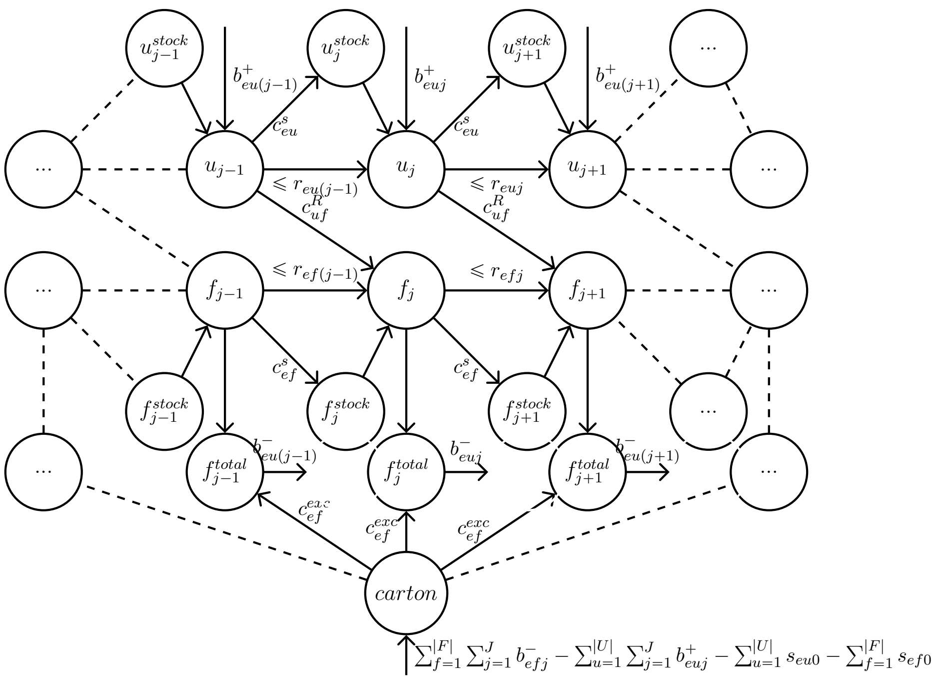
Miniprojet de recherche opérationnelle

Waël Doulazmi, Paul-Eloi Mangion, Ambroise Odonnat

Questions théoriques

1. On admet les hypothèses suivantes : $E = 1$ et $l_1 > L/2$. On note e cet unique type d'emballage. Alors chaque camion ne peut transporter qu'une pile de e et les routes ne desservent donc qu'un fournisseur. On peut modéliser le flot de pile de e comme un b-flot parcourant un graphe orienté où les sommets sont les usines, les fournisseurs et une usine (fictive) à cartons. La source de carton et toutes les usines sont reliées à chaque fournisseur de la manière suivante :



Précisions : une usine (respectivement un fournisseur) est représentée par J usines (fournisseurs) pour prendre en compte l'aspect temporel du problème. L'usine u_0 reçoit s_{eu0} et le fournisseur f_0 reçoit s_{ef0}. Les u_J et les f_J sont reliés à l'usine de carton pour que le stock final ne fuite pas du b-flot. On a $c^R_{uf} = \gamma \times d_a + c^{stop} + c^{cam}$. Lorsque ce n'est pas précisé, on admettra que le coût vaut 0 et que les contraintes sont $0 \leqslant \ \leqslant +\infty$.

2. Si la taille de l'instance ne dépend que de (U, F, J) alors les variables s, z sont de taille $O(J(|U| + |F|))$, la variable x elle est en $O(J(|U| + |F|)^5)$ puisque pour construire une route, il faut choisir une usine et jusqu'à 4 fournisseurs (donc de taille $\propto U\times$ produit de quatre termes en $\binom{F}{i}$ où $i \in [1, 4]$). Le nombre de variables est donc bien polynomial en la taille de l'instance. Si E n'était pas considéré constant, on aurait un nombre de variables exponentiel en la taille de l'instance, à cause du vecteur Q_R défini pour chaque route. La taille de l'instance étant de l'ordre de 600, on s'imagine mal résoudre directement le PLNE (11) avec un solveur (le nombre de variables est de l'ordre de la dizaines de billions). L'instance `europe.csv` représentant 10 Mo de données, rien que les variables du problèmes prendrait 1To de place : inimaginable d'avoir autant de RAM (sauf avec un super-ordinateur).

3. On fait l'hypothèse qu'au plus K camions sont utilisés par jour. Ce ne sont pas nécessairement les mêmes chaque jour aussi on notera $k \in K$ le k-ième camion au jour j. Cela assure que sur tout le dispatch au plus $K \times J$ routes sont parcourues. Une approche possible serait de remplacer la notion de parcours des routes par la notion de parcours d'arcs $l \to p$ où l est dans $U \cup F$ et p est dans F. En effet dans le PLNE (11), les variables x permettent de compter le nombre de camions parcourant une route R le jour j_R avec un chargement agrégé plus petit que L. Comme on sait qu'il y a au plus K camions par jour , on peut les indicer et pour chaque camion numéro k déterminer s'il parcourt le trajet entre une usine et un fournisseur (ou entre deux fournisseurs) pour un jour j donné. Cela revient bien au parcours d'un camion k sur les arcs $l \to p$ au jour j. Reste à introduire proprement les variables et les contraintes associées.

Dans toute la suite on notera $l \in U \cup F$ et $p \in F$ les extrémités d'un arc parcouru par un camion k. En effet, un camion part d'une usine et parcourt au plus quatre fournisseurs mais ne peut aller d'un fournisseur à une usine.

On introduit la variable binaire $x_{l \to p}^{k,j}$ qui vaut 1 si le camion k parcourt l'arc $l \to p$ au jour j et 0 sinon. On introduit la variable binaire y_j^k qui vaut 1 si le k-ième camion du jour j est utilisé. On introduit $q_{e,l \to p}^{k,j}$ la quantité d'emballage e transportée par le camion k entre l et p. Le coût de parcours d'un arc $l \to p$ est $c_{lp} = \mathbb{1}_{l \in U} c^{\text{cam}} + c^{\text{stop}} + \gamma d_{lp}$ où d_{lp} désigne la distance entre l et p.

On considère les contraintes suivantes qui assurent l'équivalence des problèmes :

- (10a) $z_{euj}^- = \sum_k \sum_{p \in F} q_{e,u \to p}^{k,j} x_{u \to p}^{k,j}$ $\forall e \in E, u \in U, j \in [J]$
- (10b) $z_{efj}^+ = \sum_k \sum_{l \in U \cup F} q_{e,l \to f}^{k,j} x_{l \to f}^{k,j}$ $\forall e \in E, f \in F, j \in [J]$
- (12) $\sum_e \sum_{l,p} q_{e,l \to p}^{k,j} x_{l \to p}^{k,j} \leqslant L \quad \forall k \in [K], j \in [J]$ liée au chargement agrégé
- (13a) $\sum_{u \in U, f \in F} x_{u \to f}^{k,j} = y_j^k \quad \forall k \in [K], j \in [J]$ liée au fait qu'un camion part forcément d'une usine
- (13b) $\sum_{l \in U \cup F, p \in F} x_{l \to p}^{k,j} \leqslant 4 y_j^k \quad \forall k \in [K], j \in [J]$ liée au nombre maximal de fournisseurs parcourus par un camion (i.e quatre)
- (13c) $\sum_{l \in U \cup F} x_{l \to f_1}^{k,j} = \sum_{f_2 \in F} x_{f_1 \to f_2}^{k,j} \quad \forall f_1 \in F, k \in [K], j \in [J]$
- (14a) $x_{l \to p}^{k,j} \in \{0,1\} \quad \forall l \in U \cup F, p \in F, k \in [K], j \in [J]$
- (14b) $y_j^k \in \{0,1\} \quad \forall k \in [K], j \in [J]$
- (14c) $q_{e,l \to p}^{k,j} \geqslant 0 \quad \forall l \in U \cup F, p \in F, e \in E, k \in [K], j \in [J]$

Le problème (11) se réécrit donc :

$$\begin{aligned} \min_{s,z,x} & \sum_{lp} c_{lp} \sum_{kj} x_{l \to p}^{k,j} + \sum_{euj} c_{eu}^s \max(s_{euj} - r_{euj}, 0) + \\ & \sum_{efj} c_{ef}^s \max(s_{efj} - r_{efj}, 0) + \sum_{efj} c_{ef}^{\text{exc}} \max(b_{efj}^- - s_{ef(j-1)}, 0) \\ & \text{s.t. } (1), (2), (6), (10), (12), (13), (14) \end{aligned}$$

Alors le nombre de variable liées à s et z demeure inchangé, mais $x_{l \to p}^{k,j}$ est un vecteur de $\{0,1\}^{(|U|+|F|)^2 \times JK}$, y_j^k est un vecteur de $\{0,1\}^{JK}$ et $q_{e,l \to p}^{k,j}$ est un vecteur de $\mathbb{R}_+^{|E| \times (|U|+|F|)^2 \times JK}$ ce qui assure que le nombre de variables du problème est en $O(|E| \times (|U|+|F|)^2 \times JK)$.

PLNE pour les flux

Une première idée pour obtenir une solution réalisable est de considérer qu'un camion ne peut transporter qu'un emballage.

Pour ce faire on change la modélisation :

- Le jour j la quantité d'emballage e fournie par l'usine u au fournisseur f vaut x_{eufj}
- Le coût associé à ce transport est : $c_{eufj} = x_{eufj} \times (c^{\text{cam}} + c^{\text{stop}} + \gamma d_{uf})$

Le nouveau problème s'écrit :

$$\min_{s,x,z} \sum_{eufj} c_{eufj} + \sum_{euj} c_{eu}^s \max(s_{euj} - r_{euj}, 0) + \sum_{efj} c_{ef}^s \max(s_{efj} - r_{efj}, 0) + \sum_{efj} c_{ef}^{\text{exc}} \max(b_{efj}^- - s_{ef(j-1)}, 0)$$

Les contraintes restent les mêmes que dans le PLNE initial à l'exception de la (10) qui devient :

$$z_{euj}^- = \sum_{f \in F} x_{eufj}, \quad z_{efj}^+ = \sum_{u \in U} x_{eufj}$$

Cette modélisation permet de réduire grandement le nombre de variables à l'instar de la question 3. En effet le nombre de variables est désormais en $O(J|E|(|U|+|F|)^2)$.

Cette modélisation permet d'obtenir une solution réalisable meilleure que lorsqu'on ne met pas de routes, mais elle a un grand désavantage : le coût d'expédition d'une pile d'emballage étant élevé, les usines sont encouragées à sur-stocker, et peu de routes sont créées.

Pour pallier ce penchant et encourager les usines à envoyer des emballages même si ce n'est pas forcément rentable en terme de coût total, on modifie le coût d'envoi d'un emballage dans la formulation PLNE : $c_{eufj} = x_{eufj} \times (c^{\text{cam}} + c^{\text{stop}} + \gamma d_{uf}) \times \frac{\ell_e}{L}$
Le terme $\frac{\ell_e}{L}$ permet de rendre réaliste le coût d'envoi d'un emballage : en effet dans le meilleur des cas l'emballage voyagera dans un camion parfaitement plein et son transport coûtera donc le prix c_{eufj}.
Le solveur Gurobi nous renvoie une solution optimale en 15 minutes sur `europe.csv`. Pour s'assurer de l'optimalité de la solution il suffit d'exécuter la commande : `termination_status(model)`
Ensuite on reconstruit les routes correspondantes de la manière suivante (algorithme 1) :

Algorithm 1: Construire des routes à partir de x le flot de camions

Data: Données de l'instance europe
Input: $x \in \mathbb{N}^{|E||U||F|J}$
Result: un vecteur de routes $\mathcal{R}$
Initialization : $\mathcal{R} \leftarrow \varnothing$
for $e \in E, u \in U, f \in F, j \in [J]$ **do**
  if $x_{eufj} > 0$ **then**
    $j_R \leftarrow j$;
    $P_R \leftarrow (u, f)$;
    $q_R \leftarrow (\mathbb{1}_{(i=e)})_{i \in E}$;
    $x_R \leftarrow x_{eufj}$;
    $R \leftarrow (j_R, P_R, q_R)$;
    Add R to $\mathcal{R}$
  end
end
return $\mathcal{R}$

Cela permet d'obtenir une solution admissible en utilisant ces routes. On obtient ainsi 20771 routes pour un coût total de **25 619 966** euros. La décomposition de ce coût est très intéressante : **6 570** euros pour les usines, **3 779 819** euros pour les fournisseurs et **21 833 577** euros pour les routes. En concentrant la suite de notre recherche sur l'amélioration de ces routes (qui sont pour l'instant loin d'être optimales puisque les emballages voyagent seuls) on pourrait grandement améliorer nos coûts.

Bin Packing

Pour améliorer nos routes une idée naturelle est d'optimiser le remplissage des camions chaque jour sur les trajets $u \to f$ pour $u \in U$, $f \in F$. Pour ce faire on extrait de $\mathcal{R}$ le sous-ensemble :
$\mathcal{R}_{juf} = \{R \in \mathcal{R},\ j_R = j,\ P_R = (u, f)\}$ Et on améliore nos routes en optimisant le remplissage des camions sur le trajet $u \to f$ du jour j (algorithme 2).
On a utilisé l'heuristique *FIRST-FIT-DECREASING* décrite dans le cours, qui fournit une $\frac{3}{2}$-approximation du problème de bin-packing et qui a l'avantage de s'implémenter facilement car intuitive. L'application de l'algorithme 2 pour tous les couples (j, u) nous a permis, comme espéré, de faire baisser notre coût de routes. Sur Europe on obtient un coût total de **10 579 336** euros, avec le même coût pour les usines et les fournisseurs et un coût de **6 792 947** euros pour les routes.

Construction de routes à plusieurs fournisseurs

Maintenant que nos trajets $u \to f$ sont optimisés, on peut envisager de construire des routes d'au plus quatre fournisseurs en mutualisant les routes obtenues. On notera **cond** le fait que pour deux routes r_1,r_2 données, le nombre de fournisseurs que parcourt r_1 est au plus 3, elles sont parcourues le même jour avec la même usine de départ et la somme des chargement agrégés de r_1 et r_2 est plus petit que L, la taille d'un camion.

On applique l'algorithme 3. Cet algorithme permet d'obtenir une solution admissible en utilisant ces routes et s'exécute en quelques secondes ($\leqslant$ 10 s). On obtient ainsi 3799 routes pour un coût total de **9 810 035** euros. Le coût se décompose ainsi : **6 570** euros pour les usines, **3 779 819** euros pour les fournisseurs et **6 023 646** euros pour les routes. On remarque une amélioration non négligeable par rapport aux routes non mutualisées. Néanmoins le coût de route demeure élevé et une autre approche pourrait permettre de meilleurs résultats.

Algorithm 2: Optimiser les routes existantes

Input: $\mathcal{R}_{juf}$
Result: $\mathcal{R}_{juf}^{\text{new}}$, un nouvel ensemble de routes, de meilleur coût
$\mathcal{R}_{juf}^{\text{new}} \leftarrow \varnothing$;
$a \leftarrow [\,]$ un vecteur d'emballages qui représente l'ensemble des piles transitant de u à f le jour j;
for $R \in \mathcal{R}_{juf}$ **do**
    $e \leftarrow \text{argmax}(q_R)$;
    On ajoute x_R fois l'emballage e au vecteur a;
end
On trie a par ordre décroissant de valeur de ℓ_e;
$(f, k) \leftarrow$ le résultat de l'algorithme du first-fit appliqué à (a, L);
for $i \in [k]$ **do**
    $Q_R \leftarrow (0, \ldots, 0)$;
    $j_R \leftarrow j$;
    $P_R \leftarrow (u, f)$;
    $x_R \leftarrow 1$;
    On ajoute $R = (j_R, P_R, Q_R)$ à $\mathcal{R}_{juf}^{\text{new}}$
end
for $i = 1$ **to** *longueur de a* **do**
    $e \leftarrow a[i]$;
    $R \leftarrow \mathcal{R}_{juf}^{\text{new}}[f(i)]$;
    $Q_{Re} \leftarrow Q_{Re} + 1$
end
return $\mathcal{R}_{juf}^{\text{new}}$

Algorithm 3: Mutualiser les routes existantes

Input: un vecteur de routes $\mathcal{R}^{prec}$
Result: un vecteur de routes d'au plus quatre fournisseurs $\mathcal{R}$
Initialization : $\mathcal{R} \leftarrow \varnothing$
for $u \in U, j \in [J]$ **do**
    $R \leftarrow$ les routes partant de u le jour j ordonnées par ordre décroissant de chargement agrégé
    $i \leftarrow 1$
    $N \leftarrow |\mathcal{R}|$
    while $i \leqslant N - 1$ **do**
        if ***cond*** **then**
            $R[i] \leftarrow R[i] + R[i+1]$
            $i \leftarrow i - 1$
        end
        $N \leftarrow |\mathcal{R}|$
        $i \leftarrow i + 1$
    end
    for $r \in R$ **do**
        $r \leftarrow$ r triée dans l'ordre croissant des distances de ses arrêts à u
        Add r to $\mathcal{R}$
    end
end
return $\mathcal{R}$

Pistes non explorées

Une idée que nous n'avons pas réussi à implémenter informatiquement : après avoir obtenu les routes $\mathcal{R}$ du premier PLNE, associer à chaque usine u l'ensemble $F_{ju} = \{f \in F,\ \exists R \in \mathcal{R},\ j_R = j,\ P_R = (u, f)\}$ puis ensuite résoudre le problème du voyageur de commerce sur le graphe de sommets $u \cup F_{ju}$, puis récupérer les 4 fournisseurs reliés à u dans le trajet obtenu. Puis résoudre à nouveau le voyageur de commerce. Cela aurait permis d'optimiser les trajectoires des camions, on aurait ensuite pu appliquer la partie bin-packing pour optimiser leur chargement.