Dial-A-Ride Problem
Modélisation PLNE

Clara Gros
Pauline Vaillant
Clémence David
Camil Zarrouk
Marouane Abia

March 2020

Contents

1 Présentation du problème

1.1 Objectifs

Ce problème vise à trouver une solution optimale vis à vis de deux objectifs $c1$ et $c2$ avec $c1 \succcurlyeq c2$:

(c1) Affecter le plus de courses possibles aux chauffeurs
(c2) Minimiser le temps total de trajet pour les chauffeurs

On se retrouve avec un problème bi-objectif et l'on va se ramener à un problème mono-objectif (voir équation 0) en utilisant la méthode lexicographique sachant que le critère 1 est prioritaire sur le critère 2.

1.2 Contraintes

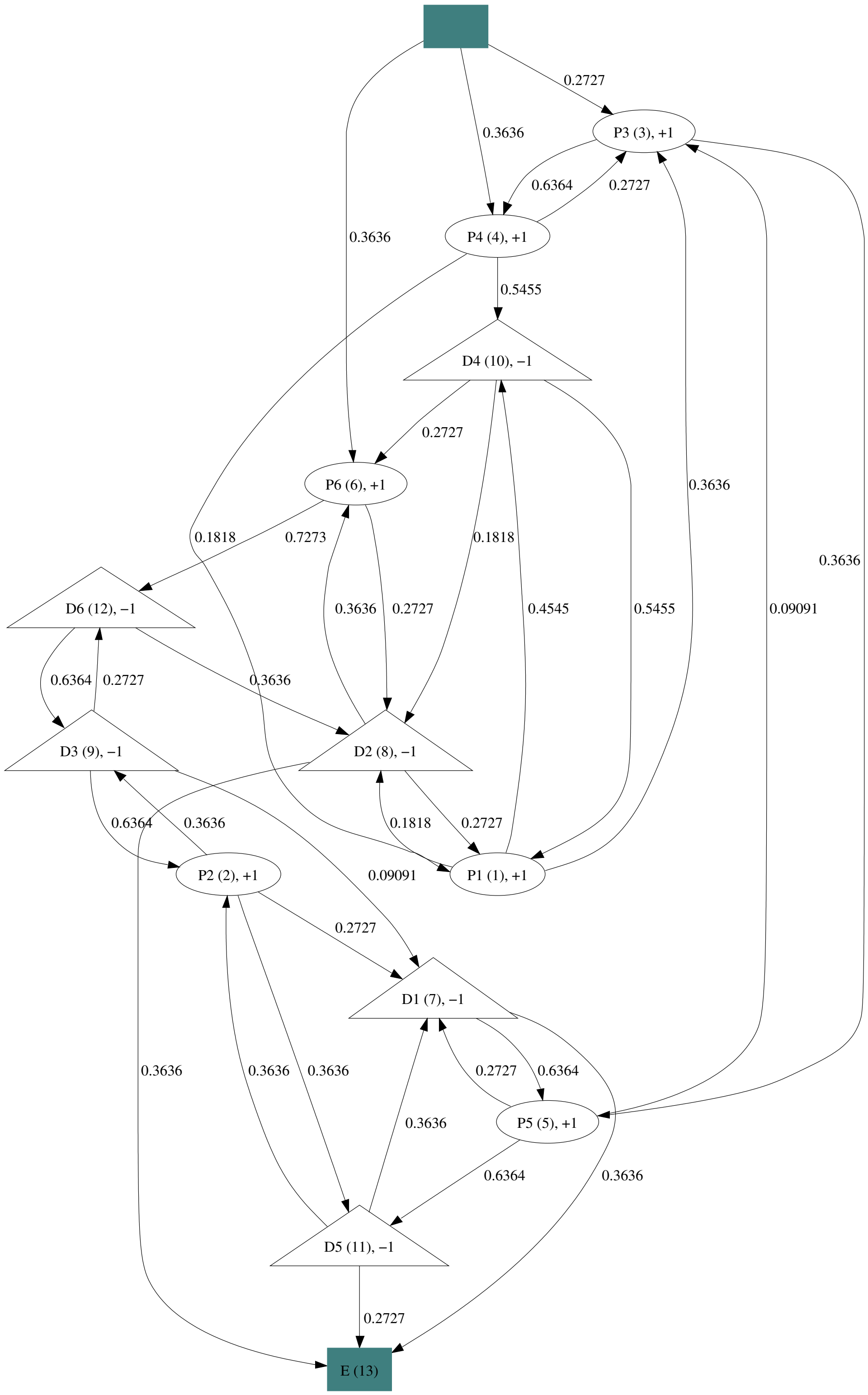
Ci-dessous une liste exhaustive des contraintes littérales imposées par le problème:

- Revenu maximal de chaque chauffeur
- Durée maximale de chaque course
- Capacité des véhicules des chauffeurs
- Fenêtres de temps pour chaque prise et dépose
- Début et fin de service de chaque chauffeur
- Temps de trajet entre le dépôt, les prises et les déposes

1.3 Illustration du problème

Voir page suivante.

The dial a ride problem (DARP)



2 Modèle mathématique

2.1 Notations

2.1.1 Modélisation par un graphe

Sur une journée, notons $n \in \mathbb{N}$ le nombre total de requêtes. Une requête (course) peut être faite par un ou plusieurs utilisateurs. Le DARP peut donc être défini sur un graphe orienté.

$$G = (V, E), \text{ où } V = P \cup D \cup \{0, 2n+1\}, P \in [\![1, n]\!], \text{ et } D \in [\![n+1, 2n]\!]$$

Les sous-ensembles P et D contiennent respectivement les nœuds de ramassage (pick up) et de dépôt (drop off), tandis que les nœuds 0 et $2n+1$ représentent les dépôts d'origine et de destination des véhicules de la flotte (confondus dans notre cas). À chaque utilisateur i est donc associé un nœud d'origine i et un nœud de destination $n+i$. L'ensemble des arcs est défini comme suit : $E = \{(i,j) : i = 0,\ j \in P$ ou $(i,j) \in P \cup D,\ i \neq j$ et $i \neq n+j$, ou $i \in D,\ j = 2n+1\}$.

Concrètement, une arête $(i,j),\ i = 0,\ j \in P$ traduit le fait qu'un bus qui part vide de l'entrepôt passe forcément en premier par une station de prise pour récupérer des gens. De même, une arête $(i,j), i \in D,\ j = 2n+1$ traduit le fait qu'un bus qui rentre à l'entrepôt en fin de tournée ne peut pas passer juste avant par une station de prise pour prendre des gens. Enfin, les arêtes de type $(i,j), (i,j) \in P \cup D,\ i \neq j$ et $i \neq n+j$ représentent les chemins que peut emprunter un bus au cours de sa tournée. La condition $i \neq j$ signifie qu'il n'y a pas d'arête bouclant sur un même noeud : un chemin faisable va d'une station à une autre bien distincte. Quant à la condition $i \neq n+j,\ \forall (i,j) \in P$, elle traduit le fait que, pour une requête i donnée, passer d'abord par la station de dépôt $n+i$ puis par la station de prise i n'a pas de sens.

2.1.2 Paramètres du problème

- M représente le nombre total de chauffeurs (véhicules) de la flotte.
- $\forall (i,j) \in V^2$, t_{ij} représente le temps de trajet entre les stations i et j.
- $\forall i \in V$, e_i représente la durée d'exécution à la station i (e.g le temps d'arrêt à la station pour embarquer ou débarquer des gens). On pose par convention $e_0 = e_{2n+1} = 0$.
- $\forall i \in P$, m_i représente le temps de trajet maximal de la course i.
- $\forall i \in V$, q_i représente le nombre de personnes chargées dans le véhicule au nœud i. Il va de soi que $q_0 = q_{2n+1} = 0$ car aucune personne ne monte ni ne descend à l'entrepôt des véhicules et $\forall i \in P$, $q_i = -q_{n+i}$ car toutes les personnes qui montent dans le véhicule à la station i (requête i) descendent à leur station d'arrivée $n+i$.

- $\forall i \in V$, r_i représente le coût de la course i. Ainsi, $\forall i \in V \backslash P$, $r_i = 0$ car le coût de la course est encaissé par le chauffeur dès la prise des passagers et le chauffeur ne gagne donc pas d'argent à l'entrepôt ni aux stations de dépose des clients.

- $\forall k \in M$, C_k représente la capacité maximale du véhicule k.

- $\forall k \in M$, (L_k, U_k) représentent respectivement la date minimale (maximale) de départ (d'arrivée) du véhicule k à l'entrepôt.

- $\forall i \in V$, (l_i, u_i) représente la fenêtre de temps à la station i.
  Les fenêtres de temps à l'entrepôt (stations 0 et $2n + 1$) représentent la date la plus tôt à laquelle le véhicule devrait partir de l'entrepôt pour servir les clients et la date la plus tard à laquelle il doit retourner au dépôt. Ainsi, $l_0 = min(min(l_i - t_{0i}), min(L_k))$ et $l_{2n+1} = min(min(l_i - t_{2n+1,i}), min(L_k))$ signifie que nous prenons la première heure de départ nécessaire comme heure de début de la fenêtre temporelle au dépôt, et de même, on prend $u_0 = max(max(u_j - t_{j,0}), max(U_k))$ et $u_{2n+1} = max(max(u_j - t_{j,2n+1}), max(U_k))$ comme étant le dernier moment de la fenêtre temporelle au dépôt.

- $\forall k \in M$, R_k représente le revenu maximal que le chauffeur k peut toucher en une tournée.

2.2 Variables de décision

- $\forall (i,j) \in V^2$, $\forall k \in M$, x_{ij}^k illustre le fait que le véhicule k se déplace ou non du nœud i au nœud j.

$$x_{ij}^k = \begin{cases} 1 \text{ si le chauffeur } k \text{ passe par } (i,j) \\ 0 \text{ sinon} \end{cases}$$

  Remarque : Si $(i,j) \notin E$, $x_{ij}^k = 0$.

- $\forall (i,k) \in V \times M$, u_i^k représente l'heure à laquelle le chauffeur k arrive à la station i.

- $\forall (i,k) \in P \times M$, m_i^k représente le temps de trajet total de la course i dans le véhicule k.

- $\forall (i,k) \in V \times M$, q_i^k représente la charge du véhicule k en partant de la station i.

2.3 Contraintes

Chaque demande est réalisée au plus une fois : seulement un véhicule peut arriver à une station de prise donnée et il n'y repart que par un seul chemin.

$$\sum_{k \in M} \sum_{j \in V} x_{ij}^k \leq 1 \ \forall i \in P \tag{1}$$

Un même véhicule qui entre dans un nœud quitte obligatoirement le nœud.

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{ji}^k = 0 \ \forall i \in P \cup D, \forall k \in M \tag{2}$$

Les contraintes 3 et 4 garantissent que l'itinéraire de chaque véhicule k commence au dépôt d'origine et se termine au même dépôt, si le véhicule fait au moins une course.

$$\sum_{j \in V} x_{0j}^k \leq 1 \ \forall k \in M \tag{3}$$

$$\sum_{i \in V} x_{i,2n+1}^k \leq 1 \ \forall k \in M \tag{4}$$

Les nœuds d'origine et de destination sont visités par le même véhicule.

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{n+i,j}^k = 0 \ \forall i \in P, \forall k \in M \tag{5}$$

Visit time La cohérence des variables de temps est assurée par la contrainte 6 et la contrainte 7 impose des fenêtres de temps définies par chaque utilisateur.

$$u_j^k \geq (u_i^k + e_i + t_{ij}) x_{ij}^k \ \forall (i,j) \in V^2, \forall k \in M \tag{6}$$

$$l_i \leq u_i^k \leq u_i \ \forall i \in P \cup D, \forall k \in M \tag{7}$$

Ride time L' égalité 8 définit le temps de parcours de chaque utilisateur, qui est limité par les contraintes 9.

$$m_i^k = u_{n+i}^k - (u_i^k + e_i) \ \forall i \in P, \forall k \in M \tag{8}$$

$$t_{i,n+i} \leq m_i^k \leq m_i \ \forall i \in P, \forall k \in M \tag{9}$$

Load constraints La cohérence des variables de charge est assurée par la contrainte 10 et la contrainte 11 impose des contraintes de capacités définies pour chaque véhicule.

$$q_j^k \geq (q_i^k + q_j) x_{ij}^k \ \forall (i,j) \in V^2, \forall k \in M \tag{10}$$

$$max(0, q_i) \leq q_i^k \leq min(C_k, C_k + q_i) \ \forall i \in V, \forall k \in M \tag{11}$$

Vehicles tour constraints Les contraintes 12 et 13 imposent respectivement une date de début et une date de fin de la tournée d'un véhicule.

$$u_0^k \geq L_k \ \forall k \in M \tag{12}$$

$$u_{2n+1}^k \leq U_k \ \forall k \in M \tag{13}$$

Cost constraint La contrainte 14 limite le revenu maximal de chaque chauffeur.

$$\sum_{j\in V}\sum_{i\in V} r_i x_{ij}^k \leq R_k \ , \forall k \in M \tag{14}$$

Avoid loop on one station :

$$\sum_{k\in M}\sum_{i\in V} x_{ii}^k = 0 \tag{15}$$

2.4 Fonction objectif

Pour établir la fonction multi objectif, de la forme

$$\max \sum_{k\in M}\sum_{(i,j)\in V^2} x_{ij}^k - \epsilon \sum_{k\in M}\sum_{(i,j)\in V^2} t_{ij}x_{ij}^k \tag{0}$$

il faut choisir la valeur du paramètre ϵ afin de garantir que les objectifs sont bien optimisés en priorisant le nombre de courses par rapport au temps de trajet. On choisit ϵ de sorte que $\epsilon \sum_{k\in M}\sum_{(i,j)\in V^2} t_{ij}x_{ij}^k \leq 1$. Pour se faire, on imagine le pire des cas où toutes les routes possibles seraient empruntées par un véhicule : c'est à dire le cas où le graphe est complet. Alors en prenant $\epsilon = \sum_{(i,j)\in V^2} t_{ij}$, on s'assure que la quantité précédente sera toujours inférieure à 1. En effet, on veut maximiser la quantité $n - \epsilon d$, avec n le nombre de courses et d la distance parcourue par l'ensemble des véhicules de la flotte. Si on note N le nombre de courses maximales et que l'on s'assure que $\epsilon d \leq 1$, alors on est sûr que pour une configuration où $n = N$, on aura $n - \epsilon d > N - 1$ et si $n \leq N - 1$, alors $n - \epsilon d \leq N - 1$. Cela montre bien que la meilleure configuration sera forcément une configuration où $n = N$. On est donc bien sûr de maximiser en priorité sur le nombre de courses. La fonction objectif devient :

$$\max \sum_{k\in M}\sum_{(i,j)\in V^2} x_{ij}^k - \epsilon \sum_{k\in M}\sum_{(i,j)\in V^2} t_{ij}x_{ij}^k$$

$$\text{with } \left\{ \ \epsilon = \textstyle\sum_{(i,j)\in V^2} t_{ij} \right.$$

2.5 Linéarisation des contraintes

La formulation des contraintes ci-dessus est non linéaire en raison des contraintes 6 et 10. En introduisant les constantes M_{ij} et Q_i^k, ces contraintes peuvent toutefois être linéarisées comme suit :

$$u_j^k \geq u_i^k + e_i + t_{ij} - M_{ij}(1 - x_{ij}^k) \ \forall (i,j) \in V^2, \forall k \in M \tag{6.1}$$

avec

$$M_{ij} = max(0, e_i + t_{ij} + u_i - l_j) \ \forall (i,j) \in V^2 \tag{6.2}$$

$$q_j^k \geq (q_i^k + q_j) - Q_i^k(1 - x_{ij}^k) \ \forall (i,j) \in V^2, \forall k \in M \tag{10.1}$$

avec

$$Q_i^k = min(C_k, C_k + q_i) \; \forall i \in V, \, \forall k \in M \tag{10.2}$$

3 Programme linéaire

$$\max \sum_{k\in M}\sum_{(i,j)\in V^2} x_{ij}^k - \epsilon \sum_{k\in M}\sum_{(i,j)\in V^2} t_{ij}x_{ij}^k$$

$$\text{s.t.} \begin{cases} \sum_{k\in M}\sum_{j\in V} x_{ij}^k \leq 1 & \forall i \in P \\ \sum_{j\in V} x_{ij}^k - \sum_{j\in V} x_{ji}^k = 0 & \forall i \in P\cup D,\, \forall k\in M \\ \sum_{j\in V} x_{0j}^k \leq 1 & \forall k \in M \\ \sum_{i\in V} x_{i,2n+1}^k \leq 1 & \forall k \in M \\ \sum_{j\in V} x_{ij}^k - \sum_{j\in V} x_{n+i,j}^k = 0 & \forall i \in P,\, \forall k \in M \\ u_j^k \geq u_i^k + e_i + t_{ij} - M_{ij}(1-x_{ij}^k) & \forall (i,j)\in V^2,\, \forall k\in M \\ l_i \leq u_i^k \leq u_i & \forall i \in P\cup D,\, \forall k \in M \\ m_i^k = u_{n+i}^k - (u_i^k + e_i) & \forall i\in P,\, \forall k\in M \\ t_{i,n+i} \leq m_i^k \leq m_i & \forall i \in P,\, \forall k\in M \\ q_j^k \geq (q_i^k + q_j) - Q_i^k(1-x_{ij}^k) & \forall (i,j)\in V^2,\, \forall k \in M \\ max(0,q_i) \leq q_i^k \leq min(C_k, C_k+q_i) & \forall i\in V,\, \forall k\in M \\ u_0^k \geq L_k & \forall k\in M \\ u_{2n+1}^k \leq U_k & \forall k \in M \\ \sum_{j\in V}\sum_{i\in V} r_i x_{ij}^k \leq R_k & \forall k\in M \\ \sum_{k\in M}\sum_{i\in V} x_{ii}^k = 0 & \\ x_{ij}^k \in \{0,1\} & \forall (i,j)\in V^2,\, \forall k\in M \end{cases}$$