

# Projet d'optimisation linéaire

Envoi d'un message crypté sur un canal avec du bruit creux

Deadline: vendredi 23 novembre

## 1 Description du problème

Alice et Bob souhaitent s'envoyer des messages cryptés via un canal contenant du bruit creux, c'est-à-dire un canal ne perturbant qu'un petit nombre des entrées du message, mais les entrées perturbées le sont très fortement. Plus précisément, Alice veut envoyer un message binaire

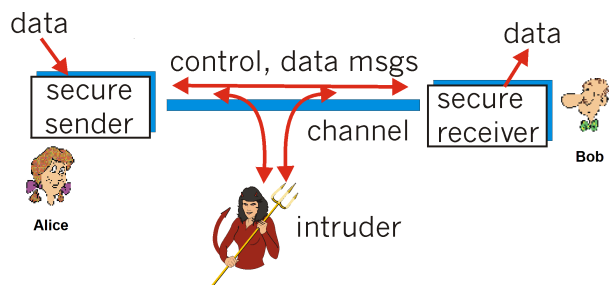


Figure 1: Illustration du problème.

$x \in \{0,1\}^p$  à Bob. Avant de commencer leur communication, Alice et Bob se sont rencontrés et se sont mis d'accord sur le choix d'une matrice d'encodage  $A \in \mathbb{R}^{m \times p}$  où  $m \geq p$  (on utilisera pour ce projet  $m = 4p$ ). Ainsi, Alice encode le message en utilisant la matrice  $A$  et envoie le message  $y = Ax \in \mathbb{R}^m$  sur le canal. Le canal va transmettre le message bruité  $y' = Ax + n$  à Bob où le vecteur de bruit  $n$  ne contient qu'un petit nombre d'entrées non nulle (par exemple, 10%). En présence de ce type de bruit, une bonne approche<sup>1</sup> est de minimiser la norme 1 de l'erreur. En termes mathématiques, afin de récupérer le message d'Alice, Bob aurait intérêt à résoudre le problème d'optimisation suivant:

$$\min_{x' \in \mathbb{R}^p} \|Ax' - y'\|_1 \quad \text{tel que} \quad x' \in \{0,1\}^p,$$

où  $\|z\|_1 = \sum_{i=1}^m |z_i|$  pour  $z \in \mathbb{R}^m$ . Malheureusement, ce problème est combinatoire et difficile à résoudre. En pratique, il est courant d'utiliser la relaxation continue suivante

$$\min_{x' \in \mathbb{R}^p} \|Ax' - y'\|_1 \quad \text{tel que} \quad 0 \leq x' \leq 1, \quad (1)$$

et d'arrondir la solution obtenue. Si le bruit n'est pas trop important,  $x' \approx x$ , ce qui permet à Bob de récupérer le message d'Alice.

L'objectif de ce projet est l'étude du problème (1), et ainsi de pouvoir décoder les messages d'Alice.

---

<sup>1</sup>Pour ceux que cela intéresse, voir l'article *Candès and Tao, Decoding by Linear Programming, IEEE Transactions on Information Theory, 2005*.

## 2 Questions

1. Modélisez le problème comme un problème d'optimisation linéaire. Expliquez votre raisonnement.
2. Ecrivez ce problème linéaire sous forme standard.
3. Utilisez Octave<sup>2</sup> et la fonction *glpk* pour déchiffrer le message fourni sur le site du cours (messagedAlice.mat)? Quel est le message envoyé par Alice?
4. La solution obtenue est-elle un sommet du polyèdre correspondant? Justifiez.
5. Générez maintenant vous-même un message: jusqu'à quel niveau de bruit peut-on déchiffrer votre message (c'est-à-dire combien d'entrées de  $y'$  peut-on perturber)? Cela est-il surprenant? Commentez très brièvement.
6. Utilisez maintenant *glpk* en imposant des variables binaires: pouvez-vous déchiffrer votre message avec un niveau de bruit supérieur?

**Consignes.** Le travail se réalise par groupe de 2 (*un* groupe de 3 est autorisé si nécessaire). Veuillez fournir avec le rapport les codes implémentés en annexe. Le rapport ne doit pas dépasser 5 pages (en dehors des annexes et page de garde). Le tout est à envoyer pour le 23 novembre à [valentin.leplat@umons.ac.be](mailto:valentin.leplat@umons.ac.be).

---

<sup>2</sup>Octave est un langage extrêmement similaire à Matlab, excepté qu'Octave est un logiciel libre (=gratuit). Une version avec une interface graphique similaire à Matlab est par exemple disponible sur la page <https://github.com/octave-de/mxeoctave.osuv.de> pour Windows. Voir [http://wiki.octave.org/Octave\\_for\\_MacOS\\_X](http://wiki.octave.org/Octave_for_MacOS_X) pour les Mac's. Cependant, il est également possible de faire le projet en Matlab en utilisant la fonction *linprog* (et *bintprog* dans le cas binaire).