**Open Book Assignment 2**

*Please read the complete assignment (right up to the final full stop). Make sure that you understand what you have read. Then start thinking how you would solve the problem.*

We want to use Genetic Algorithms to solve the problem of job scheduling with deadlines that was discussed in the class. The specifications of the problem are as follows:

You are given a set of N jobs, J(1), J(2) … J(N).

All jobs have to be completed on a single machine.

The time taken by each job is T(1), T(2) … T(N).

All jobs are present initially i.e. their arrival time is T = 0.

Each job has a deadline attached to it as D(1), D(2), … D(N).

Your algorithm must produce a schedule, S, that contains the order in which the jobs are to be executed. For example, for N = 5, a schedule like S = (2, 4, 1, 3, 5) indicates that J(2) will be executed first, followed by J(4), followed by J(1), followed by J(3) followed by J(5).

The output of your algorithm must

1. Produce a schedule such that all deadlines are satisfied, if such a schedule exists
2. In case more than one schedule exists (satisfying all deadlines) then your algorithm must output that schedule which minimizes the average turnaround time
3. In case no schedule satisfies all deadlines, then produce a schedule that minimizes the total number of jobs missing their deadlines

Your assignment:

1. Find a good representation for the solution (i.e. the schedules).
2. Define the Objective function
3. Define the fitness function
4. Define crossover operator
5. Define mutation operator
6. Execute your algorithm on some test cases
7. Submit the report, along with the code and results

Some suggestions:

1. You can try the approach suggested in one of my lectures. You will have to look at all the lectures to figure out which lecture I am referring to ☺.
2. Notice that this is a slightly peculiar problem (but very close to real life). It is peculiar because the objective function (i.e. the function that you are trying to minimize) is not unique. If you have schedules satisfying the deadline, then you need to minimize the average turnaround time. On the other hand, if no such schedule exists, you want to minimize the total number of tasks missing the deadline. These are two different objective functions. Also, *a priori*, you do not know whether any schedule exists satisfying all deadlines for a given input.
3. A sensible approach for this type of situation is to break up the problem and solve it in two steps. In the first step use GA with an objective function that minimizes the total number of jobs missing deadline (i.e. the second objective function). In case there exist schedules that satisfy all deadlines, then this objective function can find them. In case there are several such schedules then calculate the turnaround time for each schedule (can be done in O(N) once the schedule is known). Output the schedule that has the smallest turnaround time. In case no schedule exists that satisfies all deadlines, then the solution after the first step can be output.

Submission deadline: 7th July, 11.30 p.m.

Submission style: Build a report that answers each point mentioned in the item labelled “Your Assignment”. Make sure that any assumption you make is stated clearly. As usual, you are free to consult any material and each other. However, you have to write your own code and report.