Cours n°4: Modélisation de Programmes Linéaires (en Nombres Entiers), techniques générales et études de cas

Nicolas DUPIN

https://github.com/ndupin/ORteaching
http://nicolasdupin2000.wixsite.com/research

version du 20 mars 2022

Issu et étendu d'enseignements donnés à l'ENSTA Paris, Polytech' Paris-Saclay, et à l'université Paris-Saclay

Rappels : optimisation linéaire

Optimisation Linéaire : les contraintes et la fonctions objectifs sont linéaires :

$$
\begin{aligned}
& \min \sum_{i=1}^{m+p} c_i x_i \\
s.c: \quad & \sum_{i=1}^{m+p} A_{i,c} x_i \geqslant b_c \quad \forall c \in [\![1; C]\!] \\
& x \in \mathbb{N}^m \times \mathbb{R}_+^p
\end{aligned}
\tag{1}
$$

Un objectif linéaire : donné par un produit scalaire $c.x$

Un ensemble de contraintes linéaires $A.x \geqslant b$, une matrice de contraintes A, un second membre b.

A,b,c sont les données d'entrée, peuvent être paramétriques.

Par transformation d'écritures/ajouts de variables, on peut remplacer min / max, avoir des contraintes d'égalités ou d'inégalités

Rappels : optimisation linéaire

Programme Linéaire (PL, Linear Programming LP) : les variables sont continues :

$$\begin{aligned} & \min c.x \\ s.c: \quad & A.x \geqslant b \\ & x \in \mathbb{R}_+^p \end{aligned} \tag{2}$$

Programme Linéaire en Nombres Entiers (PLNE, Integer Linear Programming ILP) : les variables sont discrètes :

$$\begin{aligned} & \min c.x \\ s.c: \quad & A.x \geqslant b \\ & x \in \mathbb{N}^m \end{aligned} \tag{3}$$

Cas général : Programme Mixte en Nombres Entiers (Mixed Integer Linear Programming)

Exemples d'applications industrielles

- Saint Gobain, Découper des plaques de verres selon des demandes de grandes plaques normalisées en minimisant les pertes.
- Air Liquide, distribution de gaz liquide par voie routière
- Google, optimisation de l'usage de processeurs sur une grille d'ordinateurs
- SNCF, grilles d'horaires de trains, affectation de voies et maintenances.
- EDF, planification des maintenances des centrales nucléaires
- Air France, planification/replanification online de vols et d'équipage pour gérer des aléas.
- Renault, ordonnancement de production sur les lignes d'assemblages avec les caractéristiques individuelles de chaque véhicule.
- France Telecom, planning et routage de tournées de techniciens devant réaliser des opérations d'installation d'ADSL.
- ONERA, Gestion des prises de vue réalisées par un satellite agile d'observation de la Terre.

cf : challenge ROADEF, https ://www.roadef.org/roadef-le-challenge

PLNE de la littérature académique

- Problème de flots, multiflots dans un réseau (routier, télécom)
- Problèmes de design de réseaux.
- Problème de planifications (production, maintenance, emplois du temps).
- Optimisation conjointe production/transport .
- Problème d'ordonnancement de tâches dans un atelier (Job shop, flow shop).
- Problèmes de rangement (Bin packing 2d, 3d)

$\Longrightarrow$ Comment/pourquoi l'optimisation linéaire en nombres entiers offre-t-elle un cadre de modélisation aussi riche ?

Plan

Plan

Techniques générales de modélisation PL/PLNE

Relations PLNE, PQNE, QUBO, 3-SAT, et NP-complétude

Modélisation PLNE, problèmes d'affectation

Modélisation PLNE, problèmes de clustering discret

Modélisation PLNE de problèmes d'ordonnancement

Unit Commitment Problem, UCP, modélisation PLNE

Remarques sur le cadre linéaire

- Le cadre linéaire paraît restrictif à première vue
- PLNE $\Rightarrow$ optimisation convexe ? Non, car variables discrètes, ensemble de définition non convexe (non connexe).
- Quand on "linéarise" un PLNE, ce n'est pas forcément que l'on simplifie avec des expressions linéaires (ex : développement limités à l'ordre 1)
- Avoir des variables binaires permet de linéariser bcp de problèmes dans le cadre PLNE
- Le cadre continu de la PL permettra moins de linéarisations

Qu'est-ce que linéariser en PL/PLNE ?

- Quand on "linéarise" un PLNE, ce n'est pas forcément que l'on simplifie avec des expressions linéaires (ex : développement limités à l'ordre 1)
- Linéarisation PLNE en ajoutant des variables et/ou des contraintes, un problème non linéaire sur un espace de variables peut être rendu linéaire dans un plus grand espace vectoriel.
- Linéarisation génériques (et implémentées dans certains solveurs) pour valeur absolue, produit de variables binaires, fonction linéaire par morceaux, contraintes logiques ...

$\Longrightarrow$ Cadre de modélisation et résolution efficace pour des PL/PLNE génériques.

Retour aux lignes de production

Problème : Pour des contraintes d'installation physique et électrique, les lignes D et F ne peuvent pas fonctionner simultanément, et la ligne E ne peut fonctionner que si A fonctionne.

Ligne de prod	A	B	C	D	E	F
Coût de prod	2	6	7	2	3	4
Capacité de prod	3	4	5	2	6	6
Conso énergétique	450	450	500	200	350	250

A produire : 17 containers de masques, sans dépasser 1600 kWh de consommation électrique.

- les lignes D et F ne peuvent pas fonctionner simultanément $x_D + x_F \leqslant 1$

- la ligne E ne peut fonctionner que si la ligne A fonctionne : $x_E \leqslant x_A$ i.e. $x_E - x_A \leqslant 0$

Linéarisation de contraintes logiques entre binaires (1)

Des opérations logiques entre variables binaires $x, y, z \in \{0, 1\}$ peuvent s'écrire comme des contraintes linéaires.

$x = 1 \Longrightarrow y = 1$ se traduit par $x \leqslant y$.

$x = 1 \Longrightarrow y = 0$ se traduit par $x \leqslant 1 - y$, i.e. $x + y \leqslant 1$.

$x = 0 \Longrightarrow y = 1$ se traduit par $1 \leqslant x + y$.

$x = 0 \Longrightarrow y = 0$ se traduit par $y \leqslant x$.

Remarque : il s'agit de la même réécriture en considérant des transformations $y \leftrightarrow 1 - y$, $x \leftrightarrow 1 - x$)

Remarque : il est souvent naturel de raisonner avec des implications, on peut le faire et "convertir" après en équations linéaires (c'est une gymnastique mentale à prendre)

Linéarisation de contraintes logiques entre binaires (2)

Des opérations logiques entre variables binaires $x, y, z \in \{0, 1\}$ peuvent s'écrire comme des contraintes linéaires.

$x = 1 \Longrightarrow y = 1$ se traduit par $x \leqslant y$.

$y = 1$ OU $z = 1$ se traduit par $y + z \geqslant 1$.

$y = 1$ XOR $z = 1$ se traduit par $y + z = 1$.

NON $z = 1$ se traduit par $z = 0$ ou $z \leqslant 0$.

$y = 1$ ET $z = 1$ peut se traduire par $y + z = 2$ ou $y + z \geqslant 2$, ou sinon deux contraintes.

(et transformations $x \leftrightarrow 1 - x$, $y \leftrightarrow 1 - y$, $z \leftrightarrow 1 - z$)

Linéarisation de contraintes logiques entre binaires (3)

Des opérations logiques entre N variables binaires $x_1, \dots, x_N \in \{0,1\}$ peuvent s'écrire comme des contraintes linéaires.

$x_1 = 1$ OU ... OU $x_N = 1$ se traduit par $\displaystyle\sum_{i=1}^{N} x_i \geqslant 1$.

$x_1 = 1, \dots, x_N$ sont incompatibles se traduit par $\displaystyle\sum_{i=1}^{N} x_i \leqslant 1$.

$x_1 = 1$ XOR ... XOR $x_N = 1$ se traduit par $\displaystyle\sum_{i=1}^{N} x_i = 1$.

$x_1 = 1$ ET ... ET $x_N = 1$ peut se traduire par $\displaystyle\sum_{i=1}^{N} x_i \geqslant N$ ou $\displaystyle\sum_{i=1}^{N} x_i = N$, ou sinon N contraintes.

(et transformations $x_i \leftrightarrow 1 - x_i$)

Linéarisation de contraintes logiques entre binaires (4)

Si on a N variables binaires $x_1, \ldots, x_N \in \{0, 1\}$ toutes incompatibles.

$x_1 (= 1), \ldots, x_N (= 1)$ sont incompatibles se traduit par $\displaystyle\sum_{i=1}^{N} x_i \leqslant 1$.

On aurait aussi pu écrire l'ensemble de contraintes :

$$\forall i \in [\![1, N]\!],\ \forall j \in [\![i+1, N]\!], \quad x_i + x_j \leqslant 1 \tag{4}$$

Ces deux modélisations sont valides, et définissent les mêmes points entiers réalisables

La première version a une seule contrainte au lieu de $N(N-1)/2$, c'est sans doute mieux pour un algorithme de type simplexe.

Une autre raison de privilégier la première version, plus regroupée ...

Relaxation continue et variables incompatibles

Si on considère le PLNE suivant, avec 3 variables binaires incompatibles $x_1, x_2, x_3 \in \{0, 1\}$:

$$\max_{x_1,x_2,x_3 \in \{0,1\}} \quad 1.1x_1 + 1.2x_2 + 1.3x_3$$
$$x_1 + x_2 \leqslant 1 \qquad \text{(a)}$$
$$x_2 + x_3 \leqslant 1 \qquad \text{(b)}$$
$$x_1 + x_3 \leqslant 1 \qquad \text{(c)}$$

Le relâché continu de ce PLNE donne (par un algo de type simplexe) pour optimum $x_1 = x_2 = x_3 = \frac{1}{2}$, et une valeur de borne supérieure 1.8

Si on écrit la contrainte : $x_1 + x_2 + x_3 \leqslant 1$, la relaxation continue donne une solution entière, $x_1 = x_2 = 0$ et $x_3 = 1$, c'est l'optimum entier, de valeur 1.3

La contrainte $x_1 + x_2 + x_3 \leqslant 1$ est plus forte que (a), (b) et (c), elle les implique avec la positivité des variables, et contient moins de points continus réalisables.

A RETENIR : Regrouper les contraintes d'incompatibilité améliore la qualité de la relaxation continue. (on le formalisera et le généralisera)

Linéarisation de contraintes disjonctives (1)

On suppose que $y \in \{0,1\}$ est une variable binaire, ou une expression linéaire à valeurs binaire (cela revient au même pour avoir une linéarisation)

Contraintes disjonctives : $y = 1$ OU $\displaystyle\sum_{i=1}^{N} b_{i,c}.x_i \leqslant b_c$ pour toute contrainte $c \in [\![1, C]\!]$ (éventuellement $C > 1$)

On suppose avoir des valeurs M_c qui vérifient toujours $\sum_{i=1}^{N} b_i x_i \leqslant M_c$

Il suffit de considérer alors $\displaystyle\sum_{i=1}^{N} b_{i,c}.x_i \leqslant b_c + y(M_c - b_c)$ pour tout $c \in [\![1, C]\!]$

N.B : si on avait $y = 0$ à la place de $y = 1$, on se ramène au cas précédent avec $\overline{y} = 1 - y = 1$

N.B : Pour l'efficacité pratique, il vaut mieux prendre les valeurs M_c les plus petites possibles

Linéarisation de contraintes disjonctives (2)

À partir de deux contraintes disjonctives : $\sum_{i=1}^{N} a_i x_i \leqslant a$ OU $\sum_{i=1}^{N} b_i.x_i \leqslant b$

Une linéarisation : on introduit une variable binaire z qui imposée à 1 impliquera $\sum_{i=1}^{N} a_i x_i \leqslant a$ et qui dans le cas $z = 0$ imposera $\sum_{i} b_i x_i \leqslant b$.

On prend deux valeurs M_a, M_b qui vérifient $\sum_{i=1}^{N} a_i x_i - a \leqslant M_a$ et $\sum_{i=1}^{N} b_i x_i - b \leqslant M_b$. On considère les contraintes :

$$\sum_{i=1}^{N} a_i x_i \leqslant a + (1 - z) M_a$$

$$\sum_{i=1}^{N} b_i x_i \leqslant b + z M_b$$

N.B : si on a une expression binaire, il est plus efficace de considérer la linéarisation précédente, ce n'est pas la même linéarisation induite !

Peut on généraliser avec des systèmes de contraintes disjonctifs ?

Linéarisation de contraintes disjonctives (3)

$$\sum_{i=1}^{N} a_{i,c} x_i \leqslant a_c \text{ pour tout } c \in [\![1, C]\!] \text{ OU } \sum_{i=1}^{N} b_{i,c'} . x_i \leqslant b_{c'} \text{ pour tout } c' \in [\![1, C']\!]$$

Une linéarisation : on introduit une variable binaire z qui imposée à 1 impliquera les contraintes indicées par $c \in [\![1, C]\!]$ et qui dans le cas $z = 0$ imposera les contraintes indicées par $c' \in [\![1, C']\!]$.

On prend des valeurs $M_c^a, M_{c'}^b$ qui vérifient $\sum_{i=1}^{N} a_{i,c} x_i - a_c \leqslant M_c^a$ et $\sum_{i=1}^{N} b_{i,c'} x_i - b_{c'} \leqslant M_{c'}^b$. On considère les contraintes :

$$\sum_{i=1}^{N} a_{i,c} x_i \leqslant a_c + (1 - z) M_c^a \text{ pour tout } c \in [\![1, C]\!]$$

$$\sum_{i=1}^{N} b_{i,c'} x_i \leqslant b_{c'} + z M_{c'}^b \text{ pour tout } c' \in [\![1, C']\!]$$

$\Longrightarrow$ On ajoute bien une seule variable binaire

Linéarisation de contraintes conjonctives

Pour des contraintes $\sum_{i=1}^{N} a_i x_i \leqslant a$ ET $\sum_{i=1}^{N} b_i x_i \leqslant b$?

On écrit tout simplement les deux types de contraintes dans le programme linéaire

Les contraintes s'ajoutent indépendamment dans un modèle PLNE.

Linéarisation de contraintes d'implication (1)

On suppose que $y \in \{0,1\}$ est une variable binaire ou une expression linéaire à valeurs binaire (cela revient au même pour avoir une linéarisation)

Pour des contraintes $y = 1$ IMPLIQUE $\sum_i b_i.x_i \leqslant b$?

$y = 1 \Longrightarrow C_B$ est équivalent à (NON $y_i = 1$) OU C_B, cad $y = 0$ OU C_B

On est ramené à linéariser une contrainte avec un OU comme précédemment

Soit M_b qui vérifie $\sum_{i=1}^N b_i x_i - b \leqslant M_b$. On considère la contrainte suivante, pas d'ajout de variable binaire :

$$\sum_{i=1}^N b_i x_i \leqslant b + M(1 - y)$$

N.B : si on avait $y = 0$ à la place de $y = 1$, on se ramène au cas précédent avec $\overline{y} = 1 - y = 1$

N.B : En pratique, il faut prendre M_b le plus petit possible.

Linéarisation de contraintes d'implication (2)

Pour des contraintes $\sum_{i=1}^{N} a_i x_i \leqslant a$ IMPLIQUE $\sum_i b_i.x_i \leqslant b$?

$A \Longrightarrow B$ est équivalent à (NON A) OU B

NON $(\sum_i a_i x_i \leqslant a) = \sum_i a_i x_i > a =$ (ou $\approx$) $(\sum_i a_i.x_i \geqslant a + \varepsilon)$ avec $\varepsilon > 0$ très petit ou plus petit que le grain des données de la contrainte ($\varepsilon = 1$ si on n'a que des entiers).

On est ramené à linéariser une contrainte avec un OU comme précédemment

- $\displaystyle\sum_{i=1}^{N} a_i x_i \geqslant a + \varepsilon - zM_a$
- $\displaystyle\sum_{i=1}^{N} b_i x_i \leqslant b + (1 - z)M_b$

Où M_a, M_b vérifient $\sum_{i=1}^{N} a_i x_i - a \geqslant \varepsilon - M_a$ et $\sum_{i=1}^{N} b_i x_i - b \leqslant M_b$ et z est une variable binaire additionnelle.

Linéarisation de valeur absolue, premier exemple

- ex : $\min_{x\in[-2,7],y\in[-2,2]} |x+y+1| + y$
- On introduit $z = |x+y+1| \geqslant 0$
- $z \geqslant 0$, $z = \max(x+y+1, -x-y-1)$

$$\begin{array}{rl} & \min_{x,y,z} z + y \\ s.c: & z \geqslant x+y+1 \\ & z \geqslant -x-y-1 \\ & x \in [-2,7], y \in [-2,2], z \geqslant 0 \end{array} \tag{5}$$

Linéarisation de valeur absolue dans fonction objectif

$$\begin{array}{lll} & \displaystyle\min_{x \in \mathbb{N}^m \times \mathbb{R}_+^p} \sum_{i=1}^{m+p} c_i^0 x_i + \sum_{j=1}^{J} \left| \sum_{i=1}^{m+p} c_i^j x_i \right| & \\ s.c: & \sum_{i=1}^{m+p} A_{i,c} x_i \geqslant b_c & \forall c \in [\![1; C]\!] \end{array} \tag{6}$$

On introduit pour tout $j \in [\![1; J]\!]$, "$z_j = \left| \sum_{i=1}^{m+p} c_i^j x_i \right| \geqslant 0$"

$$\begin{array}{lll} & \displaystyle\min_{x \in \mathbb{N}^m \times \mathbb{R}_+^p, z \in \mathbb{R}_+^J} \sum_{i=1}^{m+p} c_i^0 x_i + \sum_{j=1}^{J} z_j & \\ s.c: & \sum_{i=1}^{m+p} A_{i,c} x_i \geqslant b_c & \forall c \in [\![1; C]\!] \\ & \sum_{i=1}^{m+p} c_i^j x_i \leqslant z_j & \forall j \in [\![1; J]\!] \\ & -\sum_{i=1}^{m+p} c_i^j x_i \leqslant z_j & \forall j \in [\![1; J]\!] \end{array} \tag{7}$$

$\Longrightarrow$ Linéarisation en ajoutant J variables continues et $2J$ contraintes

Linéarisation de contrainte avec valeur absolue (1)

D'une contrainte $\displaystyle\sum_{i=1}^{m+p} c_i^0 x_i + \left|\sum_{i=1}^{m+p} c_i^1 x_i\right| \leqslant b$, on a 2 contraintes linéaires :

$$\sum_{i=1}^{m+p} c_i^0 x_i + \sum_{i=1}^{m+p} c_i^1 x_i \leqslant b \text{ et } \sum_{i=1}^{m+p} c_i^0 x_i - \sum_{i=1}^{m+p} c_i^1 x_i \leqslant b$$

Preuve : $\displaystyle\left|\sum_{i=1}^{m+p} c_i^1 x_i\right| = \max\left(\sum_{i=1}^{m+p} c_i^1 x_i, -\sum_{i=1}^{m+p} c_i^1 x_i\right)$

D'une contrainte $\displaystyle\sum_{i=1}^{m+p} c_i^0 x_i + \sum_{j=1}^{J}\left|\sum_{i=1}^{m+p} c_i^j x_i\right| \leqslant b$, on a 2^J contraintes linéaires :

$$\forall \varepsilon \in \{-1;1\}^J, \quad \sum_{i=1}^{m+p} c_i^0 x_i + \varepsilon_j \sum_{i=1}^{m+p} c_i^1 x_i \leqslant b$$

Linéarisation de contrainte avec valeur absolue (2)

Avec les contraintes $\displaystyle\sum_{i=1}^{m+p} c_i^0 x_i + \left|\sum_{i=1}^{m+p} c_i^1 x_i\right| \geqslant b$, plus dur !

Cela se reformule en $\sum_{i=1}^{m+p} c_i^1 x_i \geqslant b - \sum_{i=1}^{m+p} c_i^0 x_i$ ou $\left|\sum_{i=1}^{m+p} c_i^1 x_i\right| \geqslant b - \sum_{i=1}^{m+p} c_i^0 x_i$

On introduit une variable binaire $z \in \{0;1\}$, avec $z = 1$ si $\left|\sum_{i=1}^{m+p} c_i^1 x_i\right| = \sum_{i=1}^{m+p} c_i^1 x_i$

$$\sum_{i=1}^{m+p} c_i^1 x_i \geqslant b - \sum_{i=1}^{m+p} c_i^0 x_i - Mz$$

$$-\sum_{i=1}^{m+p} c_i^1 x_i \geqslant b - \sum_{i=1}^{m+p} c_i^0 x_i - M(1-z)$$

Linéarisation de minimum-maximum dans fonction objectif

On considérant un objectif min-max (analogue pour max-min) :

$$\begin{aligned}
&\min_{x \in \mathbb{N}^m \times \mathbb{R}_+^p} \sum_{i=1}^{m+p} c_i^0 x_i + \max_{j \in [\![1,J]\!]} \left(\sum_{i=1}^{m+p} c_i^j x_i \right) \\
s.c: \quad & \sum_{i=1}^{m+p} A_{i,c} x_i \geqslant b_c \qquad\qquad \forall c \in [\![1;C]\!]
\end{aligned} \tag{8}$$

On introduit une variable additionnelle, "$z = \max_{j \in [\![1,J]\!]} \left(\sum_{i=1}^{m+p} c_i^j x_i \right)$"

$$\begin{aligned}
&\min_{x \in \mathbb{N}^m \times \mathbb{R}_+^p, z \in \mathbb{R}} \; z + \sum_{i=1}^{m+p} c_i^0 x_i \\
s.c: \quad & \sum_{i=1}^{m+p} A_{i,c} x_i \geqslant b_c \qquad\qquad \forall c \in [\![1;C]\!] \\
& \sum_{i=1}^{m+p} c_i^j x_i \leqslant z \qquad\qquad \forall j \in [\![1;J]\!]
\end{aligned} \tag{9}$$

$\Longrightarrow$ Linéarisation en ajoutant 1 variables continues et J contraintes

Linéarisation de maximum dans une contrainte d'inégalité

D'une contrainte $\displaystyle\sum_{i=1}^{m+p} c_i^0 x_i + \max_{j \in [\![1,J]\!]} \left(\sum_{i=1}^{m+p} c_i^1 x_i \right) \leqslant b$, on a J contraintes linéaires :

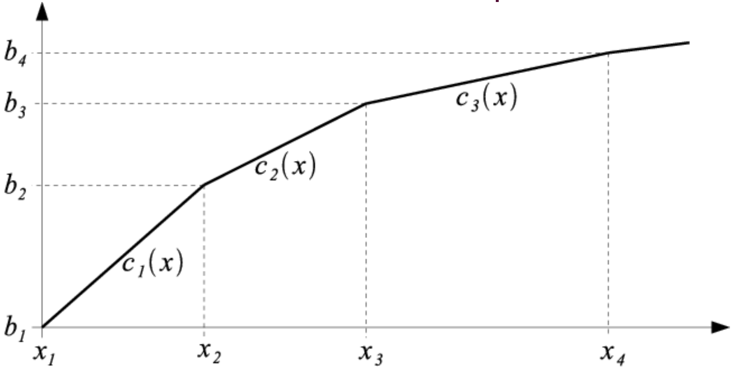
$$\forall j \in [\![1;J]\!], \quad \sum_{i=1}^{m+p} c_i^0 x_i + \sum_{i=1}^{m+p} c_i^1 x_i \leqslant b.$$

D'une contrainte $\displaystyle\sum_{i=1}^{m+p} c_i^0 x_i + \max_{j \in [\![1,J]\!]} \left(\sum_{i=1}^{m+p} c_i^1 x_i \right) \geqslant b$, on peut introduire J variables $\varepsilon_j \in \{0,1\}$ (ou $[0,1]$) et $J+1$ contraintes linéaires :

$$\forall j \in [\![1,J]\!], \quad \sum_{i=1}^{m+p} c_i^0 x_i + \varepsilon_j \sum_{i=1}^{m+p} c_i^1 x_i \leqslant b$$

$\sum_{j=1}^{J} \varepsilon_j \geqslant 1$

Fonction continue concave linéaire par morceaux



Equivalent à considérer
$f(x) = \min(c_1 x + b_1 + c_1 x_1, c_2 x + b_2 + c_2 x_2, c_3 x + b_3 + c_3 x_3, c_4 x + b_4 + c_4 x_4)$

se linéarise comme précédemment

N.B : en toute rigueur, on devrait dire "affine par morceaux" et non linéaire par morceaux. En anglais, "piecewise linear" et dans les articles de recherche et solveurs de RO.

Linéarisation de contraintes quadratiques, premier exemple

On part de l'exemple que l'on cherche à écrire en PLNE :

$$OPT = \min_{x,y,z \in \{0,1\}^3} x^2 - 2y^2 + z^2 + 3yz + 4xy + y$$
$$\text{s.c : } x + y + z \leqslant 2$$

On a déjà $x^2 = x$, $y^2 = y$ et $z^2 = z$, valant soit $0 = 0^2$ soit $1 = 1^2$

On introduit i une variable binaire, qui vaudra $i = xy \in \{0, 1\}$. On a les contraintes linéaires équivalentes :

- $i \geqslant x + y - 1$,
- $x \geqslant i$
- $y \geqslant i$

De même, on introduit y une variable binaire, qui vaudra $j = yz \in \{0, 1\}$

Linéarisation de contraintes quadratiques, premier exemple

$$OPT = \min_{x,y,z \in \{0,1\}^3} x^2 - 2y^2 + z^2 + 3yz + 4xy + y$$
s.c : $x + y + z \leqslant 2$

OPT s'écrit alors de manière équivalente en PLNE :

$$\begin{array}{rcl} OPT & = \displaystyle\min_{x,y,z,i,j \in \{0,1\}^5} & x - 2y + z + 3j + 4i + y \\ s.c: & x + y + z & \leqslant 2 \\ & x + y - 1 & \leqslant i \\ & i & \leqslant x \\ & i & \leqslant y \\ & z + y - 1 & \leqslant j \\ & j & \leqslant z \\ & j & \leqslant y \end{array} \tag{10}$$

Linéarisation de contraintes quadratiques (1)

Plus généralement :

Cas où intervient dans les contraintes un produit de variables binaires $x \times y$ où x, y sont deux variables binaires.

Une linéarisation générique si $x \neq y$: on introduit une variable binaire z qui vaudra $x \times y$. On remplace chaque occurrence de $x \times y$ par z et on ajoute les contraintes

- $z \geqslant x + y - 1$
- $x \geqslant z$
- $y \geqslant z$

Remarque : on peut déclarer les variables z comme des variables continues, avoir x et y binaires implique z binaire. Cela peut faire des différences sur les algorithmes de résolution

Retour au Sac à dos quadratique

Modélisation :

$$
\begin{array}{lll}
 & \max \sum_{i=1}^{n} c_i^{lin} x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{i,j}^{quad} x_i x_j & \\
s.c: & \sum_{i=1}^{n} m_i x_i \leqslant M & \\
\forall i & x_i \in \{0,1\} &
\end{array}
\tag{11}
$$

Ce n'est pas linéaire, à cause des produits de variables $x_i x_j$

Linéarisation ? Plus au prochain cours, c'est maintenant !

Sac à dos quadratique, formulation PLNE

$y_{i,j} = x_i x_j$ produit de variables binaires

$$
\begin{aligned}
& \max_{x,y \geqslant 0} \sum_{i=1}^{n} c_i^{lin} x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} c_{i,j}^{quad} y_{i,j} \\
s.c: \quad & \sum_{i=1}^{n} m_i x_i \leqslant M \\
\forall i < j \quad & y_{i,j} \leqslant x_i \\
\forall i < j \quad & y_{i,j} \leqslant x_j \\
\forall i < j \quad & y_{i,j} \geqslant x_i + x_j - 1 \\
\forall i \quad & x_i \in \{0,1\}
\end{aligned}
\tag{12}
$$

Remarque : on peut déclarer les variables $y_{i,j}$ comme des variables continues et positives si $c_{i,j}^{quad} \geqslant 0$: si les x_i sont tous entiers, il est est alors de même pour les $y_{i,j}$ à l'optimum.

Remarque 2 : si $c_{i,j}^{quad} = 0$, il n'y a pas besoin d'introduire des variables $y_{i,j}$

PQVB : Problème quadratique en variables binaires

$$
\begin{array}{rl}
\text{Soit un PQVB :} & \displaystyle\min_{x\in\{0,1\}^m} \sum_{i=1}^{m} c_i^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_i^{quad} x_i x_j \\
s.c : & \sum_{i=1}^{m+p} A_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} A_{i,c}^{quad} x_i x_j \geqslant a_c \quad \forall c \in [\![1; C]\!] \\
s.c : & \sum_{i=1}^{m+p} B_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} B_{i,c}^{quad} x_i x_j = b_c \quad \forall c \in [\![1; C']\!]
\end{array}
$$

1. On réduit les occurrences de $x_i^2 = x_i$
2. Il reste des produits $x_i \times x_j$ avec $i \neq j$. On linéarise en introduisant des variables $z_{i,j} = x_i \times x_j$ et les contraintes linéarisées, PQVB devient :

$$
\begin{array}{rll}
 & \displaystyle\min_{x,z} \sum_{i=1}^{m} c_i^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_i^{quad} z_{i,j} & \\
s.c : & \sum_{i=1}^{m+p} A_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} A_{i,c}^{quad} z_{i,j} \geqslant a_c & \forall c \in [\![1; C]\!] \\
 & \sum_{i=1}^{m+p} B_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} B_{i,c}^{quad} z_{i,j} = b_c & \forall c \in [\![1; C']\!] \\
 & x_i + x_j - 1 \leqslant z_{i,j} & \forall i < j \\
 & x_i \geqslant z_{i,j} & \forall i < j \\
 & x_j \geqslant z_{i,j} & \forall i < j \\
 & x \in \{0,1\}^m, z \in \{0,1\}^{\frac{m(m-1)}{2}} &
\end{array}
$$

Encore plus généralement

Cas où interviennent des produit de variables binaires $x_1 \times x_2 \times \cdots \times x_p$, dans un problème d'optimisation à valeurs binaires, objectifs et contraintes polynomiales

1. On réduit pour chaque monôme les occurrences multiples avec $x_i^p = x_i^{p-1} = \cdots = x_i^2 = x_i$, peu de linéarisation

Il reste des produits $x_1 \times x_2 \times \cdots \times x_{p'}$ tous distincts avec $p' \leqslant p$.

On linéarise en introduisant des variables $z_m = x_1 \times x_2 \times \cdots \times x_{p'}$ et les contraintes :

- $z_m \geqslant x_1 + x_2 + \cdots + x_{p'} - p' + 1$
- $x_1, x_2, \ldots, x_{p'} \geqslant z_m$

$\Longrightarrow$ On peut linéariser tout problème polynomial de degré p avec n variables binaires et m contraintes polynomiales de degré p en un PLVB avec au plus $n + \dfrac{(n-1)!}{p!(n-p)!}$ variables et $m + (p+1)\dfrac{n!}{p!(n-p)!}$ contraintes.

Remarque : on peut déclarer les variables z comme des variables continues, et avoir que n variables binaires pour la résolution du PLNE.

Linéarisation de contraintes quadratiques (2)

- Cas où intervient dans les contraintes un produit $x \times A$ où x est une variable binaire et A une variable continue bornée dans $[0, \bar{A}]$
- on introduit une variable continue y qui vaudra $x \times A$.
- $0 \leqslant y \leqslant \bar{A}x$
- En effet, si $x = 0$, alors $y = 0$. Sinon, on a bien $y \in [0, \bar{A}]$

Linéarisation de contraintes quadratiques (3)

- Cas où intervient dans les contraintes un produit $n \times A$ où $n \in [\![0, \bar{N}]\!]$ est une variable discrète et bornée et A une variable continue bornée dans $[0, \bar{A}]$.
- On se ramène au cas précédent, en écrivant $n = \displaystyle\sum_{i=1}^{N} i x_i$, où $x_i = 1$ ssi $n = i$, avec la contrainte additionnelle $\sum_{i=1}^{N} i x_i = 1$
- On a alors $n \times A = \displaystyle\sum_{i=1}^{N} i x_i \times A$, et on linéarise les produits $x_i \times A$ comme précédement
- Par contre, on n'a pas de linéarisation générale pour un produit de variables continues.

Bilan partiel

- Quand on “linéarise” un PLNE, ce n’est pas forcément que l’on simplifie avec des expressions linéaires (ex : développement limités à l’ordre 1)
- Linéarisation PLNE en ajoutant des variables et/ou des contraintes, un problème non linéaire sur un espace de variables peut être rendu linéaire dans un plus grand espace vectoriel.
- Linéarisations vues : valeur absolue, fonctions linéaire par morceaux, optimisation max-min, expressions quadratiques de produits de variables binaires, produit binaire/continu borné ...
- Les expressions logiques (implications, ou, exclusion), sur lesquelles on raisonne plus naturellement s’écrivent automatiquement en PLNE.
- Face à un problème complexe, on raisonne ”contrainte par contrainte”, transformant chaque règle en système de contraintes qui s’ajoutent, apportant une facilité de modélisation.

$\Longrightarrow$ Beaucoup de problèmes industriels complexes s’écrivent en PLNE

Plan

PLNE, PQNE, QUBO, 3-SAT

- On va montrer que d'autres classes de problèmes bien connus sont contenues dans la PLNE, et même plus précisément dans la classe des PLNE à variables binaires (noté PLVB dans la suite).
- Cela a un intérêt théorique, pour montrer que ces classes de problèmes sont "aussi durs les uns que les autres"
- Formalisation, tous ces problèmes sont NP-complets
- En pratique, ce n'est pas une solution de transformer tout problème en PLNE et utiliser les boîtes à outils de ce cours, si des algorithmes spécifiques et différents existent dans tous ces cas, il y a une raison !
- ça voudra dire qu'avec les outils de ce cours, on pourra résoudre optimalement et assez facilement des petites instances de tous ces problèmes. Pour avoir la résolution la plus efficace, les algorithmes dédiés ont été conçus pour.
- Il peut exister des similitudes dans les résolutions spécifiques de ces problèmes, cette section donne un éclairage de structures communes ou voisines.

PLNE, PLVB

PLNE, Problème linéaire en nombres entiers, s'écrit sous la forme :

$$\begin{aligned} &\min_{x \in \mathbb{N}^m} \sum_{i=1}^{m} c_i x_i \\ s.c: \quad & \sum_{i=1}^{m+p} A_{i,c} x_i \geqslant a_c \quad \forall c \in [\![1; C]\!] \\ s.c: \quad & \sum_{i=1}^{m+p} B_{i,c} x_i = b_c \quad \forall c \in [\![1; C']\!] \end{aligned} \tag{13}$$

PLVB : Problème linéaire à variables binaires, PLNE avec $x \in \{0,1\}^m$

Un PLVB est un PLNE particulier. mais réciproquement ?

Avoir que des variables binaires ?

Ce sera le cas de bcp de PLNE vus dans ce cours.

Tout PLNE à variables entières bornées (cas applicatif courant, ou on s'y ramène),s'écrit comme un PLVB.

Soit z une variable à valeur dans $[\![0; Z]\!] = [0; Z] \cap \mathbb{Z}$

On transforme de manière équivalente z en $Z + 1$ variables binaires $x_i \in \{0, 1\}$ qui indique si la valeur z prend la valeur i

On remplace z par sa valeur $z = \sum_{i=0}^{Z} i x_i$, et on a un PLNE avec que des variables binaires

PQNE, PQVB

PQNE, Problème quadratique en nombres entiers :

$$
\begin{aligned}
& \min_x \sum_{i=1}^{m} c_i^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_i^{quad} x_i x_j \\
s.c: \quad & \sum_{i=1}^{m+p} A_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} A_{i,c}^{quad} x_i x_j \geqslant a_c \quad \forall c \in [\![1; C]\!] \\
s.c: \quad & \sum_{i=1}^{m+p} B_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} B_{i,c}^{quad} x_i x_j = b_c \quad \forall c \in [\![1; C']\!] \\
& x \in \mathbb{N}^m
\end{aligned}
\tag{14}
$$

PQVB : Problème quadratique à variables binaires, PQNE avec $x \in \{0,1\}^m$

Même transformation que précédemment, PQNE peut se ramener à PQVB avec des variables binaires bornées

Un PLVB est un PQVB particulier, sans terme quadratique. Réciproquement ?

$$\text{Soit un PQVB :}\quad \min_{x\in\{0,1\}^m} \sum_{i=1}^{m} c_i^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_i^{quad} x_i x_j$$

$$\begin{array}{lll} s.c: & \sum_{i=1}^{m+p} A_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} A_{i,c}^{quad} x_i x_j \geqslant a_c & \forall c \in [\![1;C]\!] \\ s.c: & \sum_{i=1}^{m+p} B_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} B_{i,c}^{quad} x_i x_j = b_c & \forall c \in [\![1;C']\!] \end{array}$$

1. On réduit les occurrences de $x_i^2 = x_i$
2. Il reste des produits $x_i \times x_j$ avec $i \neq j$. On linéarise en introduisant des variables $z_{i,j} = x_i \times x_j$ et les contraintes linéarisées, PQVB devient :

$$\min_{x,z} \sum_{i=1}^{m} c_i^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_i^{quad} z_{i,j}$$

$$\begin{array}{lll} s.c: & \sum_{i=1}^{m+p} A_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} A_{i,c}^{quad} z_{i,j} \geqslant a_c & \forall c \in [\![1;C]\!] \\ & \sum_{i=1}^{m+p} B_{i,c}^{lin} x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} B_{i,c}^{quad} z_{i,j} = b_c & \forall c \in [\![1;C']\!] \\ & x_i + x_j - 1 \leqslant z_{i,j} & \forall i < j \\ & x_i \geqslant z_{i,j} & \forall i < j \\ & x_j \geqslant z_{i,j} & \forall i < j \\ & x \in \{0,1\}^m, z \in \{0,1\}^{\frac{m(m-1)}{2}} & \end{array}$$

PLVB et PQVB

- Un PLVB est un PQVB particulier
- Tout PQVB avec n variables binaires et m contraintes se ramène à un PLVB avec au plus $n + \frac{n(n-1)}{2} = \frac{n(n+1)}{2} = \Theta(n^2)$ variables et $m + 3\frac{n(n-1)}{2}$ contraintes.
- La transformation d'une instance PQVB en instance PLVB est bien polynomiale en fonction de l'espace mémoire de l'instance initiale
- Prouver qu'un problème est NP-complet prouve que l'autre est NP-complet aussi

Problèmes SAT et 3-SAT

Problèmes SAT : problème de satisfaction booléennes, à partir d'une expression faisant intervenir N booléens $x_1, \dots, x_N$ et leur négation $\overline{x}_1, \dots, \overline{x}_N$, écrite sous forme normale conjonctive (FNC), existe t'il une affectation des variables telle que l'expression est vraie ?

Table de vérité : 2^N cas pour tout énumérer. Question, a t'on un algorithme polynomial ? SAT est le premier problème NP-complet

3-SAT, problème de satisfaction booléennes, avec uniquement des termes de degré 3 dans la FNC (exactement ou au plus, cela revient au même, transformation polynomiale)

3-SAT est NP-complet

2-SAT, avec une FNC de degré 2, définit un problème polynomial

Problèmes 3-SAT et PLVB

Soit une instance 3-SAT, définie sur 5 booléens $v_1, \dots, v_5$ et leur négation $\neg v_1, \dots, \neg v_5$ par l'évaluation :

$(\neg v_1 \vee v_2 \vee v_4) \wedge (\neg v_1 \vee v_2 \vee \neg v_5) \wedge (\neg v_3 \vee v_4 \vee v_5) \wedge (v_1 \vee v_2 \vee v_3) \wedge (\neg v_1 \vee v_2 \vee v_4) \wedge (\neg v_1 \vee v_2 \vee \neg v_5) \wedge (\neg v_3 \vee v_4 \vee v_5)$

On définit les variables binaires $x_1, \dots, x_5$ avec $x_i = 1$ ssi on affecte au booléen v_i la valeur VRAI.

3-SAT est équivalent à regarder le système de contraintes :

$$
\begin{array}{l}
1 - x_1 + x_2 + x_4 \geqslant 1 \\
1 - x_1 + x_2 + 1 - x_5 \geqslant 1 \\
1 - x_3 + x_4 + x_5 \geqslant 1 \\
x_1 + x_2 + x_3 \geqslant 1 \\
1 - x_1 + x_2 + x_4 \geqslant 1 \\
1 - x_1 + x_2 + 1 - x_5 \geqslant 1 \\
1 - x_3 + x_4 + x_5 \geqslant 1
\end{array}
$$

Problèmes 3-SAT et PLVB

Soit une instance 3-SAT, définie sur 5 booléens $v_1, \dots, v_5$ et leur négation $\neg v_1, \dots, \neg v_5$ par l'évaluation :

$(\neg v_1 \vee v_2 \vee v_4) \wedge (\neg v_1 \vee v_2 \vee \neg v_5) \wedge (\neg v_3 \vee v_4 \vee v_5) \wedge (v_1 \vee v_2 \vee v_3) \wedge (\neg v_1 \vee v_2 \vee v_4) \wedge (\neg v_1 \vee v_2 \vee \neg v_5) \wedge (\neg v_3 \vee v_4 \vee v_5)$

On définit les variables binaires $x_1, \dots, x_5$ avec $x_i = 1$ ssi on affecte au booléen v_i la valeur VRAI. 3-SAT est équivalent à regarder le problème d'optimisation PLVB :

$$\begin{aligned}
&\mathrm{OPT} = \max_{x_i} x_1 + x_2 + x_3 \\
&1 - x_1 + x_2 + x_4 \geqslant 1 \\
&1 - x_1 + x_2 + 1 - x_5 \geqslant 1 \\
&1 - x_3 + x_4 + x_5 \geqslant 1 \\
&1 - x_1 + x_2 + x_4 \geqslant 1 \\
&1 - x_1 + x_2 + 1 - x_5 \geqslant 1 \\
&1 - x_3 + x_4 + x_5 \geqslant 1
\end{aligned}$$

OPT $\geqslant 1$ si et seulement si le problème 3-SAT original est satisfiable

Problèmes 3-SAT et PLVB, de manière générale

Soit une instance 3-SAT, sur N booléens $v_1, \ldots, v_N$ et leur négation $\neg v_1, \ldots, \neg v_N$.
Soit f la fonction à évaluer sur un sous ensemble de M clauses de degré 3, $\mathcal{C} \subset \mathcal{A}^3$ où $\mathcal{A} = \{v_1, \ldots, v_N\} \cup \{\neg v_1, \ldots, \neg v_N\}$,

$$f(x) = \bigwedge_{(a,b,c)\in\mathcal{C}} (a \vee b \vee c)$$

On définit les variables binaires $x_1, \ldots, x_N$ avec $x_i = 1$ ssi on affecte au booléen v_i la valeur VRAI. Si $a = v_i$, on note $x_a = x_i$ et si $a = \neg v_i$, on note $x_a = 1 - x_i$.
3-SAT est équivalent à regarder le système de contraintes linéaires :

$$\forall (a,b,c) \in \mathcal{C}, \quad x_a + x_b + x_c \geqslant 1$$

En transformant une contrainte en objectif comme précédemment, on se ramène à un PLNE à N variables et $M-1$ contraintes

PLVB et 3-SAT

- Tout problème 3-SAT avec N booléens et M clauses conjonctives se résout par un PLVB avec N variables et $M-1$ contraintes.
- La transformation d'une instance 3-SAT en une telle instance PLVB est bien polynomiale en fonction de l'espace mémoire de l'instance initiale
- Si on sait résoudre de manière polynomiale tout PLVB, alors on saurait résoudre de manière polynomiale toute instance 3-SAT.
- Donc PLVB est NP-complet
- Donc PLNE, PQVB et PQNE aussi !

QUBO : Quadratic Unconstrained Binary Optimization

Un QUBO est un problème qui s'écrit de la forme :

$$\min_{x \in \{0,1\}^n} \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} x_i x_j \tag{15}$$

Existence : $x \in \{0,1\}^n$ définit un compact de $\mathbb{R}^n$, la fonction quadratique est continue (et même $\mathcal{C}^\infty$ car polynôme de $\mathbb{R}_n[X, Y]$ de degré 2)

2^n possibilités, on peut tout énumérer pour des petites valeurs de n pour prendre la meilleure solution

PLNE vs QUBO

Tout PLVB à coefficients entiers et à contraintes d'égalités (hypothèses peu restrictives) peut s'écrire en un QUBO.

$$
\begin{aligned}
& \min_{x \in \{0,1\}^n} \sum_{i=1}^{n} c_i x_i \\
s.c: \quad & \sum_{i=1}^{n} A_{i,c} x_i = b_c \quad \forall c \in [\![1; C]\!]
\end{aligned}
\tag{16}
$$

avec $c_i \geqslant 0$, $A_{i,c}, b_c \in \mathbb{Z}$ se transforme en QUBO, où M est un majorant du coût optimal :

$$
\min_{x \in \{0,1\}^n} \sum_{i=1}^{n} c_i x_i + M \sum_{c=1}^{C} \left(b_c - \sum_{i=1}^{n} A_{i,c} x_i \right)^2 \tag{17}
$$

Si le QUBO a une solution optimale de coût $\geqslant M$, c'est que le PLVB n'a pas de solutions réalisables. Si le le QUBO a une solution optimale de coût $< M$, cela définit une solution réalisable du PLVB et optimale.

$\Longrightarrow$ Les QUBO comprennent une classe plus générale que les PLVB.

Démonstration

Si une solution faisable existe dans le PLNE, elle définit une solution du QUBO de cout inférieur à M.

Une solution optimale du QUBO a nécessairement un coût $\geqslant M$ si une des contraintes n'est pas vérifiée, l'optimum se trouve parmi les solutions de coûts $\geqslant M$, qui vérifient les contraintes et de coût $\sum_{i=1}^n c_i x_i$. Le QUBO donne une solution optimale du PLNE

S'il n'existe pas de solution réalisable dans le PLNE, toute solution optimale du QUBO aura un coût $\geqslant M$, ce qui permet d'affirmer.

Remarque : pas d'unicité, on peut avoir plusieurs solutions optimales à un PLNE et un QUBO. On démontre en fait que toute solution optimale du PLNE est en bijection avec les solutions optimales du QUBO de coût $\geqslant M$.

Remarques sur les hypothèses du PLVB transformé en QUBO

- Comment trouver un majorant M ?
- Avoir que des variables binaires ? Extension aux PLNE à variables entières bornées, qui se transforment en PLVB
- Avoir des coefficients entiers ?
- Avoir des contraintes d'égalité ?

Comment trouver un majorant strict M ?

La fonction de coût du PLNE est à coefficients positifs, $M = 1 + \sum_{i=1}^{n} c_i$ convient

En pratique, un intérêt numérique à avoir M le plus petit possible pour certaines méthodes numériques.

Si on a une solution réalisable, évidente ou obtenue par une méthode heuristique, sont coût définit un tel M.

Avoir que des variables binaires ?

Ce sera le cas de bcp de PLNE vus dans ce cours.

Si on a des variables entières bornées, cela marche aussi ! C'est valable pour tout PLNE après transformation.

Soit z une variable à valeur dans $[\![1;Z]\!] = [1;Z] \cap \mathbb{Z}$

on transforme de manière équivalente z en Z variables binaires $x_i \in \{0,1\}$ qui indique si la valeur z prend la valeur i

On remplace z par sa valeur $z = \sum_i ix_i$, et on a un PLNE avec que des variables binaires

remarque pas besoin d'avoir un grain de 1, si on a un ensemble discret de valeurs, ça marche aussi (mais la modélisation PLNE aura écrit naturellement écrit directement les équations linéaires avec des variables binaires)

Avoir des coefficients entiers dans les contraintes ?

Ce sera le cas de bcp de PLNE vus dans ce cours.

Vision théorique : Si on a des nombres rationnels, on multiplie par le dénominateur commun dans chaque contrainte et on a des nombres entiers dans toutes les contraintes

Vision pratique : on travaille avec une précision numérique décimale définie,n uil suffit de multiplier par la bonne puissance de 10 suivant le nombre de chiffres significatifs

(remarque, cela marche aussi pour la fonction objectif, que l'on pourra supposer entière dans la version QUBO)

Avoir des contraintes d'égalités à coefficient entiers ?

Cette fois ci les techniques de transformation d'inégalités en inégalités utilisent l'ajout de variables continues, les variables de “slack”.

Après renormalisation pour avoir des coefficient entiers, la variable de slack est entière, on est ramené au cas précédent

une contrainte $\sum_i x_i \leqslant 1$, si elle n'est pas équivalente à $\sum_i x_i = 1$ peut se réécrire comme une linéarisation $\sum_i x_i = 0$ ou $\sum_i x_i = 1$, ici c'est juste introduite une nouvelle variable y et poser la contrainte $y + \sum_i x_i = 1$.

Remarque : QUBO et PLNE

ça ne paraissait pas intuitif que les QUBO contiennent tous les PLNE de nos applications.

QUBO peut être vue comme une technique de pénalisation de contraintes

En pratique, l'utilisation de formulation QUBO est plus adaptée des problèmes faiblement contraints, difficultés numériques pour les algorithmes de résolution de pb fortement contraints.

Les contraintes ou l'aspect quadratique apportent la richesse de modélisation.

Qu'est ce qu'un Problème linéaire en variables binaires uniquement linéaire et sans contrainte ?

Optimisation linéaire en variables binaire et sans contrainte

Tout simplement :

$$\max_{x \in \{0,1\}^N} \sum_{i=1}^{N} c_i^{lin} x_i = \sum_{i=1}^{N} \max(0, c_i^{lin}) \qquad (18)$$

Les contraintes de la PLNE ou l'aspect quadratique de QUBO apportent la richesse de modélisation et la difficulté de résolution.

Plan

Problème d'affectation : première situation

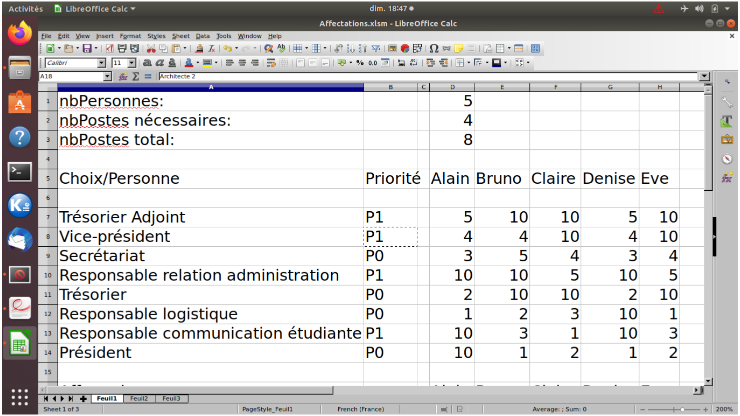
Cinq étudiants formant une association doivent définir leurs postes/responsabilités.

Certaines responsabilités sont indispensables statutairement ou en pratique, et doivent être pourvues : président, trésorier, secrétaire, responsable logistique

D'autres responsabilités sont envisagées, mais moins prioritaire, les postes ne peuvent pas tous être pourvus.

Chaque personne n classe ses cinq voeux préférés de 1 à 5. (on demande au moins trois voeux parmi les postes à pourvoir)

On souhaite calculer une affectation de postes qui minimise la somme des pondérations associées aux voeux, en affectant les postes nécessaires



nbPersonnes:		5					
nbPostes nécessaires:		4					
nbPostes total:		8					
Choix/Personne	Priorité	Alain	Bruno	Claire	Denise	Eve	
Trésorier Adjoint	P1	5	10	10	5	10	
Vice-président	P1	4	4	10	4	10	
Secrétariat	P0	3	5	4	3	4	
Responsable relation administration	P1	10	10	5	10	5	
Trésorier	P0	2	10	10	2	10	
Responsable logistique	P0	1	2	3	10	1	
Responsable communication étudiante	P1	10	3	1	10	3	
Président	P0	10	1	2	1	2	

Premier problème d'affectation, formalisation

On considère un problème d'affectation de poste, où N personnes doivent être affectées sur $P0$ postes prioritaires et $P1$ postes moins prioritaires, avec $P0 \leqslant N \leqslant P0 + P1$.

N.B : pour chaque personne n et chaque poste p, on a des coefficients $c_{n,p}$ qui indique le degré de satisfaction de n d'avoir le poste p, $c_{n,p}$ est d'autant plus bas que la satisfaction de n est grande.

On souhaite calculer une affectation de poste qui minimise la somme des pondérations associées aux voeux

N.B : On remarquera que la modélisation est générique, on calcule à une personne n et un poste m la pénalisation $c_{n,m}$ donnée pour la suite, qui pourra être tronquée à k premiers voeux et une pénalisation grande pour les autres voeux, ou des carrés telle une minimisation de moindre carrés.

Problème d'affectation, Formulation PLNE

On définit les variables avec $x_{n,p} \in \{0,1\}$ où $x_{n,p} = 1$ si et seulement si la personne n est affectée au poste p.

$$\begin{array}{lll} & \min \sum_{n,p} c_{n,p} x_{n,p} & \\ s.c: & & \\ \forall n & \sum_m x_{n,p} = 1 & (*) \\ \forall p \in \mathcal{P}_O & \sum_n x_{n,p} = 1 & (**) \\ \forall p \in \mathcal{P}_1 & \sum_n x_{n,p} \leqslant 1 & (***) \\ \forall n, m & x_{n,p} \in \{0,1\} & \end{array} \tag{19}$$

$(*)$ impose que chaque personne soit affectée à un poste. (on aurait pu avoir simplement $\geqslant 1$, c'est équivalent)

$(**)$ impose que chaque poste prioritaire est réalisée par une personne exactement.

$(***)$ impose que chaque poste est réalisé par une personne au plus.

Remarque sur la formulation PLNE

- On aurait préféré modéliser avec des variables $X_n \in \{1, M\}$, indiquant directement le poste de la personne n.
- Contrainte alors : $n \neq n' \Longrightarrow X_n \neq X_{n'}$, non linéaire
- La structure moins compacte $x_{n,m} \in \{0, 1\}$ est utilisée pour avoir des contraintes linéaires
- Une telle modélisation avec des variables $0 - 1$ est assez générique pour avoir des modèles PLNE.
- Souvent, la linéarisation revient à introduire de telles variables binaires (c'est le cas pour la fonction objectif)

Problème d'affectation, Formulation QUBO

On définit les variables avec $x_{n,m} \in \{0,1\}$ où $x_{n,m} = 1$ si et seulement si la personne n est affectée au poste m et y_m si poste m pourvu.

$$\begin{array}{lll} & \min \sum_{n,m} c_{n,m} x_{n,m} & \\ s.c: & & \\ \forall n & \sum_m x_{n,m} = 1 & (*) \\ \forall m & \sum_n x_{n,m} + y_m = 1 & (**) \\ \forall n,m & x_{n,m} \in \{0,1\} & \end{array} \tag{20}$$

Formulation QUBO :

$$\min_{x,y} \sum_{n,m} c_{n,m} x_{n,m} + S \sum_n \left(1 - \sum_m x_{n,m}\right)^2 + S \sum_m \left(1 - y_m - \sum_n x_{n,m}\right)^2 \tag{21}$$

Problème d'affectation : seconde situation

Le bureau des élèves organise un week-end d'intégration pour 200 personnes.

Plusieurs sorties facultatives avec quotas sont possibles :
- rester au centre de loisir, pas de quota
- sortie en plage, par 2 navettes de bus, quota 80 personnes
- visite culturelle , par navette bus, quota 40 personnes
- sortie accrobranche, par navette bus, quota 40 personnes
- sortie laserquest/paintball, par navette bus, quota 40 personnes

Chaque étudiant remplit ses voeux de 1 à 5

On souhaite calculer une affectation en activités qui minimise la somme des pondérations associées aux voeux.

Peut on utiliser le même modèle mathématique que précédemment ?

Problème d'affectation : seconde situation

On peut utiliser tel quel le modèle précédent en définissant 600 postes, 400 pour les restants (restant 1, restant 2,...), 80 plagistes (plagiste1, plagiste2, ...), ...

avec 200 personnes, soit une taille de matrice des coûts d'affectations $600 \times 200 = 120000$, avec 120000 variables et $800 = 600 + 200$ contraintes.

On peut formaliser autrement :
- soit $\mathcal{A}$ l'ensemble des activités. Pour tout $a \in \mathcal{A}$, on note Q_a le quota correspondant
- chaque étudiant $e \in \mathcal{E}$, fournit ses 5 choix classés, matrice $c_{a,e}$

$$\begin{array}{lll} & \min\limits_{x_{a,e} \in \{0,1\}} \sum\limits_{a,e} c_{a,e} x_{a,e} & \\ s.c : & & \\ \forall e & \sum_a x_{a,e} = 1 & (*) \\ \forall a & \sum_e x_{a,e} \leqslant Q_a & (**) \\ \forall a, e & x_{a,e} \in \{0, 1\} & \end{array} \tag{22}$$

Cela fait un PLNE avec 1000 variables et 205 contraintes.

Solution optimale de la première situation :

nbPersonnes:		5					
nbPostes nécessaires:		4					
nbPostes total:		8					
Choix/Personne	Priorité	Alain	Bruno	Claire	Denise	Eve	
Trésorier Adjoint	P1	5	10	10	5	10	
Vice-président	P1	4	4	10	4	10	
Responsable communication étudiante	P1	10	3	1	10	3	
Responsable relation administration	P1	10	10	5	10	5	
Secrétariat	P0	3	5	4	3	4	
Trésorier	P0	2	10	10	2	10	
Responsable logistique	P0	1	2	3	10	1	
Président	P0	10	1	2	1	2	
Affectation		Alain	Bruno	Claire	Denise	Eve	Poste occupé
Trésorier Adjoint	P1	0	0	0	0	0	0
Vice-président	P1	0	0	0	0	0	0
Responsable communication étudiante	P1	0	0	1	0	0	1
Responsable relation administration	P1	0	0	0	0	0	0
Secrétariat	P0	1	0	0	0	0	1
Trésorier	P0	0	0	0	1	0	1
Responsable logistique	P0	0	0	0	0	1	1
Président	P0	0	1	0	0	0	1
Objectif		8					
Préférence choix		3	1	1	2	1	
Somme var		1	1	1	1	1	

Tout va pour le mieux dans le meilleur des mondes ?

- La résolution est très facile en utilisant le solveur Excel/Open Office.
- On pourrait augmenter les tailles des problèmes, la résolution PLNE a de très bonnes propriétés sur ce problème (on l'expliquera, cas de totale unimodularité)
- La solution optimale fournie devrait contenter tout le monde

$\Longrightarrow$ Cela parait trop beau pour être vrai, où est le problème ?

Solution optimale avec d'autres valeurs numériques :

nbPersonnes:	5											
nbPostes nécessaires:	5											
nbPostes total:	5											
Choix/Personne	Alain	Bruno	Claire	Denise	Eve							
Trésorier	5	5	5	5	5							
Secrétariat	4	4	3	2	1							
Logistique	3	3	2	1	2							
Vice-président	2	2	1	3	4							
Président	1	1	5	4	3							
Affectation	Alain	Bruno	Claire	Denise	Eve	occup	Alain	Bruno	Claire	Denise	Eve	occup
Trésorier	1	0	0	0	0	1	0	1	0	0	0	1
Secrétariat	0	0	0	0	1	1	0	0	0	0	1	1
Logistique	0	0	0	1	0	1	0	0	0	1	0	1
Vice-président	0	0	1	0	0	1	0	0	1	0	0	1
Président	0	1	0	0	0	1	1	0	0	0	0	1
Objectif	9						9					
Préférence choix	5	1	1	1	1		1	5	1	1	1	
Somme var	1	1	1	1	1		1	1	1	1	1	

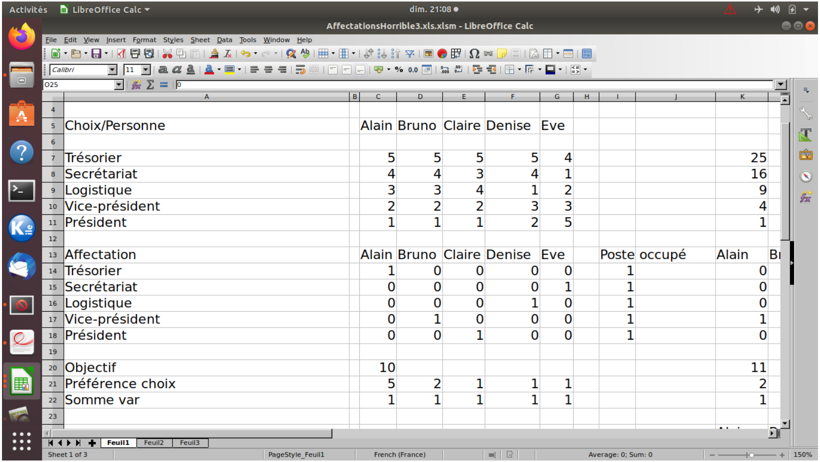
Feuil1 Feuil2 Feuil3

Analyse de la solution optimale précédente :

- 2 solutions optimales équivalentes, ou tous ont leur premier choix, sauf A ou B, qui aurait alors son pire choix.
- Solution difficile à faire accepter
- Pire, aucun moyen d'arbitrer entre A ou B aura le premier/dernier choix
- Pire, il n'y a rien de mieux à proposer sur cet exemple, personne ne veut être trésorier, le poste sera pourvu par qqn dont c'est le dernier choix. Si c'est A ou B, ça permet aux autres d'avoir leur premier choix.
- Faudrait il dégrader les choix de tout le monde, pour une équité où personne n'est satisfait ?

$\Longrightarrow$ situation intenable !

Autres valeurs numériques :

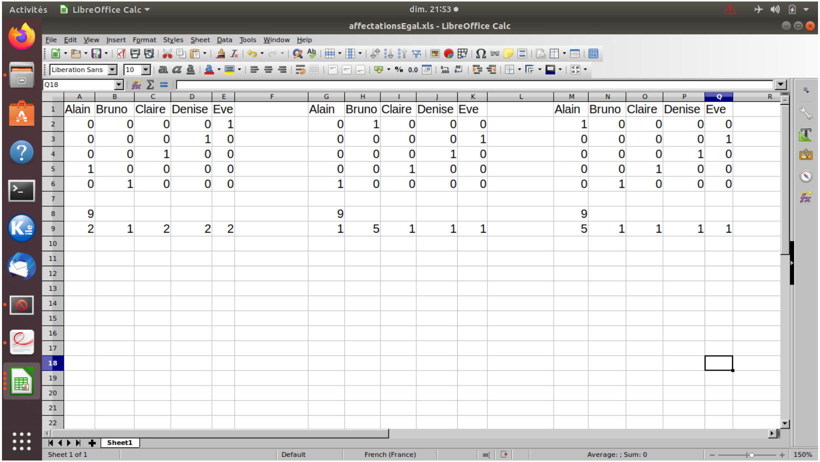


	A	C	D	E	F	G	H	I	J	K
4										
5	Choix/Personne	Alain	Bruno	Claire	Denise	Eve				
6										
7	Trésorier	5	5	5	5	4				25
8	Secrétariat	4	4	3	4	1				16
9	Logistique	3	3	4	1	2				9
10	Vice-président	2	2	2	3	3				4
11	Président	1	1	1	2	5				1
12										
13	Affectation	Alain	Bruno	Claire	Denise	Eve		Poste	occupé	Alain
14	Trésorier	1	0	0	0	0		1		0
15	Secrétariat	0	0	0	0	1		1		0
16	Logistique	0	0	0	1	0		1		0
17	Vice-président	0	1	0	0	0		1		1
18	Président	0	0	1	0	0		1		0
19										
20	Objectif	10								11
21	Préférence choix	5	2	1	1	1				2
22	Somme var	1	1	1	1	1				1
23										

$\Longrightarrow$ Même problème de solution déséquilibrée, et symétrie A - B

$\Longrightarrow$ E accepterait d'être trésorière en choix 4. Peut-on trouver d'autres solutions ?

Autres valeurs numériques :



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Alain	Bruno	Claire	Denise	Eve		Alain	Bruno	Claire	Denise	Eve		Alain	Bruno	Claire	Denise	Eve
2	0	0	0	0	1		0	1	0	0	0		1	0	0	0	0
3	0	0	0	1	0		0	0	0	0	1		0	0	0	0	1
4	0	0	1	0	0		0	0	0	1	0		0	0	0	1	0
5	1	0	0	0	0		0	0	1	0	0		0	0	1	0	0
6	0	1	0	0	0		1	0	0	0	0		0	1	0	0	0
7																	
8	9						9						9				
9	2	1	2	2	2		1	5	1	1	1		5	1	1	1	1

$\Longrightarrow$ 3 solutions optimales, très différentes, on préférerait la première intuitivement

$\Longrightarrow$ Comment le formaliser ?

Arbitrer parmi les solutions optimales

En pratique, on n'aura pas d'algorithme qui énumère toutes les solutions optimales d'un PLNE. (potentiellement un nombre exponentiel)

On peut calculer la valeur optimale (et une solution optimale associée)

On définit un indicateur de dispersion

On minimise la dispersion, sous contrainte des affectations et éventuellement que le coût est optimal (cela fait une contrainte linéaire en plus)

$\Longrightarrow$ Ce n'était pas la fonction objectif la plus adéquate qu'on a considéré

Indicateurs de dispersion

Soit C_n le choix validé pour la personne n.

On peut considérer $\max_n C_n - \min_n C_n$, la largeur de bande d'affectation. Dans le premier exemple, cette fonction équité donne la pire solution possible ...

$\max C_n$ est un indicateur d'équité, on peut cherche à minimiser la pire affectation d'un candidat. Dans le premier exemple, cela ne change rien

La fonction objectif minimise la moyenne des C_n. Et si on minimisait la variance ? (pas immédiat en PLNE)

Similairement, si on minimisait la somme des carrés ? Cela pénaliserait plus fortement les mauvais scores individuels, et cela éliminerait les solutions très déséquilibrées de l'exemple précédent. (analogue des moindre carrés)

Formulation PLNE minimisant les sommes des carrés des scores

On définit les variables avec $x_{n,p} \in \{0,1\}$ où $x_{n,p} = 1$ si et seulement si la personne n est affectée au poste p.

$$
\begin{array}{lll}
 & \min \sum_{n,p} c_{n,p}^2 x_{n,p} & \\
s.c: & & \\
\forall n & \sum_m x_{n,p} = 1 & (*) \\
\forall p \in \mathcal{P}_O & \sum_n x_{n,p} = 1 & (**) \\
\forall p \in \mathcal{P}_1 & \sum_n x_{n,p} \leqslant 1 & (***) \\
 & (\sum_{n,p} c_{n,p} x_{n,p} \leqslant OPT_1) & (****) \\
\forall n, m & x_{n,p} \in \{0,1\} &
\end{array}
\qquad (23)
$$

$\Longrightarrow$ On a un PLNE de même nature, cela reste linéaire comme les $c_{n,p}^2$ sont des paramètres et pas des variables de l'optimisation PLNE.

Autres valeurs numériques :

	A	C	D	E	F	G	K	L	M	N	O	Q
1	nbPersonnes:	5										
2	nbPostes nécessaires:	5										
3	nbPostes total:	5										
5	Choix/Personne	Alain	Bruno	Claire	Denise	Eve	Alain	Bruno	Claire	Denise	Eve	
7	Trésorier	5	5	5	5	4	25	25	25	25	16	
8	Secrétariat	4	4	3	4	1	16	16	9	16	1	
9	Logistique	3	3	4	1	2	9	9	16	1	4	
10	Vice-président	2	2	2	3	3	4	4	4	9	9	
11	Président	1	1	1	2	5	1	1	1	4	25	
13		Norme 1					Norme 2					
15	Affectation	Alain	Bruno	Claire	Denise	Eve	Alain	Bruno	Claire	Denise	Eve	Poste oc
16	Trésorier	1	0	0	0	0	0	0	0	0	1	
17	Secrétariat	0	0	0	0	1	0	0	1	0	0	
18	Logistique	0	0	0	1	0	0	0	0	1	0	
19	Vice-président	0	1	0	0	0	1	0	0	0	0	
20	Président	0	0	1	0	0	0	1	0	0	0	
22	Objectif Norme 1	10					11					
23	Préférence choix	5	2	1	1	1	2	1	3	1	4	
26	Objectif Norme 2	32					31					
27	Préférence choix											

$\Longrightarrow$ L'optimum en minimisant la somme des carrés donne une solution naturellement moins dispersée

Formulation PLNE minimisant le score max individuel

On définit les variables avec $x_{n,p} \in \{0,1\}$ où $x_{n,p} = 1$ si et seulement si la personne n est affectée au poste p.

$$
\begin{array}{lll}
 & \min \max_n \sum_p c_{n,p} x_{n,p} & \\
s.c: & & \\
\forall n & \sum_m x_{n,p} = 1 & (*) \\
\forall p \in \mathcal{P}_O & \sum_n x_{n,p} = 1 & (**) \\
\forall p \in \mathcal{P}_1 & \sum_n x_{n,p} \leqslant 1 & (***) \\
 & (\sum_{n,p} c_{n,p} x_{n,p} \leqslant OPT) & (***) \\
\forall n, m & x_{n,p} \in \{0,1\} &
\end{array}
\tag{24}
$$

$$
\begin{array}{lll}
 & \min_{C \geqslant 0} C & \\
s.c: & & \\
\forall n & \sum_p c_{n,p} x_{n,p} \leqslant C & \\
\forall n & \sum_m x_{n,p} = 1 & (*) \\
\forall p \in \mathcal{P}_O & \sum_n x_{n,p} = 1 & (**) \\
\forall p \in \mathcal{P}_1 & \sum_n x_{n,p} \leqslant 1 & (***) \\
 & (\sum_{n,p} c_{n,p} x_{n,p} \leqslant OPT) & (***) \\
\forall n, m & x_{n,p} \in \{0,1\} &
\end{array}
\tag{25}
$$

Optimisation hiérarchiques

- 3 fonctions objectifs possibles en PLNE
- On peut calculer les trois valeurs optimales
- Pour une valeur optimale, on peut considérer le PLNE en prenant un autre critère, en ajoutant la contrainte d'avoir une solution optimale sur un autre critère.
- Cela peut faire 6 solutions différentes, les opérateurs ne sont pas commutatifs (dans le cas 1, on aurait toujours la même solution)
- A priori il peut être pertinent de minimiser le score max individuel, et d'opérer une autre minimisation sur les affectations restantes
- Minimiser les moindres carrés est une approche pertinente aussi

$\Longrightarrow$ On peut générer plusieurs solutions suivant une logique rationnelle acceptable, après c'est du cas par cas pour voir si une solution convient à l'unanimité

Faiblesses inhérentes et incontournables

- Exemple 1 : on aura toujours une situation inextricable et changer de fonction objectif ne change rien

- Question d'équité : avec des symétries, choix identiques, et des Affectations finales différentes, les solutions sont équivalentes. Cela revient à tirer au sort qui perd.

- Phénomènes d'anticipation : si vous ne voulez pas vos pires choix, vous prenez en 1er choix qqch de moins demandé, en choix suivant des choix très demandés qui vous intéressent, puis des choix très demandés qui ne vous intéressent pas, et ceux que vous voulez éviter enfin, vous aurez de grandes chances d'avoir votre 1er choix.

- C'était déjà compliqué avec quelques personnes qui peuvent se mettre d'accord, mais alors de manière décentralisée avec bcp de participants ? (ex : ParcourSup)

$\Longrightarrow$ Le travail ne se limite pas à élaborer un modèle et le résoudre à l'optimalité, il faut vérifier que le modèle répond à l'application au monde réel

$\Longrightarrow$ Questions d'éthique à se poser quand on élabore ou applique un modèle d'optimisation

Bilan

Cet exemple était très riche en enseignements pour la PLNE :

- Formulation PLNE moins compacte que formulation naturelle X_n indiquant directement le poste, on utilise des variables binaires pour avoir des contraintes linéaires.
- Linéarisation min-max
- Optimisation hiérarchique, intérêt d'avoir plusieurs fonctions objectifs (pour approfondir, cours Master 2, optimisation multi-objectifs)
- La PLNE est assez souple pour changer de fonction objectif, cela pose des questions d'éthique.

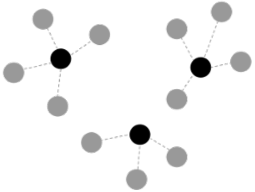
$\Longrightarrow$ Le travail ne se limite pas à élaborer un modèle et le résoudre à l'optimalité, il faut vérifier que le modèle répond à l'application au monde réel

$\Longrightarrow$ Questions d'éthique à se poser quand on élabore ou applique un modèle d'optimisation avec des implications humaines.

Plan

Problèmes de clustering discret

Données d'entrée : un ensemble X de N points dans $\mathbb{R}^d$, un entier $p < N$



p-median : On veut former une partition de X en p sous-ensembles (= cluster), choisir p éléments représentants de chacun des clusters qui minimise la somme des distances de chacun des N points au représentant du cluster dont il appartient.

k-medoids : idem que k-median, sauf que l'on minimise la somme des carrés des distances. (Variante discrète de k-means, les k-moyennes)

p-centre discret : on cherche à couvrir les points par p boules identiques de rayon r, ayant pour centre un des N points, tout en minimisant le rayon r des boules utilisées.

Min-Sum-k-radii discret : on cherche à couvrir les points par k boules (pas forcément identiques) de rayon r_i, en minimisant $\sum r_i$.

Extensions, contraintes de cardinalité : imposer un cardinal maximal pour chaque cluster, autoriser à ne pas couvrir quelques points (outliers ?)

Problèmes de clustering k-means/k-medoids

Soit $E = \{x_1, \dots, x_N\}$ un ensemble de N éléments of $\mathbb{R}^d$. On définit $\Pi_K(E)$, l'ensemble des partitions de E en K sous-ensembles :

$$\Pi_K(E) = \left\{ P \subset \mathcal{P}(E) \;\middle|\; \forall p, p' \in P, \;\; p \cap p' = \emptyset \text{ and } \bigcup_{p \in P} = E \text{ and } \mathrm{card}(P) = K \right\}$$

En définissant une fonction de coût f mesurant la dissimilarité de chaque sous ensemble de E, on définit des problèmes de clustering comme des problèmes d'optimisation indexés par $\Pi_K(E)$:

$$\min_{\pi \in \Pi_K(E)} \sum_{P \in \pi} f(P) \tag{26}$$

K-medoids : $f_{medoids}(P) = \min_{y \in P} \sum_{x \in P} \|x - y\|^2$

K-means :

$f_{means}(P) = \min_{y \in \mathbb{R}^d} \sum_{x \in P} \|x - y\|^2 = \sum_{x \in P} \left\| x - \frac{1}{\mathrm{card}(P)} \sum_{y \in P} y \right\|^2$

K-medoids est la version discrète de k-means : le centre d'un cluster, le médoïde, est un des points de E.

N.B : On utilise ici une distance issue d'une norme et on note $d(x_i, x_j) = ||x_i - x_j||$. On peut considérer une distance quelconque dans la suite

Extensions et variantes

On peut généraliser ou considérer d'autre variantes de fonctions f

K-med, α : $f_{med}^{(\alpha)}(P) = \min_{y \in P} \sum_{x \in P} \|x - y\|^{\alpha}$

K-medoids est le cas $\alpha = 2$, K-median (discret) est le cas $\alpha = 1$. On a aussi de telles généralisations avec k-means, mais le calcul du centroïde n'est plus analytique.

Boule englobante de centre discret : $f_{ctr}^{\mathcal{D}}(P, \alpha) = \min_{y \in P} \max_{x \in P} \|x - y\|^{\alpha}$

Boule englobante avec centre quelconque : $f_{ctr}^{\mathcal{C}}(P) = \min_{y \in \mathbb{R}^2} \max_{x \in P} \|x - y\|$

Avec le centre quelconque, et $\alpha = 1$, c'est le problème "min-sum-K-radii", équivalent à "min-sum-diameter".

On peut même également considérer les problèmes de clustering min-max suivants, au lieu de min-sum :

$$\min_{\pi \in \Pi_K(E)} \max_{P \in \pi} f(P) \tag{27}$$

Application : les problèmes de K-centre (discrets et continus), sont de tels problèmes min-max, avec un calcul de boule englobante pour la fonction f.

Programmation mathématique et p-median

Variables : $z_{n,p} \in \{0,1\}$: si le point x_n est affecté au cluster p.

$y_{n,p} \in \{0,1\}$: si le point n est choisi comme le représentant du cluster p.

$$
\begin{array}{ll}
 & \min \sum_{n,n',p} d(x_n, x_{n'}) z_{n,p} y_{n',p} \\
s.c : & \\
\forall p \in [\![1, P]\!], & \sum_n y_{n,p} \leqslant 1 \\
\forall n, p, & y_{n,p} \leqslant z_{n,p} \\
\forall n, & \sum_p z_{n,p} = 1 \\
\forall n, p, & z_{n,p}, y_{n,p} \in \{0,1\}
\end{array}
\tag{28}
$$

$\Longrightarrow$ formulation non linéaire, produit de variables $z_{n,p} y_{n',p}$

$\Longrightarrow$ formulation quadratique, facile à linéariser.

Une première linéarisation du p-median

Variables : $z_{n,n',p} \in \{0,1\}$: si le point x_n est affecté au cluster p où le représentant choisi est $x_{n'}$.

$y_{n,p} \in \{0,1\}$: si le point x_n est choisi comme le représentant du cluster p.

$$
\begin{array}{ll}
 & \min \sum_{n,n',p} d(x_n, x_{n'}) z_{n,n',p} \\
s.c: & \\
\forall p, \in [\![1, P]\!] & \sum_n y_{n,p} \leqslant 1 \\
\forall n, n', p, & z_{n,n',p} \leqslant y_{n',p} \\
\forall n, & \sum_{n',p} z_{n,n',p} = 1 \text{ (ou } \geqslant 1) \\
\forall n, p & z_{n,n',p} \in \{0,1\}
\end{array}
\tag{29}
$$

Une linéarisation plus économe, "par représentants"

Variables : $z_{n,n'} \in \{0,1\}$: vaut 1 si les point $x_n, x_{n'}$ sont affectés dans un même cluster, de représentant $x_{n'}$.

$z_{n,n} = 1$ indique que x_n est représentant de son cluster.

$$\begin{array}{lll} & \min \displaystyle\sum_{n,n',p} d(x_n, x_{n'}) z_{n,n'} & \\ s.c : & & \\ \forall n' & \displaystyle\sum_n z_{n',n} = 1 & (\text{ou} \geqslant 1) \\ \forall n, n', & z_{n,n'} \leqslant z_{n',n'} & \\ & \displaystyle\sum_n z_{n,n} = p & (\text{ou} \leqslant p) \\ \forall n, p & z_{n,n'} \in \{0,1\} & \end{array} \tag{30}$$

Variante k-medoids

En remplaçant $d(x_n, x_{n'})$ par $d(x_n, x_{n'})^2$, on a des formulations PLNE de K-medoids !

Cela ne change pas la linéarité, comme les $d(x_n, x_{n'})$ sont des paramètres d'entrée, et pas des variables !

Pour K-med, α, idem, il suffit de remplacer $d(x_n, x_{n'})$ par $d(x_n, x_{n'})^\alpha$ et on a des formulations PLNE de K-med, α

Variante : problème de p-centre

Variables binaires : $z_{n,n'} \in \{0,1\}$: vaut 1 si le point x_n est affectés au cluster de centre $x_{n'}$.
Variable continue : $r \geqslant 0$, le rayon des boules

$$
\begin{aligned}
\min_{r \geqslant 0} \quad & r & & & (31.1)\\
s.t: \quad d(x_n, x_{n'}) z_{n,n'} & \leqslant r & & \forall (n,n') \in [\![1,N]\!]^2 & (31.2)\\
\sum_{n'=1}^{N} z_{n',n'} & = p & & & (31.3)\\
\sum_{n'=1}^{N} z_{n,n'} & = 1 & & \forall n \in [\![1,N]\!] & (31.4)\\
z_{n,n'} & \leqslant z_{n',n'} & & \forall (n,n') \in [\![1,N]\!]^2, & (31.5)\\
z_{n,n'} & \in \{0,1\} & & \forall (n,n') \in [\![1,N]\!]^2, & (31.6)
\end{aligned}
\qquad (31)
$$

Variante : Min-Sum-k-radii

Variables binaires : $z_{n,n'} \in \{0,1\}$: vaut 1 si le point x_n est affectés au cluster de centre $x_{n'}$.

Variable continue : $r_n \geqslant 0$, le rayon des boules de centre n, 0 si le point est non sélectionné.

$$
\begin{aligned}
\min_{z,r\geqslant 0} \quad & \sum_{n=1}^{N} r_n && && (32.1)\\
s.t: \quad \sum_{n=1}^{N} d(x_n, x_{n'}) z_{n,n'} & \leqslant r_{n'} && \forall n' \in [\![1,N]\!] && (32.2)\\
\sum_{n'=1}^{N} z_{n',n'} & = k && && (32.3)\\
\sum_{n'=1}^{N} z_{n,n'} & = 1 && \forall n \in [\![1,N]\!] && (32.4)\\
z_{n,n'} & \leqslant z_{n',n'} && \forall (n,n') \in [\![1,N]\!]^2, && (32.5)\\
z_{n,n'} & \in \{0,1\} && \forall (n,n') \in [\![1,N]\!]^2, && (32.6)
\end{aligned}
\qquad (32)
$$

Variante α de Min-Sum-k-radii (1)

Si on considère $f_{ctr}^{\mathcal{D}}(P,\alpha) = \min_{y \in P} \max_{x \in P} \|x - y\|^{\alpha}$, avec $\alpha \neq 1$, on pourrait avoir envie d'écrire la formulation non-linéaire :

$$
\begin{array}{llllr}
\min\limits_{z,r \geqslant 0} & \displaystyle\sum_{n=1}^{N} r_n^{\alpha} & & & (33.1) \\
s.t: & \displaystyle\sum_{n=1}^{N} d(x_n, x_{n'}) z_{n,n'} & \leqslant r_{n'} & \forall n' \in [\![1, N]\!] & (33.2) \\
 & \displaystyle\sum_{n'=1}^{N} z_{n',n'} & = k & & (33.3) \\
 & \displaystyle\sum_{n'=1}^{N} z_{n,n'} & = 1 & \forall n \in [\![1, N]\!] & (33.4) \\
 & z_{n,n'} & \leqslant z_{n',n'} & \forall (n,n') \in [\![1, N]\!]^2, & (33.5) \\
 & z_{n,n'} & \in \{0,1\} & \forall (n,n') \in [\![1, N]\!]^2, & (33.6)
\end{array}
\quad (33)
$$

$\Longrightarrow$ Peut-on avoir un modèle linéaire ?

Variante α de Min-Sum-k-radii (2)

Variables binaires : $z_{n,n'} \in \{0,1\}$: vaut 1 si le point x_n est affectés au cluster de centre $x_{n'}$.

Variable continue : $r_n \geqslant 0$, le rayon des boules de centre n, 0 si le point est non sélectionné.

$$
\begin{aligned}
& \min_{z,r\geqslant 0} && \sum_{n=1}^{N} r_n && && (34.1) \\
& s.t: && \sum_{n=1}^{N} d(x_n, x_{n'})^{\alpha} z_{n,n'} && \leqslant r_{n'} && \forall n' \in [\![1,N]\!] && (34.2) \\
& && \sum_{n'=1}^{N} z_{n',n'} && = k && && (34.3) \\
& && \sum_{n'=1}^{N} z_{n,n'} && = 1 && \forall n \in [\![1,N]\!] && (34.4) \\
& && z_{n,n'} && \leqslant z_{n',n'} && \forall (n,n') \in [\![1,N]\!]^2, && (34.5) \\
& && z_{n,n'} && \in \{0,1\} && \forall (n,n') \in [\![1,N]\!]^2, && (34.6)
\end{aligned}
\qquad (34)
$$

Extensions de contraintes de cardinalité

On a les mêmes variables binaires $z_{n,n'} \in \{0,1\}$ dans tous les cas.

contraintes de cardinalité 1 : imposer un cardinal maximal M pour chaque cluster :

$$\sum_{n'=1}^{N} z_{n',n'} \leqslant M \tag{35}$$

contraintes de cardinalité 2 : autoriser à ne pas couvrir quelques points (outliers ?) M', par exemple $M' = 0.01 \times N$ (maximum 1% d'outliers) :

On remplace les contraintes $\displaystyle\sum_{n'=1}^{N} z_{n,n'} = 1$ pour tout n par :

$$\sum_{n=1}^{N} \sum_{n'=1}^{N} z_{n,n'} \geqslant N - M' \tag{36}$$

$\Longrightarrow$ et on peut le faire pour k-means ?

Programmation mathématique et K-means (1)

Variables : $z_{n,k} \in \{0,1\}$: si le point x_n est affecté au cluster $k \in [\![1,K]\!]$.

Le cardinal du cluster k est donné par : $\displaystyle\sum_{n=1}^{N} z_{n,k}$

Le centre continu du cluster k est donné par : $\displaystyle\frac{1}{\sum_n z_{n,k}} \sum_n z_{n,k} x_n$

$$
\begin{array}{lll}
& \displaystyle\min_{z_{n,k}} \sum_{k=1}^{K} \sum_{n=1}^{N} z_{n,k} \left\| x_n - \frac{z_{n,k}}{\sum_{n'=1}^{N} z_{n',k}} x_n \right\|^2 & \\
s.c: & & \\
\forall n, & \displaystyle\sum_{k=1}^{K} z_{n,k} = 1 & \\
\forall n,k, & z_{n,k} \in \{0,1\} &
\end{array}
\tag{37}
$$

$\Longrightarrow$ formulation non linéaire, on ne pourra pas la transformer en PLNE

$\Longrightarrow$ Il y a bien des exemples de problèmes qui ne s'écrivent pas en PLNE !

Programmation mathématique et K-means (2)

Variables binaires : $z_{n,k} \in \{0,1\}$: si le point x_n est affecté au cluster k.

Variables continues : $c_k \in \mathbb{R}^d$: le centroïde du cluster k, qui sera défini à l'optimum par $\min_{y \in \mathbb{R}^d} \sum_{x \in P} \|x - y\|^2 = \sum_{x \in P} \left\| x - \frac{1}{\text{card}(P)} \sum_{y \in P} y \right\|^2$

$$
\begin{aligned}
& \min_{z_{n,k}, c_k} \sum_{k=1}^{K} \sum_{n=1}^{N} z_{n,k} \|x_n - c_k\|^2 \\
s.c: & \\
\forall n, \quad & \sum_{k=1}^{K} z_{n,k} = 1
\end{aligned}
\tag{38}
$$

$\Longrightarrow$ formulation plus agréable, mais on ne pourra pas la transformer en PLNE, à cause de produits de variables continues $\times$ continues.

$\Longrightarrow$ Cette formulation se résout avec des solveurs d'optimisation non linéaires gérant des non convexités (BARON, GloMiQo), mais pas avec les extensions non linéaires des meilleurs solveurs de PLNE (CPLEX ou GUROBI)

Programmation mathématique et K-means (3)

Variables binaires : $z_{n,k} \in \{0,1\}$: si le point x_n est affecté au cluster k.

Variables continues :
- $c_k \in \mathbb{R}^d$: le centroïde du cluster k.
- $d_{n,k} \geqslant 0$ le carré de la distance du point x_n à son centre de cluster c_k (si $z_{n,k} = 1$) et $d_{n,k} = 0$ si $z_{n,k} = 0$)

$$\begin{aligned} & \min_{z_{n,k}, d_{n,k}, c_k} \sum_{k=1}^{K} \sum_{n=1}^{N} d_{n,k} \\ & s.c : \\ & \forall n, k, \quad d_{n,k} \geqslant \left\| x_n - c_k \right\|^2 - M_{n,k}(1 - z_{n,k}) \\ & \forall n, \quad \sum_{k=1}^{K} z_{n,k} = 1 \end{aligned} \tag{39}$$

Le "big M" suffisamment grand $M_{n,k}$ peut être $M_{n,k} = \max_{n' \in [\![1,N]\!]} \left\| x_n - x_{n'} \right\|^2$

$\Longrightarrow$ Cette formulation quadratique (convexe) se résout aussi avec les extensions non linéaires des meilleurs solveurs de PLNE (CPLEX ou GUROBI)

Plan

TODO : pour prochaine version :

ordonnancement PERT, modélisation PL et algo glouton

équilibrage de charge sur machines parallèles

RCPSP

min-up/min-down polytope

lot-sizing

Plan

Unit Commitment Problem, UCP

UCP : Planifier la production d'électricité pour les différentes unités $u \in \mathcal{U}$ de production à tout instant $t \in \mathcal{T}$, en satisfaisant les demandes en électricité et en minimisant les coûts de production.

Fonction de coût : coûts de productions proportionnel à la production C^u_{prop} et coûts fixes de fonctionnement C^u_{fix} pour tout $u \in \mathcal{U}$.

Contrainte de demande : à tout instant $t \in \mathcal{T}$, la demande en puissance D_t (connue) doit être égale à la production totale à t sur toutes les unités $u \in \mathcal{U}$.

Contrainte de production : pour toute unité $u \in \mathcal{U}$, la production est soit nulle si l'unité est hors fonctionnement, soit entre $Pmin^u_t$ et $Pmax^u_t$ à l'instant t.

$\Longrightarrow$ modèle simple de production d'électricité, à raffiner suivant cas d'étude

Formulation PLNE de l'UCP

Variables de décision :

- $P_t^u \geqslant 0$: puissance générée par l'unité u à la période t.
- $x_t^u \in \{0,1\}$: variables de "set-up", $x_t^u = 1$ si u en fonctionnement à t.

$$\min \sum_{u \in \mathcal{U}} \sum_{t \in T} C_{prop}^{u} P_t^u + \sum_{u \in \mathcal{U}} \sum_{t \in T} C_{fix}^{u} x_t^u$$

s.t :

$$\forall t \in [\![1, T]\!], \quad \sum_{u \in \mathcal{U}} P_t^u = D_t \tag{40}$$

$$\forall u \in \mathcal{U}, \quad \forall t \in [\![0, T]\!], \quad x_t^u.Pmin_t^u \leqslant P_t^u \tag{41}$$

$$\forall u \in \mathcal{U}, \quad \forall t \in [\![0, T]\!], \quad P_t^u \leqslant x_t^u.Pmax_t^u \tag{42}$$

exemple de linéarisation binaire $\times$ continu, ne pas écrire $P_t^u \times x_t^u$

Contrainte de temps minimaux de fonctionnement et d'arrêt

En pratique, il y a une inertie à démarrer ou arrêter des centrales électriques.

Contrainte de temps minimaux de fonctionnement : toute unité $u \in \mathcal{U}$ admet un temps minimal en fonctionnement L_u,

Contrainte de temps minimaux d'arrêt : toute unité $u \in \mathcal{U}$ admet un temps minimal hors fonctionnement lors d'un arrêt l_u.

En pratique, les démarrages à chaud et à froid doivent être distingués ...

Comment peut on formuler ces contraintes en PLNE ?

Temps minimal de fonctionnement et d'arrêt

Les contraintes de temps minimal de fonctionnement peuvent s'écrire :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \;\; x_t^u = 1 \text{ ET } x_{t-1}^u = 0 \Longrightarrow \forall \tau \in [\![t+1; t+L^u-1]\!], x_\tau^u = 1 \quad (43)$$

Linéarisation :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall \tau \in [\![t+1; t+L^u-1]\!], \;\; x_t^u - x_{t-1}^u \leqslant x_\tau^u \quad (44)$$

De même pour le temps minimal en arrêt :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \;\; x_t^u = 0 \text{ ET } x_{t-1}^u = 1 \Longrightarrow \forall \tau \in [\![t+1; t+l^u-1]\!], x_\tau^u = 0 \quad (45)$$

Linéarisation :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \forall \tau \in [\![t+1; t+l^u-1]\!], \;\; x_{t-1}^u - x_t^u \leqslant 1 - x_\tau^u \quad (46)$$

Ajout de coûts de démarrage

A chaque démarrage, un coût de démarrage C^u_{start} doit être pris en compte.

On introduit des variables de démarrage ("start-up") $y^u_t \in \{0,1\}$.

La nouvelle fonction de coût s'écrit alors :

$$\min \sum_{u \in \mathcal{U}} \sum_{t \in T} {C_{prop}}^u P^u_t \ + \sum_{u \in \mathcal{U}} \sum_{t \in T} {C_{fix}}^u x^u_t \ + \sum_{u \in \mathcal{U}} \sum_{t \in T} {C_{start}}^u y^u_t \tag{47}$$

On a en fait $y^u_t = \max(0, x^u_t - x^u_{t-1}) \in \{0,1\}$, l'ajout des variables y^u_t est nécessaire pour avoir une formulation linéaire. Avec $y^u_t \geqslant 0$, il n'y a qu'à ajouter les contraintes :

$$\forall u \in \mathcal{U}, \ \forall t \in [\![1, T]\!], \ x^u_t - x^u_{t-1} \leqslant y^u_t \tag{48}$$

Temps min d'arrêt et de fonctionnement avec variables de démarrage

- Takriti et al (2005) : les contraintes de temps min d'arrêt et de fonctionnement peuvent être écrites avec les variables de set-up et de start-up x_t^u, y_t^u :

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-L^u+1}^{t} y_{t'}^u \leqslant x_t^u \tag{49}$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-l^u+1}^{t} y_{t'}^u \leqslant x_{t-l^u}^u \tag{50}$$

- Résultats théoriques et pratiques de formulations : il est plus efficace de résoudre le problème avec (49),(50) plutôt que (44),(46).
- Illustre que de multiples formulations PLNE peuvent être écrites pour un pb donné, avec une incidence sur l'efficacité pratique.

Formulation mixte en nombres entiers

$$\min \sum_{u \in \mathcal{U}} \sum_{t \in T} C_{prop}^{u} P_t^u \;+ \sum_{u \in \mathcal{U}} \sum_{t \in T} C_{fix}^{u} x_t^u \;+ \sum_{u \in \mathcal{U}} \sum_{t \in T} C_{start}^{u} y_t^u$$

s.t :

$$\forall u \in \mathcal{U}, \;\; \forall t \in [\![1, T]\!], \;\; x_t^u - x_{t-1}^u \leqslant y_t^u \tag{51}$$

$$\forall t \in [\![1, T]\!], \quad \sum_{u \in \mathcal{U}} P_t^u = D_t \tag{52}$$

$$\forall u \in \mathcal{U}, \;\; \forall t \in [\![0, T]\!], \;\; x_t^u.Pmin_t^u \leqslant P_t^u \tag{53}$$

$$\forall u \in \mathcal{U}, \;\; \forall t \in [\![0, T]\!], \;\; P_t^u \leqslant x_t^u.Pmax_t^u \tag{54}$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-L^u+1}^{t} y_{t'}^u \leqslant x_t^u \tag{55}$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-l^u+1}^{t} y_{t'}^u \leqslant x_{t-l^u}^u \tag{56}$$

Ajout de coûts de d'arrêt

On considère à présent des coûts pour chaque arrêt C^u_{end}.

On introduit des variables d'arrêts ("start-up") $z^u_t \in \{0,1\}$, $z^u_t = 1$ ssi u est éteinte à l'instant t, ie $x^u_t = 0$ et $x^u_{t-1} = 1$

On a $z_{u,t} = \max(0, x^u_{t-1} - x^u_t)$. En fait, on a la relation linéaire suivante

$$\forall u \in \mathcal{U},\ \ \forall t \in [\![1, T]\!],\ \ y^u_t = z^u_t + x^u_t - x^u_{t-1} \tag{57}$$

Preuve : on énumère les quatre cas possibles sur les valeurs de x^u_t, x^u_{t-1} on calcule les valeurs induites pour y^u_t, z^u_t, et on remarque qu'on a bien l'égalité

N.B : variables x, y, z aussi utilisées dans problèmes de type "lot-sizing", ordonnancement de production et stockage autorisé

Formulation PLNE

$$\min_{x,y,P\geqslant 0} \sum_{u\in\mathcal{U}} \sum_{t\in T} C_{prop}^{u} P_t^u \;+ C_{fix}^{u} x_t^u \;+\; C_{start}^{u} y_t^u + C_{end}^{u} (y_{u,t} - x_{u,t} + x_{u,t-1})$$

s.t :

$$\forall u \in \mathcal{U},\;\; \forall t \in [\![1, T]\!],\;\; x_t^u - x_{t-1}^u \leqslant y_t^u$$

$$\forall t \in [\![1, T]\!],\;\; \sum_{u\in\mathcal{U}} P_t^u = D_t$$

$$\forall u \in \mathcal{U},\;\; \forall t \in [\![0, T]\!],\;\; x_t^u.Pmin_t^u \leqslant P_t^u$$

$$\forall u \in \mathcal{U},\;\; \forall t \in [\![0, T]\!],\;\; P_t^u \leqslant x_t^u.Pmax_t^u$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T},\quad \sum_{t'=t-L^u+1}^{t} y_{t'}^u \leqslant x_t^u$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T},\quad \sum_{t'=t-l^u+1}^{t} y_{t'}^u \leqslant x_{t-l^u}^u$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T},\;\; x_t^u, y_t^u \in \{0,1\}$$

Contraintes spécifiques

De multiple cas d'études se basent sur la structure UCP avec des contraintes additionnelles suivant les contraintes de fonctionnements, et les pas de temps de discrétisation :

- Fonction objectif suivant d'autres modèles (non linéaires)
- Gradients de puissance bornés
- Hydraulique : fonctionnement couplés le long d'un fleuve, contraintes de débit et délais dus au débits
- Unités nucléaire : temps maximal à fonctionnement à puissance non maximale (modulation de puissance)
- Participations aux services systèmes
- Maintenances imposant un arrêt de la production. (les exos 6-7 du TD4 donnent des idées sur de telles contraintes)

Prise en compte d'aléas dans un modèle UCP

En pratique, on peut considérer des aléas dans l'"optimisation, pour ne pas baser l'optimisation sur un scénario nominal avantageux et avoir une mauvaise décision en pratique après réalisation de certains aléas.

Pire : le planning peut ne plus être réalisable

Aléas sur les demandes et les niveaux de consommation (ajustés aux différentes échelles de l'optimisation énergétique)

Aléas sur les capacités de production, ex : pannes.

Les décisions de mise en fonctionnement/arrêt sont impactantes et ne peuvent pas être réorganisées facilement avec inertie des temps minimums d'arrêt et de fonctionnement.

On cherche à avoir de la manoeuvrabilité, pour pouvoir réagir à des aléas

UCP et programmation stochastique à 2 niveaux

Modèle : plusieurs scénarios discrets en nombre S sont élaborés pour être représentatifs de la réalité, chaque scénario s a une probabilité π_s d'apparition.

Les décisions de set-up et de start-up x, y doivent être communes à tous les scénarios

Les puissances peuvent être recalculées/adaptées pour chaque scénario, on a des variables de puissance pour chaque scénario.

Contrainte de demande : à tout instant $t \in \mathcal{T}$, la demande en puissance $D_{t,s}$ doit être égale à la production totale à t sur toutes les unités $u \in \mathcal{U}$ pour le scénario s.

Contrainte de production : pour toute unité $u \in \mathcal{U}$, la production est soit nulle si l'unité est hors fonctionnement, soit entre $Pmin^u_{t,s}$ et $Pmax^u_{t,s}$ à l'instant t pour le scénario s.

On minimise l'espérance du coût de production, et étant réalisable sur tous les scénarios

Formulation PLNE

$$\min_{x,y,P\geqslant 0} \sum_{u\in\mathcal{U}}\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}} \pi_s C_{prop}^{u} P_{t,s}^{u} \;+ \sum_{u\in\mathcal{U}}\sum_{t\in T} C_{fix}^{u} x_t^u \;+ \sum_{u\in\mathcal{U}}\sum_{t\in T} C_{start}^{u} y_t^u$$

s.t :

$$\forall u \in \mathcal{U},\ \forall t \in [\![1, T]\!],\ x_t^u - x_{t-1}^u \leqslant y_t^u \qquad (58)$$

$$\forall t \in [\![1, T]\!],\ \forall s \in \mathcal{S},\ \sum_{u\in\mathcal{U}} P_{t,s}^u = D_{t,s} \qquad (59)$$

$$\forall u \in \mathcal{U},\ \forall s \in \mathcal{S},\ \forall t \in [\![0, T]\!],\ x_t^u.Pmin_{t,s}^u \leqslant P_{t,s}^u \qquad (60)$$

$$\forall u \in \mathcal{U},\ \forall s \in \mathcal{S},\ \forall t \in [\![0, T]\!],\ P_{t,s}^u \leqslant x_t^u.Pmax_{t,s}^u \qquad (61)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T},\quad \sum_{t'=t-L^u+1}^{t} y_{t'}^u \leqslant x_t^u \qquad (62)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T},\quad \sum_{t'=t-l^u+1}^{t} y_{t'}^u \leqslant x_{t-l^u}^u \qquad (63)$$

UCP et programmation robuste à 2 niveaux

Modèle : plusieurs scénarios discrets en nombre S sont élaborés pour être représentatifs de la réalité.

On minimise le coût de production, en étant réalisable sur tous les scénarios, sur le pire cas pouvant arriver. (mesure averse au risque)

Analogie de théorie des jeux, comme si un adversaire fictif choisissait le pire aléa qui nous arriverait après avoir pris nos décisions : optimisation min-max

Les puissances peuvent être recalculées/adaptées pour chaque scénario, on a des variables de puissance pour chaque scénario, recours. C'est un problème min-max-min, le calcul des puissances se fait en connaissant les aléas.

N.B : l'optimisation robuste ne se limite pas à avoir un nombre fini de scénarios, c'est un de ses grands avantages, de ne considérer que des bornes pour les aléas. Les méthodes de résolution dépassent ce cours, la formalisation suivante permet de l'appréhender sur le cas fini, illustrant la linéarisation PLNE.

Formulation Programmation mathématique

$$\min_{x,y \geqslant 0} \max_{s \in \mathcal{S}} \min_{P \geqslant 0} \sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{T}} C_{prop}^{u} P_{t,s}^{u} + \sum_{u \in \mathcal{U}} \sum_{t \in T} C_{fix}^{u} x_t^u + \sum_{u \in \mathcal{U}} \sum_{t \in T} C_{start}^{u} y_t^u$$

s.t :

$$\forall u \in \mathcal{U}, \ \forall t \in [\![1, T]\!], \ x_t^u - x_{t-1}^u \leqslant y_t^u \quad (64)$$

$$\forall t \in [\![1, T]\!], \ \forall s \in \mathcal{S}, \ \sum_{u \in \mathcal{U}} P_{t,s}^u = D_{t,s} \quad (65)$$

$$\forall u \in \mathcal{U}, \ \forall s \in \mathcal{S}, \ \forall t \in [\![0, T]\!], \ x_t^u.Pmin_{t,s}^u \leqslant P_{t,s}^u \quad (66)$$

$$\forall u \in \mathcal{U}, \ \forall s \in \mathcal{S}, \ \forall t \in [\![0, T]\!], \ P_{t,s}^u \leqslant x_t^u.Pmax_{t,s}^u \quad (67)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-L^u+1}^{t} y_{t'}^u \leqslant x_t^u \quad (68)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-l^u+1}^{t} y_{t'}^u \leqslant x_{t-l^u}^u \quad (69)$$

Linéarisation PLNE

$$\min_{x,y,P,C\geqslant 0} C^{rob} + \sum_{u\in\mathcal{U}}\sum_{t\in T} C_{fix}^{u} x_t^u \;\; + \sum_{u\in\mathcal{U}}\sum_{t\in T} C_{start}^{u} y_t^u$$

s.t :

$$\forall s \in \mathcal{S}, \quad \sum_{u\in\mathcal{U}}\sum_{t\in\mathcal{T}} C_{prop}^{u} P_{t,s}^u \leqslant C^{rob} \qquad (70)$$

$$\forall u \in \mathcal{U}, \;\; \forall t \in [\![1, T]\!], \;\; x_t^u - x_{t-1}^u \leqslant y_t^u \qquad (71)$$

$$\forall t \in [\![1, T]\!], \;\; \forall s \in \mathcal{S}, \quad \sum_{u\in\mathcal{U}} P_{t,s}^u = D_{t,s} \qquad (72)$$

$$\forall u \in \mathcal{U}, \;\; \forall s \in \mathcal{S}, \;\; \forall t \in [\![0, T]\!], \;\; x_t^u.Pmin_{t,s}^u \leqslant P_{t,s}^u \qquad (73)$$

$$\forall u \in \mathcal{U}, \;\; \forall s \in \mathcal{S}, \;\; \forall t \in [\![0, T]\!], \;\; P_{t,s}^u \leqslant x_t^u.Pmax_{t,s}^u \qquad (74)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-L^u+1}^{t} y_{t'}^u \leqslant x_t^u \qquad (75)$$

$$\forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad \sum_{t'=t-l^u+1}^{t} y_{t'}^u \leqslant x_{t-l^u}^u \qquad (76)$$

CONCLUSIONS

Bilan

- Linéarisation PLNE en ajoutant des variables et/ou des contraintes, un problème non linéaire sur un espace de variables peut être rendu linéaire dans un plus grand espace vectoriel.
- Linéarisations vues : valeur absolue, fonctions linéaire par morceaux, optimisation max-min, expressions quadratiques de produits de variables binaires, produit binaire/continu borné ...
- Les expressions logiques (implications, ou, exclusion), sur lesquelles on raisonne plus naturellement s'écrivent automatiquement en PLNE.
- Une grande richesse pour modéliser des problèmes d'optimisation et de nombreuses variantes en PLNE
  $\Longrightarrow$ Beaucoup de problèmes industriels complexes s'écrivent en PLNE
  $\Longrightarrow$ Beaucoup de variantes de fonctions objectifs en PLNE
- Résoudre un PLNE général, ou même déterminer si une solution réalisable existe, est un problème NP-complet ! C'est la contre-partie de la facilité/richesse de modélisation PLNE.

Perspectives pour ce cours

- TD : modélisation de PL/PLNE plus complexes issus d'applications réelles, à l'aide des règles de linéarisations.
- Prochain cours : modélisations PLNE de problèmes d'optimisation dans les graphes.
- Seconde partie du module : pour résoudre de manière générique des problèmes modélisés en PLNE