

The Truck & Freightier Routing Problem

ORO - Optimization in Transportation and Logistics

Project at home

Romain BERNARD, Pauline CHATELLIER

2018

n.b : Pour exécuter la résolution, simplement lancer `include("modelFinal.jl")` dans un interpréteur Julia. Le choix de l'instance se fait en utilisant son numéro, trouvable dans les tables de résultats en section 5. Le script python requiert le chemin de l'instance au début du fichier pour s'exécuter et retourner la carte correspondant aux résultats. Les traits bleus correspondent au petit camion, et les rouges au gros camion.

1 Project Organization

The project is to be made by pair. The goal is to propose a mixed integer linear program to model this problem and to implement it.

A report (based on the current document for which LaTeX sources are given), and a code in either Julia or Python are expected at the end.

Allowed solvers : CBC, Gurobi, CPLEX.

2 Problem description

The Truck & Freightier Routing Problem consists in delivering goods starting from a principal distribution center to the multiple customers in a city. The specificity of the problem is that the delivery of goods is performed by an innovative type of truck¹, where a small electric truck (called city freighter) can travel inside a larger truck (called truck in the following). The city implements an innovative logistics delivery policy and some streets are forbidden to traditional trucks. As a result, some customers can be served only by the city freighter.

In the Truck & Freightier Routing Problem (TFRP), customers are all delivered from a depot located outside of the city. To travel to the city center, the large truck leaves this depot, carrying the city freighter. Both vehicles can separate and join again at dedicated parking areas. In terms of capacity, the large truck can be considered as having an infinite capacity. The city freighter has a more limited capacity, but it can be resupplied several times by the large truck at parking areas. The day is considered to start at time 0 and end at the end of the time horizon T . All trucks should be back at the depot by this time.

Each customer has a location, a quantity and a time window. All customer should be served before the end of their time window. If a truck arrives at a customer before the opening of the time window, it has to wait until this time.

The routing costs are considered to be equal to the sum of vehicle routing traveling times. Note that no cost is accounted for when the small city freighter travels within the large truck.

¹<http://bil.libner.com>

The problem consists of designing the routes of the truck and city freighter, including determining when they travel together and separately and where they separate and join, such that all routes start and end at the depot, customers are served within their respective time windows, vehicles capacity are respected, and traveling costs are minimized.

This problem is illustrated on Figure 1.

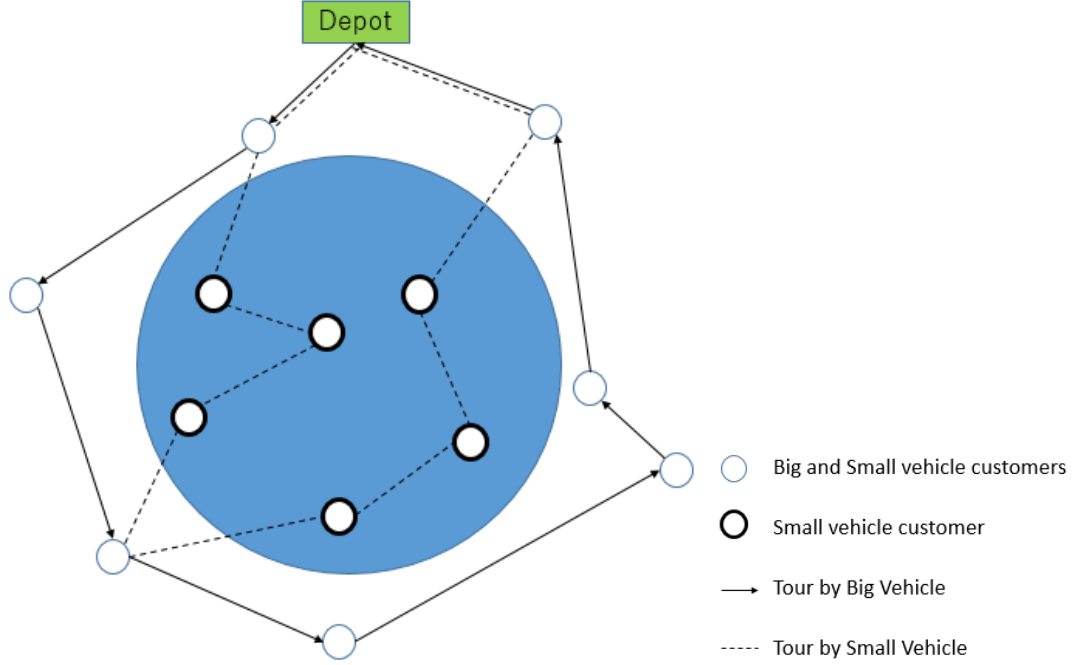


Figure 1: Typical solution of the truck and freighter routing problem

3 Notation

To model the problem, the following notation is given

3.1 Sets and data

Sets:

- $J = \{1, \dots, n\}$: set of customers.
- vertex 0 and $n + 1$ model the depot.
- $J_S \subset J$: nodes which can be served by the small truck only
- P : set of transfer nodes (parking areas)
- $V = \{0; n + 1\} \cup J \cup P$

Data:

- c_{ij}^B : cost of going from node $i \in V$ to node $j \in V$ with the big truck
- c_{ij}^S : cost of going from node $i \in V$ to node $j \in V$ with the small truck alone
- q_j demand of customer $j \in J$

- Q capacity of the small truck
- t_{ij} : time of traveling from node $i \in V$ to node $j \in V$
- s_j : duration of service at customer $j \in J$
- $[a_j, b_j]$: customer $j \in J$ time window
- T time horizon.

In this project we assume that $c_{ij}^B = t_{ij}$. For the city freighter, a ratio α is given such that the city freighter travelling time on an arête (i, j) is $\alpha \times t_{ij}$ and $c_{ij}^S = \alpha \times t_{ij}$.

3.2 Model

Variables de décision :

$$x_{i,j}^k = \begin{cases} 1, & \text{si le véhicule } k \text{ emprunte l'arête } (i, j), \text{ si } k = 3 \text{ les deux véhicules empruntent cet arête} \\ 0, & \text{otherwise} \end{cases} \quad k \in \{1, \dots, 3\}, i \in V \setminus \{n+1\}, j \in V \setminus \{0\}$$

i et j ne sont pas définis sur la totalité de l'ensemble V puisque l'on sait déjà qu'aucun trajet n'arrivera sur le nœud 0 correspondant au dépôt de départ. De la même manière aucun trajet ne partira du nœud $n+1$ correspondant au dépôt d'arrivée.

u_i : capacité du petit véhicule en arrivant au nœud i $i \in V$

w_i^k : temps de passage du véhicule k au nœud i $k \in \{1, 2\}, i \in V$

$$\min z = \sum_{i \in V \setminus \{n+1\}, j \in V \setminus \{0\}} t_{ij} x_{i,j}^1 + \alpha t_{ij} x_{i,j}^2 - \alpha t_{ij} x_{i,j}^3$$

such that:

$$x_{i,j}^k \geq x_{i,j}^3 \quad \forall k \in \{1, 2\}, \forall i, j \in J \quad (1)$$

$$\sum_{j \in V \setminus \{i, n+1\}} x_{i,j}^k = \sum_{j \in V \setminus \{0, i\}} x_{j,i}^k \quad \forall k \in \{1, 2\}, \forall i \in J \cup P \quad (2)$$

$$\sum_{j \in V \setminus \{0\}} x_{0,j}^1 = \sum_{j \in V \setminus \{n+1\}} x_{j,n+1}^1 = 1 \quad (3)$$

$$x_{0,j}^1 = x_{0,j}^2 \quad \forall j \in V \setminus \{0\} \quad (4)$$

$$x_{j,n+1}^1 = x_{j,n+1}^2 \quad \forall j \in V \setminus \{n+1\} \quad (5)$$

$$x_{j,i}^1 + \sum_{l \in V \setminus \{j, n+1\}} x_{l,i}^2 \leq 1 \quad \forall i \in J, \forall j \in V \setminus \{n+1\} \quad (6)$$

$$x_{i,j}^1 + \sum_{l \in V \setminus \{0, j\}} x_{i,l}^2 \leq 1 \quad \forall i \in J, \forall j \in V \setminus \{0\} \quad (7)$$

$$\sum_{k \in \{1, 2\}} \sum_{j \in V \setminus \{0, i\}} x_{i,j}^k \geq 1 \quad \forall i \in J \quad (8)$$

$$\sum_{j \in V \setminus \{0, i\}} x_{i,j}^1 = 0 \quad \forall i \in J_S \quad (9)$$

$$u_j \leq u_i - q_i + (1 - x_{i,j}^2)Q \quad \forall i \in J, \forall j \in V \setminus \{0\} \quad (10)$$

$$u_j \leq Q + (1 - x_{i,j}^2)q_j \quad \forall i \in P, \forall j \in V \setminus \{0\} \quad (11)$$

$$a_i \leq w_i^k \leq b_i \quad \forall k \in \{1, 2\}, \forall i \in V \quad (12)$$

$$w_i^1 + s + t_{i,j} - w_j^1 \leq (1 - x_{i,j}^k) M_{i,j}^1 \quad \forall i \in V \setminus \{n+1\}, \forall j \in V \setminus \{1, i\} \quad (13)$$

$$w_i^2 + s + \alpha t_{i,j} - w_j^2 \leq (1 - x_{i,j}^k) M_{i,j}^2 \quad \forall i \in V \setminus \{n+1\}, \forall j \in V \setminus \{1, i\} \quad (14)$$

$$w_i^1 = w_i^2 \quad \forall i \in \{0, n+1\} \cup P \quad (15)$$

3.3 Comments

Description des contraintes :

- (1) : Lien entre $x_{i,j}^1$, $x_{i,j}^2$ et $x_{i,j}^3$. Lorsque $x_{i,j}^1$ et $x_{i,j}^2$ sont égales à 1, $x_{i,j}^3$ l'est aussi, sinon $x_{i,j}^3$ vaut 0. Cet ensemble de contraintes fonctionne puisque par défaut $x_{i,j}^3$ prends la valeur 1 car elle intervient dans la fonction objectif en minimisant le coût.
- (2) : Conservation des flots sur tous les nœuds excepté le dépôt.
- (3) : Le gros véhicule part une fois du dépôt et y revient une fois.
- (4) : Le gros véhicule et le petit véhicule partent ensemble du dépôt.
- (5) : Le gros véhicule et le petit véhicule reviennent ensemble au dépôt.
- (6) : Pour tous les nœuds qui ne sont pas des parkings, si le gros véhicule arrive sur l'un de ces nœuds on interdit au petit véhicule d'arriver sur ce même nœud en étant parti d'un nœud différent de celui duquel était parti le gros véhicule.
- (7) : Équivalent de la contrainte (6) pour les arêtes partant des nœuds.
- (8) : Satisfaction de la demande, chaque nœud doit être visité au moins par un véhicule.
- (9) : Le gros véhicule ne peut pas visiter les nœuds $\in J_S$.
- (10) : Le petit véhicule doit avoir une capacité résiduelle suffisante pour visiter un nœud.
- (11) : La capacité du petit véhicule est réinitialisée à sa valeur maximale Q après la visite d'un parking.
- (12) : La visite de chacun des nœuds doit se faire dans la fenêtre de temps correspondante à chacun de ces nœuds.
- (13) : Contrainte de temps du gros véhicule, à chaque nœud le véhicule met un temps de service s_i à fournir le client, il met également un temps $t_{i,j}$ pour aller du nœud i au nœud j . Cette contrainte est une linéarisation de l'équation $x_{i,j}^1(w_i^1 + s + t_{i,j} - w_j^1) \leq 0$ avec $M_{i,j}^1 = \text{Max}(b_i + s_i + t_{i,j} - a_j)$.
- (14) : Contrainte de temps du petit véhicule, équivalente à la contrainte (13), le petit véhicule met cependant $\alpha t_{i,j}$ à effectuer le trajet entre i et j . Avec donc $M_{i,j}^2 = \text{Max}(b_i + s_i + \alpha t_{i,j} - a_j)$.
- (15) : Le gros véhicule et le petit véhicule partent et reviennent en même temps au dépôt. Ils se rencontrent également au même moment aux parkings.

4 Valid inequalities

- $x_{j,i}^2 + \sum_{l \in V \setminus \{j, n+1\}} x_{l,i}^1 \leq 1 \quad \forall i \in J, \forall j \in V \setminus \{n+1\}$
- $x_{i,j}^2 + \sum_{l \in V \setminus \{0, j\}} x_{i,l}^1 \leq 1 \quad \forall i \in J, \forall j \in V \setminus \{0\}$

Ces deux contraintes sont présentes dans le modèle original uniquement du point de vue du gros véhicule. En ajoutant l'inégalité du point de vue du petit véhicule, on renforce la contrainte en aidant le solveur dans le cas où les variables ne sont pas encore fixées pour le premier véhicule.

Ajouter la contrainte (13) du papier (1) (contrainte de co-circuit) a été étudié pour renforcer les contraintes touchant aux parkings. Effectivement, ce sont ces derniers qui sont considérées comme des "coupes" du graphe (edge-cut set) en servant de point de réunion aux deux véhicules car ça implique qu'une des arêtes entrante ou sortant du parking rassemble les véhicules, donnant un nombre d'arêtes impair. Cependant, nous n'avons pas eu le temps d'implémenter correctement cette contrainte dans le modèle pour la tester.

5 Experimentations

Toutes les mesures ont été effectuées sur le même ordinateur avec le solveur Cbc. Les temps correspondent uniquement aux temps pris par la commande `solve` de JuMP.

Passage 1.0 \rightarrow 2.0 :

- Optimisation de la création des sous-ensembles nécessaires aux contraintes
- Suppression de l'interdiction du chemin dépôt vers dépôt
- Suppression contrainte de boucles (arêtes de i vers i)
- Suppression contrainte redondante interdisant les départs séparés aux parkings et la séparation hors parking
- Rassemblement des contraintes de début, de fin et d'arrivée aux parkings en une seule contrainte (15)

La motivation de cette deuxième version du modèle était de retirer un maximum de contraintes redondantes, certaines étant des reliquats de précédents essais lors de la création du modèle, on remarque cependant qu'à part sur quelques instances où le gain de temps est significatif (4,9 et 11), les autres sont équivalentes ou ralenties d'un facteur allant de 2 à 3 (sans compter l'instance 5)

Passage 2.0 \rightarrow 3.0 :

- Ajout de la contrainte interdisant les trajets dépôt \rightarrow dépôt
- On pose les contraintes empêchant les deux véhicules de se retrouver ou de se quitter aux nœuds qui ne sont pas des parkings (\implies rend la contrainte plus forte, cf section 4)

Enfin, pour arriver à la dernière version, nous avons ajouté sélectivement certaines contraintes supprimées jusqu'à atteindre ce qui nous semble être le meilleur compromis entre les temps obtenus des différentes instances.

n°	Nom	z	1.0	2.0	3.0
1	C1-2-8	4749.9	3.56s	3.47s	3.59s
2	C1-3-10	4351.5	10.04s	35.28s	7.86s
3	C1-3-12	4602.3	8.55s	39.28s	17.66s
4	C2-2-8	3987.2	5.50s	2.99s	2.36s
5	C2-3-10	4052.3	0.34s	13.61s	0.29s
6	C2-3-12	4892.1	37.12s	76.25s	43.96s
7	R1-2-8	Infaisable	N/A	N/A	N/A
8	R1-3-10	7956.8	11.26s	18.13s	16.21s
9	R1-3-12	7669.9	50.92s	34.13s	47.09s
10	R2-2-8	6165.3	2.60s	2.75s	2.48s
11	R2-3-10	5293.8	27.69s	11.25s	8.24s
12	R2-3-12	7601.3	63.55s	163.21	80.39s

5.1 Instance description

12 instances are provided with denominations `T_N_P.txt` where T denotes the instance type with respect to the geographical distribution of points (clustered (C) or random (R)), N denotes the number of customers and P denotes the number of parking areas in the problem. Make sure to read the `readme.txt` file for instances description.

The shared instance parameters are the small truck capacity, the speed ratio α , the length of the time horizon, a common time window width, and the service duration at each customer. Note that all customers are considered to have the same time window width, hence, only the opening time is given in instances. The default values to be taken are the following:

- $Q = 400$
- $\alpha = 2$
- $T = 32400sec.$
- Time window width $\delta = 7200sec.$
- Service duration $s_i = 300sec., \forall i \in J.$

5.2 Evaluation of modeling options

Pour expérimenter le modèle, plusieurs exécutions de la résolution ont été faites pour avoir une idée du temps de résolution, et ce, sur plusieurs instances. Ce qui nous a permis de voir la réaction du modèle selon ces dernières à cause de leurs caractéristiques respectives.

5.3 Instance and parameter analysis

5.3.1 Caractéristiques des instances

Au fur et à mesure des tests, on a pu remarquer des changements notables en fonction de nos contraintes et de l'ajout ou non de nos inégalités valides. Le changement est particulièrement important entre les versions 1.0 et 2.0 où la suppression de contraintes liées aux parkings (entre autres, mais notamment celle-ci) augmentent énormément les temps de résolution des instances contenant plus de deux parkings. Ceci s'explique par le fait qu'en cas de présence de seulement deux parkings, leur position dans la tournée est généralement très proche du début et de la fin, pour permettre aux véhicules de se rejoindre une fois leur tournée individuelle effectuée.

Les instances à distribution de points aléatoires (R) se montrent aussi être les moins stables face aux changements de modèles, avec des delta entre versions globalement plus élevés que dans le cas des instances groupées (C), cette instabilité plus prononcée est logique car les instances (C) disposent d'un chemin pour le petit camion généralement assez identifiable.

5.3.2 Modification des paramètres

Ici, on teste les différentes instances disponible en modifiant les paramètres un à un pour voir les variations de valeur objectif :

n°	Nom	$z_{default}$	$z_{Q=150}$	$z_{\alpha=3}$
1	C1-2-8	4749.9	4978.7	6186.3
2	C1-3-10	4351.5	5645.3	5599.4
3	C1-3-12	4602.3	5066.3	5884.1
4	C2-2-8	3987.2	Infaisable	5049.2
5	C2-3-10	4052.3	4421.5	5258.2
6	C2-3-12	4892.1	5955.4	5701.9
7	R1-2-8	Infaisable	Infaisable	Infaisable
8	R1-3-10	7956.8	8411	9964.4
9	R1-3-12	7669.9	8256.3	9733.3
10	R2-2-8	6165.3	6165.3	7736.9
11	R2-3-10	5293.8	5293.8	6559.6
12	R2-3-12	7601.3	7747.9	9794.8

Ici, on a peu fait varier Q , la capacité du petit camion, car n'ayant pas dupliqué les parkings, on voulait que les instances restent faisables. Passer à 150 a cependant suffi à modifier les objectifs obtenus, les sous tournées devenant trop grosses pour être effectués d'une traite par le petit camion, forçant à passer au parking pour les instances à trois parkings, ou passer la main au gros camion pour les nœuds communs aux deux camions.

À noter que les instances mettent significativement plus de temps à se résoudre avec une capacité resserrée, particulièrement sur les instances à trois parkings. Effectivement, il faut alors générer deux sous tournées puis revenir au gros camion, là où les instances à deux parkings correspondent à de petits changements, ou deviennent infaisables.

En faisant varier α , on a obtenu l'effet attendu avec une augmentation relativement petite de ce dernier : une augmentation du coût et pas de changement dans les solutions.

Ce changement a cependant eu l'effet inattendu d'augmenter grandement la vitesse de résolution des instances, dépassant souvent les meilleurs temps trouvés sur ces dernières. Ne connaissant pas les mécanismes internes à Cbc, on ne peut que supposer que l'augmentation du coût de déplacer le petit camion puisse permettre d'éliminer certains chemins plus facilement qu'avec les valeurs par défaut.

References

- [1] José Manuel Belenguer and Enrique Benavent and Antonio Martínez and Christian Prins and Caroline Prohon and Juan Guillermo Villegas, 2014, *A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots*, Transportation Science.