

# OPTIMISATION DU TRANSPORT DE PERSONNES INTÉGRANT DES CONTRAINTES D'INCOMPATIBILITÉ ET DE PAUSES DES CHAUFFEURS

par

Anas AMADDAH

RAPPORT DE PROJET PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE  
SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION DE  
LA MAÎTRISE EN GENIE DE LA PRODUCTION AUTOMATISÉE  
M.ing

MONTREAL, LE 14 NOVEMBRE 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

©Tous droits réservés, Anas AMADDAH, 2020



©Tous droits réservés

Cette licence signifie qu'il est interdit de reproduire, d'enregistrer ou de diffuser en tout ou en partie, le présent document. Le lecteur qui désire imprimer ou conserver sur un autre media une partie importante de ce document, doit obligatoirement en demander l'autorisation à l'auteur.



Cette licence [Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/) signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette œuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'œuvre n'ait pas été modifié.

**PRÉSENTATION DU JURY**

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Marc PAQUET directeur de projet  
Département génie des systèmes à l'École de technologie supérieure

M. Mustapha Ouhimmou, membre du jury  
Département génie des systèmes à l'École de technologie supérieure



## **REMERCIEMENTS**

Mes remerciements vont tout d'abord à mon professeur, Monsieur Marc Paquet, qui m'a permis d'effectuer ce travail de maîtrise, qui a cru en ma capacité pour l'accomplir, et qui m'a accompagné tout le long de mon projet. Sa disponibilité à mon égard et son excellent support académique furent fondamentaux pour ma réussite.

Je souhaite aussi remercier Monsieur Romain Bonnifet, et Quentin Massoteau de l'entreprise FASTERCOM, m'ayant guidé au début du projet à définir le problème que nous devons résoudre.

J'exprime beaucoup de reconnaissance envers mes parents et ma famille pour leur support et leurs encouragements continus.

Enfin, je remercie mes amis pour leur soutien durant les moments difficiles.



# **OPTIMISATION DU TRANSPORT DE PERSONNES INTEGRANT DES CONTRAINTES D'INCOMPATIBILITE ET DE PAUSES DES CHAUFFEURS**

ANAS AMADDAH

## **RÉSUMÉ**

Ayant pour objectif de proposer un modèle d'optimisation de transport adapté pour les personnes, nous allons à travers ce rapport proposer une grande étude bibliographique qui nous permettra de bien comprendre le problème. Nous allons définir au début de notre revue de littérature ce qu'est un problème d'optimisation de transport en recherche opérationnelle de manière générale, avant de montrer les différences et les particularités de chacun de ses variantes. Nous allons ensuite mettre l'accent sur les contraintes qui constituent notre problème, à savoir les contraintes d'incompatibilité et la contrainte des pauses pour chauffeurs en étudiant des cas proches de notre problème. Nous parlerons aussi des différentes méthodes utilisées dans la résolution de ces problèmes. Ensuite, dans notre partie de méthodologie, nous commencerons par définir notre problème plus en détails, et voir la méthodologie suivie afin de construire notre modèle, en analysant de manière détaillée un modèle existant et en proposant des modifications afin de l'adapter à notre problème, nous décrirons ensuite la méthode utilisée pour la résolution de celui-ci ainsi que les outils que nous utiliserons pour notre résolution. Enfin, nous allons présenter le modèle que nous proposons, détaillant nos choix d'équations, de paramètres et de variables. Nous allons ensuite montrer et expliquer les résultats obtenus, ayant pu avoir une optimisation en respectant les contraintes. Nous finirons par une discussion, dans laquelle nous allons parler des résultats obtenus, des lacunes ou difficultés rencontrés et des recommandations possibles pour une amélioration du modèle dans un projet futur.





# **PASSENGER TRANSPORTATION OPTIMISATION WITH INCOMPATIBILITY AND DRIVER BREAKS CONSTRAINTS**

ANAS AMADDAH

## **ABSTRACT**

With the aim of proposing an optimization model of adapted transport for passenger, we are going through this report to propose a large bibliographic study that will allow us to fully understand the problem. At the beginning of our literature review, we will define what a transport optimization problem is in general, seen from operational research side of view, before showing the differences and peculiarities of each of its variants. We will then focus on the constraints that constitute our problem, namely the constraints of incompatibility and drivers breaks by studying cases from literature that have those interests as well. We will also talk about the different methods used in solving these problems. Then, in our methodology part, we will start by defining our problem with more details, and see the methodology followed in order to build our model, by analyzing in detail an existing model and proposing modifications in order to adapt our model to our problem, we will then describe the method used for the resolution of the studied model as well as the tools that we will use for the resolution of our model. Finally, we will present the model that we propose, detailing our choices of equations, parameters and variables. We will then show and explain the results obtained, having been able to have an optimization while respecting the constraints. We will end with a discussion, in which we will talk about the results obtained, the gaps or difficulties encountered and possible recommendations for an improvement of the model in a future project.



## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
CHAPITRE 1    REVUE DE LITTERATURE.....	3
1.1    Introduction.....	3
1.2    Les problèmes de routage de véhicule (VRP).....	3
1.3    Les variantes des problèmes VRP.....	6
1.3.1    Les problèmes de cueillette et de livraison.....	6
1.3.2    Les problèmes de transport sur demande (DARP) .....	8
1.4    Domaines d'application .....	9
1.4.1    Domaine industriel.....	9
1.4.2    E-commerce .....	10
1.4.3    Co-voiturage et transport en commun.....	10
1.4.4    Domaine de la santé.....	11
1.5    Contraintes du problème .....	12
1.5.1    Incompatibilités.....	12
1.5.2    Contraintes des chauffeurs.....	16
1.6    Méthodes utilisées en résolution des problèmes VRP .....	18
1.6.1    Les méthodes exactes.....	19
1.6.1.1    La méthode de séparation et évaluation : (Branch and Bound) .	19
1.6.1.2    La méthode des coupes planes.....	20
1.6.1.3    La méthode Branch and Cut .....	20
1.6.1.4    La méthode de la génération de colonnes .....	21
1.6.2    Méthodes heuristiques .....	21
1.6.3    Métaheuristiques .....	22
1.6.3.1    Le recuit simulé (simulated annealing).....	22
1.6.3.2    La recherche avec liste de tabous (Tabu Search).....	24
1.6.3.3    Les algorithmes génétiques.....	24
1.6.3.4    Les algorithmes de colonies de fourmis.....	26
1.7    Conclusion .....	28
CHAPITRE 2    METHODOLOGIE.....	29
2.1    Description du problème .....	29
2.2    Modélisation .....	31
2.3    Méthode de résolution du problème .....	38
2.3.1    Génération de colonnes.....	38
2.3.2    Heuristique de recherche dans le voisinage.....	40
2.4    Outils de résolutions .....	41
2.5    Conclusion .....	42
CHAPITRE 3    PRESENTATION DE LA SOLUTION, ET ANALYSE DU MODELE.....	43
3.1    Modèle proposé.....	43

3.2	Jeux de données .....	47
3.3	Présentation des solutions obtenues.....	48
3.4	Discussion.....	53
3.4.1	Analyse de la solution obtenue .....	53
3.4.2	Obstacles et recommandations.....	54
3.5	Conclusion .....	55
CONCLUSION.....		57
ANNEXE I	Tableaux de données.....	59
ANNEXE II	Code LINGO.....	61
ANNEXE III	DEBUG Solveur Lingo.....	65
BIBLIOGRAPHIE.....		67

## LISTE DES TABLEAUX

	Page
Tableau 1: Valeurs situation 1 .....	48
Tableau 2: Valeurs situation 2 .....	48
Tableau 3: Valeurs retournées dans la situation 1 .....	49
Tableau 4: Trajet du véhicule 2 .....	50
Tableau 5: Valeurs retournées situation 2.....	51
Tableau 6: Trajet du véhicule 2 .....	52
Tableau 7: Trajet du véhicule 3 .....	52
Tableau 8: Temps de trajets entre différents nœud et nombre de clients par nœud.....	59
Tableau 9: Coûts des trajets .....	59
Tableau 10: Tableau des incompatibilités.....	60
Tableau 11: Véhicules disponibles et capacité des véhicules.....	60



## LISTE DES FIGURES

	Page
Figure 1: Exemple de trajet optimisés dans un VRP (Caceres-Cruz, Arias, Guimarans, Riera, & Juan, 2014).....	5
Figure 2: Fonctionnement de l'algorithme de recuit simulé (Sidi Mohamed Douiri,2011).....	23
Figure 3: Fonctionnement d'un algorithme génétique (Sidi Mohamed Douiri,2011).....	25
Figure 4: Expérience du pont double (Sidi Mohamed Douiri,2011) .....	26
Figure 5: Description des trajets possibles des véhicules (cas de notre modèle).....	30
Figure 6: Description des trajets possibles des véhicules .....	31
Figure 7: Solution situation 1.....	65
Figure 8: Solution situation 2.....	65





## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

**DARP** : Dial-a-Ride Problems (Problèmes de transport adapté)

**VRP** : Vehicle Routing Problems (Problèmes de routage de véhicule)

**RVRP** : Rich Vehicle Routing Problems (Problèmes de routage de véhicule riches)

**CVRP** : Capacitated Vehicle Routing Problems (Problèmes de routage de véhicule capacitifs)

**VRPTW** : Vehicle Routing Problems with Time Windows (Problèmes de routage de véhicule avec fenêtres de temps)

**AVRP** : Asymmetric Vehicle Routing Problems (Problèmes de routage de véhicule asymétriques)

**HVRP** : Heterogenous Vehicle Routing Problems (Problèmes de routage de véhicule Hétérogènes)

**OVRP** : Open Vehicle Routing Problems (Problèmes de routage de véhicule ouverts)

**SDVRP** : Split-Delivery Vehicle Routing Problems (Problèmes de routage de véhicule avec livraison partagée)

**PDP** : Pick-up and Delivery Problems (Problèmes de cueillette et livraison)

**MCDARP** : Multi-Criteria Dial-a-Ride Problems (Problèmes de transport adapté avec multicritères)

**GA** : Genetic Algorithm (Algorithme génétique)

**TS** : Tabu Search (recherche Tabou)



## INTRODUCTION

Lors des dernières décennies, le domaine du transport a connu une croissance notable. Cette croissance a engendré plusieurs défis et problèmes. Ces derniers se manifestent principalement par des soucis économiques et environnementaux. En effet, en raison de l'augmentation du nombre de véhicules qui circulent simultanément sur la route, cela a donné naissance à de nouvelles problématiques. D'une part, les soucis environnementaux liés à la forte émission de gaz à effet de serre et à la pollution. D'une autre part, les soucis économiques liés aux pertes dues aux retards et aux déplacements non nécessaires.

Au fil de ce rapport, nous porterons notre attention sur les différentes problématiques liées à l'optimisation des flottes de véhicules et plus précisément celles dédiées au transport de personnes. Pour ce faire, une synthèse des différents problèmes d'optimisation des flottes de véhicules a été réalisée, ainsi qu'une présentation des méthodes de modélisation et de résolution des modèles d'optimisation. Finalement, nous proposerons un modèle d'optimisation pouvant répondre à cette problématique.

Le cas traité ici est un type particulier de transport, celui du transport adapté des personnes. La particularité de celui-ci par rapport aux autres types de transport de personnes réside dans le fait qu'il s'adapte aux demandes et aux contraintes de ses clients.

On vise à travers ce projet de proposer un modèle d'optimisation de transport adapté, celui-ci en tenant compte de deux types de contraintes. La première est une contrainte d'incompatibilité entre les clients, la deuxième est une contrainte qui gère les pauses déjeuner des chauffeurs de véhicules.

Notre rapport se compose de trois chapitres. Le premier chapitre est dédié à la revue de littérature, dans celui-ci, nous mettrons en situation notre problème, en commençant par définir les problèmes de routages de véhicules et leurs domaines d'application et voir les différentes variantes de ce problème. Nous présenterons aussi ce qui a été traité dans la littérature en rapport avec les deux contraintes que nous souhaitons traiter. Nous finirons par citer les différentes méthodes de résolution utilisées.

Dans le deuxième chapitre, nous donnerons une présentation détaillée de notre problème et la méthodologie avec laquelle nous traiterons ce dernier, à travers l'étude d'un exemple vu dans la littérature et à partir duquel nous avons pu bâtir notre modèle. Nous verrons le modèle proposé ainsi que les méthodes de résolution utilisées.

Dans le troisième et dernier chapitre, nous allons présenter le modèle que nous avons proposé, en justifiant le choix de nos paramètres, variables et équations. Nous essayerons de valider le modèle à travers un exemple en donnant un scénario. Nous finirons le chapitre par une discussion sur ce que nous avons pu réaliser et les améliorations que l'on pourrait rapporter sur notre modèle.

## **CHAPITRE 1**

### **REVUE DE LITTERATURE**

#### **1.1 Introduction**

Dans cette partie, nous verrons plusieurs articles traitant l'optimisation des tournées de véhicules, partant au début d'un contexte général, expliquant les types de problèmes d'optimisation de transport qui existent avec les multiples différences et particularité de chaque type. Nous aborderons par la suite les problèmes de transport sur demande DARP (Dial-a-Ride Problems) liés à notre sujet puis à des cas particuliers traitant les contraintes que l'on veut modéliser en proposant notre modèle, en analysant les modèles proposés par les articles cités. Nous finirons par décrire les différentes méthodes utilisés pour la résolution de ces types de problèmes.

#### **1.2 Les problèmes de routage de véhicule (VRP)**

Le problème que nous allons résoudre à travers notre projet est un problème de routage de véhicule (VRP) ou « Vehicule Routing Problem ». Ce genre de problème a été créé afin de gérer et optimiser les tournées de véhicules. Les recherches traitant ce genre de problème sont nombreuses. L'article de Laporte (Laporte, 2009) donne une vaste revue de littérature de recherches faites dans les VRP en général, commençant par citer l'article de Dantzig et Ramser (Dantzig & Ramser, 1959) qui est considéré comme la première publication traitant des problèmes VRP. Pour commencer notre revue de littérature, on cite l'article de Caceres et al. (Caceres-Cruz, Arias, Guimarans, Riera, & Juan, 2014), qui, dans celui-ci, font une enquête globale sur les différents types de VRP existants, et les particularités de chacun de ces types, citant qu'il existe plusieurs variantes de problèmes VRP, qui varient selon les paramètres et les contraintes du problème. Dans cet article on nous donne aussi la notion de RVRP ou « Rich VRP » et ce sont des VRP qui prennent de multiples types de contraintes. On nous décrit aussi dans cet article plusieurs nominations et abréviations mises en valeurs dans d'autres études. Nous notons que dans le même article, on nous fait une présentation d'un modèle mathématique simple en ne prenant en considération dans sa fonction-objectif que le coût des

distances parcourus par tous les véhicules, avec les contraintes basiques telles que le respect du nombre de véhicules, de leur capacité, et de leur disponibilité. L'article nous propose le modèle le plus simple de VRP, qui est un problème de routage de véhicules avec capacité CVRP « Capacitated Vehicle Routing Problem » et qui est considéré la base de tout modèle VRP que l'on verra par la suite. Le modèle général est décrit comme suit :

On considère un ensemble  $\Omega$  de  $n+1$  nœuds, avec le nœud 0 qui définit le dépôt et les  $n$  autres nœuds des nœuds de collecte ou de destination de ce qui est transporté avec  $\Omega^* = \Omega - \{0\}$ , soit  $q_i > 0$  la demande en chaque point  $i$ , avec  $q_0 = 0$ . On prend aussi  $A = \{(i, j) : i, j \in \Omega; i < j\}$  l'ensemble des arcs reliant tous les nœuds de  $\Omega$ , chaque couple  $(i, j)$  est défini par un coût de transport entre  $i$  et  $j$   $c_{ij} > 0$ . On considère une flotte de véhicules composé de  $V$  véhicules ( $V \geq 1$ ) avec une capacité  $Q > \max \{q_i : i \in \Omega\}$ . La suite de la formulation proposée dans l'article se base sur la formulation de (Toth et Vigo, 2002). Ils proposent ici plusieurs types de véhicules  $M = \{1, \dots, m\}$  et chaque type  $k \in M$  a un  $m_k$  nombre de places disponibles, défini par une capacité  $Q_k$ . On donne  $x_{ij}^k$  la variable qui définit si l'arc  $(i, j)$  est utilisé par un véhicule de type  $k$ , et la variable  $y_{ij}^k$  indique le nombre d'unités à bord du véhicule qui vient de visiter le point  $i$  et allant vers  $j$ .

La fonction-objectif du modèle de Toth et Vigo (2002) est définie comme suit :

$$\min \sum_{k \in M} \sum_{\substack{i, j \in \Omega \\ i \neq j}} c_{ij}^k x_{ij}^k \quad (1)$$

Sujet aux contraintes suivantes :

$$\sum_{j \in \Omega^*} x_{0j}^k = \sum_{i \in \Omega^*} x_{i0}^k \quad \forall k \in M \quad (2)$$

$$\sum_{k \in M} \sum_{i \in \Omega^*} x_{ij}^k = 1 \quad \forall j \in \Omega^* \quad (3)$$

$$\sum_{j \in \Omega} x_{iu}^k = \sum_{j \in \Omega} x_{uj}^k \quad \forall u \in \Omega^*, \forall k \in M \quad (4)$$

$$\sum_{j \in \Omega^*} x_{0j}^k < m_k \quad \forall k \in M \quad (5)$$

$$\sum_{i \in \Omega} y_{ij}^k - \sum_{i \in \Omega} y_{ji}^k = q_j \sum_{i \in \Omega} x_{ij}^k \quad \forall j \in \Omega^*, \forall k \in M \quad (6)$$

$$0 \leq q_i x_{ij}^k \leq y_{ij}^k \leq (Q_k - q_i) x_{ij}^k \quad \forall i, j \in \Omega, i \neq j, \forall k \in M \quad (7)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in \Omega, i \neq j, \forall k \in M \quad (8)$$

La fonction-objectif minimise la somme des coûts des trajets effectués, la contrainte (2) impose que le nombre de véhicules qui sortent du dépôt est le même que celui qui y revient, les équations (3) et (4) veulent dire que chaque nœud est visité une seule fois, et qui si un point est visité, celui-ci doit être le point de départ vers le point suivant. La contrainte (5) impose que le nombre de véhicules utilisé ne dépasse pas le nombre de véhicules disponibles dans la flotte. L'équation (6) implique que le nombre d'unités à bord du véhicule arrivant au point  $j$  ( $y_{ij}^k$ ) moins la demande en ce point-là, est égale au nombre d'unités à bord du véhicule  $n$  sortant de ce dernier. La contrainte (7) limite la quantité  $y_{ij}^k$  à bord du véhicule, qui est supérieure ou égale à la demande au nœud suivant  $q_j$ , et ne dépasse pas la capacité maximale du véhicule de type  $k$ .

L'article propose le schéma suivant qui schématise un problème VRP donnant comme solution des routes en se basant sur le principe des nœuds comme vu dans l'exemple précédent :

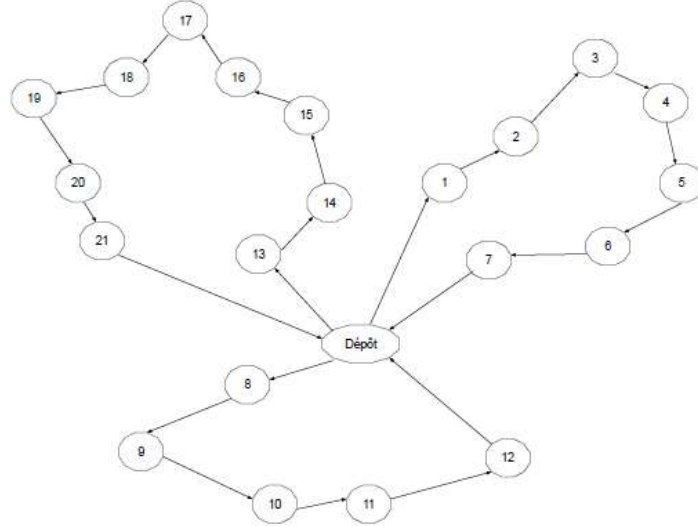


Figure 1: Exemple de trajet optimisés dans un VRP (Caceres-Cruz, Arias, Guimarans, Riera, & Juan, 2014)

### 1.3 Les variantes des problèmes VRP

Il existe plusieurs variantes de problèmes VRP, ceux-ci varient selon les paramètres et les contraintes dont on dispose, par exemple, il existe une variante de problèmes de routage de véhicule avec gestion des fenêtres de temps ou VRPTW (« Time Windows ») que l'on verra par la suite avec un exemple de problème tenant cette contrainte en considération. Une autre variante existante s'appelle « the green VRP » ou les VRP verts comme dans les articles de (Erdoğan & Miller-Hooks, 2012) et (Kopfer, Schonberger, & Kopfer, 2014), les VRP asymétriques (AVRP) où la matrice des coûts entre les nœuds est asymétrique, les VRP à flotte hétérogènes (HVRP), ce dernier est largement traité, puisqu'il traite des problèmes plus ou moins réalistes en considérant une flotte de véhicule avec différents types de véhicules. Les problèmes avec multi-dépôts, dans ceux-ci, les routes peuvent avoir différents points de départ ou d'arrivées avec différents dépôts considérés. Les VRP ouverts « Open VRP » (OVRP), ce sont des problèmes dans lesquels les routes peuvent finir sur n'importe quel nœud. Les problèmes de livraison fractionnée ou « split-delivery » (SDVRP) sont des problèmes dans lesquelles les nœuds de livraison peuvent être servis ou visités par plusieurs véhicules dans des routes différentes. Les problèmes stochastiques sont des problèmes dans lesquels on y ajoute des paramètres stochastiques en introduisant l'aspect aléatoire qui donne plus de réalisme au problème. L'article de Young et al. (Yong, Boudouh, Grunder, & Deyun, 2018) donne un exemple de résolution de ce type de problèmes dans le domaine de la santé. Plusieurs de ces types de variantes peuvent être regroupées dans des problèmes dits hybrides, ces derniers peuvent avoir les caractéristiques de plusieurs variantes de problèmes de routage de véhicules.

#### 1.3.1 Les problèmes de cueillette et de livraison

Un des problèmes qui les plus utilisés et qui nous intéressent le plus est le problème de cueillette et de livraison « pick-up and delivery problems » (PDP), ce sont des problèmes où on gère à la fois la cueillette et la livraison de biens ou services à partir ou vers des points différents de l'espace.

Il existe plusieurs types de PDPs et ce sont des problèmes en rapport avec le transport de biens ou de personnes à partir d'une origine vers une destination tout deux données.



Dans leur article, Bebeglia et al. (Bebeglia, Cordeau, & Laporte, 2010) commencent par nous rappeler les trois grands groupes des PDPs, le premier rassemblant des problèmes « plusieurs à plusieurs » (« many-to-many problems »), c'est un mode où chaque point peut être considéré à la fois origine ou destination. Le deuxième groupe contient des problèmes de « un à plusieurs à un » (« one to many to one ») et ce sont généralement les problèmes où l'on a un point de production par exemple, plusieurs dépôts dans d'endroits différents et puis à partir de chaque dépôt un point de destination. Et puis en fin le troisième qui regroupe les problèmes « un à un » (« one to one problems ») où chaque élément transporté (bien ou personne) a une origine et une destination connues. Cet article rappelle aussi qu'un problème peut être statique comme il peut être dynamique, statique étant un problème où l'on connaît déjà tous les paramètres avant que les chemins soient définis, ce qui donne un résultat optimal statique. Dans un problème dynamique, certains paramètres peuvent changer en cours de route, ce qui fait que le résultat doit changer et se mettre à jour à chaque fois qu'un paramètre est changé, on trouve ce type de problème plus dans le transport de personnes. L'article lui-même nous présente le cadre général du concept d'un PDP dynamique de type un à un, tel que le problème est décrit comme suit : à chaque instant  $t$ , le véhicule est soit en train de servir un client, d'attendre un client ou de se diriger vers un client pour le servir. Et que dans le processus, deux types de décision rentrent en action, attendre ou aller lorsqu'un client est servi, et accepter ou refuser lorsqu'une nouvelle demande est reçue. Ce qui rend le concept dit « dynamique » c'est cette capacité à mettre à jour la solution dépendamment de ces décisions prises au fur et à mesure. L'article aussi va présenter les deux approches basiques de solutions différentes, la première consiste à résoudre le problème comme si c'était un problème statique à chaque fois qu'une décision est prise et que la solution doit être modifiée, la deuxième approche est de faire l'algorithme de résolution du problème statique une seule fois au début, et puis grâce à des méthodes heuristiques. Il présente aussi d'autres concepts tels que la performance d'un algorithme, l'anticipation de nouvelles demandes, le détournement de véhicules (la capacité de gérer les détours des véhicules pendant une action), et le degré de dynamisme d'un problème.

### 1.3.2 Les problèmes de transport sur demande (DARP)

En ce qui concerne notre projet, il s'agit plus spécifiquement d'un problème de personnes pouvant être classifiés en plusieurs types, où on essaierait plus spécialement de gérer à long terme les incompatibilités entre les patients et les véhicules, les patients et les chauffeurs et aussi les patients avec d'autres patients. Ces types de problèmes sont des cas particuliers qui rentrent dans la catégorie connue sous le nom de DARP (« Dial-a-Ride Problems ») ou problèmes de transport sur demande comme défini dans la thèse de Ichoua (Ichoua, 2001). L'article de Ho et al. (Ho et al., 2018) fait lui aussi une enquête récente sur les développements récents dans les DARP, dans cet article, les auteurs présentent plusieurs études récentes sur des problèmes DARP et montrent que dernièrement, de plus en plus d'études se font dans cette direction là en différenciant les problèmes dans différentes catégories. L'une des catégories spécifiées est la catégorie « Manpower requirements » ou « besoin en mains d'œuvre », dans celle-ci, on trouve comme référence l'article de (Parragh, Cordeau, Doerner, & Hartl, 2012). Dans cet article, on décrit un modèle qui gère différents types de véhicules et différents types de patients, et l'affectation des patients se fera selon le besoin en siège, en chaise roulante ou en civière, la fonction-objectif ici reste une fonction de coût, mais ici, on utilise cinq variables binaire de décision, une qui définit si un arc est traversé par un véhicule, une pour définir si un chauffeur est défini pour un véhicule, une qui définit si un préposé est défini pour un véhicule le matin, et une autre s'il est affecté pour l'après-midi, et une autre pour définir s'il y a lieu d'une pause dans un vecteur  $i$ . Cet article gère les pauses de midi et les changements de véhicule pour les chauffeurs, et gère en même temps les compatibilités des patients avec les véhicules et les chauffeurs, on peut aussi dire qu'il gère les incompatibilités entre les patients vus qu'on prend en compte pour chaque patient un nombre de personnes l'accompagnant, ce qui donnera parfois une impossibilité de transporter 2 patients si le nombre total de personnes dépasse la capacité du véhicule.

L'article de Lehoue et al. (Lehoue, Masson, Parragh, Peton, & Tricoire, 2014) traite lui aussi le problème DARP lié au transport de patients handicapés mais inclut la notion de l'analyse multicritères à un DARP et le problème devient un MCDARP (« Multi-Criteria DARP »), le principe ici d'un MCDARP est que dans la fonction-objectif du modèle, nous avons plusieurs

termes ceci étant des critères les critères pris en considération sont les suivants : le temps de trajet excédentaire maximum, la moyenne des temps excédentaire par parcours, le temps d'attente moyen, la moyenne des temps des parcours et le taux d'occupation moyen des véhicules. Dans la fonction-objectif, chaque critère est multiplié par un coefficient, l'ajout de ces coefficients rend le problème plus flexible et ainsi la solution dépendra de ce que l'entreprise de transport choisira de prioriser.

Pour mieux se familiariser avec les problèmes DARP, nous avons aussi pu voir plusieurs autres articles qui traitent de différents problèmes DARP d'un point de vue différent, comme dans l'article de Mauri et al. (Mauri, Antonio, & Lorena, 2009), qui présente un modèle d'optimisation portant sur la qualité du service du patient (prenant en considération des contraintes comme le temps de voyage maximal et le temps d'attente maximal pour un patient). L'article de Ta Yin et al. (T.-Y. Hu, Zheng, & Liao, 2017) lui aussi prend en compte ladite qualité du service, en incluant aussi le facteur de vitesse. L'article de Liu et al. (Liu, Luo, & Lim, 2015) présente un modèle réaliste qui différencie les types de besoins des clients, et les véhicules associés, là on inclue aussi la contrainte de la durée de voyage max pour un patient. Et puis en fin dans la fonction-objectif de l'article de Paolo Detti et al. (Detti, Detti, Papalini, & Lara, 2017), on représente la qualité du service par un coût (en multipliant les temps d'attentes par un coefficient qu'on peut varier selon l'importance donnée à la qualité du service).

## **1.4 Domaines d'application**

Comme évoqué dans la partie précédente, plusieurs domaines ont été sujet de recherches pour l'optimisation des flux de transport en utilisant les méthodes de recherche opérationnelle. Nous allons citer dans cette partie des exemples d'articles qui ont proposé des modèles d'optimisation dans des domaines différents.

### **1.4.1 Domaine industriel**

Si on prend l'exemple de l'industrie, cette dernière connaît aujourd'hui une grande concurrence, ce qui pousse les entreprises à optimiser au maximum tous les aspects qui rentrent dans leur chaîne logistique, afin de réduire les différents coûts, maximiser le profit et ainsi renforcer leur compétitivité. Ces dernières années, grâce au développement des outils informatiques et de la

recherche opérationnelle, plusieurs méthodes mathématiques d'aide à la décision commencent à être utilisées pour la résolution de ce que l'on appelle des problèmes d'organisation des flottes de véhicules en cueillette et livraison des biens ou des services.

L'optimisation des tournées de véhicules a connu ses débuts dans le domaine industriel, avec l'article de Dantzig et Ramster (Dantzig & Ramser, 1959). Madankumar et Rajendran (Madankumar & Rajendran, 2018) ont donné un exemple de modélisation d'une chaîne logistique dans une industrie de production de semi-conducteurs. Dans leur article, ils proposent deux modèles en programmation linéaires mixtes en utilisant dans un premier lieu les contraintes basiques nécessaires à la minimisation des coûts de l'opération de transport, mais dans le deuxième modèle, ils intègrent la notion des prix des carburants qui n'est pas la même entre les stations, ce qui donne plus de modification quant à la fonction-objectif aussi où il faudrait alors minimiser la somme des coûts de transport plus celle du ravitaillement en carburant.

#### **1.4.2 E-commerce**

Les entreprises de l'E-commerce ont eux aussi profité de recherches visant à optimiser leurs flux de livraison et ainsi augmenter leurs bénéfices. Dans l'article de Zheng et al. (Zheng, Liu, Gu, & Bai, 2017), le problème étudié est un problème dynamique simple, où l'on essaye d'optimiser le trajet des véhicules faisant des livraisons avec l'introduction de nouveaux clients et ainsi de nouveaux points de livraison. L'exemple propose 10 véhicules servant chacun 10 destinations différentes, après avoir optimisés les trajets une première fois, on essaye d'inclure 10 nouvelles destinations, l'idée ici est d'inclure pour chaque véhicule le client qui aura le moins d'impact sur le coût total de son voyage, mais aussi l'ordre dans lequel le véhicule va livrer ses clients en incluant la nouvelle destination.

#### **1.4.3 Co-voiturage et transport en commun**

Pour des fins économiques et environnementales, la réduction de l'utilisation des véhicules conventionnels est une priorité dans les pays les plus développés, ce qui a poussé les chercheurs

dans ce domaine à développer des algorithmes permettant d'optimiser les transports en commun ou le transport de personnes en général. Dans la même direction, nous voyons également récemment le développement de véhicules autonomes ou de navettes autonomes pouvant ainsi être utilisés dans le domaine du transport en commun.

Une autre application des modèles de VRP qui est plus récente est le co-voiturage, qui est aussi un concept qui peut être optimisé en le considérant comme un cas particulier de problème de cueillette et de livraison (PDP), ce dernier ayant comme paramètres pour chacun des clients, une intervalle de temps à laquelle il fera le départ de son origine et une intervalle dans laquelle il devra être à sa destination, ce qui donne plus de contraintes au problème normal, un domaine qui a connu ces dernières années un grand développement et fait l'objet de plusieurs recherches, Mahmoudi et Xuesong (Mahmoudi & Xuesong, 2016) ont fait l'étude dans leur article en se basant sur un exemple simple de 6 nœuds, et fait l'expérimentation avec six scénarios différents. Toujours dans le co-voiturage, Fagnant et Kockelman (Fagnant & Kockelman, 2018) ont fait une étude sur l'impact de la mise en service des véhicules autonomes pour les voyages de tous les jours et sur l'impact qui peuvent avoir sur l'utilisations des véhicules conventionnels. Dans leur article, Zihan et al. (Zihan, Ying, Mahmassani, & Shuang, 2017) essayent de donner une solution de co-voiturage optimal, en analysant la structure routière de la ville étudié (l'exemple ici de la ville de Chicago), après un balayage de la ville, celle-ci est redéfinie par des nœuds, des liens et des zones de trafic, pour à la fin obtenir des trajets de navettes optimaux (ou regroupements) pour des potentiels co-voitureurs.

#### **1.4.4     Domaine de la santé**

Un autre domaine où l'optimisation du transport deviens de plus en plus utilisé est le domaine de la santé. Comme on le sait tous, le transport des patients doit être fait de la manière la plus fluide et optimale possible, avec le grand risque que peut poser un retard d'ambulance, ou d'une équipe médicale en général. Différents chercheurs dans le domaine ont essayé d'analyser ces opérations là et ont proposer des modèles d'optimisation.

On trouve notamment dans la littérature plusieurs exemples qui traitent l'optimisation des tournées pour les soins à domicile. Un premier exemple est donné par Ran et al. (Ran, Xiaolan,

Augusto, & Rodriguez, 2013) qui présentent dans leur article plusieurs algorithmes dont deux en programmation linéaire mixte simples, et deux autres avec des méthodes heuristiques, la première s'agit d'un algorithme génétique (GA), et la deuxième d'une recherche avec liste tabous (TS). Le deuxième exemple s'agit à peu près du même problème, mais cette fois-ci, Young et al. (Yong et al., 2018) vont utiliser un modèle stochastique, ne prenant pas le temps de service d'un client ou la durée des trajets comme des valeurs connues (constantes) mais plutôt aléatoires avec une distribution normale, pour se rapprocher de la réalité. Ce changement a fait que nous avons une fonction-objectif constituée de deux parties, une partie fixe et une partie dite stochastique.

## **1.5 Contraintes du problème**

Après avoir vu des généralités sur les VRP, les PDP et les DARP, nous verrons maintenant ce qui a été réalisé par certains chercheurs du domaine et qui est en liaison avec les contraintes de notre problème, qui sont les contraintes d'incompatibilité et celles des chauffeurs.

### **1.5.1 Incompatibilités**

Les incompatibilités sont les incompatibilités entre des clients ou des biens qui sont à bord du véhicule, en mettant par exemple des conditions qui ne laisseraient pas un véhicule servir un client si celui-ci est incompatible avec un autre qui est déjà à bord du véhicule. Bien que la littérature concernant ce type de contraintes soit limitée, nous avons pu trouver un exemple qui traite à peu près le même problème que l'on veut traiter, mais ce dernier en appliquant la notion de « compatibilité » où l'on vérifie la compatibilité et non l'incompatibilité des personnes à bord du véhicule. C'est l'exemple donné dans l'article de Ramachandran (Ramachandran, 2009), dans ce dernier, on gère un problème de compatibilité entre deux types de personnes à transporter, ici c'est des employés femmes et hommes. Dans le modèle, on donne en premier lieu la fonction-objectif et les contraintes de faisabilité des trajets, puis en arrivant aux contraintes gérant la compatibilité, on le fait de deux manières différentes. La première est une gestion d'incompatibilité complète, où on ne fait que séparer les deux types, n'ayant jamais

deux employés de sexes différents en même temps dans un même véhicule. Ce qui est intéressant c'est la deuxième proposition où on donne un scénario où l'on a une condition qui permettrait d'avoir les deux types dans le même véhicule, on appelle ça dans l'article une exclusion conditionnelle.

La revue de littérature concernant les incompatibilités reste toujours limitée et pas trop abordée par les différents chercheurs dans le domaine comme cité au début de l'article de Hu et al. (Z.-H. Hu, Sheu, Zhao, & Lu, 2015), où on traite quand même un problème d'incompatibilité en VRPSD dans le milieu de l'industrie de la restauration en Chine. Ici, il s'agit de ramasser des produits de différentes locations, en faisant attention aux incompatibilités de certains produits par rapport à d'autres lors du ramassage, les produits dont on parle dans l'article sont justement des produits de vaisselle, où il ne faudrait pas avoir de la vaisselle désinfectée et utilisée dans un même véhicule. Dans l'article, nous avons un dépôt et des clients, chaque client a une demande en vaisselle désinfectée et de la vaisselle utilisée qu'il faudrait collecter de chez ce client. Le modèle a deux étapes, la première étant de trouver les routes faisables avec les paramètres réels que l'on a avant le départ. Ce qui se passera dans cette phase est que les demandes de collecte de vaisselle utilisée ne seront juste pas satisfaites si le véhicule contient toujours de la vaisselle désinfectée à bord. On utilise un ensemble de routes  $R = (R_1, R_2 \dots R_M)$  pour les routes prises par chacun des véhicules et un ensemble  $R_k = (R_{k,1}, R_{k,2}, \dots, R_{k,nk})$  avec  $R_{k,1}$  le premier clients visité dans la route  $K$ . L'objectif est de trouver un ensemble de routes faisables en essayant de minimiser les coûts de transportation, les contraintes utilisées lors de cette phase sont juste des contraintes de quantité à bord du véhicule pour que celles-ci ne dépassent pas la capacité  $Q$  de ce dernier. Dans la deuxième étape on utilise des variables stochastiques pour formuler les paramètres des demandes de vaisselle utilisée à ramasser, celles-ci seront modélisés ainsi en raison des contraintes d'incompatibilité de ces dernières et l'incertitude de servir ces demandes. On inclut ici aussi un paramètre de qualité de service  $z^S$ , qui est évalué par le ratio des clients qui seront visités plus qu'une fois pour servir les demandes de collecte et de livraison, un deuxième ratio est pris en considération qui va être le ratio d'incompatibilité  $z^P$ , et puis un troisième qui sera un ratio de coûts, qui comprend deux parties, le coût des routes obtenu dans la première étape, et celui qui prend en considération

les routes prises  $R^*$  qui n'ont pas pu satisfaire toutes les demandes de collecte en premier lieu. La fonction-objectif est une fonction de minimisation de coût qui inclut ces trois critères. Le modèle est présenté comme suit :

**Ensembles :**

$V = \{0, 1, 2, \dots, n\}$  Ensemble des points  $i$  avec  $i=0$  le dépôt (point de départ des véhicules)

On note  $V^C$  l'ensemble  $V$  sans le dépôt

$A = \{(i, j) : i, j \in V ; i \neq j\}$  : ensemble des arcs  $(i, j)$

$K = \{0, 1, 2, \dots, M\}$  Ensemble des véhicules de la flotte

$R = \{R_1, R_2, \dots, R_M\}$  est l'ensemble des routes faites par tous les véhicules,

**Paramètres :**

$m$  : Nombre de véhicules de la flotte

$Q$  : Capacité des véhicules

$P_i$  : Demande en vaisselle propre chez un client  $i$

$D_i$  : Collecte de vaisselle usée de chez un client  $i$

Avec :  $P_i \geq 0, D_i \geq 0$  et  $P_i + D_i > 0$

$C(R_{k,l}, R_{k,l+1})$  : Cout du trajet pris par le véhicule  $k$  entre les points  $l$  et  $l+1$

**Éléments de sortie du modèle :**

$R_k = \{R_{k,1}, R_{k,2}, \dots, R_{k,nk}\}$  : représente le trajet fait un véhicule  $k$ , tel que  $R_{k,1}$  représente le 1er client et  $nk$  c'est le nombre de clients visités par ce véhicule.

La notation  $i \in R_k$  veut dire que le client  $i$  est servi dans la route  $k$  (ou par le véhicule  $k$ ).

Le problème est résolu en deux parties sous raison qu'il est possible de visiter un clients  $i$  deux fois, une pour la livraison et une pour la demande. Dans la première, on fait notre fonction-objectif de manière simple sur la faisabilité des routes sous la condition qu'on ne dépasse pas la capacité  $Q$  à bord du véhicule et qu'on ne mette pas ensemble la vaisselle propre et la vaisselle usée.

La fonction-objectif est la suivante :



$$\min_{R \in R} z^c(R) = \sum_{k \in K} \sum_{l=0}^{n_k} C(R_{k,l}, R_{k,l+1}) \quad (1)$$

Ici dans la fonction-objectif (1) on minimise les coûts en choisissant les trajets entre  $l$  et  $l+1$  les moins coûteux.

Pour nos contraintes ici, elles sont les suivantes :

$$\sum_{i \in R_K} P_i \leq Q \quad (2)$$

$$\sum_{i \in R_K} D_i \leq Q \quad (3)$$

$$\sum_{i \in R_K} P_i - \sum_{i \in R_K} D_i \leq Q \quad (4)$$

Les contraintes (2), (3) et (4) définissent les routes qui seront faisable dans une route  $k$  où l'on ne peut pas remplir le véhicule d'une demande  $P$  supérieur à la capacité  $Q$ , ou d'une livraison  $D$  supérieure à la capacité  $Q$ , et on ne peut pas avoir les 2 dans un même véhicule dans un certain point  $i$ .

Le modèle que l'on vient de présenter fait que plusieurs demandes de collecte de vaisselle usées ne seront pas satisfaites. Pour que le modèle satisfait ces demandes, on fera une deuxième phase où il y aura la collecte de la vaisselle usée de chez les clients non satisfaits. On inclut alors un  $z^s$  comme indicateur de qualité de service, celui-ci est évalué par le ratio du nombre de clients visité 2 fois par rapport au nombre total de client.

Dans l'article on note ces collectes par  $(\bar{p}_i)$ , et c'est une donnée stochastique qui suit une distribution statistique.

- Soit  $U_i^D$  la quantité de vaisselle propre restante sur le véhicule à l'arrivée chez le  $i^{\text{ème}}$  client
- Soit  $U_i^P$  la quantité de vaisselle usée collectée
- Le facteur  $\rho_i$  est le degré d'incompatibilité chez le client  $i$
- $T_{i,i+1}$  est le temps du trajet entre le client  $i$  et  $i+1$
- Soit  $\bar{\rho}_i$  le seuil d'incompatibilité pour un véhicule tel que  $\rho_i \leq \bar{\rho}_i$

- L'incompatibilité totale est juste la somme de toutes les incompatibilités qu'on a eues chez chaque client, et est modélisé sur l'équation (5)

$$z^P = \rho = \int_t \rho_i dt = \sum_i \rho_i T_{i,i+1} \quad (5)$$

A la fin du modèle on présente la fonction-objectif (6), qui serait la minimisation d'un  $z$  global, qui inclut à la fois le  $z^S$ , le  $z^P$  et le  $z^C$  vu dans la première partie.

$$\min_{R \in R, R^* \in R^*} z = (z^C(R, R^*), z^S(R, R^*), z^P(R, R^*)) \quad (6)$$

### Remarques :

Dans cet article, l'incompatibilité est gérée de la sorte car dans le problème, nous avons le droit de visiter un client plusieurs fois, ce qui n'est probablement pas possible dans la situation que nous aurons à gérer en transport de personnes sauf si par exemple, nous avons plusieurs clients dans un même endroit et que certains sont incompatibles à d'autres qui sont déjà à bord du véhicule.

Dans cet article, on donne un facteur de contamination, en autorisation, selon ce facteur, d'avoir de la vaisselle propre et usée dans un même véhicule pour optimiser les trajets encore plus, sans pour autant contaminer la vaisselle propre en respectant le seuil donné :

$$\rho_i = \frac{\min\{U_i^D, U_i^P\}}{\max\{U_i^D, U_i^P\}} \left/ \left( 2 - \frac{U_i^D + U_i^P}{Q} \right) \right.,$$

C'est une relation que l'on trouve plus tard dans l'article (on appelle cela le facteur de contamination). Si on arrive à quantifier ce facteur d'incompatibilité, l'approche de la fonction-objectif (6) peut être reprise dans une situation de transport de personne, où on essaierait de satisfaire le maximum de demandes avec un coût minimal, ça nous rappelle un peu l'approche multicritères que l'on a vu sur l'article de (Lehuede et al., 2014), c'est une fonction qui prend en considération les coûts, le taux de revisite d'un même client et le taux d'incompatibilité chez les clients qui n'ont pas été satisfaits qu'après une deuxième visite.

### 1.5.2 Contraintes des chauffeurs

Ce qu'on compte faire aussi dans notre problème c'est de modéliser les contraintes pour chauffeur, afin qu'on arrive à long terme à pouvoir gérer les chauffeurs séparément des

véhicules, puisque dans la plupart du temps, les chauffeurs sont souvent liés à des véhicules et on parle rarement de compétences de chauffeurs, de notion de pauses des chauffeurs, ou de limite d'heures de travail de ces derniers. On propose ici un exemple de modèle qui gère quelques contraintes des chauffeurs, à savoir les pauses de midi, et les heures de travail max par journée qu'un chauffeur peut avoir.

Dans le modèle de (Parragh et al., 2012), on utilise deux ensembles principaux, un ensemble  $V$  contenant tous les points de nos destinations, et un ensemble  $A$  qui est l'ensemble des arcs  $(i, j)$  de toutes les liaisons possibles entre les différentes destinations, pour chaque arc  $(i, j)$ , nous avons des variable  $c_{i,j}$  et  $t_{i,j}$  qui y sont liées définissant le cout du trajet et la durée de celui-ci. Nous avons aussi deux autres ensembles  $P$  pour les points de cueillette (« pickup ») et  $D$  pour les points de destination, avec  $P = \{1, \dots, n\}$  et  $D = \{n + 1, \dots, 2n\}$ .

Pour le transport d'un patient il existe 3 modes de transport  $\mathbf{r}$ , mis dans l'ensemble  $R$  (Ho et al., 2018), ces modes varieront selon la demande du patient. Soit  $q_i^r$  la charge d'un patient  $i$  demandant le mode de transport  $r$ , pour les passagers souhaitant faire le trajet sur des banc normaux on aura  $q_i^1 = 1$ , si le patient a besoin d'une civière  $q_i^2 = 1$  et si le patient a besoin d'une chaise roulante  $q_i^3 = 1$ , chaque patient pourrait avoir besoin d'un accompagnateur d'où le  $q_i^0 = 1$ . En plus de ça, ce problème propose aussi la possibilité d'avoir des médecins accompagnateurs qui pourront accompagner le patient, une variable binaire  $a_i$  est introduite pour designer si un patient a besoin d'un accompagnateur ou pas, en cas de besoin,  $a_i = 1$ .

Pour chaque patient, nous avons un intervalle de temps  $[e_i, l_i]$  qui définissent le temps de départ minimal et le temps d'arrivée maximal à la destination, pour chaque patient on définit aussi une durée de trajet maximale  $L$ .

Le problème dispose aussi d'un ensemble  $K$  de véhicules spécialisés, on associe chaque véhicule  $k$  à une constante  $C^{r,k}$ , nous disposons ici de deux types de véhicules, des véhicules de type 1, avec 6 bancs normaux pour patients, un banc pour le personnel et une chaise roulante, le type 2 contient lui deux bancs pour le personnel, une chaise roulante, une civière et un banc normal pour patients. Comme contrainte, les patients qui ont besoins de chaises roulantes ne

peuvent pas utiliser les bancs normaux ou la civière, les autres patients peuvent utiliser la civière en cas de disponibilité de celle-ci et que les places assises sont complètes.

Les chauffeurs ont un maximum de 8,5 heures de travail par jour, avec  $H = 30$  minutes de pause à prendre dans un intervalle  $[e_H, l_H]$ , la pause déjeuner peut être prise dans chacun des points  $i$ . Certain chauffeurs  $m_d$  ( $m_d < m$ ) peuvent en cas où le besoin en chauffeurs est atteint, travailler en tant qu'accompagnateurs. Le nombre d'accompagnateurs disponible est  $m_a$ , ces derniers ne peuvent travailler qu'une demi-journée par jour, s'ils travaillent le matin dans un véhicule, ils doivent être déposés au dépôt  $j = 2n + 1$  dans une intervalle de temps limité  $[e_{2n+1}, l_{2n+1}]$ , un accompagnateur qui travaille l'après-midi devrais être pris du dépôt  $2n+1$  dans cette même intervalle de temps.

Sans le modèle, on considère aussi un ensemble  $u \in \{0, \dots, m_d\}$  le nombre de chauffeurs qui pourraient travailler comme accompagnateurs additionnels. La variable  $B_i^k$  défini le début du service du véhicule  $k$  dans le point  $i$  et  $Q_i^{r,k}$  la charge du véhicule  $k$  au point  $i$  respectant le mode  $r$ . La variable  $W_H^k$  représente le temps d'attente avant la pause déjeuner du véhicule  $k$ .

## 1.6 Méthodes utilisées en résolution des problèmes VRP

Il existe plusieurs méthodes de résolution de problèmes, chacune de ces méthodes a ses propres particularités et donne des résultats pouvant ne pas être parfaitement les mêmes, le temps de calcul est aussi un paramètre qui change lorsqu'on change de méthode de résolution. Nous allons présenter ici les différentes méthodes de résolution qui existent et leurs particularités. Ces méthodes peuvent être classifiées en trois grands groupes, les méthodes exactes, les méthodes heuristiques et les méthodes méta-heuristiques. Nous donnerons ici une description globale de ces méthodes en détaillant pour certains les algorithmes de résolution.

### **1.6.1 Les méthodes exactes**

Les méthodes de résolution exactes sont des méthodes qui arrivent à résoudre un problème de recherche opérationnelle en donnant la solution optimale exacte et absolue, celles-ci ont pour contraintes le temps de calcul qui pourrait être plus long dépendamment de la complexité du problème. On cite ici quelques exemples de méthodes exactes dans les sections suivantes.

#### **1.6.1.1 La méthode de séparation et évaluation : (Branch and Bound)**

Dans cette méthode, on essaye de trouver une solution en suivant une arborescence par séparation et évaluation des différentes solutions possibles. Tout le processus de résolution peut être représenté par un arbre avec des nœuds et des feuilles. La résolution par cette méthode se fait en se basant sur les trois axes suivants, l'évaluation, la séparation et la stratégie de parcours.

- **L'évaluation**

L'évaluation est le point qui permet de réduire le champ de recherche des solutions. Puisque les solutions sont représentées par des branches d'arbres, on explore chacun des nœuds se trouvant dans l'arborescence en commençant des niveaux plus hauts. On évalue la solution trouvée dans chacun des nœuds et on la compare à la solution optimale déjà trouvée (un coût minimal ou un bénéfice maximal), si la solution évaluée dans un nœud est moins optimale que la solution optimale déjà trouvée, on ne considère pas l'évaluation des branches à niveau plus bas de ce nœud.

- **La séparation**

Le problème est séparé en plusieurs sous-problèmes, on essaye alors de résoudre chacun de ces sous-problèmes, puis on essaye de garder chacune des meilleures solutions trouvées. Un arbre de solutions réduit est ainsi reconstruit en ne gardant que ces solutions. On appelle l'ensemble des nœuds de cet arbre l'ensemble des nœuds actifs.

- **La stratégie du parcours :**

Trois stratégies sont utilisées afin de trouver la solution optimale dans l'arbre de solution. La première consiste à aller chercher la solution dans les sommets des branches qui ont fait le

moins de séparations, la méthode s'appelle « la largeur d'abord » (Sidi Mohamed Douiri, 2011) et est la moins efficace. La deuxième stratégie va cependant chercher la solution au sommets les plus profonds, c'est une stratégie qui a plus de chance de mener vers la solution optimale, on appelle ça la stratégie du « plus profond d'abord ». La troisième solution s'intitule « le meilleur d'abord », dans cette dernière, on cherche la solution dans les sous-problèmes ayant la meilleure borne.

### 1.6.1.2 La méthode des coupes planes

C'est une méthode qui a été développée par (Schrijver, 2000), elle peut être utilisée pour la résolution des problèmes d'optimisation combinatoire se formulant sous la forme d'un programme linéaire (POC) :

$$\min \{c^T x : Ax \geq b, x \in R^n\}$$

Quand le problème d'optimisation combinatoire (POC) est complexe, celui-ci pourrait utiliser un énorme espace mémoire et rendre la résolution avec un solveur de programmation linéaire plus compliquée. Ce qu'on fait alors est qu'on diminue dans un premier temps quelques-unes des contraintes de sorte à relaxer le problème et avoir des solutions qui violent certaines contraintes initiales du problème, on l'appelle alors le problème relaxé (POCR). La résolution du problème se fait ainsi en commençant par un problème relaxé, puis en y rajoutant à chaque fois une des contraintes précédemment non tenues en compte dans l'itération précédente, chacune de ces contraintes ajoutées s'appelle alors une coupe plane (d'où le nom de la méthode).

### 1.6.1.3 La méthode Branch and Cut

Cette méthode est une combinaison des deux méthodes citées juste avant et elle est utilisée pour résoudre les programmes linéaires en nombres entiers. Dans celle-ci, on commence par la relaxation du problème, en ayant cette première solution du problème relaxé, on y applique la méthode des coupes planes. Si la solution obtenue n'est pas entière, le problème va alors être divisé en sous-problèmes, et là, on applique la résolution suivant la méthode du Branch and Bound, en évaluant les solutions des différents problèmes.

#### 1.6.1.4 La méthode de la génération de colonnes

Dans cette méthode de résolution, on n'utilise pas forcément toutes les variables du programme linéaire pour résoudre le problème, on crée alors à partir du problème principal un autre problème réduit qui contient un ensemble réduit de variables. Le problème initial s'appelle le problème maître, alors que le problème réduit s'appelle problème restreint. La résolution du problème restreint ne donne forcément pas la solution optimale, mais fournira des solutions optimales de variables duales  $v_i$  liées aux contraintes, ces variables nous permettront d'obtenir les colonnes à rajouter. Pour savoir si une colonne améliore la solution (par réduction de coût), on calcule une valeur appelé coût réduit, avec une formule :

$$\bar{c}_i = c_i - \sum_{j=1}^n a_{ij} v_j$$

afin d'atteindre la solution optimale, on rajoute au problème des variables aidant à améliorer la solution. Un sous-problème est alors créé, celui de la recherche de la variable (ou colonne) qui nous donnera la meilleure solution.

#### 1.6.2 Méthodes heuristiques

Les méthodes heuristiques proposent généralement des algorithmes permettant de trouver des solutions plus rapidement d'un problème d'optimisation combinatoire. La solution n'est forcément pas la solution optimale exacte mais elle est approximative. Pour la résolution de plusieurs problèmes, la solution approximative donnée par une méthode heuristique peut être utilisée pour faciliter et accélérer le processus de résolution du problème d'une méthode exacte. Il existe 2 catégories de problèmes heuristiques, Des méthodes constructives, et ce sont des méthodes dans lesquels, à partir d'une solution initiale, on rajoute des éléments et ainsi trouver une solution optimale. La deuxième catégorie de méthodes sont appelées des méthodes de fouilles locales, dans lesquelles, à partir d'une solution non optimale, on cherche de chercher des solutions plus intéressantes dans son voisinage.

### 1.6.3 Métaheuristiques

Les métaheuristiques ont été conçus afin de compenser les lacunes des méthodes heuristiques et exactes. Par rapport aux méthodes heuristiques, la solution proposée par les métaheuristiques est une solution qui est plus optimale et plus proche de la solution exacte, et en même temps, c'est une méthode qui prend moins de calcul pour la résolution, et ainsi nécessite moins d'espace mémoire, ce qui simplifie la résolution des problèmes beaucoup plus complexes. Le rapport entre la qualité du résultat et le temps de résolution s'avère plus intéressant que celui donné par les autres types de résolution. Ces méthodes de résolution sont généralement inspirées de phénomènes naturels en biologie ou éthologiques. Il existe deux types de métaheuristiques, nous avons des méthodes qui essaient d'itérativement chercher une meilleure solution à partir de solutions aux alentours et ce type de méthodes sont nommées méthodes de recherche locale ou méthodes de trajectoire, les méthodes les plus utilisées de cette catégorie sont la recherche avec liste de tabous (Tabou search) et le recuit simulé (Simulated Annealing). Le deuxième type de métaheuristique fait une analogie avec des populations, tel les algorithmes génétiques et les algorithmes de colonies de fourmis. Nous allons alors expliquer le principe de fonctionnement de quelques-unes de ces méthodes

#### 1.6.3.1 Le recuit simulé (simulated annealing)

La méthode tient ses origines de la mécanique statistique, elle vient à la base du processus de recuit des métaux, elle est basée sur l'analogie du phénomène du refroidissement progressif des métaux chauffés pour avoir une cristallisation à un meilleur état. Le principe de la méthode est d'essayer de parcourir tout l'espace de solution une à une de manière itérative. À partir d'une solution générée aléatoirement, avec une « Énergie initiale »  $E_0$ , et une « Température initiale »  $T_0$  élevée. On effectue un changement sur la solution après chaque itération, ce qui varie l'énergie du système créant un  $\Delta E$ .  $\Delta E$  doit être négative pour que la nouvelle solution soit acceptée puisque selon l'analogie du recuit simulé, cette solution permet de diminuer l'énergie et ainsi améliorer la qualité de cristallisation du métal. Sinon, si  $\Delta E$  est positif, il existe une probabilité  $P$  pour que la solution soit acceptée qui est calculée comme suit :



$$P(E, T) = \exp \left( -\frac{\Delta E}{T} \right)$$

On compare la probabilité  $P$  à un nombre  $\varepsilon \in [0,1]$ , la solution est acceptée si  $P \leq \varepsilon$ .

Pour expliquer le fonctionnement de l'algorithme, on propose le schéma suivant :

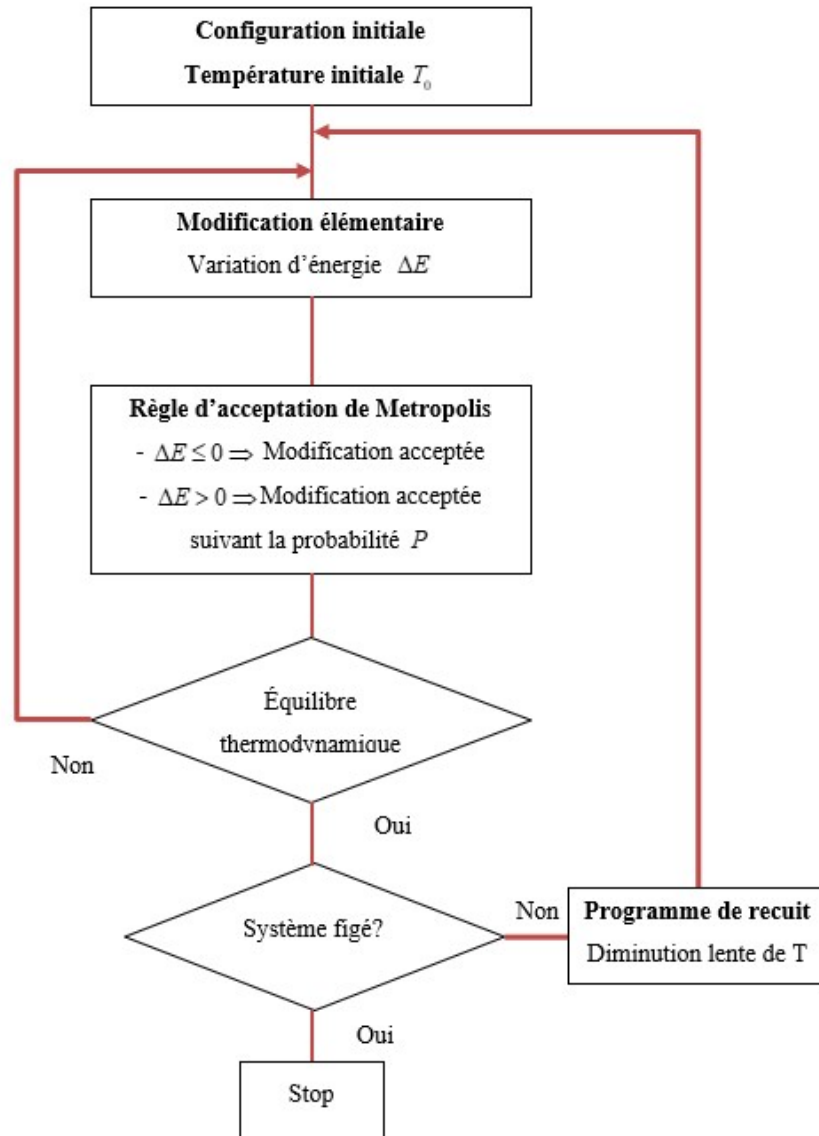


Figure 2: Fonctionnement de l'algorithme de recuit simulé (Sidi Mohamed Douiri, 2011)

### 1.6.3.2 La recherche avec liste de tabous (Tabu Search)

Comme la méthode de recuit simulé, la recherche Tabou est aussi une méthode qui, à partir d'une solution initiale, essaye de trouver des ensembles de solutions optimales dans le voisinage. Elle est connue pour son efficacité face aux problèmes difficiles. La particularité de cette méthode est le fait qu'elle évite de se faire piéger dans la répétition d'un cycle ou dans un minimum local. C'est une technique qui a été initiée par Glover et Laguna (Glover & Laguna, 1998), et Hansen (Hansen, 1986).

Le principe de la méthode est comme suit : on prend d'abord une solution initiale  $S$  appartenant à un ensemble de solutions local  $S$ , on génère des sous-ensembles de solution  $N(s)$  appartenant au voisinage  $S$ . Nous avons une fonction d'évaluation qui va nous aider à retenir la solution qui améliore la valeur de  $f$ , choisie de  $N(s)$ .

Parfois, des solutions n'améliorant pas la solution peuvent être acceptées. D'où la mise en œuvre d'une liste de déplacements tabous dans le voisinage (Tabu list)  $T$  avec les  $k$  dernières solutions visitées. Le choix d'une nouvelle solution se fait de sorte à éviter de réutiliser une solution déjà présente dans la liste, t une fois la liste pleine, l'ajout d'une nouvelle solution écrasera la solution la plus ancienne de la liste. L'itération va continuer à se faire jusqu'à ce qu'un critère d'arrêt soit satisfait, on peut mettre comme critère d'arrêt un nombre maximal d'itération ou un temps de calcul maximal. Une fois la liste faite, on prendra la valeur optimale comme meilleure solution donnée par cette méthode.

### 1.6.3.3 Les algorithmes génétiques

Les algorithmes génétiques sont des algorithmes d'optimisation fondées à partir d l'analogie faite avec les concepts de la génétique en sciences naturelle. Ces concepts ont été initialement adaptés à l'optimisation grâce à (Holland, 1992) et aussi David Goldberg (Goldberg, 1989). Même en optimisation, les termes qu'on utilise sont les mêmes utilisés en génétique, on garde notamment le lexique tel « individu » pour définir une solution potentielle, « population » pour définir un ensemble de solutions, « génotype » pour une représentation de la solution, ou

« gène » pour une partie du génotype. On prend aussi pleins de termes comme « parent », « enfant », « croisement », « mutation », « génération » ...

Le principe de la méthode est le suivant : on commence par définir une population de solutions potentielles initiale (Chromosomes) choisie aléatoirement. On évalue ces chromosomes, puis à partir de cette évaluation, on procède à la création d'un nouvel ensemble de solutions potentielles (ou nouvelle population) en utilisant les opérateurs évolutionnels tels que la sélection, le croisement et la mutation. Ce qui se passe c'est que certaines solutions (individus) vont être supprimées à travers plusieurs sélections, ou reproduites jusqu'à avoir à la fin une nouvelle population avec les individus survivants (les mieux adaptés, ou en termes d'optimisation, les meilleures solutions de la génération). Ce même cycle sera effectué plusieurs fois jusqu'à avoir une solution satisfaisante.

On peut schématiser l'algorithme de la manière suivante :

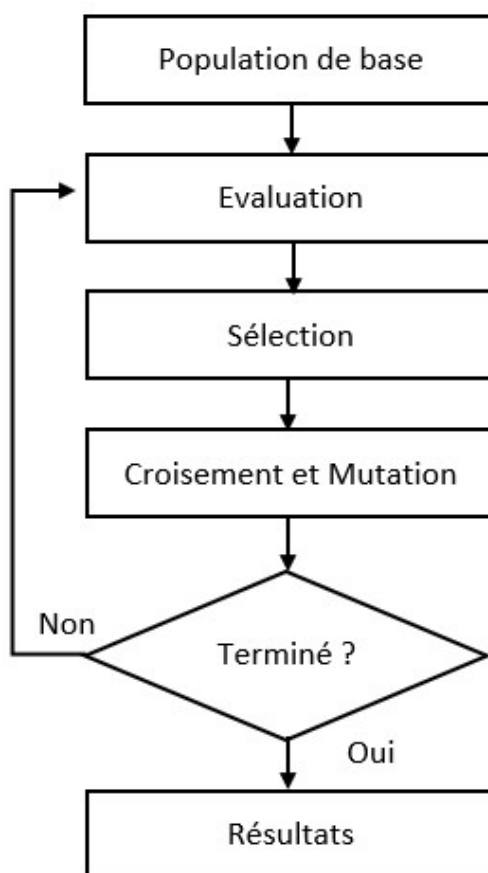


Figure 3: Fonctionnement d'un algorithme génétique (Sidi Mohamed Douiri,2011)

#### 1.6.3.4 Les algorithmes de colonies de fourmis

Ces algorithmes ont été introduits par Colorni, Dorigo et Maniezzo (Colorni, Dorigo, & Maniezzo), et le problème de voyageur de commerce fut l'une de ses premières applications, il a aussi été appliqué dans des problèmes de routage de véhicules VRP comme dans l'article de Mazzeo et Loiseau (Mazzeo & Loiseau, 2004). Il s'agit là encore d'algorithmes itératifs à population dans lequel un savoir commun est partagé entre les différents individus (ou solution potentielle), ce savoir se transmet entre les différents individus, les guidant à faire des choix à suivre et à éviter. L'analogie est faite notamment avec les colonies de fourmis puisque ces dernières aussi transmettent l'information (chemin le plus court entre autres) aux autres fourmis à travers une substance chimique appelée « phéromone », chaque fourmi en dépose une certaine quantité dans son chemin parcouru, les fourmis qui viendront après, auront plus de chance de choisir le chemin où se trouve une quantité plus grande de phéromone. Et ainsi, si on revient dans notre monde d'optimisation, on trouverait la meilleure solution.

Pour mieux expliquer le phénomène, Gauss et Deneubourg (Goss, Aron, Deneubourg, & Pasteels, 1989) ont réalisé une expérience appelée expérience du pont double, l'expérience consiste à mettre 2 branches de formes et longueurs différentes entre un nid de fourmis et de la nourriture.

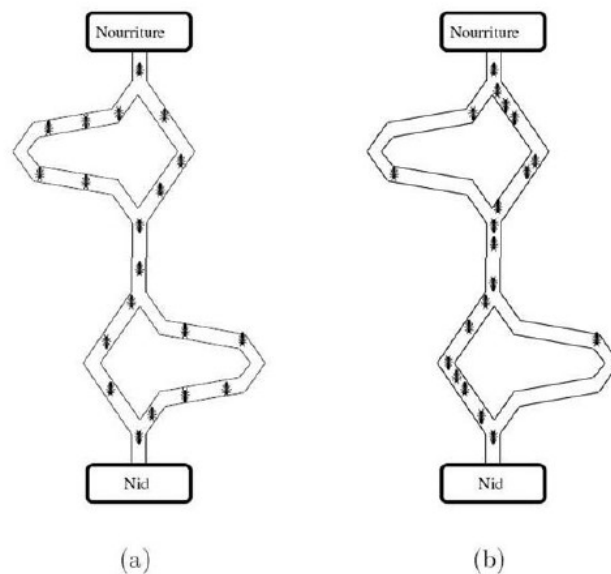


Figure 4: Expérience du pont double (Sidi Mohamed Douiri, 2011)

En sortant du nid, les fourmis font des allers-retours en prenant aléatoirement au début l'un des deux chemins. Dans la figure ci-dessus, on illustre la différence entre le début et la fin de l'expérience quant à l'utilisation des deux chemins et on remarque qu'à la fin de l'expérience, la majorité des fourmis prennent le chemin le plus court. Ceci est dû au fait que les fourmis ayant pris le chemin le plus court au début on fait plus d'aller-retour, et ainsi déposé une quantité supérieure phéromone ce qui a attiré plus de fourmis à prendre ce chemin.

En pratique, si on utilise cette méthode de résolution sur un VRP, les fourmis « artificielles » construisent des solutions en choisissant successivement les nœuds à visiter, jusqu'à ce que tous les nœuds soient visités. A chaque fois que le choix d'un nœud conduit à une incohérence ou infaisabilité (entraînant à enfreindre une des contraintes), on revient au dépôt, et une nouvelle route doit être commencée. Pour le choix d'un nœud, deux aspects entrent en compte, celui de l'optimisation donnée par le choix de ce nœud dans les itérations qui ont précédés, qui est un donné par un premier facteur  $\tau_{ij}$  qui est la valeur du phéromone menant de  $i$  à  $j$ , et à quel point le choix de ce nœud serait prometteur d'être la prochaine destination, donné aussi par un facteur de visibilité  $\eta_{ij} = \frac{1}{d_{ij}}$ .

La probabilité qu'une fourmi  $k$  se déplace d'un nœud  $i$  à un autre  $j$  est donnée par la valeur suivante :

$$p_{ij}^k(t) = \frac{(\tau_{ij}(t))^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in S_i^k} (\tau_{il}(t))^\alpha \cdot (\eta_{il})^\beta}$$

Les exposants  $\alpha$  et  $\beta$  influencent sur l'importance de chacun des facteurs (phéromone ou visibilité). Si  $\alpha = 0$ , la visibilité sera le seul paramètre pris en considération, ce qui mènera à toujours choisir que les nœuds les plus proches. Si  $\beta=0$ , on ne prendra compte que de la quantité de phéromone entre les nœuds et on négligera le facteur de visibilité, ce qui pourrait diminuer la qualité de la solution finale trouvée.

Quand la totalité des fourmis ont fini par construire une solution chacune, on dépose une quantité de phéromone  $\Delta\tau_{ij}^k$  pour chaque fourmi  $k$  sur son trajet. Quand un chemin  $(i,j)$  se

trouve dans la tournée d'une fourmi  $k$ , on calcule la quantité de phéromone sur ce trajet comme suit :

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L^k(t)}$$

Avec  $Q$  une constante et  $L^k(t)$  la longueur totale de la tournée.

Si on regarde bien la relation, on peut remarquer que la quantité de phéromone déposée sur une tournée augmente plus la longueur de totale de la tournée est petit, ceci va dans le sens où l'on voudrait minimiser la longueur des trajets dans notre problème d'optimisation de tournées, donc en choisissant les routes avec une plus grande quantité de phéromone, la longueur du trajet sera diminuée.

La méthode utilise aussi un autre paramètre, qui est le taux d'évaporation de phéromone  $\rho$  (avec  $0 < \rho < 1$ ), en intégrant le concept d'évaporation des pistes de phéromones. Celui-ci a été conçu pour éviter de converger vers des optima locaux de mauvaise qualité. La relation ci-après décrit ce concept :

$$\tau_{ij}(t+1) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

Où  $t$  représente une itération, et  $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$  avec  $m$  le nombre de fourmis.

## 1.7 Conclusion

Nous avons pu voir à travers ce chapitre plusieurs articles traitant des sujets en rapport avec notre projet, allant de recherches sur le thème global de l'optimisation du transport à la définition des variantes de ces problèmes. Nous avons aussi pu voir des articles de chercheurs qui ont fait des études en tenant compte de contraintes se rapprochant des contraintes que l'on souhaite prendre en considération dans notre projet.

Dans le prochain chapitre, nous commencerons par décrire notre problème et nous discuterons la méthodologie suivie dans la formulation de notre modèle d'optimisation en s'inspirant d'un des articles vus dans ce chapitre.

## CHAPITRE 2

### METHODOLOGIE

Dans cette partie, nous allons discuter de la modélisation de notre problème en décrivant le problème de l'entreprise et en proposant le scénario que l'on veut modéliser. Nous allons après expliquer le modèle en présentant celui utilisé dans l'article de (Parragh et al., 2012) dont on s'inspire, en décrivant tous les paramètres utilisés dans celui-ci, la fonction-objectif et les équations contraintes et en essayant d'expliquer en détail chaque point en montrant la relation avec le modèle que nous allons proposer. Nous allons aussi définir les paramètres de ce modèle qui ne seront pas utilisés dans le nôtre, et les changements que l'on a apporté à ce modèle pour que le nôtre soit plus simple et plus adapté à la situation de l'entreprise. Nous présenterons après les méthodes de résolutions suivies pour la résolution de ce problème, en expliquant les grandes lignes. Puis à la fin, nous parleront des outils avec lesquels nous avons travaillé pour résoudre le problème

#### 2.1 Description du problème

Dans cette partie, nous allons discuter de la modélisation de notre problème en décrivant le problème de l'entreprise, et en expliquant en détail un modèle que l'on a pris comme source d'inspiration afin de proposer un modèle plus simple et qui répond au besoin de l'entreprise, et qui regroupera à la fois des contraintes d'incompatibilités entre les clients et les contraintes des pauses pour les chauffeurs ainsi que leur période de travail maximale en une journée.

Pour notre problème, on imaginera un scénario, où l'on a un ensemble de points de ramassage de clients (patients, employés, écoliers ...) comme dans le deuxième exemple, et un ensemble de points de dépôts de ces clients, tel que  $j = n + i$ , on considère que dans chaque point  $i$  on a un nombre de client à ramasser ou à déposer  $d_i$ , avec  $d_i \geq 0$  si c'est un point de cueillette et  $d_{i+n} = -d_i$  ce qui veut dire que chaque nœud de livraison  $d_{i+n}$  correspond au dépôt de clients cueillis dans le nœud  $d_i$ .

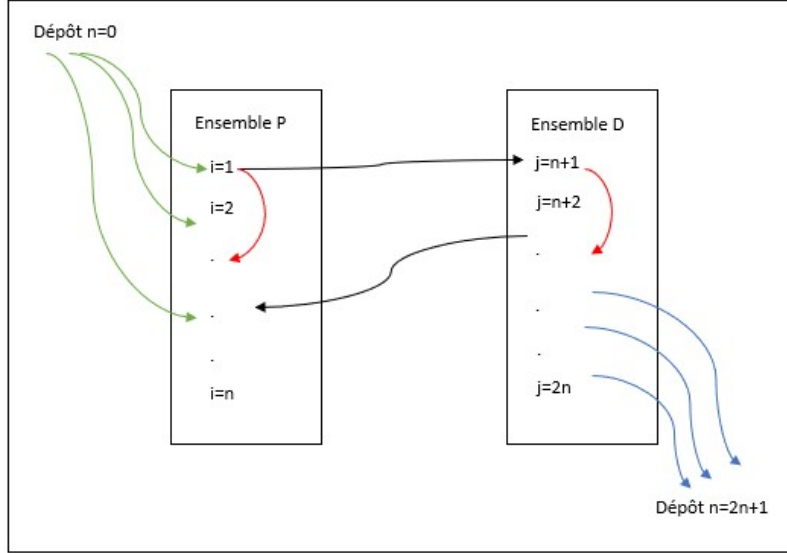


Figure 5: Description des trajets possibles des véhicules (cas de notre modèle)

Nous avons un dépôt  $i=0$  avec  $m$  véhicules, nous avons un ensemble de points de ramassage de clients  $P$ , et de destination de chacun de ses clients  $D$ . Ce qu'on explique par les flèches ici c'est que le premier point à visiter après le départ doit être un point de ramassage (vert), et que le dernier avant le dépôt final est un point de destination du dernier client (bleu). On peut aller directement d'un point de cueillette à un point de destination d'un des personnes à bord ou d'aller à un point de cueillette après avoir déposé un client (noir). Il est possible de visiter des points  $i$  successifs ou des points  $j$  successifs en vérifiant l'incompatibilité des clients à bord du véhicule (rouge).

En ce qui est de nos contraintes d'incompatibilités, nous essayeront de travailler avec une matrice d'incompatibilité entre tous les clients présents dans les points de cueillette  $P_i$ , en tenant en considération que si un client  $i$  est à bord du véhicule, aucun autre client  $j$  incompatible ( $inc_{i,j} = 1$ ) ne pourra être ramassé à bord du véhicule tant que ce véhicule n'est pas arrivé à la destination  $d_i$  soit  $j = n + i$ . Dans la contrainte, on vérifiera aussi qu'aucune incompatibilité ne peut se trouver dans un même véhicule, en d'autres termes, la somme des incompatibilités dans un même véhicule est égale à 0.



Pour les contraintes de chauffeurs, on va programmer des pauses qui doivent se tenir à des intervalles d'heures précises  $e_H$  et  $l_H$ , ces pauses peuvent être prises à n'importe quel point  $i$  et ont une durée déterminée  $L$ .

## 2.2 Modélisation

Comme vu en revue de littérature, le problème de (Parragh et al., 2012) traite un problème qui gère plusieurs types de véhicules, différents non seulement par leurs capacités mais aussi par la nature des équipements disponibles à bord. Le problème gère aussi la possibilité d'avoir des employés accompagnateurs à bord du véhicule si l'un des clients transportés en a besoin, la notion des pauses ne concerne pas seulement les chauffeurs mais aussi ces accompagnateurs, pour qui le modèle propose un dépôt de midi, où il faudra faire ces pauses à midi si un des clients transportés par un certain véhicule demande à avoir un accompagnateur que ce soit le matin ou l'après-midi. La figure ci-dessous décrit le scénario des trajets pris en considération dans le modèle.

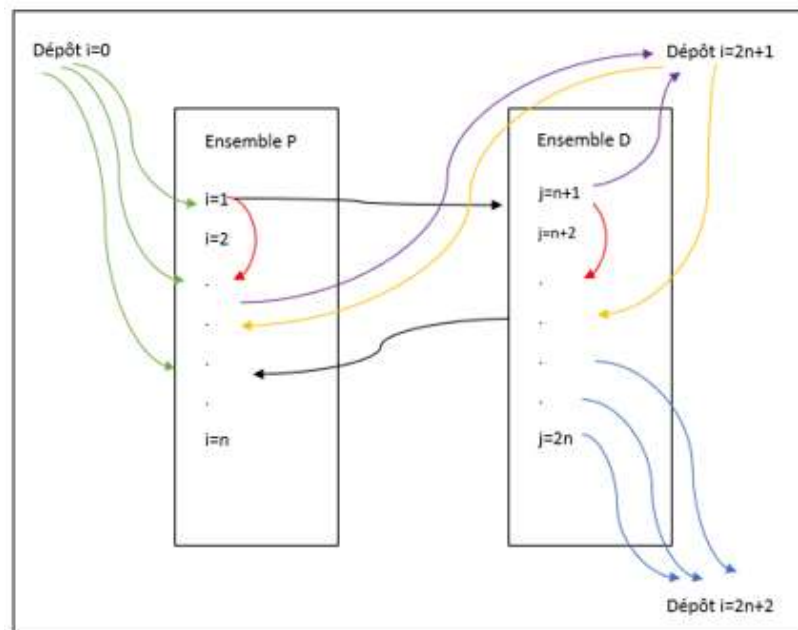


Figure 6: Description des trajets possibles des véhicules

Comme dans le schémas présenté lors de la description du problème (2.1) Nous avons un dépôt  $i=0$  avec  $m$  véhicules, nous avons un ensemble de points de ramassage de clients  $P$ , et de destination de chacun de ses clients  $D$ . On rajoute ici un dépôt  $i = 2n + 1$  où on va effectuer la pause déjeuner si le dans le véhicule  $V$  on aura un employé accompagnateur, ceci arrivera si l'un des clients  $i$  visité par le véhicule  $V$  a besoin d'un employé accompagnateur ou pas. Comme dans (2.1) les flèches en vert, bleu, noir et rouge veulent dire la même chose. Ici, on rajoute la possibilité d'aller directement d'un point  $i$  de cueillette ou d'un point de destination  $j$  au dépôt de midi (violet). Et aussi, il est possible de commencer l'après-midi en allant à un point de cueillette ou de destination (orange).

Le modèle du problème présenté dans l'article de (Parragh et al., 2012) est formulé comme suit :

- **Formulation du modèle :**

**Ensembles :**

$P = \{1, \dots, n\}$  : Ensemble des points de cueillette

$D = \{n+1, \dots, 2n\}$  : Ensemble des points de destination

$V = P \cup D \cup \{0, 2n+1, 2n+2\}$  : Ensemble des nœuds

$A = \{(i, j) : i \in V / \{2n+2\}, j \in V / \{0\}, i \neq j\}$  : Ensemble des arcs  $(i, j)$

$R = \{0, 1, 2, 3\}$  : Ensemble des modes de transport

$K = \{1, \dots, m\}$  : Ensemble des véhicules

**Indices :**

$i \in V$  : Noeud  $i$  appartenant à  $V$  (on put y ramasser ou déposer des clients)

$k \in K$  : Véhicule  $m$  appartenant à la flotte de véhicules

$r \in R$  : Mode de transport demandé par le patient

**Paramètres :**

$C_{i,j}$  : Cout du trajet  $(i, j)$

$t_{i,j}$  : Temps du trajet  $(i, j)$

$C^{r,k}$  : nombre de ressources  $r$  dans le véhicule  $k$

$e_i$  : borne inférieure de la fenêtre de temps du point de cueillette ou de dépôt

$l_i$  : Borne supérieure de la fenêtre de temps du point de cueillette ou de dépôt

$d_i$  : Le temps de service au point  $i$

$a_i$  : Paramètre binaire qui définit si oui ou non un patient  $i$  a besoin d'un employé accompagnateur

$a_i = 1$  : si on en a besoin,  $a_i = 0$  sinon

$q_i^r$  : constante qui définit le besoin du patient  $i$  tel que :

$q_i^1 = 1$  : si le patient a besoin d'un banc normal

$q_i^2 = 1$  : si le patient a besoin d'une chaise roulante

$q_i^3 = 1$  : si le patient a besoin d'une civière

$q_i^4 = 1$  : si le patient a besoin d'une personne accompagnatrice

$L$  : Temps de conduite maximale d'un chauffeur par jour

$T$  : Durée maximum d'une route de tous les véhicules

$H$  : Durée de la pause

### **Variables de décision :**

$x_{ij}^k = 1$  si un arc  $(i, j)$  est traversé par un véhicule  $k$ , 0 sinon

$Q_i^{r,k}$  : charge du véhicule  $k$  au point  $i$  respectant le mode  $r$

$z_0^k = 1$  si un chauffeur est affecté au véhicule  $k$ , 0 sinon

$z_1^k = 1$  si un accompagnateur est désigné sur un véhicule  $k$  le matin, 0 sinon

$z_2^k = 1$  si un accompagnateur est désigné sur un véhicule  $k$  l'après-midi, 0 sinon

$v_i^k = 1$  si une pause déjeuner est faite dans le point  $i$ , 0 sinon

### **Fonction-objectif :**

La fonction-objectif est une fonction de minimisation de couts :

$$\min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} C_{ij} x_{ij}^k \quad (1)$$

### **Contraintes :**

Dans l'article, les contraintes du problème sont mises en groupes pour donner plus de clarté au modèle. Le premier groupe de contraintes est celui où l'on trouve les contraintes de base pour la gestion des demandes et des trajets de la manière la plus basique

$$\sum_{k \in K} \sum_{j \in V'} x_{ij}^k = 1 \quad \forall i \in P \quad (2)$$

$$\sum_{j \in V'} x_{ij}^k - \sum_{j \in V'} x_{n+i,j}^k = 0 \quad \forall i \in P, k \in K \quad (3)$$

$$\sum_{i \in V'} x_{ij}^k - \sum_{i \in V'} x_{ji}^k = 0 \quad \forall j \in P \cup D \cup \{2n+1\}, k \in K \quad (4)$$

$$\sum_{j \in V'} x_{0j}^k = z_0^k \quad \forall k \in K \quad (5)$$

$$\sum_{i \in V'} x_{i,2n+2}^k = z_0^k \quad \forall k \in K \quad (6)$$

$$x_{ij}^k \in \{0,1\} \quad \forall i, j \in V, k \in K \quad (7)$$

$$z_0^k \in \{0,1\} \quad \forall k \in K \quad (8)$$

Dans l'équation contrainte (2), on assure que chaque demande va être servie une fois, la contrainte (3) signifie que chaque couple  $(i,j)$  est visité par le même véhicule, la contrainte (4) assure la conservation des flux, les équation (5) et (6) assure la contrainte du fait que si un chauffeur est désigné pour un véhicule, le véhicule doit commencer  $i=0$  et finir  $j=2n+2$  dans le dépôt, (7) et (8) sont la définition des deux variables binaires  $z_0^k$  et  $x_{ij}^k$ .

Dans le deuxième groupe, on met les contraintes liées aux accompagnateurs :

$$\sum_{i \in V'} x_{i,2n+1}^k = \max \{z_1^k, z_2^k\} \quad \forall k \in K \quad (9)$$

$$Q_{0,2n+1}^{0,k} \geq z_1^k \quad \forall k \in K \quad (10)$$

$$x_{i,2n+1}^k = 1 \Rightarrow Q_{2n+1}^{0,k} \geq Q_i^{0,k} - z_1^k + z_2^k \quad \forall i \in V, k \in K \quad (11)$$

$$x_{i,2n+1}^k = 1 \Rightarrow Q_{2n+1}^{r,k} \geq Q_i^{r,k} \quad \forall i \in V, k \in K, r \in R / \{0\} \quad (12)$$

$$z_1^k + z_2^k \geq a_i \sum_{j \in V} x_{ij}^k \quad \forall i \in V, k \in K \quad (13)$$

$$z_1^k, z_2^k \in \{0, 1\} \quad \forall k \in K \quad (14)$$

Dans la contrainte (9) on fait en sorte que le dépôt de midi (2n+1) est utilisé si un accompagnateur est désigné pour un véhicule donné soit pour le matin ou l'après-midi, si un accompagnateur est désigné pour un véhicule k, les contraintes (10), (11) et (12) sont là pour pouvoir lui assurer un banc dans le véhicule quel que soit le mode du véhicule, la contrainte (13) définit le fait qu'un patient qui demande à avoir un accompagnateur ne pourrait être visité que si un accompagnateur est présent dans le véhicule, la contrainte (14) définit nos variables binaires  $z_1^k$  et  $z_2^k$ . Il est à noter que ces contraintes-là ne seront pas présentes dans notre modèle puisqu'on ne tiendra pas compte des paramètres liés aux accompagnateurs.

Les contraintes suivantes ont un rapport avec la disponibilité des véhicules de la flotte et la gestion de la capacité de ces derniers.

$$x_{ij}^k = 1 \Rightarrow Q_j^{r,k} \geq Q_i^{r,k} + q_j \quad \forall i \in V, j \in V / \{2n+1\}, k \in K, r \in R \quad (15)$$

$$\sum_{r'=r}^2 Q_i^{r',k} \leq \sum_{r'=r}^2 C^{r',k} \quad \forall i \in V, k \in K, r \in R / \{3\} \quad (16)$$

$$Q_i^{3,k} \leq C^{3,k} \quad \forall i \in V, k \in K \quad (17)$$

$$Q_i^{r,k} \geq 0 \quad \forall i \in V, k \in K, r \in R \quad (18)$$

La contrainte (15) assure que si on se déplace d'un point i à un point j, la charge au point j sera supérieure ou égale à la charge dans le point i qui la précède, la contrainte (16) vérifie la disponibilité de bancs d'où de civières suffisantes en cas de besoin du patient. La contrainte (17) assure la disponibilité d'une chaise roulante au cas où le patient la demande. La contrainte (18) est une contrainte de la positivité de la valeur Q qui définit la charge. Dans notre problème, puisqu'on ne gère pas différents types de véhicules, les contraintes (15) et (16) ne sont pas incluses dans notre modèle. La contrainte (17) sera modifiée de sorte à ce que le nombre de places dans le véhicule ne soit pas dépassé

Les contraintes suivantes définissent le début de service pour chaque point :

$$x_i^{r,k} = 1 \wedge v_i^k = 0 \Rightarrow B_j^k \geq B_i^k + d_i + t_{ij} \quad \forall i, j \in V, k \in K \quad (19)$$

$$x_{ij}^k = 1 \wedge v_i^k = 1 \Rightarrow B_j^k \geq B_i^k + d_i + t_{ij} + W_H^k + H \quad \forall i, j \in V, k \in K \quad (20)$$

$$B_{n+i}^k \geq B_i^k \quad \forall i \in P, k \in K \quad (21)$$

$$B_{2n+2}^k - B_0^k \leq T \quad \forall k \in K \quad (22)$$

$$\sum_{i \in V} v_i^k \geq z_0^k \quad \forall k \in K \quad (23)$$

$$v_i^k \in \{0, 1\} \quad \forall i \in V, k \in K \quad (24)$$

Les contraintes (19) et (20) sont là pour la gestion des pauses, tel que si un point  $i$  est choisi pour être le point où on va faire la pause déjeuner, l'heure de début du service au point  $j$  est supérieure ou égale à l'heure du service au points  $i$  plus le temps d'attente avant la pause plus le temps de pause actuel, en plus de la durée du trajet entre le point  $i$  et  $j$ . La contrainte (21) assure qu'on visite un patient pour le prendre de son point de demande avant son lieu de dépôt. La contrainte (22) assure que la durée totale du trajet d'un véhicule ne dépasse pas la limite maximale  $T$  (exemple de  $T=8,5h$ ). La contrainte (23) signifie que si un chauffeur est affecté à un véhicule, il faudrait prévoir une pause pendant le trajet. La contrainte (24) définit la variable binaire  $v_i^k$ .

Pour un respect total des données du problème, plusieurs autres contraintes sont données par les inégalités suivantes :

$$e_i \leq B_i^k \leq l_i \quad \forall i \in V, k \in K \quad (25)$$

$$z_1^k = 1 \wedge z_2^k = 0 \Rightarrow B_0^k \leq B_i^k \leq B_{2n+1}^k \quad \forall i \in V_a, k \in K \quad (26)$$

$$z_1^k = 0 \wedge z_2^k = 1 \Rightarrow B_{2n+1}^k \leq B_i^k \leq B_{2n+2}^k \quad \forall i \in V_a, k \in K \quad (27)$$

$$v_i^k = 1 \Rightarrow e_H \leq B_i^k + d_i + W_H^k \leq l_H \quad \forall i \in V, k \in K \quad (28)$$

$$W_H^k \geq 0 \quad \forall k \in K \quad (29)$$

La contrainte (25) limite l'heure du début de service chez un patient  $i$  par dans l'intervalle  $[e_i, l_i]$ , La contrainte (26) délimite les heures de service durant le matin quant à la contrainte (27) délimite les heures de service pendant l'après-midi. La contrainte (28) fait de sorte que la pause déjeuner aie bien lieu entre  $e_H$  et  $l_H$ , l'heure donnée où la pause déjeuner devra être effectuée. (29) est la contrainte de positivité de la variable  $w_H^k$  (temps d'attente). Dans notre modèle, nous n'avons pas d'employés accompagnateurs, ce qui fait que la pause de midi ne se fera pas au dépôt  $2n+1$ , les contraintes (26) et (27) ne seront pas présents dans notre modèle. Nous avons aussi retiré les limites pour le service dans nœud pour réduire la complexité du problème, on ne trouvera alors pas la contrainte (25) dans notre modèle.

Le dernier groupe de contraintes gère la disponibilité du personnel (chauffeurs et accompagnateurs) et leur affectation aux véhicules :

$$\sum_{k \in K} z_0^k \geq m_d - u \quad (30)$$

$$\sum_{k \in K} z_1^k + \sum_{k \in K} z_2^k \geq m_d + u \quad (31)$$

$$u \in \{0, \dots, m_d\} \quad (32)$$

Les inégalités (30) et (31) limitent le nombre de chauffeurs et accompagnateurs qui peuvent être assignés à un véhicule. Chaque chauffeur qui va travailler en tant qu'accompagnateur au lieu de conduire devra être employé soit le matin soit l'après-midi, ce qui veut dire que le nombre de conducteur disponible va diminuer alors que le nombre d'accompagnateur lui va augmenter de la même valeur  $u$ , qui définit le nombre de conducteurs qui sont employés comme accompagnateurs, et qui, selon (32) est un nombre entier positif inférieur à  $m_d$ . Ce groupe de contraintes a entièrement été enlevé dans notre problème vu que tout le concept d'employés accompagnateurs a été enlevé.

## 2.3 Méthode de résolution du problème

Dans cette partie, nous allons parler des méthodes de résolution utilisées pour la résolution de ce problème. Bien que celui-ci soit complexe, nous essayerons d'expliquer les grandes lignes des méthodes utilisées.

Pour la résolution de ce problème, deux méthodes sont comparées dans l'article, la première méthode présentée est la méthode utilisant la génération de colonnes, qui est une méthode exacte. La deuxième méthode est une méthode heuristique, où l'on va chercher une solution optimale dans le voisinage d'une solution générée.

### 2.3.1 Génération de colonnes

Comme nous l'avons expliqué avant, cette méthode consiste à reformuler en quelques sortes le problème et ainsi créer le problème restreint. Celui-ci est défini de la manière suivante :

#### **Ensembles :**

Un ensemble  $T$  a été créé et celui-ci définit l'ensemble des types de véhicules disponibles et  $\Omega_t$  l'ensemble des routes faisables par les véhicules de type  $t \in T$ , avec aussi l'ensemble  $\Omega$  de toutes les routes  $\omega$  de tous les types de véhicules.

#### **Paramètres:**

$m_t$  : le nombre de véhicules de type  $t$  disponibles.

$c_\omega$  : Coût de la route  $\omega$

#### **Variables:**

$b_{i\omega}$  : Nombre de fois que le nœud  $i$  est traversé

$g_\omega$  : Nombre d'employés accompagnateurs dont on aurait besoin pour la route  $\omega$

$y_\omega = 1$  : seulement si la route  $\omega$  est utilisée dans la solution

#### **Nouvelle fonction-objectif:**



$$\min \sum_{\omega \in \Omega} c_{\omega} y_{\omega} \quad (33)$$

La fonction-objectif est une minimisation des coûts totaux des routes choisies.

Elle est soumise aux contraintes suivantes :

$$\sum_{\omega \in \Omega} b_{i\omega} y_{\omega} = 1 \quad \forall i \in P \quad (34)$$

$$\sum_{\omega \in \Omega_t} y_{\omega} \leq m_t \quad \forall t \in T \quad (35)$$

$$\sum_{\omega \in \Omega} y_{\omega} \leq m_d - u \quad (36)$$

$$\sum_{\omega \in \Omega} g_{\omega} y_{\omega} \leq m_a + u \quad (37)$$

$$u \geq 0 \quad (38)$$

$$y_{\omega} \in \{0,1\} \quad \forall \omega \in \Omega \quad (39)$$

La contrainte (34) garantie que chaque demande est servie exactement une fois, la contrainte (35) ne permet pas de dépasser le nombre de véhicules disponibles pour chaque type. La contrainte (36) assure que le nombre d véhicules ne dépasse pas le nombre de chauffeurs disponibles. La contrainte (37) gère la contrainte des employés accompagnateurs.

La contrainte (39) a été remplacé par la contrainte (40), celle-ci aidant à programmer un le programme linéaire relaxé

$$y_{\omega} \geq 0 \quad \forall \omega \in \Omega \quad (40)$$

Le problème étant difficile à résoudre, avec la grandeur de l'ensemble  $\Omega$ , cet ensemble va être réduit et nous allons travailler avec un ensemble plus petit  $\Omega' \subset \Omega$ . Pour la génération de colonnes, le problème sera divisé en  $|T|$  sous-problèmes, un pour chaque type de véhicules.

Soit  $\pi_i$ ,  $\sigma_t$ ,  $\lambda$  et  $\phi$  les variables duales associées aux contraintes (34) pour  $i$ , (35) pour  $t$ , (36) et (37), le coût réduit de la colonne  $y_{\omega}$  pour une route  $\omega \in \Omega_t$  est donné par la relation suivante :

$$\bar{c}_{\omega} = c_{\omega} - \sum_{i \in P} b_{i\omega} \pi_i - \sigma_t - \lambda - g_{\omega} \phi \quad \forall \omega \in \Omega \quad (41)$$

Ici, l'objectif du sous-problème  $t$  est de trouver une route  $\omega \in \Omega_t$  pour chaque type de véhicule  $t$  avec un cout réduit minimum  $\bar{c}_\omega$ , et il est sujet aux contraintes de (3) à (29), l'indice  $k$  sera ignoré, et  $z_0 = 1$ . Pour les capacités  $C^{r,k}$ , on mettra  $C^{r,t}$ , avec  $t$  le type de véhicule.

L'algorithme suivant est suivi pour la résolution du problème en utilisant la méthode de génération de colonne :

Dans la première phase d'initiation, les colonnes initiales sont générées et ajoutées à  $\Omega'$ , plusieurs opération de préparation sont appliquées pour la simplification du calcul. Une fois cette phase finie, en appliquant le calcul de la solution, on récupère les valeurs des variables duales associées aux contraintes. A partir de ces valeurs, des heuristiques sont ainsi utilisé pour générer les colonnes à couts réduits. Les colonnes à cout négatifs sont ajoutées à l'ensemble  $\Omega'$  et on refait le calcul. Les itérations seront arrêtées lorsque l'on trouve aucune nouvelle colonne avec réduction de cout, dans cette situation, la solution optimale du problème a été trouvée.

### 2.3.2 Heuristique de recherche dans le voisinage

La méthode utilisée est une méthode heuristique, consistant à la recherche de solutions dans le voisinage d'une solution initiale.

La résolution par cette méthode consiste en quatre phases, une phase d'initialisation, la deuxième phase est une opération de sélections à travers des opérateurs, la troisième phase est celle ou on répète les itérations, et puis en fin la phase de décision.

Dans la première phase d'initialisation, on génère au début une solution initiale qui sera généralement infaisable. Toutes les routes des véhicules seront construites dans cette phase, en affectant toutes les demandes aux véhicules qui les satisfont. On arrête l'affectation des demandes une fois la capacité maximale est atteinte et on commence à affecter les prochaines demandes selon le classement dans le tableau des données dans le prochain véhicule. Cette phase finit une fois toutes les demandes seront affectées. Cela étant, on vérifie les routes om l'on a des demandes pour des employés accompagnateurs (selon l'affectation qui a été faite), si c'est le cas, le dépôt de midi est ajouté dans la route et un employé accompagnateur est affecté à ce véhicule suivant si la demande est faite pour le matin ou l'après-midi.

Dans la deuxième phase qui est la phase de sélection, on utilise des opérateurs afin de modifier les affectations faites dans la phase initiale. Ces opérations sont tous les changements qui seront appliqués sur les affectations des demandes. Par exemple on essaye de prendre une demande, et on voit d'abord quelle est la meilleure route à laquelle on peut affecter cette demande, après, on vérifie la meilleure position possible où l'on affectera cette demande dans la route choisie. Dans la troisième phase, on répète les itérations sur toutes les demandes à satisfaire dans le problème, et après chaque itération, une réparation sera faite de sorte à ne pas avoir des véhicules sans conducteur affecté, si on tombe dans cette situation, les demandes de ce véhicule seront dispatchées sur les autres routes. Après chaque modification, on vérifie si la solution a été améliorée, cette solution doit répondre à un critère d'acceptation pour que le voisinage de recherche soit celui de la nouvelle solution trouvée. On continue à modifier les routes et vérifier les solutions jusqu'à ce qu'on arrive plus à trouver des améliorations dans la solution. A la fin, après le choix de la solution, on effectue une évaluation des routes, cette fois, c'est une vérification des routes définies. On vérifie les positions des pauses de midi pour voir si on pourrait avoir une meilleure possibilité, voir même si on peut les retirer de la route si après les itérations, cette route n'a plus aucune demande d'un employé accompagnateur. Et comme ça, notre solution finale est définie.

## **2.4 Outils de résolutions**

Pour la résolution des problèmes d'optimisation, plusieurs outils et solveurs sont généralement utilisés, l'un des plus utilisés est CPLEX, c'est ce dernier qui a été utilisé pour la résolution de ce problème-là,

Pour notre version simplifiée du modèle, nous avons choisi de travailler avec le solveur LINGO, ce dernier étant le plus conseillé pour les débutants en programmation linéaire. Il offre la possibilité de gérer plusieurs types de problèmes, bien que CPLEX soit plus performant, et est généralement utilisé pour les problèmes les plus complexes.

Nous avons aussi utilisé Microsoft EXCEL, où nous avons pu insérer nos tableaux de données, ainsi que les sorties de variables de notre programme. La syntaxe du solveur LINGO faire la

liaison avec EXCEL, en définissant nos ensembles, nos paramètres et nos variables de décision.

## **2.5 Conclusion**

Nous avons pu à travers ce chapitre définir et détailler notre problème et discuter notre méthodologie quant à sa résolution, en détaillant le modèle d'optimisation duquel nous nous sommes inspirés afin de proposer le nôtre, et aussi en expliquant les méthodes de résolutions utilisées dans l'article de Paragh et al. (Parragh et al., 2012). Nous avons fini par citer les outils que nous allons utiliser pour la résolution de notre problème.

Dans le prochain chapitre, nous allons justement proposer notre modèle d'optimisation en détaillant les données et les équations du modèle. Nous discuterons aussi des résultats obtenus à travers celui-ci et finirons par proposer des améliorations à effectuer pour une résolution plus optimale.

## CHAPITRE 3

### PRESENTATION DE LA SOLUTION, ET ANALYSE DU MODELE

Après avoir fait une synthèse à propos des méthodologies suivies pour traiter ce problème, en modélisation et aussi en résolution en discutant les différentes méthodes, nous allons dans cette partie présenter et décrire la solution que nous avons proposé. Pour la présentation des résultats, nous allons la faire suivant 2 cas d'études, un cas où le programme choisira un seul véhicule entre les trois, et un deuxième où l'on aura à utiliser deux véhicules, nous allons expliquer les résultats obtenus selon les données choisies dans chaque cas. Nous finirons par donner une analyse de ce qui a été réalisé et des propositions que l'on pourrait proposer pour améliorer ce projet dans un futur projet.

#### 3.1 Modèle proposé

##### Ensembles :

$P = \{1, \dots, n\}$  : Ensemble des points de ramassage des clients

$D = \{n+1, \dots, 2n\}$  : Ensemble des points de destinations des clients

$R = P \cup D \cup \{0, 2n+2\}$  Ensemble des nœuds  $i$  avec  $i=0$  le dépôt (point de départ des véhicules) et  $i = 2n+2$  le dépôt final

$A = \{(i, j) : i \in R / \{2n+2\}, j \in R / \{0\}, i \neq j\}$  : Ensemble des arcs  $(i, j)$

$V = \{1, \dots, m\}$  : Ensemble des véhicules  $V$  dans la flotte des véhicules

##### Indices :

$v \in V$  : véhicules appartenant à l'ensemble  $V$  des véhicules

$i \in R$  : Nœuds appartenant à l'ensemble  $R$ , on peut y ramasser ou déposer des clients

##### Paramètres :

$inc_{i,j}$  : Incompatibilité entre les clients ramassés aux points  $i$  et  $j$  ( $i$  et  $j$  appartiennent à  $P$  et  $inc_{i,j} = 1$  s'ils sont incompatibles, 0 sinon)

$c_{i,j}$  : Cout du trajet entre les points  $i$  et  $j$

$t_{i,j}$  : Durée du trajet entre les points  $i$  et  $j$

$m$  : Nombre de véhicules dans la flotte

$p$  : Somme des clients à ramasser dans tous les points de cueillette.

$d_i$  : Nombre de clients collectés ou déposés au nœuds  $i$  ( $d_i \geq 1$  si  $i \in P$  et  $d_i \leq -1$  si  $i \in D$ )

$q^v$  : Capacité max d'un véhicule  $v$

$H_{DS}$  : Heure de début du service (le chauffeur doit faire son départ au moins à  $H_{DS}$ )

$H_{FS}$  : Heure de fin du service (le chauffeur doit retourner au dépôt final avant  $H_{FS}$ )

$e_H$  : heure minimale pour le début d'une pause

$l_H$  : Heure maximale pour la fin d'une pause

$L$  : Durée de la pause

#### **Variables de décision :**

$X_{i,j}^v$  : Variables binaire égale à 1 si le trajet  $(i,j)$  est fait par le véhicule  $v$

$Y_i^v$  : Variable qui définit le nombre de personne à bord du véhicule  $v$  sortant du point  $i$

$Z_v = 1$  si un chauffeur est affecté au véhicule  $v$ , 0 sinon

$B_i^v = 1$  si une pause déjeuner est faite dans le point  $i$ , 0 sinon

$S_i^v$  : Heure du service du véhicule au point  $i \in R$  (lieu d cueillette ou destination)

#### **Fonction-objectif : Minimiser le cout total des trajets effectués :**

L'équation (1) représente notre fonction-objectif. Il s'agit ici de minimiser le cout de toutes les routes effectuées par les véhicules de la flotte.

$$\min z = \sum_{v \in V} \sum_{i \in R} \sum_{j \in R} c_{i,j} X_{i,j}^v \quad (1)$$

#### **Soumise aux contraintes suivantes :**

Pour présenter les contraintes du problème, nous allons les séparer en des groupes de contraintes. Dans le premier groupe, on met les contraintes visant à gérer le flux des trajets. L'équation (2) signifie qu'un point de ramassage de client  $i$  doit être visité exactement une

fois et par un seul véhicule dans la totalité des trajets des véhicules de la flotte, l'équation (3) signifie que chaque couple  $(i, j)$  est visité par le même véhicule, (4) assure la conservation des flux, les équation (5) et (6) assure la contrainte du fait que si un chauffeur est désigné pour un véhicule, le véhicule doit commencer à  $i=0$  et finir à  $j=2n+2$  dans le dépôt, et aussi impose que le premier trajet soit fait à un point de Pickup ( $j \in P$ ) et le dernier trajet soit fait à partir d'un point de livraison ( $i \in D$ ), (7) et (8) sont la définition des deux variables binaires  $Z_v$  et  $X_{ij}^v$ , et (9) la contrainte de positivité de  $Y_i^v$

$$\sum_{v \in V} \sum_{j \in R} X_{i,j}^v = 1 \quad \forall i \in P \quad (2)$$

$$\sum_{j \in R} X_{ij}^v - \sum_{j \in R} X_{n+i,j}^v = 0 \quad \forall i \in P, v \in V \quad (3)$$

$$\sum_{i \in R} X_{i,j}^v - \sum_{i \in R} X_{j,i}^v = 0 \quad \forall j \in P \cup D, v \in V \quad (4)$$

$$\sum_{j \in P} X_{0,j}^v = Z_v \quad \forall v \in V \quad (5)$$

$$\sum_{i \in D} X_{i,2n+1}^v = Z_v \quad \forall v \in V \quad (6)$$

$$X_{i,j}^v \in \{0,1\} \quad \forall v \in V; \forall i, j \in R \quad (7)$$

$$Z_v \in \{0,1\} \quad \forall v \in V \quad (8)$$

$$Y_i^v \geq 0 \quad \forall v \in V; \forall i \in R \quad (9)$$

L'équation (10) signifie qu'on ne peut pas avoir plus de véhicules  $V$  qui sortent du point de départ  $i=0$  que de véhicule  $V$  que l'on a de disponible (au nombre  $m$ ). L'équation (11) signifie qu'on ne peut pas avoir plus de véhicules  $V$  qui arrivent au point de destination point de départ  $j=n$  que de véhicule  $V$  que l'on a de Véhicules disponible.

$$\sum_{v \in V} \sum_{i \in R} X_{0,i}^v \leq m \quad (10)$$

$$\sum_{v \in V} \sum_{i \in R} X_{i,2n+1}^v \leq m \quad (11)$$

Dans le deuxième groupe de contraintes, nous trouvons les contraintes qui gèrent la charge des véhicules, et font en sorte que les véhicules respectent leur capacité maximale. Les contraintes (12) et (13) assurent que si on se déplace d'un point  $i$  à un point  $j$ , la charge au point  $j$  sera égale à la charge dans le véhicule dans le trajet précédent plus ou moins le nombre de personnes cueillis ou déposés dans ce nœud. La contrainte (14) limite le nombre de personnes à bord du véhicule, tel que le nombre de passagers à bord d'un véhicule  $v$  ne peut pas dépasser sa capacité  $q^v$ .

$$Y_j^v \geq Y_i^v + d_j - p(1 - X_{i,j}^v) \quad \forall i \in R, j \in P, v \in V \quad (12)$$

$$Y_j^v \geq Y_i^v + d_j - p(1 - X_{i,j}^v) \quad \forall i \in R, j \in D, v \in V \quad (13)$$

$$Y_i^v \leq q^v \quad \forall v \in V; \forall i, j \in R \quad (14)$$

Le groupe de contraintes suivant gère le temps d'arrivée du véhicule dans chaque nœud et aussi la contrainte des pauses des chauffeurs. Les contraintes (15) et (16) gèrent les cas particuliers du premier et dernier trajet du véhicule. Les équations (17) et (18) imposent que l'heure du début et de la fin du service du chauffeur soient respectés. L'inéquation (19) inclut la notion des pauses, tel que si un point  $i$  est choisi pour être le point où on va faire la pause déjeuner (soit  $B_i^v = 1$ ), l'heure de début du service au point  $j$  est supérieure ou égale à l'heure du service au point  $i$  plus le temps de pause actuel  $L$ , en plus de la durée du trajet entre le point  $i$  et  $j$ . Les inéquations (20) et (21) assurent que si le temps d'arrivée à un nœud  $S_i^v$  est entre  $e_H$  et  $l_H$  (heures limites pour prendre une pause), le chauffeur effectuera une pause en ce point et ainsi la variable  $B_i^v$  prend la valeur 1. La contrainte (22) signifie que si un chauffeur est affecté à un véhicule, il faudrait prévoir une pause pendant le trajet. La contrainte (23) impose que le temps d'arrivée à un point de livraison doit être supérieure à celle de la cueillette du même client. La contrainte (24) définit la variable  $B_i^v$  comme variable binaire.

$$S_j^v \geq S_0^v + X_{0,j}^v \times t_{0,j} - (1 - X_{0,j}^v) \times M \quad j \in P, v \in V \quad (15)$$

$$S_{2n+1}^v \geq S_i^v + X_{i,2n+1}^v \times t_{i,2n+1} - (1 - X_{i,2n+1}^v) \times M \quad \forall i \in D, v \in V \quad (16)$$



$$S_0^v \geq H_{DS} \quad \forall i \in P, v \in V \quad (17)$$

$$S_{2n+2}^v \leq H_{FS} \quad \forall v \in V \quad (18)$$

$$S_j^v \geq S_i^v + X_{i,j}^v \times t_{i,j} - (1 - X_{i,j}^v) \times M + B_i^v \times L \quad \forall i, v \in P \cup D, v \in V \quad (19)$$

$$S_i^v \geq B_i^v \times e_H \quad \forall i \in R, v \in V \quad (20)$$

$$S_i^v \leq (1 - B_i^v) \times M + l_H - L \quad \forall i \in R, v \in V \quad (21)$$

$$\sum_{i \in R} B_i^v \geq Z_v \quad \forall v \in V \quad (22)$$

$$S_{i+5}^v \geq S_i^v \quad \forall i \in P, v \in V \quad (23)$$

$$B_i^v \in \{0, 1\} \quad \forall i \in R, v \in V \quad (24)$$

L'équation (24) est là pour la gestion des incompatibilités, assurant que pour toutes les routes  $v$  (ou les véhicules  $v$ ), tous les clients ne soient pas incompatibles  $inc_{i,j} = 0$ .

$$\sum_{v \in V} \sum_{i \in P} \sum_{j \in P} inc_{i,j} X_{i,j}^v = 0 \quad (25)$$

### 3.2 Jeux de données

Comme introduit au début de ce chapitre, nous avons essayé de trouver une application de ce modèle et voir les résultats que ça peut donner vu que nous n'avons pas des données d'un exemple réel et nous devons nous mêmes proposer des données, et avons fini par sortir avec deux situations, la première dans laquelle on n'aurait à utiliser qu'un seul véhicule et une deuxième où l'on va affecter deux chauffeurs ce qui implique l'utilisation de deux véhicules. Nos données sont organisées sous forme de tableaux sur un fichier Excel (Tableaux de l'[annexe I](#)), ce fichier étant lié au programme codé sur Lingo ([Annexe II](#)). Nous allons alors dans cette partie montrer les données choisies dans chacune des situations et expliquer la différence. Les paramètres d'entrée de notre programme sont les paramètres de coût, de temps de trajet, des capacités des véhicules, des incompatibilités entre clients. Mais aussi, on peut aussi

modifier les limites de temps comme l'heure de départ des véhicules, l'heure du début de la pause ou de la fin de la pause, et la limite maximale du retour des véhicules au dépôt.

Pour obtenir les solutions que nous avons eu, nous avons choisi de modifier dans chacune des situations l'heure du retour au dépôt, et les limites inférieures et supérieures de l'intervalle de temps où le chauffeur doit prendre sa pause. On montre dans les tableaux ci-dessous les valeurs que nous avons choisies pour chaque situation :

eH	IH	Hds	Hfs
750	810	510	1105

Tableau 1: Valeurs situation 1

eH	IH	Hds	Hfs
620	680	510	800

Tableau 2: Valeurs situation 2

Ces changements nous permettent de tester notre programme pour deux situations différentes, celle où un seul véhicule suffirait pour servir tous les clients, et une deuxième où il faut utiliser un deuxième véhicule en même temps. Il est à noter que l'on aurait aussi pu modifier les temps de trajet entre les différents nœuds en mettant des valeurs plus grandes, et aussi modifier les capacités des véhicules en mettant des véhicules avec une capacité réduite. On aurait aussi pu augmenter le nombre de points à visiter.

### 3.3 Présentation des solutions obtenues

#### **Situation 1 :**

Dans la première situation, nous avons eu comme résultat le choix d'un seul véhicule, et c'est le véhicule pouvant transporter le plus grand nombre de passagers. Ici, ce même véhicule peut, en respectant un trajet optimal à partir de chaque nœud, répondre au besoin de tous les clients et traverser tous les nœuds dans une même journée en respectant la contrainte de pause et l'heure maximale du retour au dépôt.

X <sub>vi,j</sub>		R												Y <sub>vi</sub>	S <sub>iv</sub>	B <sub>iv</sub>
		1	2	3	4	5	6	7	8	9	10	11	12			
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	510	0
2	2	0	0	0	0	1	0	0	0	0	0	0	0	2	545	0
2	3	0	0	0	0	0	0	1	0	0	0	0	0	4	769	1
2	4	0	0	0	0	0	1	0	0	0	0	0	0	3	664	0
2	5	0	0	0	0	0	0	0	0	0	1	0	0	4	591	0
2	6	0	0	0	0	0	0	0	0	0	0	1	0	4	699	0
2	7	0	0	0	0	0	0	0	1	0	0	0	0	2	834	0
2	8	0	0	0	0	0	0	0	0	1	0	0	0	1	873	0
2	9	0	0	0	0	0	0	0	0	0	0	0	1	0	909	0
2	10	0	0	0	1	0	0	0	0	0	0	0	0	2	630	0
2	11	0	0	1	0	0	0	0	0	0	0	0	0	3	734	0
2	12	0	0	0	0	0	0	0	0	0	0	0	0	0	945	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	5	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
3	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tableau 3: Valeurs retournées dans la situation 1

Le tableau 3 est obtenu dans notre situation 1. Ici, on remarque que le véhicule choisi est le véhicule 2 qui a une capacité de transportation de 4 passagers en même temps. Pour voir le résultat de manière plus claire, nous proposons un nouveau tableau qui représente le trajet du véhicule choisi, en montrant à chaque nœud le temps du trajet, l'heure d'arrivée au nœud et la charge du véhicule dans ce nœud.

Nœud de départ	Nœud d'arrivée	Durée du trajet	Charge du véhicule	Début du trajet	Pause
1	2	35	0	510	
2	5	46	2	545	
5	10	39	4	591	
10	4	34	2	630	
4	6	35	3	664	
6	11	35	4	699	
11	3	35	3	734	
3	7	35	4	769	Oui
7	8	39	2	834	
8	9	36	1	873	
9	12	36	0	909	
Heure de retour au dépôt 12				945	

Tableau 4: Trajet du véhicule 2

On remarque que la contrainte de la capacité du véhicule a été respectée, et aussi, la pause a été faite dans le nœud 3, c'est pour cela que nous remarquons un décalage de 30 minutes avant le départ du point qui suit (le nœud 8). Le cout total de tout le circuit proposé est de 407 \$ (Voir [Annexe III](#)).

### **Situation 2 :**

Dans la deuxième situation, nous avons pu affecter deux véhicules. Pour avoir cette solution, nous avons réduit le temps de retour des véhicules au dépôt, dans le premier exemple, la limite qu'on avait fixée était 1105, dans cette deuxième situation, la limite est de 800, nous avons aussi modifié les limites inférieure et supérieure du temps om le chauffeur devrait prendre une pause, ici le chauffeur peut prendre une pause dans l'intervalle entre 620 et 680. On montre ci-dessous le tableau avec les valeurs retournées sur notre fichier Excel :

Xvi,j		R												Yvi	Siv	Biv
		1	2	3	4	5	6	7	8	9	10	11	12			
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	510	0
1	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
1	3	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	510	0
2	2	0	0	1	0	0	0	0	0	0	0	0	0	2	545	0
2	3	0	0	0	0	0	0	1	0	0	0	0	0	3	585	0
2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	7	0	0	0	0	0	0	0	1	0	0	0	0	1	620	1
2	8	0	0	0	0	0	0	0	0	0	0	0	1	0	689	0
2	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	12	0	0	0	0	0	0	0	0	0	0	0	0	0	735	0
3	1	0	0	0	1	0	0	0	0	0	0	0	0	0	510	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0	0	0	1	0	0	0	1	544	0
3	5	0	0	0	0	0	0	0	0	0	1	0	0	2	702	0
3	6	0	0	0	0	0	0	0	0	0	0	1	0	1	613	0
3	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	9	0	0	0	0	0	1	0	0	0	0	0	0	0	579	0
3	10	0	0	0	0	0	0	0	0	0	0	0	1	1	741	0
3	11	0	0	0	0	1	0	0	0	0	0	0	0	0	648	1
3	12	0	0	0	0	0	0	0	0	0	0	0	0	0	787	0

Tableau 5: Valeurs retournées situation 2

On remarque ici que les véhicules 2 et 3 ont été choisis pour répondre au besoin des clients. Comme nous l'avons fait pour la situation 1, nous allons représenter de manière plus claire les trajets des véhicules 2 et 3 à travers les tableaux suivants :

Nœud de départ	Nœud d'arrivée	Durée du trajet	Charge du véhicule	Début du trajet	Pause
1	2	35	0	510	
2	3	40	2	545	
3	7	35	3	585	
7	8	39	1	620	Oui
8	12	46	0	689	
Heure de retour au dépôt 12 du véhicule 2				735	

Tableau 6: Trajet du véhicule 2

Dans ce cas, le véhicule 2 fait une pause dans le nœud 7, et retourne au dépôt à la minute 735 soit avant la limite fixée qui est de 800.

Nœud de départ	Nœud d'arrivée	Durée du trajet	Charge du véhicule	Début du trajet	Pause
1	4	34	0	510	
4	9	35	1	544	
9	6	34	0	579	
6	11	35	1	613	
11	5	34	0	648	Oui
5	10	39	2	702	
10	12	46	0	741	
Heure de retour au dépôt 12				787	

Tableau 7: Trajet du véhicule 3

Ici, le véhicule 3 fait sa pause au nœud 11, et reviens au dépôt 12 à la minute 787.

Notons aussi que pour ce deuxième essai, le temps de calcul était de plus d'une heure, Le cout total cette fois-ci est de 456 \$ (Voir solution en [Annexe III](#)).

### 3.4 Discussion

#### 3.4.1 Analyse de la solution obtenue

Dans les deux situations que nous avons proposées, l'utilisateur, un gestionnaire de transport, a automatiquement le choix optimal des routes qu'il devrait planifier pour chacun des véhicules disponibles dans sa flotte. Les avantages de l'utilisation d'une application pareille est le fait que non seulement elle permet de gagner du temps et permettre d'avoir une planification de trajet, mais aussi, la solution proposée est la solution optimale, et prend aussi en considération les pauses des chauffeurs.

Si on analyse bien la solution obtenue dans les deux situations, on trouve que le véhicule choisi à chaque fois le trajet qui coûte le moins (vérifier dans les tableaux 8 et 9 des temps et coûts de trajet dans l'[annexe I](#)). Si on prend par exemple les tableaux 6 et 7, quand la charge du véhicule est égale à 0, le prochain nœud que le véhicule ira visiter est un nœud de collecte, Si la charge du véhicule est égale à la capacité du véhicule, le prochain nœud à visiter est un nœud où on va déposer des clients. Si la charge du véhicule n'est pas égale à 0 et est inférieure à la capacité du véhicule, on a deux situations différentes, si le nœud le plus proche est un nœud est un nœud de collecte, le véhicule va donc choisir de se diriger vers ce nœud en premier comme pour le véhicule dans le tableau 6. Si le nœud le plus proche est le nœud où l'on va déposer un client déjà à bord du véhicule, le véhicule prendra sa direction comme prochain trajet.

Concernant la contrainte des pauses, dans la situation 1 nous avons défini comme limites de prise de pause les valeurs 750 et 810. On remarque alors dans le tableau 4 que la pause a été prise une fois que l'arrivé au nœud 3 était à la minute 769. On remarque aussi que le départ à partir du nœud d'après (le nœud 7) a été décalé de 30 minutes.

Dans la situation 2, nous avons modifié le temps de retour au dépôt final, et aussi changé les limites pour l'heure de prise de pause, les limites fixées cette fois-ci sont 620 et 680, et on remarque dans les tableaux 6 et 7 que les pauses ont été respectivement prises à 620 et 648 pour les 2 véhicules. Le départ à partir du nœud d'après est là aussi décalé de 30 minutes.

### 3.4.2 Obstacles et recommandations

Nous sommes arrivés à obtenir une solution optimale en tenant compte des capacités des véhicules et en prenant en considération les pauses des chauffeurs, néanmoins, avec la complexité du problème de cueillette et de livraison, la gestion des incompatibilités présente toujours un obstacle pour le moment. Ceci signifie que notre modèle n'est toujours pas complet et peut être amélioré. L'une des raisons pour lesquelles on n'y arrive pas est que l'équation (25) ne suffit peut-être pas pour éliminer la possibilité d'avoir des personnes incompatibles à bord du véhicule. Même en respectant l'équation, il se peut qu'un client qu'on prend dans le point  $k$  soit compatible avec le client dans  $j$ , sans que le premier ne soit compatible avec le client  $i$ , ce cas ne serait pas pris en considération par cette équation et il pourrait y avoir un cas où l'on aurait les 3 dans le même véhicule en même temps. Pour résoudre ce problème, on peut penser à ajouter une autre variable  $f_i$  qui définit si un client pris dans le point  $i$  est à bord du véhicule en temps actuel, avec  $f_i = 1$  quand le temps est compris entre  $s_i^v$  et  $s_{n+i}^v$ , puis ajouter une condition lors du ramassage d'un nouveau client, et de vérifier le tableau des  $f_i$  et voir ceux qui sont égales à 1, puis vérifier l'incompatibilité entre  $j$  et les  $i$  dont le  $f_i = 1$ .

Si l'on a fait attention aux résultats donnés dans la situation 1, les clients que nous avons pris à bord du véhicule dans le nœud 2 ont été déposés après avoir servis les clients des nœuds 5, 4 et 6 alors que les premiers ont été à bord du véhicule pendant que l'on récupère et que l'on dépose les autres. Cela respecte bien les contraintes que l'on a mis en place, mais dans la vie réelle, ceci pourrait avoir un effet négatif sur la qualité du service et sur la satisfaction du client. Pour améliorer ce problème, peut-être que l'on pourrait ajouter une contrainte de délai maximale dans lequel on devrait déposer un client que l'on a pris à bord du véhicule.

Concernant d'autres points qui auraient pu être pris en considération, on aurait pu ajouter comme donnée un coût à l'utilisation de chaque véhicule, soit l'utilisation d'un véhicule à 4 places est logiquement plus grande que celle d'un véhicule à 2 places par exemple. Ceci pourrait affecter le résultat qui, en ce moment, favorise les véhicules à 4 places dans toutes les situations.



### **3.5 Conclusion**

A travers ce dernier chapitre, nous avons pu présenter notre modèle, en donnant une explication à chacune des contraintes. Nous avons testé le modèle sur deux situations différentes donc nous avons commencé par décrire les différences entre les données entrées dans chacune des situations, puis nous avons présenté les résultats obtenus en fonction des données que l'on a présenté. En analysant ces résultats, nous avons pu voir que la solution s'adapte en fonction de la limite de temps disponible avant le retour du véhicule au dépôt, pouvant ainsi affecter plusieurs véhicules au lieu d'un seul pour satisfaire toutes les demandes de transport et ainsi assurer une bonne gestion de la flotte disponible. Aussi, la capacité des véhicules joue un rôle dans le choix du véhicule à affecter pour faire le trajet, le programme choisi d'affecter le véhicule avec la plus grande capacité, celui-ci donnant la possibilité de prendre plus de clients à bord avant de les déposer si cela revient moins cher en termes de cout et aussi en termes de temps. Nous avons fini le chapitre par décrire les obstacles rencontrés, spécifiquement concernant la programmation des contraintes d'incompatibilités, en proposant des solutions et des pistes d'amélioration pour des projets futurs.



## CONCLUSION

Nous avons pu à travers notre recherche faire une étude sur les différents modèles d'optimisation liés au transport en général, puis au transport adapté de personnes plus en particulier. Nous avons aussi fait une vue d'ensemble sur les différentes méthodes utilisées dans la résolution de ces problèmes. Enfin, nous avons pu construire un modèle permettant à la fois de prendre en considération les contraintes d'incompatibilités entre clients et les pauses des chauffeurs.

Nous avons pu apprendre à modéliser et programmer un cas particulier d'un modèle d'optimisation de transport avec cueillette et livraison, en respectant toutes les contraintes de bases comme la gestion des flux et des capacités de véhicules. Nous avons aussi pu introduire la notion de temps dans le modèle, chose qui nous a permis de maîtriser les pauses des chauffeurs et limiter les trajets pour chaque véhicule.

Nous avons testé le modèle proposé avec deux situations différentes, dans la première où l'on a vu qu'un seul véhicule peut être suffisant pour répondre au besoin de transport de tous les clients que l'on avait dans notre tableau de données en choisissant le véhicule avec la plus grande capacité. Dans la deuxième situation où l'on a réduit l'heure de retour au dépôt, on a vu que cette fois, sous la contrainte de temps, le programme a dû affecter deux véhicules afin de répondre au besoin des clients en une même journée pour avoir une solution optimale.

Dans notre résolution, nous avons eu de la difficulté à programmer la contrainte d'incompatibilité entre clients, chose que l'on a expliqué dans notre dernier chapitre, proposant ainsi une méthode que l'on pourrait utiliser dans un projet futur afin de pouvoir réaliser une solution plus optimale. Afin d'améliorer notre modèle, on pourrait aussi ajouter comme contrainte les intervalles de temps pour la collecte ou le dépôt de chaque client. Aussi, on pourrait adapter le modèle à plusieurs utilisations spécifiques, comme dans le transport de personnes dans le domaine de la santé que ce soit pour le personnel de la santé ou les patients, en ajoutant des contraintes de priorité par exemple.



## ANNEXE I

### Tableaux de données

ti,j		di	R											
			1	2	3	4	5	6	7	8	9	10	11	12
R	1	0	0	35	37	34	48	59	54	68	50	51	55	60
	2	2	35	0	40	35	46	55	48	34	35	44	56	41
	3	1	37	40	0	50	35	46	35	44	56	46	35	38
	4	1	34	35	50	0	44	35	56	45	35	34	44	48
	5	2	48	46	35	44	0	35	42	55	48	39	38	45
	6	1	59	55	46	35	35	0	55	46	34	44	35	56
	7	-2	54	48	35	56	42	55	0	39	37	48	59	49
	8	-1	68	34	44	45	55	46	39	0	36	44	55	46
	9	-1	50	35	56	35	48	34	37	36	0	45	54	36
	10	-2	51	44	46	34	39	44	48	44	45	0	54	46
	11	-1	55	56	35	44	38	35	59	55	54	54	0	47
	12	0	60	41	44	38	48	56	49	46	36	46	47	0

Tableau 8: Temps de trajets entre différents nœud et nombre de clients par nœud

ci,j		R											
		1	2	3	4	5	6	7	8	9	10	11	12
R	1	0	35	37	34	48	59	54	68	50	51	55	60
	2	35	0	40	35	46	55	48	34	35	44	56	41
	3	37	40	0	50	35	46	35	44	56	46	35	38
	4	34	35	50	0	44	35	56	45	35	34	44	48
	5	48	46	35	44	0	35	42	55	48	39	38	45
	6	59	55	46	35	35	0	55	46	34	44	35	56
	7	54	48	35	56	42	55	0	39	37	48	59	49
	8	68	34	44	45	55	46	39	0	36	44	55	46
	9	50	35	56	35	48	34	37	36	0	45	54	36
	10	51	44	46	34	39	44	48	44	45	0	54	46
	11	55	56	35	44	38	35	59	55	54	54	0	47
	12	60	41	38	48	45	56	49	46	36	46	47	0

Tableau 9: Coûts des trajets

INC	2	3	4	5	6
2	0	0	1	0	0
3	0	0	0	1	1
4	1	0	0	0	0
5	0	1	0	0	1
6	0	1	0	1	0

Tableau 10: Tableau des incompatibilités

V	qv
1	2
2	4
3	3

Tableau 11: Véhicules disponibles et capacité des véhicules

## ANNEXE II

### Code LINGO

Sets:

!Definition des ensembles et classification des variables;

!Tous les noeuds du problème (P U D U {0,2n+1});  
RR:di;

!Capacité de chaque véhicule  
Zv variable binaire qui définit si un chauffeur est désigné pour un véhicule  
Whv temps d'attente d'un véhicule avant la pause;  
Vehicules:qv,Zv;

! cout des trajets, durée des trajets;  
Arcs(RR,RR):cij,tij;

!Variable binaire si un véhicule v effectue le trajet entre i et j  
Nombre de personnes à bord du véhicule v sortant du point i et allant vers j;  
Trajets(Vehicules,RR,RR):Xvij,Kvij;

!heure d'arrivée du véhicule v pour servir le point i (ramassage ou depot)  
Variable binaire qui définit si le véhicule v fait une pause au point i;  
Service(Vehicules,RR):Siv,Biv,Yvi;

! Gestion des incompatibilités;  
Pickups;  
inc(Pickups,Pickups);

ENDSETS

DATA:

!Ensembles;

RR=@OLE();  
Vehicules=@OLE();

!Pickups=@OLE();

!Paramètres dans le fichier Excel;

di=@OLE();  
qv=@OLE();  
m=@OLE();  
np=@OLE();  
cij=@OLE();  
tij=@OLE();  
Hds=@OLE();

```

Hfs=@OLE();
eh=@OLE();
lh=@OLE();
LL=@OLE();

!inc=@OLE();!Matrice d'incompatibilité ;

!Plages de résultats renvoyés par LINGO;

@OLE()=Xvij;
@OLE()=Yvi;
@OLE()=Siv;
@OLE()=Zv;
@OLE()=Biv;

ENDDATA

!Fonction de minimisation des couts de transport;

[EQ1] MIN=@SUM(Trajets(v,i,j)|i#NE#j: cij(i,j)*Xvij(v,i,j));

!Sujet aux contraintes;

!Partie 1: Contraintes de base (gestion du flux);
!Equ (2) du modèle;
@FOR(RR(i)|i#GE#2 #AND# i#LE#6:
    [EQ2_I] @SUM(Service(v,j)|j#NE#i #AND# j#GE#2 #AND# j#LE#11:
        Xvij(v,i,j))=1
    );

!Equ (3) du modèle;
@FOR(Service(v,i)| i#GE#2 #AND# i#LE#6:
    [EQ3_V_I] @SUM(RR(j)|j#NE#i #AND# j#GE#2 #AND# j#LE#11: Xvij(v,i,j))-
    @SUM(RR(j)|j#NE#i+5 #AND# j#GE#2 #AND# j#LE#12: Xvij(v,i+5,j))=0
);

!Equ (4) du modèle;
@FOR(Vehicules(v):
    @FOR(RR(j) | j#GE#2 #AND# j#LE#11:
        [EQ4_V_J] @SUM(RR(i)|i#NE#j #AND# i#GE#1 #AND#
            i#LE#11:Xvij(v,i,j))- @SUM(RR(i)|i#NE#j:Xvij(v,j,i))=0
        )
    );

!Equ (5) et (6) du modèle;
@FOR(Vehicules(v):
    [EQ5_V] @SUM(Arcs(i,j)|i#EQ#1 #AND# j#GE#2 #AND# j#LE#6 :
        Xvij(v,i,j))=Zv(v)
    );

@FOR(Vehicules(v):
    [EQ6_V] @SUM(Arcs(i,j)|j#EQ#12 #AND# i#GE#7 #AND# i#LE#11 :
        Xvij(v,i,j))=Zv(v)
    );

```



```

!Equ (7) du modèle;
@FOR(Trajets:[EQ7_V_I_J] @BIN(Xvij)); ! Variable binaire =1 si le vehicule v
effectue le trajet i,j

!Equ (8) du modèle;
@FOR(Vehicules:[EQ8_V] @BIN(Zv)); !Variable binaire =1 si un chauffeur est
assigné à v;

!Equ (9) du modèle;
@FOR(Service(v,i):
    [EQ9_V_I] Yvi(v,i)>=0    !Contrainte de positivité de la variable
Yvij;
);

@FOR(Trajets(v,i,j) | i#NE#j:
    [EQ_LINK] Xvij(v,i,j) <= Zv(v)
);
!Equ (10) du modèle;
[EQ10] @SUM(Vehicules(v):@SUM(RR(j)|j#GT#1 #AND# j#LT#12:Xvij(v,1,j)))<=m;

!Equ (11) du modèle;
[EQ11] @SUM(Vehicules(v):@SUM(RR(i)|i#GT#1#AND#i#LT#12:Xvij(v,i,12)))<=m;

!Partie 2: Gestion de la charge des vehicules

!Equ (12) et (13) du modèle test 3;
@FOR(Trajets(v,i,j)|i#NE#j #AND# i#NE#12 #AND# j#GE#2 #AND# j#LE#6:
    [EQ12_V_I_J] Yvi(v,j)>=Yvi(v,i)+di(j)-7*(1-Xvij(v,i,j))
);

@FOR(Trajets(v,i,j)|i#NE#j #AND# i#NE#12 #AND# j#GE#7 #AND# j#LE#11:
    [EQ13_V_I_J] Yvi(v,j)>=Yvi(v,i)+di(j)-7*(1-Xvij(v,i,j))
);

!Equ (14) du modèle;
@FOR(Service(v,i):
    [EQ14_V_I] Yvi(v,i)<=qv(v)
);

!Partie 3: Gestion des temps de service et des pauses

!Equ (15) (16) du modèle;
@FOR(Trajets(v,i,j) | i#EQ#1 #AND# j#GE#2 #AND# j#LE#6:
    [EQ_15_V_1_I] Siv(v,j)>=Siv(v,i)+Xvij(v,i,j)*tij(i,j)-(1-Xvij)*1000
);

@FOR(Trajets(v,i,j) | j#EQ#12:
    [EQ_16_V_1_I] Siv(v,j)>=Siv(v,i)+Xvij(v,i,j)*tij(i,j)-(1-Xvij)*1000
);

!Equ (17) (18) du modèle;
@FOR(Service(v,i) | i#EQ#1:

```

```

[EQ_17_V_1_I]      Siv(v,i)=Hds
);

@FOR(Service(v,i):
[EQ_18_V_1_I]      Siv(v,i)<=Hfs
);

!Equ (19) en tenant compte des pauses et de la variable Biv;

@FOR(Trajets(v,i,j) | i#GE#2 #AND# i#LE#11 #AND# j#NE#i #AND# j#GE#2 #AND#
j#LE#11:
[EQ_19_V_I_J]      Siv(v,j)>=Siv(v,i)+Xvij(v,i,j)*tij(i,j)-(1-
Xvij)*10000+Biv(v,i)*LL
);

!Conditions pour que Biv soit egal à 1 (Equ 20 et 21);
@FOR(Service(v,i):
[EQ_20_V_I_J]      siv(v,i)>=Biv(v,i)*eh
);

@FOR(Service(v,i):
[EQ_21_V_I_J]      siv(v,i)<=(1-Biv(v,i))*1000+lh-LL
);

!Equation 22, si un chauffeur est affecté il faut avoir une pause;
@FOR(Vehicules(v):
[EQ_22_V_I_J]      @Sum(RR(i) | i#NE#1 #AND# i#NE#12:Biv(v,i))>=Zv(v)
);

!Equ (23) du modèle;
@FOR(Service(v,i) | i#GE#2 #AND# i#LE#6 :
[EQ_23_V_I]      Siv(v,i+5)>=Siv(v,i)
);

!Equ (24) du modèle;
!Variable binaire Biv qui indique si on a une pause au point i;
@FOR(Trajets:[EQ24_V_I_J] @BIN(Biv));

!Equ (25) du modèle : Gestion des incompatibilités;
!@FOR(Vehicules(v):
@FOR(Arcs(i,j) | j#LE#6 #AND# j#GE#2 #AND# i#GE#2 #AND# i#LE#6:
Inc(i,j)*Xvij(v,i,j)=0
);
!);

```

### ANNEXE III

#### DEBUG Solvreur Lingo

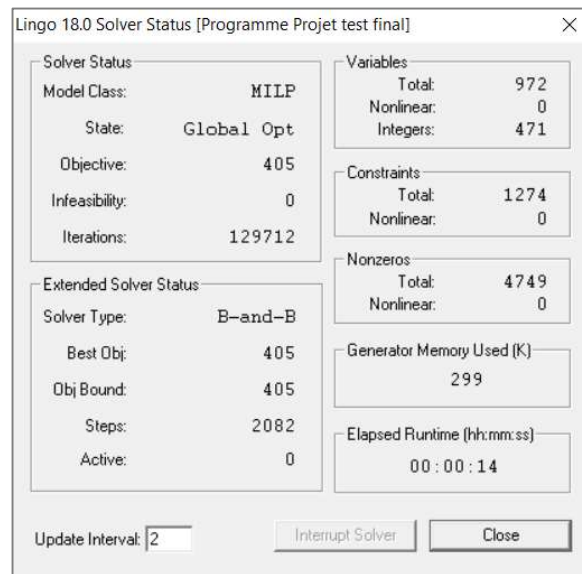


Figure 7: Solution situation 1

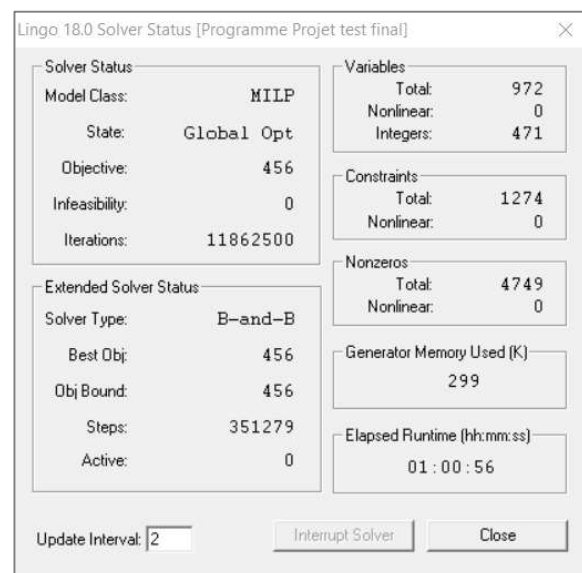


Figure 8: Solution situation 2



## BIBLIOGRAPHIE

- Berbeglia, G., Cordeau, J. F., & Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1), 8-15. doi: 10.1016/j.ejor.2009.04.024. Repéré à <http://dx.doi.org/10.1016/j.ejor.2009.04.024>
- Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., & Juan, A. A. (2014). Rich Vehicle Routing Problem: Survey. *ACM Computing Surveys*, 47(2), 32 (28 pp.). doi: 10.1145/2666003. Repéré à <http://dx.doi.org/10.1145/2666003>
- Colormi, A., Dorigo, M., & Maniezzo, V. An Investigation of some Properties of an " Ant Algorithm". Dans (1992 éd., Vol. 92).
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80-91.
- Detti, Detti, P., Papalini, F., & Lara, G. Z. M. d. (2017). A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega (Oxford)*, 70, 1-14. Repéré à [HTTP://openurl.uquebec.ca:9003/ets?ctx\\_ver=Z39.88-2004&ctx\\_enc=info%3Aofi%2Fenc%3AUTF-8&ctx\\_tim=2020-04-23T20%3A20%3A35IST&url\\_ver=Z39.88-2004&url\\_ctx\\_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx&rft\\_id=info%3Aid%2Fprimo.exlibrisgroup.com%3Aprimo3-Article-elsevier\\_s&rft\\_val\\_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rft.genre=article&rft.atitle=A%20multi-depot%20dial-a-ride%20problem%20with%20heterogeneous%20vehicles%20and%20compatibility%20constraints%20in%20healthcare&rft.jtitle=Omega&rft.btitle=&rft.aulast=Detti&rft.auinit=P&rft.auinit1=P&rft.auinitm=&rft.ausuffix=&rft.au=Detti%2C%20Paolo&rft.aucorp=&rft.date=2017-07&rft.volume=70&rft.issue=&rft.part=&rft.quarter=&rft.ssn=&rft.spage=1&rft.epage=14&rft.pages=1-14&rft.artnum=&rft.issn=0305-0483&rft.eissn=1873-5274&rft.isbn=&rft.sici=&rft.coden=&rft\\_id=info%3Adoi%2F10.1016%2Fj.omega.2016.08.008&rft\\_id=info%3Aoi%2F&rft\\_id=info%3Apmid%2F&rft.object\\_id=&sv\\_c\\_val\\_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Asch\\_svc&rft.eisbn=&req.language=fr&rft\\_dat=%3Celsevier\\_s%3Edoi\\_10\\_1016\\_j\\_omega\\_2016\\_08\\_008%3C%2Felsevier\\_s%3E&rft\\_pqid=&rft\\_%3Cprovider\\_id%3E=%7B%7Bcontrol%2F%3Cprovider\\_id%3E%7D%7D](HTTP://openurl.uquebec.ca:9003/ets?ctx_ver=Z39.88-2004&ctx_enc=info%3Aofi%2Fenc%3AUTF-8&ctx_tim=2020-04-23T20%3A20%3A35IST&url_ver=Z39.88-2004&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx&rft_id=info%3Aid%2Fprimo.exlibrisgroup.com%3Aprimo3-Article-elsevier_s&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rft.genre=article&rft.atitle=A%20multi-depot%20dial-a-ride%20problem%20with%20heterogeneous%20vehicles%20and%20compatibility%20constraints%20in%20healthcare&rft.jtitle=Omega&rft.btitle=&rft.aulast=Detti&rft.auinit=P&rft.auinit1=P&rft.auinitm=&rft.ausuffix=&rft.au=Detti%2C%20Paolo&rft.aucorp=&rft.date=2017-07&rft.volume=70&rft.issue=&rft.part=&rft.quarter=&rft.ssn=&rft.spage=1&rft.epage=14&rft.pages=1-14&rft.artnum=&rft.issn=0305-0483&rft.eissn=1873-5274&rft.isbn=&rft.sici=&rft.coden=&rft_id=info%3Adoi%2F10.1016%2Fj.omega.2016.08.008&rft_id=info%3Aoi%2F&rft_id=info%3Apmid%2F&rft.object_id=&sv_c_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Asch_svc&rft.eisbn=&req.language=fr&rft_dat=%3Celsevier_s%3Edoi_10_1016_j_omega_2016_08_008%3C%2Felsevier_s%3E&rft_pqid=&rft_%3Cprovider_id%3E=%7B%7Bcontrol%2F%3Cprovider_id%3E%7D%7D)
- Erdoğan, S., & Miller-Hooks, E. (2012). A Green Vehicle Routing Problem. *Transportation Research Part E: Logistics and Transportation Review*, 109, 100–114. doi: 10.1016/j.tre.2011.08.001

- Fagnant, D. J., & Kockelman, K. M. (2018). Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. *Transportation*, 45(1), 143-158. doi: 10.1007/s11116-016-9729-z. Repéré à <http://dx.doi.org/10.1007/s11116-016-9729-z>
- Glover, F., & Laguna, M. (1998). Tabu search. Dans *Handbook of combinatorial optimization* (pp. 2093-2229). Springer.
- Goldberg, D. E. (1989). Genetic algorithms in search. *Optimization, and Machine Learning*.
- Goss, S., Aron, S., Deneubourg, J.-L., & Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76(12), 579-581.
- Hansen, P. (1986). The steepest ascent mildest descent heuristic for combinatorial programming. Dans (pp. 70-145).
- Ho, S. C., Szeto, W. Y., Yong-Hong, K., Leung, J. M. Y., Petering, M., & Tou, T. W. H. (2018). A survey of dial-a-ride problems: literature review and recent developments. *Transportation Research Part B: Methodological*, 111, 395-421. doi: 10.1016/j.trb.2018.02.001. Repéré à <http://dx.doi.org/10.1016/j.trb.2018.02.001>
- Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- Hu, T.-Y., Zheng, G.-C., & Liao, T.-Y. (2017). A multi-objective model for dial-a-ride problems with service quality and eco-efficiency. Dans *20th IEEE International Conference on Intelligent Transportation Systems, ITSC 2017, October 16, 2017 - October 19, 2017* (Vol. 2018-March, pp. 1-6). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ITSC.2017.8317630. Repéré à <http://dx.doi.org/10.1109/ITSC.2017.8317630>
- Hu, Z.-H., Sheu, J.-B., Zhao, L., & Lu, C.-C. (2015). A dynamic closed-loop vehicle routing problem with uncertainty and incompatible goods. *Transportation Research Part C: Emerging Technologies*, 55, 273-297. doi: 10.1016/j.trc.2015.01.010. Repéré à <http://dx.doi.org/10.1016/j.trc.2015.01.010>
- Ichoua, S. (2001). *Problèmes de gestion de flottes de véhicules en temps réel*. Université de Montréal.
- Kopfer, H. W., Schonberger, J., & Kopfer, H. (2014). Reducing greenhouse gas emissions of a heterogeneous vehicle fleet. *Flexible Services and Manufacturing Journal*, 26(1-2), 221-248. doi: 10.1007/s10696-013-9180-9. Repéré à <http://dx.doi.org/10.1007/s10696-013-9180-9>
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4), 408-416. doi: 10.1287/trsc.1090.0301. Repéré à <http://dx.doi.org/10.1287/trsc.1090.0301>

- Lehuede, F., Masson, R., Parragh, S. N., Peton, O., & Tricoire, F. (2014). A multi-criteria large neighbourhood search for the transportation of disabled people. *Journal of the Operational Research Society*, 65(7), 983-1000. doi: 10.1057/jors.2013.17. Repéré à <http://dx.doi.org/10.1057/jors.2013.17>
- Liu, M., Luo, Z., & Lim, A. (2015). A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B: Methodological*, 81, 267-288. doi: 10.1016/j.trb.2015.05.009. Repéré à <http://dx.doi.org/10.1016/j.trb.2015.05.009>
- Madankumar, S., & Rajendran, C. (2018). Mathematical models for green vehicle routing problems with pickup and delivery: a case of semiconductor supply chain. *Computers & Operations Research*, 89, 183-192. doi: 10.1016/j.cor.2016.03.013. Repéré à <http://dx.doi.org/10.1016/j.cor.2016.03.013>
- Mahmoudi, M., & Xuesong, Z. (2016). Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state-space-time network representations. *Transportation Research Part B: Methodological*, 89, 19-42. doi: 10.1016/j.trb.2016.03.009. Repéré à <http://dx.doi.org/10.1016/j.trb.2016.03.009>
- Mauri, G. R., Antonio, L., & Lorena, N. (2009). Customers' satisfaction in a dial-a-ride problem. *IEEE Intelligent Transportation Systems Magazine*, 1(3), 6-14. doi: 10.1109/MITS.2009.934641. Repéré à <http://dx.doi.org/10.1109/MITS.2009.934641>
- Mazzeo, S., & Loiseau, I. (2004). An ant colony algorithm for the capacitated vehicle routing. *Electronic Notes in Discrete Mathematics*, 18, 181-186.
- Parragh, S. N., Cordeau, J. F., Doerner, K. F., & Hartl, R. E. (2012). Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR Spectrum*, 34(3), 593-633. doi: 10.1007/s00291-010-0229-9. Repéré à <http://dx.doi.org/10.1007/s00291-010-0229-9>
- Ramachandran, P. (2009). Vehicle routing problem with load compatibility constraints. Dans (pp. 339-343). IEEE. doi: 10.1109/ieem.2009.5373342. Repéré à <http://dx.doi.org/10.1109/IEEM.2009.5373342>
- Ran, L., Xiaolan, X., Augusto, V., & Rodriguez, C. (2013). Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research*, 230(3), 475-486. doi: 10.1016/j.ejor.2013.04.044. Repéré à <http://dx.doi.org/10.1016/j.ejor.2013.04.044>
- Schrijver, A. (2000). Theory of Linear and Integer Programming. *JOURNAL-OPERATIONAL RESEARCH SOCIETY*, 51(7), 892-892.

- Sidi Mohamed Douiri, S. E., Halima Lakhbab.(2011). *Cours des Méthodes de Résolution Exactes Heuristiques et Métaheuristiques*. Repéré à [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj2jv\\_81OjqAhVESN8KHfcTCXYQFjAAegQIChAB&url=ftp%3A%2F%2Fftp.fsr.ac.ma%2Fcours%2Fmaths%2FSouad%2520Bernoussi%2FCours%2520C2SI.pdf&usg=AOvVaw31pj2c\\_5n1IvmL9WP3dZ9](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj2jv_81OjqAhVESN8KHfcTCXYQFjAAegQIChAB&url=ftp%3A%2F%2Fftp.fsr.ac.ma%2Fcours%2Fmaths%2FSouad%2520Bernoussi%2FCours%2520C2SI.pdf&usg=AOvVaw31pj2c_5n1IvmL9WP3dZ9)
- Toth, P., & Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(1-3), 487-512. doi: 10.1016/S0166-218X(01)00351-1. Repéré à [http://dx.doi.org/10.1016/S0166-218X\(01\)00351-1](http://dx.doi.org/10.1016/S0166-218X(01)00351-1)
- Yong, S., Boudouh, T., Grunder, O., & Deyun, W. (2018). Modeling and solving simultaneous delivery and pick-up problem with stochastic travel and service times in home health care. *Expert Systems with Applications*, 102, 218-233. doi: 10.1016/j.eswa.2018.02.025. Repéré à <http://dx.doi.org/10.1016/j.eswa.2018.02.025>
- Zheng, J., Liu, G., Gu, Z., & Bai, X. (2017). Delivery vehicle routing problem with simultaneous delivery and pickup in E-commerce environment. Dans *2017 29th Chinese Control and Decision Conference (CCDC)*, 28-30 May 2017 (pp. 5728-5733). IEEE. doi: 10.1109/CCDC.2017.7978189. Repéré à <http://dx.doi.org/10.1109/CCDC.2017.7978189>
- Zihan, H., Ying, C., Mahmassani, H. S., & Shuang, X. (2017). Commuter ride-sharing using topology-based vehicle trajectory clustering: methodology, application and impact evaluation. *Transportation Research, Part C: Emerging Technologies*, 85, 573-590. doi: 10.1016/j.trc.2017.10.020. Repéré à <http://dx.doi.org/10.1016/j.trc.2017.10.020>