## Tutorial session 3 - Integer Linear Programming

Many optimization problems can be formulated as ILP problems. This session concerns a permutation problem which can be efficienty solved with an ILP solver.

**Problem.** $n$ objects are spread on the floor and we want to store them in $n$ boxes (see Figure 1) in such a way that the sum of the moving distances is minimized.

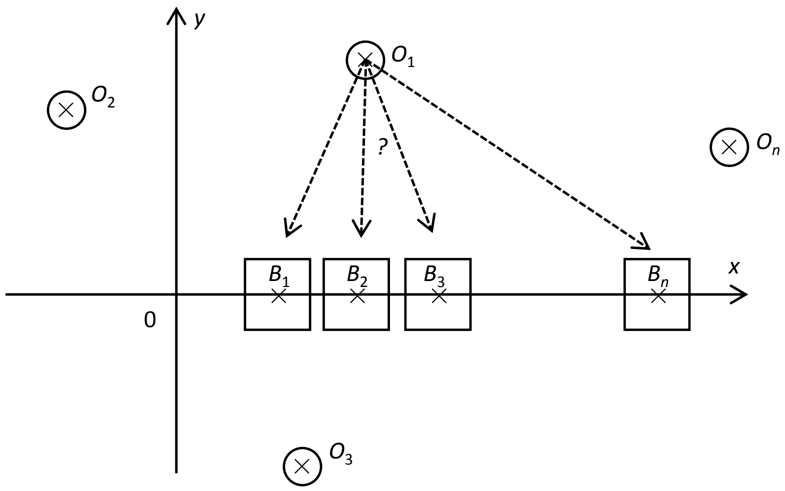


FIGURE 1 – Positions in the plane $(x, y)$ of the boxes and of the objects to be moved. For $i \in \{1, \ldots, n\}$, the point $B_i$ is the position of box $i$. In the same way, for $j \in \{1, \ldots, n\}$, the point $O_j$ is the position of object $j$.

Problem $(\mathcal{P}_0)$ consists in moving objects in such a way that :

— there is only one object per box,

— each object is in a box,

— the total moving distance is minimized. It is calculated by summing the distances between each object and its associated box. We consider the Euclidean distance (when moving the object $j$ to the box $i$, the distance is $\|O_j - B_i\|$).

The main objective of this exercise is to consider various constraints, to formulate constrained problems as ILP problems and to solve them. The developed scripts have to be tested with the files `PositionsBoxes.txt` and `PositionsObjects.txt` which give the coordinates in the plane $(x, y)$ of points $B_i$ and $O_j$.

The Matlab command `intlinprog` allows to solve ILP and MILP problems. For each resolution, please indicate the box number associated to each object and the optimized distance. You can also illustrate the solution you obtained with a figure. In order to help you the provided Matlab script `PlotSolution.m` called in the script `Main.m` permits a graphical representation of a solution.

Necessary files (data, scripts) may be downloaded from the class webpage.

**Questions.**

Let $\boldsymbol{x} \in \{0,1\}^{n\times n}$ be the binary matrix coding whether the box $i$ contains the object $j$ ($x_{i,j} = 1$) or not ($x_{i,j} = 0$).

**Q1.** Preliminary question : write linear equations which model that the box $i$ contains an object and only one and that the object $j$ is in a box and only one.

**Q2.** Formulate the optimization problem $(\mathcal{P}_0)$ as an ILP problem. Let $L$ be the matrix involved in the definition of the linear constraints. Show that $L$ is a totally unimodular matrix.
The Matlab command `linprog` allows to solve LP problems. Use this command to solve the problem for the given example and verify that the obtained solution is integer-valued.
The triangular inequality alows to construct a necessary condition that must satisfy the solution : what is this condition ?

**Q3.** Problem $(\mathcal{P}_1)$.
Modify the scripts developed for $(\mathcal{P}_0)$ to take into account the following constraint : object 1 can be neither in box 1 nor in box $n$. Is the use of the command `linprog` is still relevant ?

**Q4.** Problem $(\mathcal{P}_2)$.
Modify the scripts developed for $(\mathcal{P}_0)$ to take into account the following constraint : object 2 must be in the box located just to the right of the box containing object 3.

**Q5.** Problem $(\mathcal{P}_3)$.
What represent the constraints $\forall i, \quad x_{i,4}+\sum_{|k|\leq 3} x_{i+k,5} \leq 1$ ? Modify the scripts developed for $(\mathcal{P}_0)$ to take into account these new constraints.

**Q6.** Problem $(\mathcal{P}_4)$.
There must be at most 2 boxes between the boxes containing objects 6 and 7. Modify your scripts to consider this new constraint.

**Q7.** We would like to test if the obtained solution is unique. Propose a strategy and conclude on the uniqueness of the solutions for problems $(\mathcal{P}_2)$ and $(\mathcal{P}_3)$.