses”.

Steps:

1. Create lists of all courses assigned to each professor.
2. For each professor, if the number of unique courses is more than 3, there is a clash.

7. Section Taking more than 5 Courses

Statement: “No section can have more than 5 courses in a semester”.

Steps:

1. Create lists of all courses taken by each section.
2. For each section, if the number of unique courses is more than 5, there is a clash.

8. Course Two Lectures per Week

Statement: “Each course would have two lectures per week not on the same or adjacent days”.

Steps:

1. Create lists of all first and second lectures taken in a week by each section-course pair for theory courses.
2. For each section-course pair, if the number of weekly lecture pairs is more than 1 then there are more than 2 lectures per week, which is a constraint violation.
3. If the day difference between the two lectures is less than 2 then the lectures violate the “same or adjacent days” constraint.

9. Consecutive Lab Lecture Slots

Statement: “Lab lectures should be conducted in two consecutive slots.”

Steps:

1. Create lists of all first and second lectures taken in a week by each section-course pair for lab courses.
2. For each section-course pair, if the number of weekly lecture pairs is more than 1 then there are more than 2 lectures per week, which is a constraint violation.
3. If the day difference between the two lectures is not equal to 1 or if the timeslot different is not 1, then the lecture slots are not consecutive i.e., constraint violation.

10. 15 Min Breaks between Classes

Statement: “15 mins breaks allowed between consecutive classes to ensure that there is sufficient time for transitions between classes.”

Steps:

1. Create a list of timeslots such that there is a difference of 15 minutes between consecutive timeslots.
2. This as such to maximize number of lecture slots in a day.
3. This acts as an always satisfied implicit constraint for all theory courses.

11. Classes in Same Classroom

Statement: “A class should be held in the same classroom across the whole week”.

Steps:

1. For each class of week, check if the first lecture room is the same as the second lectures room.
2. Since this is a soft constraint, we will limit its impact by setting a strict limit of 5 violation values. Calculated using:

   total violations / No. of classes * soft constraint weight

12. Theory/Lab in Morning/Afternoon Session

Statement: “All the theory classes should be taught in the morning session and all the lab sessions should be done in the afternoon session”.

Steps:

1. Create a list of all theory course lectures and check if the starting time of the timeslot falls within the limits of morning session timings.
2. Create a list of all theory course lectures and check if the starting time of the timeslot falls within the limits of morning session timings.
3. Since this is a soft constraint, we will limit its impact by setting a strict limit of 5 violation values. Calculated using:

   total violations / No. of lectures * soft constraint weight

13. Professor/Student Floor Traversal

Statement: “Teachers/students may be facilitated by minimizing the number of floors they have to traverse. That is, scheduled classes should be on the same floor for either party.”

Steps:

1. Create a list of all lecture rooms assigned to each professor and to each section.
2. For each lecture room list for both professors and sections, count as violation if the number of unique floors in list is equal to one.
3. Since this is a soft constraint, we will limit its impact by setting a strict limit of 5 violation values. Calculated using:

   total violations / No. of lectures * soft constraint weight

14. Continuous Teaching Blocks

Statement: “Teachers may prefer longer blocks of continuous teaching time to minimize interruptions and maximize productivity except when the courses are different”.

Steps:

1. Create a list of lecture timeslots assigned to each professor for a course.
2. The lectures are in continuous teaching blocks only if the difference between the maximum timeslot value and the minimum time slot value is equal to the number of lecture timeslots, otherwise it is a violation.
3. Since this is a soft constraint, we will limit its impact by setting a strict limit of 5 violation values. Calculated using:

   total violations / No. of lectures * soft constraint weight

Genetic Algorithm Functions

Gene Structure:

Attributes:

Each gene of a chromosome in this problem represents a complete class of the weekly schedule, defined by the following attributes as:

1. Course
2. Section
3. Professor
4. First Lecture Room
5. Second Lecture Room
6. First Lecture Timeslot
7. Second Lecture Timeslot

Each attribute is an integer that indexes its respective array. The array contains the rest of the information of the gene. For example, First Lecture Timeslot is an integer that indexes timeslots list, where each element is a Timeslot object whose attributes are the day and time.

Gene encoding

Two levels of encoding are employed.

1. By default, all gene attributes are integer encoded.
2. A second level of encoding uses binary encodings of those integers.

Chromosome Structure:

Each chromosome represents all the classes of an entire week. Thus, it can be interpreted as the timetable schedule of a week. The number of classes to be scheduled is definable.

Fitness Function:

The fitness function is calculated as:

1. Take the sum of total violations of hard and soft constraints.
2. If all hard constraints are satisfied, neglect soft constraint values for earlier termination.
3. Otherwise, the fitness value is the negative of the rounded-weighted sum of hard and soft constraint violations defined as:
    - round(hard_violations + 0.4 * soft_violations)

Note: Soft constraint evaluations are suppressed using value limits and lower weights. This ensures that the algorithm prioritizes solving the hard constraints first. Thus, satisfying soft constraints leads to a solution with desirable properties but not always one that has the necessary properties.

Mutation:

Two mutation functions have been implemented:

1. Mutates an entire chromosome by completely randomizing all of its attributes.
2. Mutates specific attributes of a chromosome when mutation probability is satisfied.

Selection:

Two parent selection functions have been implemented:

1. Tournament Selection: Randomly select individuals from the population and then conduct tournaments based on fitness values. Individual with the highest fitness is selected.
2. Rank-Based Selection: Assign probability to individuals based on their rank and then randomly select an individual based on these probabilities, favoring individuals with higher ranks.

Crossover:

Three crossover functions have been implemented. Each performs a crossover based on probability.

1. Single Point Crossover: Swaps entire continuous sections at a single point.
2. Two-Point Crossover: Divides chromosome into 3 sections (2 points) and alternates them.
3. Uniform Crossover: Randomly select genes from each parent and swap them.

Encoding Analysis

Initial integer encodings of gene attributes are further encoded using BitArray from bitstring library is used to limit space consumption even more. However, it must be noted that while binary encoding uses significantly less space (about half), it is drastically slower in execution. The reason is that the fitness value calculations access requires the integer values to compute results. Thus, the binary encodings are converted to integers so frequently that the execution times are increased by about 8x.

Program Execution and Output

Parameters:

n_iter (Max Generations): 1500
n_pop (Population Size): 50
n_slots (Classes per week): 22 i.e, 44 classes
r_cross (Crossover Probability): 0.95 i.e., 95%
r_mut (Mutation Probability): 0.2 i.e., 20%
Selection: Tournament Selection
Mutation: Entire Gene Mutation
Crossover: Single Point

Execution output:

> 0, new best 11 17 6 2 13 1 4 3 5 12 5 18 14 10 20 15 6 2 12 14 13 9 = -37
> 0, new best 10 17 10 15 10 15 4 15 4 21 20 19 1 5 19 12 10 4 1 3 2 12 = -32
> 0, new best 6 0 19 2 10 11 18 19 3 7 19 8 3 7 11 14 0 2 11 2 7 7 = -25
> 1, new best 6 0 19 2 10 11 18 19 3 21 19 19 3 7 11 14 10 2 1 2 7 7 = -24
> 4, new best 7 18 6 10 21 15 7 20 6 21 20 16 3 7 17 12 0 2 2 2 19 7 = -23
> 12, new best 7 4 21 10 21 15 13 19 6 2 13 7 3 7 17 21 0 2 2 17 19 7 = -20
> 17, new best 7 4 6 10 21 15 11 20 6 21 13 7 3 1 17 0 20 8 12 2 19 7 = -18
> 20, new best 7 4 6 10 21 15 11 20 6 13 13 7 3 1 17 0 13 8 12 14 20 7 = -17
> 36, new best 7 4 6 10 20 15 5 20 1 13 13 7 3 1 17 0 3 8 0 19 20 7 = -16
> 74, new best 7 4 6 10 0 15 15 14 3 13 13 14 3 1 17 0 3 8 0 19 14 7 = -15
> 87, new best 7 14 6 10 0 15 15 14 3 13 13 14 3 1 15 0 3 8 0 19 19 7 = -14
> 100, new best 7 14 6 10 0 15 15 14 3 20 13 14 3 1 15 0 3 8 0 19 19 7 = -12
> 174, new best 7 14 6 10 0 15 11 14 3 0 13 14 3 1 15 0 9 18 0 19 19 20 = -11
> 235, new best 20 14 6 10 0 15 15 14 3 0 13 1 3 1 15 0 9 18 0 3 19 20 = -10
> 262, new best 20 9 6 10 0 15 15 14 3 0 13 1 3 1 15 0 7 18 0 19 19 20 = -9
> 333, new best 20 9 6 10 0 15 16 14 3 0 13 1 3 1 15 0 7 18 0 19 19 20 = -8
> 363, new best 20 9 6 10 0 15 16 14 3 0 13 10 3 1 15 0 7 18 0 19 19 20 = -7
> 517, new best 20 9 6 10 13 15 16 14 3 12 19 10 3 1 15 0 7 18 0 16 19 20 = -6
Stopped at Generation: 1499

Best Chromosome: 20 9 6 10 13 15 16 14 3 12 19 10 3 1 15 0 7 18 0 16 19 20 = -6

Verifying Constraint Satisfaction of Solution
Hard Constraint Violations

Classes only in Free Classrooms : 0
Class Capacity Accomodates Section : 0
Professor Given Two Lectures at once : 0
Section Given Two Rooms at once : 0
Room Given Two Sections at once : 0
Maximum 3 Courses per Professor : 0

Maximum 5 Courses per Section : 0
Two Course Lectures per week : 0
Two Consequtive Slots for Labs : 0

Soft Constraint Violations

Class in Same Classroom : 5
Courses in Morning/Afternoon : 3
Same Floor Schedule : 4
Continuous Lecture Blocks : 4

Detailed Best Chromosome:

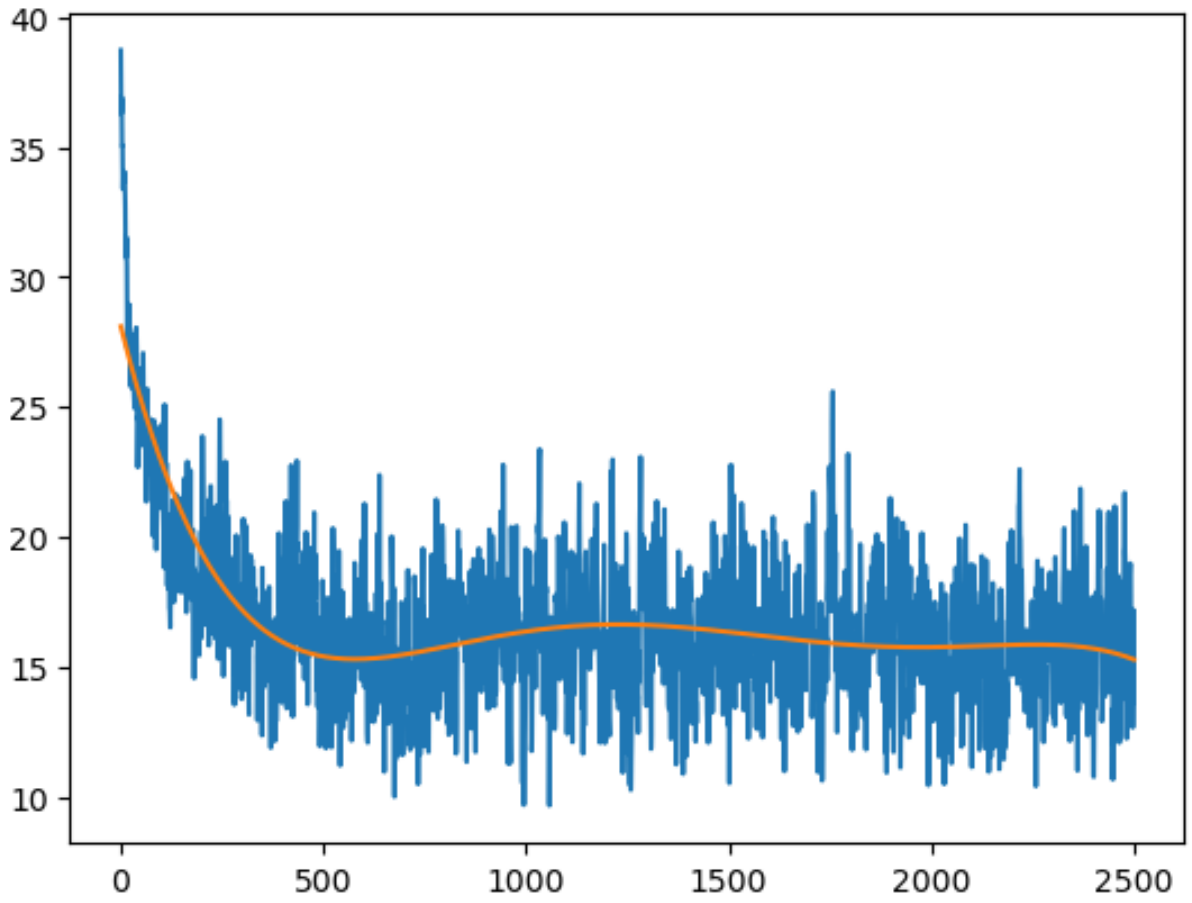
Encoded Best Solution:
Minimum Average Population Fitness:
Generation: 1057, Value: 9.66

Encoded Best Solution:
Class(course:20, section:5, professor:2, first_lecture:8, first_lecture_room:4, second_lecture:18, second_lecture_room:9)
Class(course:9, section:3, professor:5, first_lecture:20, first_lecture_room:7, second_lecture:21, second_lecture_room:4)
Class(course:6, section:6, professor:1, first_lecture:9, first_lecture_room:11, second_lecture:25, second_lecture_room:3)
Class(course:10, section:7, professor:9, first_lecture:17, first_lecture_room:1, second_lecture:27, second_lecture_room:12)
Class(course:13, section:8, professor:7, first_lecture:5, first_lecture_room:5, second_lecture:15, second_lecture_room:0)
Class(course:15, section:8, professor:7, first_lecture:26, first_lecture_room:1, second_lecture:1, second_lecture_room:8)
Class(course:16, section:8, professor:12, first_lecture:17, first_lecture_room:12, second_lecture:25, second_lecture_room:4)
Class(course:14, section:8, professor:1, first_lecture:27, first_lecture_room:2, second_lecture:28, second_lecture_room:12)
Class(course:3, section:3, professor:1, first_lecture:8, first_lecture_room:3, second_lecture:22, second_lecture_room:0)
Class(course:12, section:7, professor:3, first_lecture:26, first_lecture_room:11, second_lecture:14, second_lecture_room:6)
Class(course:19, section:0, professor:3, first_lecture:2, first_lecture_room:0, second_lecture:17, second_lecture_room:14)
Class(course:10, section:6, professor:5, first_lecture:26, first_lecture_room:15, second_lecture:13, second_lecture_room:11)
Class(course:3, section:0, professor:12, first_lecture:13, first_lecture_room:15, second_lecture:0, second_lecture_room:7)
Class(course:1, section:9, professor:6, first_lecture:3, first_lecture_room:14, second_lecture:2, second_lecture_room:14)

Class(course:15, section:4, professor:7, first_lecture:22, first_lecture_room:12, second_lecture:10, second_lecture_room:1)
Class(course:0, section:6, professor:4, first_lecture:28, first_lecture_room:14, second_lecture:2, second_lecture_room:13)
Class(course:7, section:6, professor:13, first_lecture:18, first_lecture_room:8, second_lecture:7, second_lecture_room:12)
Class(course:18, section:7, professor:5, first_lecture:18, first_lecture_room:11, second_lecture:4, second_lecture_room:8)
Class(course:0, section:0, professor:14, first_lecture:3, first_lecture_room:6, second_lecture:16, second_lecture_room:7)
Class(course:16, section:3, professor:11, first_lecture:25, first_lecture_room:14, second_lecture:15, second_lecture_room:7)
Class(course:19, section:1, professor:2, first_lecture:13, first_lecture_room:6, second_lecture:26, second_lecture_room:9)
Class(course:20, section:7, professor:8, first_lecture:11, first_lecture_room:4, second_lecture:23, second_lecture_room:4)

Generation Fitness Value Evolution:

Minimum Average Population Fitness:
`Generation: 1057, Value: 9.66`



Output Timetable:

	Section	Course Code	Professor	Time Slot	Day
C202	CS-A	SE110	Arshad Islam	830	Monday
C203	DS-C	MT224	Sadia Nauman	1005	Monday
C301	CS-A	SS152	Kashif Munir	1140	Monday
C405	SE-A	EE227L	Shams Farooq	1140	Monday
C404	DS-A	CS217	Sidra Khalid	1140	Monday
C405	SE-A	EE227L	Shams Farooq	1315	Monday
C201	CS-A	CS217	Adnan Tariq	1315	Monday
C203	DS-B	MG223	Shehreyar Rashid	1450	Monday
C306	DS-C	DS301	Sadia Nauman	1625	Monday
C403	DS-A	CS302	Asif Naeem	1005	Tuesday
C305	AI-B	SS118	Naveed Ahmad	1140	Tuesday
C304	CS-D	SE110	Umair Arshad	1140	Tuesday
C402	DS-A	CS220	Umair Arshad	1315	Tuesday
C302	AI-A	MT224	Sadia Nauman	1450	Tuesday
C305	DS-B	SS118	Ejaz Ahmed	1625	Tuesday
C402	DS-A	CS307	Shehreyar Rashid	1005	Wednesday
C406	CS-A	SE110	Arshad Islam	1005	Wednesday
C201	CS-B	SS152	Naveed Ahmad	1005	Wednesday
C201	DS-B	AI201	Kashif Munir	1140	Wednesday

C301	DS-C	DS301	Sadia Nauman	1315	Wednesday
C202	CS-D	SS113	Hammad Majeed	1315	Wednesday
C202	CS-A	CS217	Adnan Tariq	1450	Wednesday
C302	DS-B	CS307	Javaria Imtiaz	1625	Wednesday
C403	DS-C	SS113	Arshad Islam	1625	Wednesday
C405	CS-A	SS152	Kashif Munir	1625	Wednesday
C204	AI-B	SS118	Naveed Ahmad	830	Thursday
C203	DS-A	CS302	Asif Naeem	830	Thursday
C402	DS-B	MG223	Shehreyar Rashid	830	Thursday
C202	CS-D	SS111L	Shehreyar Rashid	1140	Thursday
C305	CS-D	SS111L	Shehreyar Rashid	1315	Thursday
C301	CS-D	SE110	Umair Arshad	1450	Thursday
C403	AI-A	MT224	Sadia Nauman	1450	Thursday
C305	DS-B	SS118	Ejaz Ahmed	1625	Thursday
C304	DS-A	CS220	Umair Arshad	1005	Friday
C305	DS-C	SS113	Arshad Islam	1005	Friday
C405	CS-D	SS113	Hammad Majeed	1005	Friday
C302	DS-C	MT224	Sadia Nauman	1140	Friday
C402	DS-B	AI201	Kashif Munir	1140	Friday
C406	DS-A	CS307	Shehreyar Rashid	1140	Friday
C204	CS-B	SS152	Naveed Ahmad	1140	Friday
C403	DS-B	CS307	Javaria Imtiaz	1315	Friday
C303	DS-C	CS218L	Umair Arshad	1315	Friday
C403	DS-C	CS218L	Umair Arshad	1450	Friday
C405	DS-A	CS217	Sidra Khalid	1450	Friday