Model for the truck-and-freighter routing problem

Dorian Dumez

December 16, 2017

1 Modèle

1.1 Données du problème

Les constantes et les données du problèmes sont notées :

- Le nœuds sont :
  - le nœud 1 est le dépôt
  - $J = \{1, ..., n\}$ est l'ensemble des clients, où :
    - $J_S \subset J$ est les clients ne pouvant être livré que par le petit
    - $J_B \subset J$ est les clients ne pouvant être livré que par le gros
    - $J_R = J \backslash (J_B \cup J_S)$ mes clients pouvant être livré par les 2
  - P est l'ensemble des parking
  - $V = \{0, n+1\} \cup J \cup P$, $E = V^2$
- les demandes et capacités sont :
  - $\forall j \in J$: q_j la demande du client j
  - $Q^B = \infty$ la capacité du gros
  - Q^S la capacité du petit
- les coûts sont :
  - $\forall (i,j) \in E$: c_{ij}^B le coût du trajet (i,j) avec le gros (ou les deux ensembles)
  - $\forall (i,j) \in E$: c_{ij}^S le coût du trajet (i,j) avec le petit
- les trucs de restrictions temporelles sont :
  - $\forall (i,j) \in E$: t_{ij} le temps de trajet de (i,j)
  - $\forall j \in J$: $[a_j, b_j]$ la fenêtre de temps du client j
  - e la date finale à laquelle tout le monde doit être rentré
  - sd la durée du service chez un client
  - ta le temps d'assemblage des 2 véhicules (fixé au même temps que la durée d'un service chez un client)
  - r le ratio de vitesse entre les 2 véhicules

1.2 Modèle de base

Les variables sont :

- pour la gestion du tour du gros :
  - $\forall (i,j) \in E : X_{ij} \in \{0,1\}$ le gros vas de i à j
  - $\forall i \in V : S_i \in \mathbf{R}^+$ le gros commence à servir i à cette date ou à être sur le parking. S_1 est un cas particulier, il représente la date de retour au dépôt
- pour la gestion de la k-ieme tournée du petit($k \in]K]$, K max de tournées du petit):
  - $\forall (i,j) \in E : x_{ij}^k \in \{0,1\}$ le petit vas de i à j
  - $\forall i \in V : s_i^k \in \mathbf{R}^+$ le petit commence à servir i à cette date ou à être sur le parking
- pour la gestion de la synchronisation petit/gros :
  - $\forall k \in]K] : \forall i \in P : d_i^k \in \{0,1\}$ la tourné k du petit démarre en i
  - $\forall k \in]K] : \forall i \in P : f_i^k \in \{0,1\}$ la tourné k du petit fini en i

Les contraintes sont :

- 2 : tous les clients sont servis
- 3, 4, 12, 22 : les fenêtre de temps des livraisons sont respectées
- 8, 9, 10 : les contrainte de sequentialité du gros dans les divers cas
- 18, 19 : contrainte de sequentalité du petit
- 11 : la date maximale de retour au dépôt
- 5 et 7 : respectivement qu'il faut partir du dépôt et y revenir
- 13, 15 et 14, 17 : respectivement que le petit doit partir et revenir au gros
- 20 et 21 : respectivement que les séparations et fusion se font bien au même moments
- 6 et 16 : contrainte de flot respectivement sur les déplacements du gros et du petit
- 23 : contrainte de capacité du petit

$$\min \quad \sum_{(i,j)\in V} (c_{ij}^B X_{ij} + \sum_{k\in]K]} (c_{ij}^S x_{ij}^k)) \tag{1}$$

$$\text{sc} \quad \forall j \in J : \sum_{i\in V} (X_{ij} + \sum_{k\in]K]} x_{ij}^k) = 1 \tag{2}$$

$$\forall i \in J_R : \forall k \in]K] : a_i \leqslant s_i^k \leqslant b_i \tag{3}$$

$$\forall i \in J_R : a_i \leqslant S_i \leqslant b_i \tag{4}$$

$$\sum_{j\in V} X_{1,j} = 1 \tag{5}$$

$$\forall h \in J \cup P : \sum_{i\in V} (X_{ih}) - \sum_{j\in V} (X_{hj}) = 0 \tag{6}$$

$$\sum_{i\in V} X_{i,1} = 1 \tag{7}$$

$$\forall i \in J : \forall j \in V : S_i + t_{ij} + sd - M(1 - X_{ij}) \leqslant S_j \tag{8}$$

$$\forall i \in P : \forall j \in V : S_i + t_{ij} + ta - M(1 - X_{ij}) \leqslant S_j \tag{9}$$

$$\forall j \in V : 0 + t_{1j} - M(1 - X_{1j}) \leqslant S_j \tag{10}$$

$$S_1 \leqslant e \tag{11}$$

$$\forall i \in J_B : a_i \leqslant S_i \leqslant b_i \tag{12}$$

$$\forall k \in]K] : \forall j \in P : d_j^k \leqslant \sum_{i\in V} X_{ij} \tag{13}$$

$$\forall k \in]K] : \forall j \in P : f_j^k \leqslant \sum_{i\in V} X_{ij} \tag{14}$$

$$\forall k \in]K] : \forall i \in P : \sum_{j\in V} x_{ij}^k = d_i^k \tag{15}$$

$$\forall k \in]K] : \forall h \in J : \sum_{i\in V} (x_{ih}^k) - \sum_{j\in V} (x_{hj}^k) = 0 \tag{16}$$

$$\forall k \in]K] : \forall j \in P : \sum_{i\in V} x_{ij}^k = f_j^k \tag{17}$$

$$\forall k \in]K] : \forall i \in J : \forall j \in V : s_i^k + r * t_{ij} + sd - M(1 - x_{ij}^k) \leqslant s_j^k \tag{18}$$

$$\forall k \in]K] : \forall i \in P : \forall j \in V : s_i^k + r * t_{ij} + a - M(1 - x_{ij}^k) \leqslant s_j^k \tag{19}$$

$$\forall k \in]K] : \forall i \in P : s_i^k \geqslant S_i - M.(1 - d_i^k) \tag{20}$$

$$\forall k \in]K] : \forall i \in P : \forall j \in V : s_i^k + a + t_{ij} \leqslant S_j + M.(1 - f_i^k) + M.(1 - X_{ij}) \tag{21}$$

$$\forall k \in]K] : \forall i \in J_S : a_i \leqslant s_i^k \leqslant b_i \tag{22}$$

$$\forall k \in]K] : \sum_{j\in J} (q_j \sum_{i\in V} x_{ij}^k) \leqslant Q_S \tag{23}$$

1.3 Ajouts au modèle

Les variables que l'on fixe (sans celles ci le modèle est faut) sont :

- $\forall k \in]K] : \forall i \in V : x^k_{i1}, x^k_{1i}, s^k_1 = 0$ car le petit n'interagit pas avec le dépôt
- $\forall i \in V :$ enlever (i,i) de V
- $\forall k \in]K] : \forall i \in J_B : \forall j \in V : x^k_{ij}, x^k_{ji} = 0$ car le petit n'a rien à faire chez les gros clients
- $\forall i \in J_S : \forall j \in V : X_{ij}, X_{ji} = 0$ car le gros n'a rien à faire chez les petits clients
- $\forall kin]K] : \forall i,j \in P : x^k_{ij} = 0$ car le petit n'a aucun intérêt à aller d'un parking à un autre

Les grandes valeurs à fixer :

- $M = e$ car c'est le plus grand temps possible
- $K = |J_S|/2$, pour être sûr d'avoir une borne supérieur sur le nombre de tours du petit il faudrait prendre $|J_S| + |J_R|$ mais cela devient vraiment trop prohibitif.
- le nombre de duplication des parkings : 2K, car un petit peu finir sa tourné sur le même parking sur lequel il l'a commencé et que pendant ce temps le gros ai aussi fait une tournée

Les contraintes redondantes sont :

- Une contrainte de capacité plus forte pour le petit basé sur les cover maximaux de knapsack. Un cover est un ensemble d'objet qui respecte la capacité, il est dit maximal si lui ajouter un objet fait dépasser la capacité. On adapte cela à chaque tournée de petit en générant des covers maximaux de clients. Alors pour chaque covers et chaque objet qui ne s'y trouve pas on interdit que le petit se rende (dans la même tourné) dans cet ensemble de client.
- On interdit toute action dans un sous-tour du petit si aucun départ n'est fixé. C'est à dire que si $\sum\limits_{i \in P} d^k_i = 0$ pour un k donné alors $\forall i \in P : f^k_i = 0$ et $\forall (i,j) \in E : x^k_{ij} = 0$. Malheureusement cette contrainte augmente les temps de résolutions.

Les symétries que l'on casse sont :

- pour les duplications de parkings : forcer à prendre en premiers les première duplications, et dans l'ordre $\Rightarrow$ contraintes sur les S_i et les X_{ij}
- prendre les premiers sous tours et dans l'ordre :

$$\forall k \in]K-1] : \forall i,j \in P : s^k_i \leqslant s^{k+1}_j \tag{24}$$

- utiliser les parking dans l'ordre,i.e copie 1 après l'original, copie 2 après copie 1 ... Autant sur le fait d'utiliser le parking (donc sur les $\sum\limits_{k=1}^{K} d^k_i$) que sur leurs dates d'utilisations

Remarque : avec ce modèle quand le petit véhicule rejoint le gros à un parking juste pour se ré-approvisionner on compte le temps d'assemblage 2 fois (assemblage + désassemblage). En effet un ré-approvisionnement est compté comme un nouveau "mini-VRP" du petit dans le modèle.

Remarque : la fonction objectif ne prend pas en compte l'heure à laquelle l'équipe rentre au dépôt. Par conséquent si la contrainte de taille de la journée n'est pas restrictive des temps d'attentes très larges restent optimaux.

2 Tests

instance	$\lvert J\rvert$	$\lvert JS\rvert$	$\lvert JR\rvert$	$\lvert P\rvert$	temps
1	6	3	3	2	0.7
2	8	4	4	3	4.51
3	10	5	5	4	16.56
4	12	6	6	5	> 600
5	12	6	6	2	> 600
6	7	5	2	5	> 600
7	8	2	6	5	2.29
8	10	2	8	5	305

Table 1 – temps d'exécution de Cplex en seconde selon les paramètres

Les expérimentation ont été faite avec julia 0.6 sur un asus rog G75VX-T4113H tournant avec un processeur Intel Core i7-3630QM (2.4 GHz - Cache 6 Mo) et une mémoire vive DDR3 de 8Go.

On note que Cplex trouve asse vite une solution qu'il va conserver jusqu'à la fin. On peut donc supposer qu'il peine à démontrer l'optimalité de sa solution. Cela laisse à penser qu'il faudrait rajouter des contraintes pour casser d'autres symétries ou encore rajouter des inégalités valides pour améliorer la relaxation linéaire.

Même si la capacité du modèle à gérer des clients pour les 2 camions plutôt que ceux réservé au petit est plus importante on note qu'avec plus de 8 clients dans J_R la résolution prend plus de 10mn. Il y a donc du travail autour de tous les clients mais surtout autour de ceux de J_S.