
Algorithmic Methods for Mathematical Models

— Course Project —

---

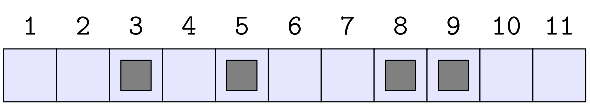
1 Problem Statement

A well-known company seeks our advice in order to schedule the production of its best-selling car. When customers buy a car, they can equip it with a set of *options* $\mathcal{O}$: sunroof, GPS, parking sensors, etc. We say that a *class* is a set of such options. There exists a fixed set of classes C and we have a Boolean matrix $ClassOption_{c,o}$ that indicates whether class $c \in \mathcal{C}$ has option $o \in \mathcal{O}$. Moreover, for each class c we know the exact number ($carsOfClass_c$) of cars of that class to be produced.

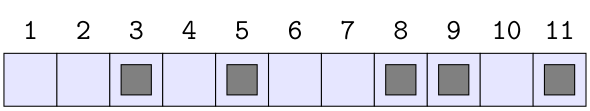
All cars go through the same production line and the only thing we need to decide is the order in which they will be put in such a line. For notational purposes, let us assume that we have a set of positions $\mathcal{P} = \{1, 2, \ldots, nPositions\}$ in the line to fill. Obviously, $nPositions$ is equal to the total number of cars that have to be produced.

We would like to schedule the production of the car in the following variations of the above problem:

A. In this case, we assume that not all sequences are correct. For every option o we have two integers k_o and m_o that indicate that for every window of k_o consecutive cars, at most m_o can have option o. For example, if for every 4 consecutive cars at most 2 can have option o, the following sequence of 11 cars is correct:



where a grey square indicates a car with option o. However, the order



is not correct because the window $[8 \ldots 11]$ contains 3 cars.

B. Let us now assume that we also want to avoid changing too often the class in the line. We say that a position p has a *change* if $p-1$ and $p+1$ exist, the classes at positions $p-1$ and p are different from each other and the classes at positions p and $p+1$ are also different from each other. The goal is to minimize the number of positions where there is a change.

C. Finally, let us consider the problem in case A but assume now that the constraint stating that for every k_o consecutive cars at most m_o can have option o is soft, i.e., it can be violated. However, we want to minimize the number of times these constraints are violated. More specifically, we say that an option o has a *violation* at position p if there is a window of size exactly k_o starting at p which contains more than m_o cars with option o. The goal is to minimize the number of pairs (o, p) such that option o has a violation at position p.

2 Tasks

1. Create an OPL project to solve case A of the problem. Since this is a feasibility problem, a constant can be used as the objective function.

   Problem data will be found in the file `projectAB.dat`. Use the file `initialModel.mod` as the initial model and complete it. Save this file as `modelA.mod`.

   Hint: Use the Boolean variables

   - $pc_{p,c}$ (for each $p \in \mathcal{P}$, $c \in \mathcal{C}$) and
   - $po_{p,o}$ (for each $p \in \mathcal{P}$, $o \in \mathcal{O}$)

   stating whether a car of class c is assigned to position p, and whether a class with option o is assigned to position p, respectively.

2. Using again the data from `projectAB.dat`, modify your OPL model in order to deal with the situation of case B. Auxiliary variables might be helpful. Save this file as `modelB.mod`.

3. Use the data from file `projectC.dat` and create an OPL model to solve case C of the problem. Again, auxiliary variables might be helpful. Save this file as `modelC.mod`.

4. Due to the complexity of the optimization problem B, we are considering using Greedy and GRASP. Specify the pseudocode of an algorithm for the Greedy and GRASP *constructive phase*, including the candidates, the greedy function $q(\cdot)$, and the equation describing the RCL.

5. Let us assume that the algorithms specified in the previous item are being executed and the following sequence of cars is already selected:

Position	1	2	3	4	5
Class	1	1	2		

   Run the algorithms proposed in the previous item to decide the next two cars in the sequence.

3 Report

Prepare a report including the following items:

1. Formal statement of the problem for all three cases A, B, and C.

2. Integer linear models used in Tasks 1, 2, and 3, including the definition of sets and parameters, the model itself and a short description of the objective function and every constraint. Do not include OPL code, but rather its mathematical formulation.

3. Pseudocode of the algorithms used in Task 4. Specify $q(\cdot)$ using mathematical notation and a short descriptive text. For each iteration of the algorithms used in Task 5, compute the value of the proposed greedy function $q(\cdot)$ for all the candidates and write down the RCLs (assume $\alpha = 0.5$) for the GRASP case.