**Exam Scheduling Using MoEA Framework**

StudentsAndCoursesDataStore class parses the data and populates relevant lists.

A custom problem class ScheduleProblem is defined which overrides newSolution and evaluate methods. NewSolution generates candidate solutions for problems. A solution is a list of random slots for courses, a list of 135 random numbers between 0-35 in this case.

A single day has 2 slots, and in our case we have 18 days(Chromosome size == 18). So 18\*2 == 36 slots for all 18 days. Relevant day and slot number is calculated as follows:

if slotNumber 17 is selected, 17/2 = 8th day and 17 %2 = 1st slot.

All 135 courses are mapped on to a chromosome at slot numbers as indicated by the solution array(array of 135 slots) and fitness is calculated. Fitness1(objective funtion 1) is number of clashes on this particular day + number of students having 2 exams on this day. Fitness2(objective function 2) is number of days with over 1200 students in any slot. We have to minimize both these objective functions.

Population size is kept at 20, crossover rate at 0.9 and maximum evaluations are kept at 20000. All these parameters can be changed as needed in Driver class.

After evaluations are completed a single non-dominated solution is obtained, which is then mapped onto the chromosome. This chromosome is the fittest of all and is printed with all scheduled courses.