Algorithmic Methods for Mathematical Models **June 20th, 2017**

IMPORTANT: Write each exercise in a different sheet of paper.

---

1. *(6 points and 2 hours)* [For this problem you will need to use CPLEX]

   A public hospital needs to design the working schedule of their nurses. As a first approximation, we are asked to help in designing the schedule of a single day. We know, for each hour h, that at least $demand_h$ nurses should be working at the hospital. We have available a set of $nNurses$ nurses and we need to determine at which hours each nurse should be working. However, there are some limitations that should be taken into account:

   - Each nurse should work at least $minHours$ hours.
   - Each nurse should work at most $maxHours$ hours.
   - Each nurse should work at most $maxConsec$ consecutive hours.

   a) Taking into account these data, build an integer linear program to find out at which hours each nurse should be working.

   Create an OPL project to solve this problem. Problem data can be found in file `exam.dat`. Use file `initialModel.mod`, where you can see which variables are needed for this question, as the initial model and complete it. Save this file as `modelA.mod` and send it through "El Racó".

   b) After examining the schedule we have found, the hospital director decides that the number of nurses that work some hour should be minimized. That means that nurses are now allowed to work zero hours. Additionally, no nurse can stay at the hospital for more than $maxPresence$ hours (e.g. if $maxPresence$ is 7, it is OK that a nurse works at 2am and also at 8am, but it not possible that he/she works at 2am and also at 9am). Change the previous OPL model to consider the new objective function and the new constraint. Save this file as `modelB.mod` and send it through "El Racó". Use `exam.dat` again as problem data.

   c) The hospital director is not happy yet with the schedule found and decides to add one last constraint: no nurse can rest for more than one consecutive hour (e.g. working at 8am, resting at 9am and 10am, and working again at 11am is not allowed, since there are two consecutive resting hours). Change the previous OPL model to consider the new objective function and the new constraint. Save this file as `modelC.mod` and send it through "El Racó". Use `exam.dat` again as problem data.

   (*Hint:* one possible solution consists in introducing, for each nurse h and hour h, three variables $worksBefore_{n,h}$, $worksAfter_{n,h}$ and $rests_{n,h}$, meaning that the nurse works at some hour before h, at some hour after h, and rests at h, respectively.)

2. *(4 points, 1 hour and 15 minutes)* [Metaheuristics]

   For this question we consider the problem defined in 1b), but we will modify the objective function: every nurse used has a cost of C_n, and every excess nurse per hour has a cost of C_h. The goal is to minimize the total cost. For example, assuming that $C_n = 100$ and $C_h = 1$, a solution requiring 6 nurses that match the exact demand at every hour (i.e. no excess) will have a cost of 600. A solution needing 7 nurses where 2 excess nurses are allocated at 7pm and 1 excess nurse at 8pm has a cost of 703.

   As a first approach, we want to provide a greedy constructive algorithm. We define a solution as a list of pairs $<nurse, hour>$. At every iteration of the algorithm, we compute the greedy cost for every candidate pair $<nurse, hour>$ and select one with minimum greedy cost to be added to the solution. Recall that we want to satisfy **all constraints** of 1b) and hence, the algorithm will stop when all constraints are satisfied.

   a) Define the function and the algorithm (if necessary) to compute the greedy cost of a candidate pair.

   b) Let us assume that we are constructing a solution for an instance with 3 nurses (n_1, n_2 and n_3) and parameters: $minHours = 5, maxHours = 8, maxConsec = 3, demand_h = 1$ for all hours h, and $maxPresence = 10$. At a given constructive iteration, we have already found the following partial solution, ordered by nurse (top to bottom) and hour (left to right) for the sake of readability:

      $<n_1, 8h>$ $\quad$ $<n_1, 9h>$ $\quad$ $<n_1, 10h>$
      $<n_2, 12h>$ $\quad$ $<n_2, 13h>$ $\quad$ $<n_2, 15h>$ $\quad$ $<n_2, 16h>$ $\quad$ $<n_2, 17h>$

      Using the greedy cost function you proposed in 2a), and starting from the partial solution above detailed, compute the greedy cost of the following candidate pairs:

      - $<n_1, 11h>$
      - $<n_2, 11h>$
      - $<n_3, 11h>$
      - $<n_1, 16h>$
      - $<n_1, 18h>$

      Which of the previous pairs would you select according to the minimum greedy cost criterion?

   c) Let us now focus on producing another algorithm for solving the same problem, but now based on the BRKGA meta-heuristic.

      Define the structure of the chromosome that encodes a solution and write the pseudo-code of the decoder algorithm needed to both convert a chromosome into a solution of the problem, and compute the fitness value. Note: you are free to define a solution as you wish, i.e., there is no need to define a solution a list of pairs $<nurse, hour>$.