Exam Schedule Generator Using Local Search Algorithms

Instructions:

- You are required to submit following three things in a ZIP file (naming convention must be ***RollNumbers_Section_Project***),
  1. Word file containing your code and sample screenshots
  2. A Jupyter Notebook/pycharm file that contains your complete well commented code.
  3. A project report, containing your group members name and roll number.

- You may make a group of maximum **3 people** for this project. However, groups can't be changed once submitted.

- Evaluation of the project will be individual based and not group based. Therefore, every member of the group must put equal efforts into it.

- Late Submissions will **not be accepted.**

- Plagiarism will not be tolerated, either done from the internet or from any fellow classmate (same or different section) and will lead to zero or **negative absolutes** in the project**.**

Problem

The assignment is to find generic solution that will facilitate generating **exam schedule** for university using local search algorithms.

- You have to write code from scratch.
- Make sure your code is well documented
- You cannot use any built-in library for the implementation of Local Search Algorithms except Pandas and NumPy etc
- You must use the best suited Local Search Algorithm for the problem.

The success of solution is estimated on fulfillment of given constraints and criteria. You have to submit a well commented .ipynb/py file with a one-page report of your implementation in pdf format.

You are provided with two folders containing an actual dataset and a smaller dataset (for testing purposes only). Data of Teachers, Students, Rooms, Courses and Registered Courses for each Student is given in Excel Files.

Using the dataset, you are required to find an optimized examination timetable for **two weeks** and also for **three weeks.**

Constraints

There are set of constraints that need to be fulfilled.

Hard Constraints

- An exam will be scheduled for each course.
- A student is enrolled in at least 3 courses. A student cannot give more than 1 exam at a time.
- Exam will not be held on weekends.
- Each exam must be held between 9 am and 5 pm
- Each exam must be invigilated by a teacher. A teacher cannot invigilate two exams at the same time.
- A teacher cannot invigilate two exams in a row.

The above-mentioned constraints must be satisfied.

Soft Constraints

- All students and teachers shall be given a break on Friday from 1-2.
- A student shall not give more than 1 exam consecutively.
- If a student is enrolled in a MG course and a CS course, it is preferred that their MG course exam be held before their CS course exam.
- Two hours of break in the week such that at least half the faculty is free in one slot and the rest of the faculty is free in the other slot so the faculty meetings shall be held in parts as they are now.

Input & Output

Input data for each exam are *teachers' names, students', exam duration*, *courses (course codes)*, and list of *allowed classrooms*.

Output data are classroom and starting time for each exam along with course code and invigilating teacher. Time is determined by day (Monday to Friday) and start hour of the exam.

- Output will be a timetable/solution which satisfies all hard constraints and the maximum number of fulfilled soft constraints.
- You have to display a list of all hard and soft constraints which are fulfilled in the output.
- Don't forget to show fitness values at each iteration.

Evaluation Criteria

Requirement	**Marks**
Fulfillment of hard constraint	10*6 = 60
Fulfillment of soft constraint	05*4 = 20
Quality of code and comments	20
.ipynb/py file containing code in python	50