Computational Methods in Optimization

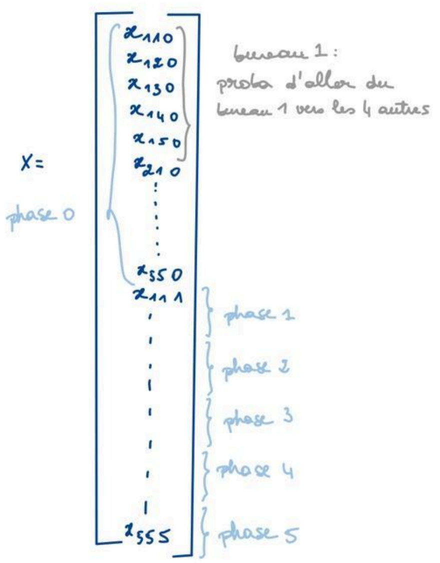
Report Course project - 2023/2024

Mouna AHAMDY – Dany MOSTEFAI – Roc de LAROUZIERE – Colette LEMAIRE

Task: A short report in PDF file explaining their approach to modeling and solving the optimization problems at hand.

LP

To create the Linear program, the first question that we asked ourselves was how to represent the movements between the offices during the phases. We first thought of implementing a 3D-tensor, but then we realized that it is impossible for cvxpy to manipulate a 3D tensor, so we dealt with a vector that is actually a representation of the 3D-tensor but "flattened".
Indeed, we represented our variable as a vector $X \in \mathbb{R}^N$ where $N = n \times n \times P$ (P = number of phases) such that each xi,j,p is the proportion of people of the office i that goes to the office j at a phase p.



Then we started to think about the different constraints we could implement. We noticed that since the xijp's represented some proportions, we would have the following 3 constraints :

The first constraint is about the positivity of X because each xijps have to be superior or equal to zero since they represent the movement of going from i to j, so it would be a nonsense to use negative values.

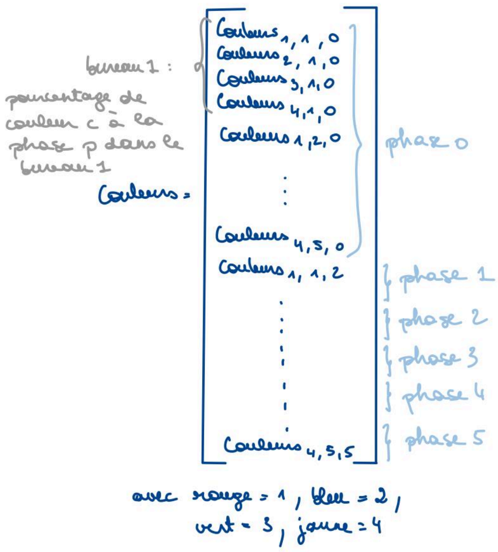
$$\sum_{j=1}^{n} x_{i,j,p} = \begin{cases} 0 \text{ if i was is renovation at phase } (p-1) \\ 1 \text{ otherwise} \end{cases} \qquad \forall j = 1, \ldots, n, \forall p = 0, \ldots P$$

After working on it, we implemented the constraint above. We say that the sum of all the probabilities to go from one office to all the others is zero if at the previous phase, the departure office i was in renovation. Indeed, if it was in renovation at phase (p-1), it was empty at the end of phase p-1, and there can't be any movement from this office to the others at phase p. We also say that the movement from an office to the other ones is 1 which we could interpret in the way that an office sends a total of 1 between itself (if can moves to itself (which of course we don't count as a move but we still have to implement it) and the other offices .

$$\sum_{i=1}^{n} x_{i,j,p} = \begin{cases} 0 \text{ if j is in renovation or j is the office N at phase P - 1} \\ 1 \text{ otherwise} \end{cases} \qquad \forall i = 1, \ldots, n, \forall p = 0, \ldots P$$

In this third constraint, we say that the sum of all the probabilities to go from all the offices to an office in renovation is zero because if the office j is in renovation, we can't go in it. The sum also returns zero if we are in the last phase (last phase has index P-1) because we cannot go to the office of wing N (it is asked by the graph). Otherwise it returns 1 which we can interpret in the way that every office has a capacity to receive a total of proportions from itself and the other offices, of exactly 1.

After implementing the constraints directly on the xijps, we needed to add the colors in our linear program. To do that, we created a vector Couleurs $\in \mathbb{R}$^Q, where $Q=4\times n\times P$, that represents the proportions of colors of each offices at each phase.



We needed to add the color constraints.

$$\sum_{c=1}^{4} Couleurs_{c,i,p} = 1 \quad \forall i = 1, \ldots, n; \forall p = 0, \ldots, P \text{ s.t i is not in renovation or the office N in last phase}$$

The sum of the colors of a given office at a given phase is equal to 1, unless it is empty.
We don't take into consideration the color white, for any phase. Each office has a proportion distribution of the different colors equal to 1, unless it is in renovation in which case the office is empty and hence has no colors in it.

$$\sum_{i=1}^{n} Couleurs_{c,i,p} = 1 \quad \forall c = 1, \ldots, 4; \forall p = 0, \ldots, P$$

This means that the sum of the proportions of a specific color in all the offices at a given phase is equal to 1.

Finally, the last constraint is :

$$\sum_{j=1}^{n} W_{c,j,p-1} = Couleurs_{c,j,p} \quad \forall c = 1, \dots, 4; \forall p = 1, \dots, P$$

$$\begin{cases} W_{i,c,j,p} <= Couleurs_{c,i,p-1} \\ W_{i,c,j,p} <= X_{i,j,p} \\ W_{i,c,j,p} >= 0 \\ W_{i,c,j,p} >= Couleurs_{c,i,p-1} + X_{i,j,p} - 1 \end{cases} \quad \forall c = 1, \dots, 4; \forall i = 1, \dots, n; \forall j = 1, \dots, n; p = 1, \dots P$$

We calculate the product X*Y thanks to the McCormick cut inequalities, similarly to the method we saw in class. The vector W is defined in the linear program as a 4-dimensional tensor (which in reality will necessarily be a vector obtained by flattening), but which in our implementation has been fragmented into several vector variables to avoid manipulating 4 indices.

We have also used an A matrix in the implementation, which combines all the constraints on X. The handling of indices differs somewhat: in our linear program, indices start at 1 for the sake of clarity, but our implementation considers indices starting at 0.

For the objective, we chose to minimize the movements of the offices during the whole renovation, the xijps. But this means that we don't have to consider the xijps such that i==j because it corresponds to the movement of i staying to i. This particularity is represented by the vector c.

SECOND GRAPH

The objective function is the same as the 1st graph with a vector c adapted to the 13 offices.

The constraints on the xijps follow the same reasoning as the constraints of the xijps for the 1st graph, except for the second constraint where there is an inequality (meaning that an office can receive a proportion of people of at most 1).

However those next constraints differ :

$$\sum_{i=1}^{n} Couleurs_{c,i,p} = nb\_color(c) \quad \forall c = 1, \dots, 4; \forall p = 0, \dots, P$$

This constraint means that, for the sum of all the proportions of a specific color in all the offices at a given phase is equal to 2 or 3 depending on the color. For example, if we focus on the blue color, at phase 0, 3 offices are colored with blue which means that at all the other phases, the sum of all the proportions of blue in all the offices will have to be equal to 3.

$$\sum_{c=1}^{4} Couleurs_{c,i,p} <= 1 \quad \forall i = 1, \ldots, n; \forall p = 0, \ldots, P \text{ s.t i is not in renovation or the offices A1, A2 or B3 in last phase}$$

For this constraint, what differs from the 1st graph is that we have a <= sign. This is because at the beginning of the renovation, 3 offices are in renovation, but then at phase 3, there are just two offices in renovation, which means that there might be an office(or more) at phase 3 that has less than a distribution of colors equal to 1.

Note that in the second graph, we use A but also A_p depending on which constraint we are dealing with. This is because some constraints have to be equal to 1 and others have to be just <=1 (because of the change of phases with 3 offices in renovation to phases of 2 offices in renovation).

QP

For this part, all the constraints are the same. But the objective function slightly changes. We added the term λ×‖$Couleurs\{p{-}{>}p{+}1\}{-}zF$)‖^2 where the ∑ is from p=0 to P, to the previous objective function. It quantifies how the allocation differs from the final one for all the steps. The vector zF represents the final allocation of the offices and the vector $Couleurs\{p{-}{>}p{+}1\}$ is all the offices colors at phase p :it corresponds to the portion of all colors in the phase p. We multiply the sum of the differences of each term by λ, that is the weight of the penalty.

SDP

To model the SDP, we needed to express the constraint U - uuT ≽ 0 that is not DCP.

$$V = \begin{bmatrix} U & u \\ u^T & z \end{bmatrix}$$

So we have considered the matrix :
Saying that U - uuT is positive semi definite is equivalent to saying that V is positive semidefinite.

Hence, we consider the variable V, instead of X, in our problem, and z that is equal to 1.

For the constraints, we know that each element of the diagonal U is equal to 1 (ensuring that the coefficients of u are between -1 and 1), entailing that the diagonal of V is an array of 1, we also forced V to be symmetric (in the definition of the variable with cvx.Variable), and that the last coefficient of V, that is, the coefficient at indices (N + 1, N + 1) is equal to z.

$$V \succcurlyeq 0$$
$$V_{N+1,N+1} = z$$
$$diag(V) = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}$$
$$z = 1$$

For the constraints on the movements and the colors, it is the same idea as the problems above with a slight change, for X is not directly embedded in V, but its value exists through the vector u.
Indeed, we know for a fact that u = 2X - 1, so, for instance, when we say that the sum of X_ijp is equal to 0, we need to replace the vector X by u, which is the N-th column of V.
So we need to sum over the coefficients of X, which is a sum over the coefficients of u, plus n, divided by 2. By moving the constants to the right insight, we obtain that the sum of coefficients of u should be equal to -n, and 2 - n if the sum over the coefficients of X were equal to 1.
We can apply this same logic to the other constraints, including the constraints over the colors, just by using the corresponding value Xijp expressed with u.

$$\sum_{j=1}^{n} V_{i,j,p}^{N} = \begin{cases} -n \text{ if i was is renovation at phase } (p-1) \\ 2-n \text{ otherwise} \end{cases} \quad \forall j = 1, \dots, n; \forall p = 0, \dots P$$
$$\sum_{i=1}^{n} V_{i,j,p}^{N} = \begin{cases} -n \text{ if j is in renovation or j is the office N1 at phase (p - 1)} \\ 2-n \text{ otherwise} \end{cases} \quad \forall i = 1, \dots, n; \forall p = 0, \dots P$$

$$\begin{cases} W_{i,c,j,p} <= Couleurs_{c,i,p-1} \\ W_{i,c,j,p} <= \frac{V_{i,j,p}^{N}+1}{2} \\ W_{i,c,j,p} >= 0 \\ W_{i,c,j,p} >= Couleurs_{c,i,p-1} + \frac{V_{i,j,p}^{N}+1}{2} - 1 \end{cases} \quad \forall c = 1, \dots, 4; \forall i = 1, \dots, n; \forall j = 1, \dots, n; p = 1, \dots P$$

And to finish, for the objective, we need to compute the trace of the product of some symmetric matrix with V, we have chosen the matrix:

$$C \in \mathbb{S}^{N \times N} \text{ s.t } C = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \dots & 0 \end{bmatrix}$$

So when, we compute Tr(CV), we compute two times the sum of the coefficients that we are interested in (the column/line of the matrix C is exactly the vector c that we used in the previous problem (minus the dimension corresponding to z) that doesn't take into account the movement of offices into themselves).

So when we compute the optimal value of the problem, we just have to compute $(\mathrm{Tr}(CV)/ 2) / 2 + N$.

For the second graph, the idea is the same.

Finally, for the quadratic SDP problem, we just took our SDPs problems, and added the quadratic term in the objective.