Correction du modèle frontal

Vianney Bœuf

Année 2015

On gardera bien en tête que le modèle proposé dans ce document ne correspond qu'à *une* des modélisations possibles de ce problème sous forme d'un problème linéaire en nombre entiers. Ce n'est pas le modèle utilisant le moins de variables (ou de contraintes), mais celui qui nous semble le plus clair. Il est tout à fait acceptable de continuer le travail avec votre propre modèle s'il est correct ; dans ce cas, envoyez-moi tout de même un mail pour être sûr qu'il n'y a pas d'erreurs dans le modèle.

Pour simplifier les notations, on notera :

— $d_B(w,i) = d(B_w, L_i)$ la distance entre la base B_w d'un technicien $w \in W$ et le lieu L_i d'une tâche $i \in J$;

— $d_L(i,j) = d(L_i, L_j)$ la distance entre les lieux L_i et L_j de deux tâches distinctes i et j de J.

et, de la même façon :

— $T_B(w,i) = V^{-1}d(B_w, L_i)$ le temps nécessaire au technicien $w \in W$ pour se rendre de sa base au lieu L_i de la tâche $i \in J$;

— $T_L(i,j) = V^{-1}d(L_i, L_j)$ le temps nécessaire à un technicien pour se rendre entre les lieux L_i et L_j de deux tâches distinctes i et j de J.

On rappelle que tous les techniciens se déplacent à la même vitesse V.

1 Variables modélisant les trajets

Comme dans le modèle du voyageur de commerce vu en cours, on va introduire, pour le trajet de chaque technicien, une variable binaire pour chaque arc possible traduisant le fait que l'arc est emprunté par le chemin. Plus précisément, on introduit les variables binaires suivantes :

— n_{ij}^w vaut 1 si le technicien w réalise la tâche j juste après la tâche i ;

— f_i^w vaut 1 si le technicien w réalise la tâche i en premier ;

— ℓ_i^w vaut 1 si le technicien w réalise la tâche i en dernier.

Ces variables doivent vérifier un ensemble de contraintes pour s'assurer que le trajet de chaque technicien est cohérent. On remarquera l'analogie entre ces variables et un « flot », défini pour chaque technicien, dont la source et le puits seraient la base du technicien, et les nœuds sont les tâches à réaliser.

Cette constatation va nous guider pour l'écriture de la première famille de contrainte, qui correspond à une « loi des nœuds » écrite pour chaque technicien : si le trajet d'un technicien « entre » dans une tâche (un nœud), soit en tant que première tâche, soit après une autre, alors il doit en « sortir » (pour rentrer à sa base ou pour faire une autre tâche).

$$\forall w \in W, \forall i \in J \quad f_i^w + \sum_{j \neq i} n_{ji}^w = \ell_i^w + \sum_{j \neq i} n_{ij}^w \tag{1}$$

De plus, la valeur du flot à la source et au puits doit être la même : si le technicien part de la base (c'est-à-dire qu'il réalise une première tâche), alors il doit également en réaliser une dernière :

$$\forall w \in W, \quad \sum_{i \in J} f_i^w = \sum_{i \in J} \ell_i^w \tag{2}$$

Il reste à s'assurer que le trajet du technicien ne réalise qu'une seule « boucle » au plus, c'est-à-dire que le technicien ne réalise qu'une tâche au plus à la fois. Attention, un technicien peut parfaitement ne réaliser aucune tâche. Au vu des contraintes précédentes, il suffit de vérifier que chaque technicien réalise au plus une tâche initiale.

$$\forall w \in W, \quad \sum_{i \in J} f_i^w \leq 1 \tag{3}$$

Si toutes les contraintes précédentes sont respectées, alors chaque technicien réalise au plus un trajet cohérent entre les différentes tâches.

2 Réalisation des tâches

Par simplicité de notation, on introduit une variable binaire x_i^w valant 1 si et seulement si le technicien w réalise la tâche i. On remarquera, grâce aux contraintes précédentes, que la valeur de x_i^w peut s'exprimer simplement par :

$$\forall w \in W, \forall i \in J, \quad x_i^w = f_i^w + \sum_{j \neq i} n_{ji}^w \tag{4}$$

les variables x servent donc uniquement à simplifier les notations.

On peut alors vérifier que chaque tâche est effectuée exactement une fois, par exemple à l'aide de la contrainte suivante :

$$\forall i \in J, \quad \sum_{w \in W} x_i^w = 1$$

Toutefois, il ne sera pas nécessaire d'imposer que chaque tâche est effectuée au plus une fois car on peut démontrer que pour toute solution admissible telle qu'une tâche est visitée au moins deux fois, il existe une solution admissible telle que cette tâche n'est visitée qu'une fois et dont la valeur est strictement inférieure. Cela donne donc la contrainte :

$$\forall i \in J, \quad \sum_{w \in W} x_i^w \geq 1 \tag{5}$$

Il reste à vérifier que chaque technicien dispose des compétences nécessaires à la réalisation des tâches sur sont trajet et à s'assurer que les fenêtres de temps technicien et intervention sont respectées pour satisfaire les contraintes du problème.

3 Respect des contraintes de compétences

Pour chaque technicien $w \in W$ et pour chaque tâche $i \in J$, introduisons une donnée S_i^w valant 1 si le technicien w a la compétence nécessaire à la réalisation de la tâche i. Les contraintes de compétences s'expriment en terme de logique : si $S_i^w = 0$, alors $x_i^w = 0$. Une implication logique reliant des variables binaires peut se modéliser sous forme d'inégalité dans un modèle linéaire ; dans ce cas, on introduit les contraintes suivantes :

$$\forall w \in W, \forall i \in J, \quad x_i^w \leq S_i^w \tag{6}$$

On pourra toutefois remarquer que, les compétences des techniciens étant des données du problème, il est tout à fait possible d'introduire les variables f_i^w, n_{ij}^w et ℓ_i^w *uniquement* dans le cas où le technicien w a les compétences nécessaires pour réaliser les tâches i et j. Ceci diminue évidemment le nombre de variables du problème et rend la contrainte ci-dessus inutile. Ce type de simplification est difficile à réaliser avec un modeleur, mais tout à fait possible lorsque l'on programme directement en C++.

4 Respect des horaires du technicien

Introduisons maintenant une variable continue t_i pour chaque tâche $i \in J$, correspondant à l'instant auquel un technicien commence à réaliser la tâche i. Supposons la tâche j réalisée par le technicien w. Pour que la fenêtre de temps soit respectée, il faut que le technicien ait le temps d'arriver au lieu de la tâche i avant t_i. Il y a deux cas possibles : si le technicien réalise la tâche i en premier, et s'il réalise la tâche i après une autre tâche j. En termes de logique :

$$\begin{array}{rcl} f_i^w = 1 & \Rightarrow & T_w^{\text{start}} + T_B(w,i) \leq t_j \\ n_{ji}^w = 1 & \Rightarrow & t_i + D_i + T_J(i,j) \leq t_j \end{array}$$

Ce type de contrainte se modélise encore en utilisant des contraintes d'inégalités. Pour s'assurer que le membre de droite (la contrainte d'inégalité) est toujours vérifiée lorsque le membre de gauche (le prédicat logique) ne l'est pas, on utilise la méthode du « big M », qui consiste à introduire une constante M grande devant toutes les durées du problème. On vérifiera sans difficulté que les contraintes logiques ci-dessus sont alors équivalentes aux contraintes linéaires suivantes :

$$\forall w \in W, \forall i \in J \quad t_s^w + T_B(w,i) \leq t_i + M(1 - f_i^w) \tag{7}$$

$$\forall w \in W, \forall i \in J, \forall j \neq i \quad t_i + D_i + T_L(i,j) \leq t_j + M(1 - n_{ji}^w). \tag{8}$$

Enfin, il reste à vérifier que chaque technicien a bien le temps de rentrer à sa base une fois la dernière tâche effectuée, soit :

$$\forall i \in J, \forall w \in W \quad t_i + D_i + T_B(w,i) \leq T_w^{\text{end}} + M(1 - \ell_i^w) \tag{9}$$

5 Respect des fenêtres de temps intervention

À l'aide des variables précédentes, il est aisé de vérifier que les fenêtres de temps sont respectées, grâce à la règle suivante :

$$\forall i \in J \quad r_i = 1 \Rightarrow t_i^{\min} \leq t_i \leq t_i^{\max}$$

Les r_i étant des données du problème, il n'y a pas besoin de modifier la contrainte :

$$\forall i \in J \text{ t.q. } r_i = 1, \quad t_i^{\min} \leq t_i + M(1 - r_i) \tag{10}$$

$$\forall i \in J \text{ t.q. } r_i = 1, \quad t_i \leq t_i^{\max} + M(1 - r_i) \tag{11}$$

6 Contraintes de sous-tours

On remarquera que ce modèle ne requiert pas de contraintes de sous-tours supplémentaires. En effet, grâce aux contraintes de respect des fenêtres de temps (8), on peut vérifier que la valeur de la variable t_i croit *strictement* sur le trajet d'un technicien. Par conséquent, si un tel trajet contenait un cycle, il n'y aurait pas d'affectation admissible des valeurs de t_i sur les points de ce cycle.

7 Fonction objectif

La somme des distances parcourues peut s'écrire [1] :

$$f(n,f,l) = \sum_{w \in W} \sum_{i \in J} d_B(w,i)(f_i^w + \ell_i^w) + \sum_{\substack{i,j \\ j \neq i}} d_L(i,j) \left(\sum_{w \in W} n_{ij}^w \right). \tag{12}$$

1. Comme précisé en séance du 18 septembre, les données de pénalité présentes dans la description du problème sont à ignorer pour ce modèle. La notation sera clémente pour le rendu du modèle frontal en cas de confusion sur les pénalités.

8 Modèle complet

Finalement, le modèle complet s'écrit :

$$\min_{n,f,\ell,x,t} \quad \sum_{i\in J}\sum_{w\in W} d_B(w,i)(f_i^w+\ell_i^w) + \sum_{\substack{i,j\\ j\neq i}} d_L(i,j)\left(\sum_{w\in W} n_{ij}^w\right)$$

$$\begin{array}{lll}
s.c. & f_i^w + \sum_{j\neq i} n_{ji}^w = \ell_i^w + \sum_{j\neq i} n_{ij}^w & \forall w\in W,\ \forall i\in J \\
& \sum_{i\in J} f_i^w = \sum_{i\in J} \ell_i^w & \forall w\in W \\
& \sum_{i\in J} f_i^w \leq 1 & \forall w\in W \\
& & \\
& x_i^w = f_i^w + \sum_{j\neq i} n_{ji}^w & \forall w\in W,\ \forall i\in J \\
& \sum_{w\in W} x_i^w \geq 1 & \forall i\in J \\
& x_i^w \leq S_i^w & \forall w\in W,\ \forall i\in J \\
& & \\
& T_w^{\text{start}} + T_B(w,i) \leq t_i + M(1-f_i^w) & \forall w\in W,\ \forall i\in J \\
& t_i + D_i + T_L(i,j) \leq t_j + M(1-n_{ji}^w) & \forall w\in W,\ \forall i\in J,\ \forall j\neq i \\
& t_i + D_i + T_B(w,i) \leq T_w^{\text{end}} + M(1-\ell_i^w) & \forall w\in W,\ \forall i\in J \\
& & \\
& t_i^{\min} \leq t_i & \forall i\in J \text{ t.q. } r_i = 1 \\
& t_i \leq t_i^{\max} & \forall i\in J \text{ t.q. } r_i = 1 \\
& & \\
& n_{ij}^w \in \{0,1\} & \forall w\in W,\ \forall i\in J,\ \forall j\neq i \\
& f_i^w, \ell_i^w, x_i^w \in \{0,1\}^3 & \forall w\in W,\ \forall i\in J \\
& t_i \in \mathbb{N}, \quad 0\leq t_i \leq M & \forall i\in J
\end{array}$$

Il s'agit bien d'un programme linéaire en nombres entiers.