

Instructions:

Following rules will be enforced for this assignment

- You might work in a group of at most Two students
- You might be asked to explain your approach and implementation details during a detailed evaluation.

Please remember that PLAGIARISM is INTOLERABLE and anyone found involved in it will get -3 marks (i.e. 100% penalty) in this assignment.

Exam Schedule Generation by Searching (Local and Evolutionary)

Manually generating a Mid-term exam schedule for NUCES-FAST is an involved task as a diverse set of constraints must be enforced while creating the schedule. In this assignment we are going to test the famous nature inspired **Genetic Algorithm combined with local search** for solving the scheduling problem.

Major inputs, for creating an exam schedule of n students registered in m courses, will be as follows

- A file named "**registration.data**" that contains course registration status in the form of a 2D $m \times n$ array with space separated entries stored in row major order. Entry $a[i][j]$ is 1 if student j is registered in course i and 0 otherwise.
- A file named "**capacity.room**" containing a space separated list of room capacities available for scheduling.
- A file named "**general.info**" containing a number specifying total exam days followed by a single number giving the exam slots per day for each room.

Our main job in this assignment includes the design and implementation of

- An efficient representation of a chromosome (representation of a complete solution)
- Defining the crossover and mutation operator for your representation of chromosome.
- Defining fitness function
- A generation of chromosome population
- GA for solving the problem (i.e. repeatedly creating the next generation from existing generation until a termination criteria is met)
- Refining the solution using local search

As any reasonable exam schedule will have all of the following properties therefore your fitness function must consider all of these properties for computing fitness of a chromosome.

1. All exams must be scheduled within the given number of days.
2. Total students taking exam in one given slot must be less than the total room capacity.
3. Number of students having two exams in one given slot must be minimized.
4. Number of students having exams on two consecutive slots must be minimized.
5. Not even a single student can have more than two exams in one slot.
6. Not even a single student can have more than two exams in consecutive slot.
7. Not even a single student can have more than three exams in one given day.
8. ...