

Rapport d'étape du projet ECMA

HASSAN Lucas, NGUYEN Tito

décembre 2015

1 Modélisation du problème sous la forme d'un PL

1.1 Modélisation naïve du problème

Dans un premier temps, on ne s'intéresse pas à la contrainte de connexité du problème. Le problème se modélise alors naturellement de la façon suivante :

$$\begin{aligned} \max \quad & \sum_{(i,j) \in M} x_{ij} \\ \text{s.t.} \quad & H^p(M) + H^a(M) \geq 2 \end{aligned} \tag{1}$$

$$\text{Où } H^p(M) = \frac{\sum_{(i,j) \in M} H_{ij}^p C_{ij}^p x_{ij}}{\sum_{(i,j) \in M} C_{ij}^p x_{ij}} \text{ et } H^a(M) = \frac{\sum_{(i,j) \in M} H_{ij}^a C_{ij}^a x_{ij}}{\sum_{(i,j) \in M} C_{ij}^a x_{ij}}$$

1.2 Linéarisation de la modélisation

On remarque que la contrainte définie est fractionnaire. Ainsi on choisit la linéariser de la manière suivante :

La contrainte s'écrit aussi :

$$\sum_{(i,j) \in M} H_{ij}^p C_{ij}^p x_{ij} \left(\sum_{(i,j) \in M} C_{ij}^a x_{ij} \right) + \sum_{(i,j) \in M} H_{ij}^a C_{ij}^a x_{ij} \left(\sum_{(i,j) \in M} C_{ij}^p x_{ij} \right) \geq 2 \left(\sum_{(k,l) \in M} C_{kl}^a x_{kl} \right) \left(\sum_{(k,l) \in M} C_{kl}^p x_{kl} \right) \tag{2}$$

On constate que cette formulation n'est équivalente à la première que si $\sum_{(k,l) \in M} C_{kl}^p x_{kl}$ et $\sum_{(k,l) \in M} C_{kl}^a x_{kl}$ sont non nulles.

Il faut donc ajouter les contraintes :

$$\begin{aligned} \sum_{(k,l) \in M} C_{kl}^p x_{kl} &\geq 0 \\ \sum_{(k,l) \in M} C_{kl}^a x_{kl} &\geq 0 \end{aligned} \tag{3}$$

Définissons les variables réelles p et a telle que :

$$\begin{aligned} p &= \left(\sum_{(i,j) \in M} C_{ij}^p x_{ij} \right) \\ a &= \left(\sum_{(i,j) \in M} C_{ij}^a x_{ij} \right) \end{aligned} \tag{4}$$

Les contraintes du problème d'optimisation s'écrivent alors :

$$\begin{cases} \sum_{(i,j) \in M} H_{ij}^p C_{ij}^p a x_{ij} + \sum_{(i,j) \in M} H_{ij}^a C_{ij}^a p x_{ij} \geq 2 \left(\sum_{(i,j) \in M} C_{ij}^a p x_{ij} \right) \\ p = \left(\sum_{(i,j) \in M} C_{ij}^p x_{ij} \right) \\ a = \left(\sum_{(i,j) \in M} C_{ij}^a x_{ij} \right) \end{cases} \tag{5}$$

On définit ensuite $\forall (i, j) \in M \quad a_{ij} = ax_{ij}$ et $p_{ij} = px_{ij}$

On sait que l'égalité $a_{ij} = ax_{ij}$ s'écrit encore :

$$\begin{aligned} a_{ij} &\leq a_{max}x_{ij} \\ a_{ij} &\leq a \\ a_{ij} &\geq (x_{ij} - 1)a_{max} + a \\ a_{ij} &\geq 0 \end{aligned} \tag{6}$$

où a_{max} est une borne supérieur de a . Une borne supérieur triviale de a est ici $\sum_{(i,j) \in M} C_{ij}^a$. Il en est de même pour la contrainte $p_{ij} = px_{ij}$.

Ainsi, on définit le problème linéarisé :

$$\left\{ \begin{array}{ll} \max \sum_{(i,j) \in M} x_{ij} & \\ \sum_{(i,j) \in M} H_{ij}^p C_{ij}^p a_{ij} + \sum_{(i,j) \in M} H_{ij}^a C_{ij}^a p_{ij} \geq 2 \left(\sum_{(i,j) \in M} C_{ij}^a p_{ij} \right) & \\ \forall (i, j) \in M \quad a_{ij} \leq a_{max}x_{ij} & p_{ij} \leq p_{max}x_{ij} \\ \forall (i, j) \in M \quad a_{ij} \leq a & p_{ij} \leq p \\ \forall (i, j) \in M \quad a_{ij} \geq (x_{ij} - 1)a_{max} + a & p_{ij} \geq (x_{ij} - 1)p_{max} + p \\ \forall (i, j) \in M \quad a_{ij} \geq 0 & p_{ij} \geq 0 \\ a = \left(\sum_{(i,j) \in M} C_{ij}^a x_{ij} \right) & p = \left(\sum_{(i,j) \in M} C_{ij}^p x_{ij} \right) \\ p > 0 & a > 0 \end{array} \right. \tag{7}$$

1.3 Prise en compte de la contrainte de connexité

Une commune peut être représentée par un graphe non orienté tel que chaque chaque maille (i, j) est un nœud du graphe, les nœuds étant reliés par des arrêtes si les mailles correspondante sont adjacentes dans $Z = \{(i, j) \in M \mid x_{ij} = 1\}$. On dira qu'un nœud (i, j) est à la distance k d'un nœud (m, n) si il existe une chaîne de taille k allant de (i, j) vers (m, n) .

Soit $\forall (i, j, k) \in M \times \llbracket 0; Card(M) - 1 \rrbracket \quad S_{i,j,k}$ qui vaut 1 si le nœud (i, j) est à la distance k d'un nœud central (i_0, j_0) et 0 sinon. On lui associe les contraintes suivante :

$$\left\{ \begin{array}{l} \sum_{(i,j) \in M} S_{i,j,0} = 1 \\ \sum_{(k) \in M} S_{i,j,k} = x_{ij} \\ \forall (i, j, k) \in M \times N \quad S_{i,j,k} \leq S_{i-1,j,k-1} + S_{i+1,j,k-1} + S_{i,j-1,k-1} + S_{i,j+1,k-1} \end{array} \right. \tag{8}$$

Où $N = \llbracket 0; Card(M) - 1 \rrbracket$

- La première contrainte assure de choisir un et un seul nœud central
- La seconde contrainte assure de n'attribuer de distance que pour les mailles de Z
- La dernière contrainte assure que (i, j) ne peut être à la distance k du nœud centrale que si au moins l'un de ses voisins est à la distance $k - 1$.

Le problème global s'écrit donc :

$$\left\{ \begin{array}{llll}
\max & \sum_{(i,j) \in M} x_{ij} & & \\
& \sum_{(i,j) \in M} H_{ij}^p C_{ij}^p a_{ij} + \sum_{(i,j) \in M} H_{ij}^a C_{ij}^a p_{ij} \geq 2 \left(\sum_{(i,j) \in M} C_{ij}^a p_{ij} \right) & & \\
& \forall (i,j) \in M & a_{ij} \leq a_{max} x_{ij} & p_{ij} \leq p_{max} x_{ij} \\
& \forall (i,j) \in M & a_{ij} \leq a & p_{ij} \leq p \\
& \forall (i,j) \in M & a_{ij} \geq (x_{ij} - 1) a_{max} + a & p_{ij} \geq (x_{ij} - 1) p_{max} + p \\
& \forall (i,j) \in M & a_{ij} \geq 0 & p_{ij} \geq 0 \\
& & a = \left(\sum_{(i,j) \in M} C_{ij}^a x_{ij} \right) & p = \left(\sum_{(i,j) \in M} C_{ij}^p x_{ij} \right) \\
& & p > 0 & a > 0 \\
& \forall k \in N & \sum_{(k) \in M} S_{ijk} = x_{ij} & \sum_{(i,j) \in M} S_{i,j,0} = 1 \\
& \forall (i,j,k) \in M \times N & S_{i,j,k} \leq S_{i-1,j,k-1} + S_{i+1,j,k-1} + S_{i,j-1,k-1} + S_{i,j+1,k-1} &
\end{array} \right. \quad (9)$$

(Les effets de bords pour la dernière contrainte ne sont pas spécifiés pour simplifier les notations)

1.4 Remarque

Les résultats seront présentés dans le rapport final, mais on peut déjà constater que la contrainte de connexité engendre un nombre très important de variable de décision (De l'ordre de $Card(M)^2$). Même sur des instance de taille très raisonnable (10x10), on génère 10000 variables. Une première question sera donc de se demander si on peut limiter le nombre de variable à ajouter pour définir les contraintes de connexité.

De plus, on peut remarquer que dans la solution final, les valeurs des S_{ijk} sont presque toutes nulles. On a donc fortement intérêt à brancher de préférence S_{ijk} vers sa valeur inférieure (nulle ici) dans le branch and cut.

Les résultats et d'autres améliorations présentées dans le rapport final.

2 Résolution du problème par décomposition-coordination

Lorsque qu'un problème est trop complexe pour être résolu de manière frontal (nombre de variables et/ou de contraintes trop élevé), il peut être intéressant d'utiliser des méthodes dites de décomposition-coordination qui consistent à définir une suite de sous problèmes plus simple à résoudre dont les solutions successives convergent vers la solution du problème initiale.

Ces méthodes s'appliquent particulièrement lorsque qu'une ou plusieurs contraintes du problème se trouvent être couplante. L'application des techniques de décomposition-coordination peut permettre de faire apparaître des problèmes découplés et donc beaucoup plus facile à résoudre.

On choisira ici de présenter quelques méthodes usuelles et de présenter les résultats de leur implémentation sur le problème considéré. Pour plus de détail sur les méthodes de décomposition-coordination on pourra consulter à profit le support de cours accompagnant le cours "Optimisation des grands systèmes de Pierre CARPENTIER (<http://perso.ensta-paristech.fr/~pcarpent/A8-4/>)

2.1 Considération générale sur le problème

La formulation du problème tel que présenté plus haut nous montre que les seules contraintes couplant S et les autres variables du problème sont les contraintes :

$$\forall (i,j) \in M \quad \sum_{(k) \in M} S_{ijk} = x_{ij} \quad (10)$$

Le coût est quand à lui totalement découplé (S n'y intervient pas). On sait que la difficulté de résoudre le problème global vient surtout de l'apparition des variables S et des contraintes liées à la connexité. Il apparaît donc intéressant d'essayer de découpler le problème en S et le reste du problème.

Soit : $S_{ad} \subset \{0,1\}^{Card(M)} \times \{0,1\}^N$ tel que $\forall(i,j,k) \in M \times N$, S_{ijk} respecte les contraintes suivante :

$$\begin{cases} \sum_{(i,j) \in M} S_{i,j,0} = 1 \\ S_{i,j,k} \leq S_{i-1,j,k-1} + S_{i+1,j,k-1} + S_{i,j-1,k-1} + S_{i,j+1,k-1} \end{cases} \quad (11)$$

De même on définit $X_{ad} \subset \{0,1\}^{Card(M)} \times \mathbb{R}^{Card(M)} \times \mathbb{R}^{Card(M)} \times \mathbb{R} \times \mathbb{R}$ tel que pour $(x, A, P, a, p) \in X_{ad}$, $\forall(i,j)$, x_{ij} , A_{ij} , P_{ij} , a , p respectent les contraintes suivante :

$$\begin{cases} \sum_{(i,j) \in M} H_{ij}^p C_{ij}^p a_{ij} + \sum_{(i,j) \in M} H_{ij}^a C_{ij}^a p_{ij} \geq 2 \left(\sum_{(i,j) \in M} C_{ij}^a p_{ij} \right) \\ a_{ij} \leq a_{max} x_{ij} & p_{ij} \leq p_{max} x_{ij} \\ a_{ij} \leq a & p_{ij} \leq p \\ a_{ij} \geq (x_{ij} - 1) a_{max} + a & p_{ij} \geq (x_{ij} - 1) p_{max} + p \\ a_{ij} \geq 0 & p_{ij} \geq 0 \\ & a = \left(\sum_{(i,j) \in M} C_{ij}^a x_{ij} \right) & p = \left(\sum_{(i,j) \in M} C_{ij}^p x_{ij} \right) \\ & p > 0 & a > 0 \end{cases} \quad (12)$$

Le problème global s'écrit alors :

$$\begin{cases} \max_{\substack{(x,A,P,a,p) \in X_{ad} \\ S \in S_{ad}}} \sum_{(i,j) \in M} x_{ij} \\ \forall(i,j) \in M \quad \sum_{(k) \in N} S_{ijk} = x_{ij} \end{cases} \quad (13)$$

Pour simplifier l'écriture, les variables A , P , p et a étant muette dans cette formulation, on écrira le problème comme suis :

$$\begin{cases} \max_{\substack{x \in X_{ad} \\ S \in S_{ad}}} \sum_{(i,j) \in M} x_{ij} \\ \forall(i,j) \in M \quad \sum_{k \in N} S_{ijk} = x_{ij} \end{cases} \quad (14)$$

2.2 Décomposition par les prix

Intuitivement la méthode de méthode de décomposition par les prix consiste à donner $\forall(i,j) \in M$ un prix $\lambda_{ij}^l \in \mathbb{R}$ à la quantité $\sum_{k \in N} S_{ijk} - x_{ij}$ et à résoudre les sous-problèmes :

$$\begin{cases} \max_{\substack{x \in X_{ad} \\ S \in S_{ad}}} \sum_{(i,j) \in M} x_{ij} + \sum_{(i,j) \in M} [\lambda_{ij}^l \sum_{k \in N} S_{ijk} - x_{ij}] \end{cases} \quad (15)$$

On montre qu'on peut trouver une matrice de prix λ_{ij}^* tel que la valeur de la solution associée soit égale à la valeur de la solution du problème initiale. Dans ce cas, on a nécessairement $\forall(i,j) \in M \quad \sum_{k \in N} S_{ijk} - x_{ij} = 0$ et la solution trouvée sera donc solution optimal du problème initial.

Cette algorithmme correspond à l'algorithme de relaxation lagrangienne. Ainsi on met à jour $\lambda^l = (\lambda_{ij}^l)_{(i,j) \in M}$ comme suit :

$$\forall(i,j) \in M \forall l \in \mathbb{N} \quad \lambda_{ij}^{l+1} = \lambda_{ij}^l + \rho_l \left(\sum_{k \in N} S_{ijk} - x_{ij} \right)$$

Pour assurer la convergence de l'algorithme on devra choisir la suite ρ_l de pas positive, convergeant vers 0 et tel que sa série diverge.

Appliquons cette décomposition à notre problème :

$$\begin{cases} \max_{\substack{x \in X_{ad} \\ S \in S_{ad}}} \sum_{(i,j) \in M} x_{ij} + \sum_{(i,j) \in M} [\lambda_{ij}^l \sum_{k \in N} S_{ijk} - x_{ij}] \end{cases} \quad (16)$$

$$\left\{ \max_{\substack{x \in X_{ad} \\ S \in S_{ad}}} \sum_{(i,j) \in M} x_{ij}(1 - \lambda_{ij}^l) \sum_{(i,j) \in M} [\lambda_{ij}^l \sum_{k \in N} S_{ijk}] \right. \quad (17)$$

$$\left\{ \max_{x \in X_{ad}} \sum_{(i,j) \in M} x_{ij}(1 - \lambda_{ij}^l) + \max_{S \in S_{ad}} \sum_{(i,j) \in M} [\lambda_{ij}^l \sum_{k \in N} S_{ijk}] \right. \quad (18)$$

On a donc bien découplé le problème en x et en S . Notons $P1$ et $P2$ les sous problèmes respectivement en x et S qui viennent d'être définis.

Interprétation : à faire ...

2.3 Décomposition par prédiction