

HEC MONTRÉAL

**Algorithmes d'optimisation pour la gestion d'une flotte de taxis
électriques**

par

Charly Robinson La Rocca

Sciences de la gestion

(Option gestion des opérations)

*Mémoire présenté en vue de l'obtention du grade
de maîtrise ès sciences (M. Sc.)*

Décembre 2016

© Charly Robinson La Rocca, 2016

Sommaire

La venue de nouvelles technologies de l'information et de la communication dans le domaine des transports a provoqué l'émergence de nouveaux joueurs qui mettent à profit la puissance de ces technologies. Parmi ceux-ci, nous retrouvons Téo Taxi, un projet de Taxelco, qui se démarque par son modèle d'affaires unique au monde : les chauffeurs de taxi de Téo Taxi sont salariés au lieu d'être travailleurs autonomes. Ce système original génère de nouveaux défis en termes de gestion de flotte et d'optimisation des processus. Ce mémoire approche ce problème de front et propose des solutions à trois problèmes opérationnels : la répartition, la relocalisation et la gestion de l'autonomie des taxis électriques. Pour évaluer la performance de nos algorithmes, nous avons conçu un simulateur à événements discrets. L'outil modélise le comportement de chaque taxi et déduit l'efficacité relative des algorithmes en fonction du temps d'attente moyen et du revenu horaire par taxi. Nous avons démontré qu'une répartition par affectations peut générer des gains de l'ordre de 20% en termes de revenu par rapport à la stratégie naïve (une requête est assignée au taxi libre le plus proche au moment de l'appel). Nous proposons également une stratégie de relocalisation des taxis libres basée sur des recherches faites sur les ambulances. Enfin, pour la gestion de l'autonomie, nous présentons une méthode qui utilise des appels à la borne pour stabiliser leur utilisation dans le temps.

Mots clés : optimisation, modélisation, heuristiques, simulation, service de taxi, répartition, problème de localisation, problème de recharge, véhicules électriques, tournées de véhicules.

Abstract

The advent of new technologies in the area of transportation has led to the establishment of new entities that benefit from their capabilities. Among them, we can name Téo Taxi, a Taxelco project, which uses a unique business model : taxi drivers are employees instead of being self-employed. This shift entails new challenges regarding fleet management and process optimization. This thesis tackles this issue and proposes solutions to three operational problems : the dispatch, the relocation and the charging of electric taxis. To evaluate the performance of our algorithms, we designed a discrete-event simulator. The application models taxi behaviors and computes relative performance according to average waiting time and hourly income per taxi. We showed that dispatching taxis based on the solution to an assignment problem can provide a 20% increase in income relative to the simplistic approach (where requests are assigned to the closest taxi). Moreover, we adapted a relocation strategy designed for ambulances in order to respect the taxi's cases. At last, to manage taxi batteries efficiently, we used a strategy that involves terminal calls. This strategy stabilizes the average state of charge and eases operations overall.

Keywords : optimization, heuristics, simulation, taxi service, dispatch, location problem, charging problem, electric vehicles, routing problem.

Table des matières

1	Introduction	1
1.1	Problématique	2
1.2	Objectifs	4
1.3	Organisation du mémoire	5
2	Revue de la littérature	7
2.1	Stratégies de répartition	10
2.2	Stratégies d’assignation aux stations de recharge	12
2.3	Stratégies de localisation	14
3	Méthodologie	19
3.1	Conception d’un simulateur	19
3.1.1	Importation et traitement des données	20
3.1.2	Initialisation des objets	22
3.1.3	La simulation	22
3.1.4	Processus de répartition	23
3.1.5	Indices de performance	24
3.1.6	Interface du simulateur	24
3.2	Modélisation du comportement client	25
3.3	Conception d’un moteur de routage	27
3.3.1	Méthode par discrétisation spatiale	28
3.3.2	Méthode par régression linéaire	31
3.4	Algorithmes de répartition	32
3.4.1	Méthode de caractérisation des algorithmes	33
3.4.2	Algorithme de répartition avec rayon	35
3.4.3	Algorithme de répartition avec file d’attente	36
3.4.4	Algorithme de répartition par affectation	37
3.5	Algorithme de gestion de l’autonomie	39

3.6	Algorithmes de localisation	41
3.6.1	Modèles exacts	41
3.6.2	Heuristique	45
4	Présentation des résultats	47
4.1	Validation du simulateur	47
4.1.1	Validation théorique	47
4.1.2	Validation expérimentale	49
4.2	Caractérisation des algorithmes de répartition	50
4.2.1	L'algorithme de répartition avec rayon	51
4.2.2	L'algorithme de répartition avec file d'attente	52
4.2.3	L'algorithme de répartition par affectation	53
4.3	Algorithme de gestion de l'autonomie	54
4.4	Caractérisation de l'algorithme de localisation	58
5	Conclusion	63
5.1	Synthèse des résultats	63
5.2	Contribution	64
5.3	Limites	64
5.4	Perspectives	65

Table des figures

1	Processus de répartition	23
2	Interface du simulateur	25
3	Le taux de <i>Balking</i> des clients en fonction du temps d'attente affiché	26
4	Illustration de la fonction <code>add_vertex()</code>	29
5	Maillage de la zone généré par CGAL avec un critère de 2.7 km pour la taille des régions	29
6	Maillage de la zone raffiné avec <code>crit=1000</code>	29
7	Illustration des voisins utilisés pour faire l'interpolation	30
8	Régression linéaire du temps de déplacement en fonction de la distance	32
9	Évolution d'un indice de performance dans le temps	34
10	Fonction de sélectivité	36
11	Distribution de la demande en pourcentage arrondi à l'unité	42
12	Journées types et leur demande	50
13	Revenus horaires réels et simulés pour la journée du 20 juin	50
14	Revenus horaires réels et simulés pour la journée du 22 juin	50
15	Performance relative de l'algorithme de répartition avec rayon fixe de 2500 m	52
16	Performance relative de l'algorithme de répartition avec rayon fixe de 3500 m	52
17	Performance relative de l'algorithme de répartition avec fonction de sélectivité. Pente = 650 m/(\$/h/taxi)	53
18	Performance relative de l'algorithme de répartition avec fonction de sélectivité. Pente = 1000 m/(\$/h/taxi)	53
19	Performance relative de l'algorithme de répartition avec file d'attente	54
20	Performance relative de l'algorithme de répartition avec affectation	55
21	Performance relative de l'heuristique pour gestion de l'autonomie avec un seuil $T2 =$ 75%	56
22	Performance relative de l'heuristique pour gestion de l'autonomie avec un seuil $T2 =$ 50%	56
23	Évolution temporelle du niveau de charge avec un seuil $T2 = 0\%$	57
24	Évolution temporelle du niveau de charge avec un seuil $T2 = 75\%$	57

25	Temps de calcul pour la résolution du problème de localisation avec deux taxis	59
26	Temps de calcul pour la résolution du problème de localisation avec cinq taxis	59
27	Performance relative de l'heuristique de localisation avec une période de 10 minutes . .	60
28	Performance relative de l'heuristique de localisation avec une période de 5 minutes . . .	61
29	Performance relative de l'heuristique de localisation avec une période de 60 minutes . .	61

Liste des tableaux

1	Description des champs de la table <i>rides</i>	21
2	Description des indices de performance	24
3	Résultats de l'évaluation de la précision de la méthode par discrétisation spatiale	30
4	Temps de déplacement en secondes pour un ensemble de courses semblables	31
5	Résultats de l'évaluation de la précision de la méthode par régression linéaire	32
6	Capacités des stations de recharge	40

Liste des abréviations

nt	Nombre de taxis
dd	Densité de la demande (nombre de clients par heure)
dt	Un pas de temps
ETA	Délai entre le moment de la requête et l'arrivée du taxi au point d'origine (<i>estimated time of arrival</i>).
Rh	Revenu horaire par taxi
TU	Taux d'utilisation de la flotte
SOC	Niveau de charge d'un taxi électrique (<i>state of charge</i>)

1 Introduction

Le service de taxi est un des moyens de transport les plus utilisés dans les régions urbaines. Plusieurs éléments font de lui un service indispensable. Contrairement au transport en commun, les points de cueillette et de dépôt ne sont pas fixes. De plus, il n'est pas contraint par un horaire prédéterminé. Cette flexibilité vient avec un prix. C'est pourquoi le service de taxi est considéré par plusieurs comme une solution de rechange lorsque le transport en commun n'est pas disponible. Il est particulièrement apprécié par le milieu des affaires pour le confort et l'intimité qu'il offre.

L'industrie du taxi à Montréal est encadrée par une réglementation gérée par l'état : la Loi concernant les services de transport par taxi (Gouvernement du Québec, 2015b). Elle a comme objectifs d'améliorer la qualité des services offerts à la population, de protéger le consommateur et d'assurer la pérennité de l'industrie. La réglementation encadre les questions d'inspection, de tarification, de permis et les obligations des différents acteurs. De plus, les chauffeurs de taxi doivent déboursier un montant substantiel pour se procurer une licence. Elle agit comme barrière à l'entrée. Par défaut, le chauffeur de taxi est un travailleur autonome et cela a des implications importantes sur la gestion de l'offre de taxi. Dans un monde où chaque taxi agit de façon autonome, l'information et le contrôle sont décentralisés. Dans ces conditions, il est difficile de répondre à la demande efficacement. Les entreprises de répartition de taxis comme Taxi Diamond ou Taxi Hochelaga aident à synchroniser l'offre avec la demande de taxi. Le chauffeur peut décider de payer un montant mensuel fixe en échange d'information sur la demande.

La réglementation du service de taxi est critiquée par plusieurs (le lecteur peut se référer à Salanova et al. (2011) pour une revue de la littérature sur le sujet). Elle limite notre capacité à optimiser le système puisque le contrôle des véhicules est partiellement décentralisé. La compagnie de taxi n'a qu'un contrôle limité sur sa flotte : les chauffeurs sont libres de choisir leurs horaires et leurs déplacements. Dans l'optique d'améliorer le niveau de service, Taxelco a lancé le projet Téo Taxi qui propose un nouveau modèle d'affaires. Téo Taxi agit comme répartiteur, mais il est également le propriétaire des licences de taxi et des véhicules. Ainsi, les chauffeurs de taxi sont salariés au lieu d'être travailleurs autonomes. L'avantage majeur de ce modèle est que l'information et le contrôle sont centralisés. Le potentiel d'optimisation est beaucoup plus important dans ce cas. Il y a moins d'incertitude par rap-

port aux comportements des chauffeurs. Nous pouvons leur donner des indications et prédire avec confiance que l'action sera réalisée. Le chauffeur n'a plus l'option de refuser une requête. Les directives aux chauffeurs peuvent prendre plusieurs formes. Il y a évidemment les ordres de cueillette : se déplacer vers un client et l'amener à sa destination. Les taxis devront également être repositionnés de manière à anticiper la demande. Un ordre de relocalisation pourrait être donné périodiquement aux taxis libres. Par ailleurs, tous leurs véhicules sont électriques. Cela entraîne des défis supplémentaires en particulier quant à la gestion de la charge des batteries. Pour gérer efficacement cet enjeu, le système devra donner des indications aux chauffeurs pour les retours aux stations de recharge.

Pour convenir à ce nouveau modèle d'affaires, les stratégies de gestion de flotte de taxis doivent être adaptées. Les opportunités de recherche sont importantes. Pour le moment, la littérature s'intéresse exclusivement aux enjeux qui touchent les entreprises de taxi conventionnelles où le contrôle est décentralisé. Ce mémoire explore l'autre côté de la médaille. Une gestion centralisée d'une flotte présente en théorie plusieurs avantages. Il reste à le démontrer. Pour valider le modèle d'affaires, il est nécessaire d'optimiser les stratégies et les méthodes de gestion de flotte. À cette fin, des algorithmes qui automatisent la prise de décision devront être conçus. Un travail important doit être fait pour découvrir les solutions profitables. Ce mémoire en propose plusieurs qui pourraient être implémentées par Téo Taxi. L'optimisation du transport est critique à plusieurs égards. Elle permet avant tout de maximiser les profits de l'entreprise et de satisfaire les besoins des citoyens. Lorsque le déplacement des taxis est fait de façon optimale, nous limitons les gaspillages (énergie, usure des actifs, etc.) et le trafic. Cela favorise l'innovation chez les concurrents et stimule l'économie de la région.

1.1 Problématique

L'industrie du taxi a subi plusieurs bouleversements au cours des dernières années. Ces bouleversements s'expliquent entre autres par l'apparition de plusieurs nouveaux acteurs sur le marché. Nous retrouvons Communauto, Car2go, UberX et Téo Taxi. Les deux premières entreprises offrent des véhicules en libre-service (VLS) :

« Avant la tenue de la consultation publique de 2013, deux entreprises privées avaient manifesté de l'intérêt pour développer un système de VLS. Ces projets ont vu le jour : Communauto (le projet Auto-

mobile : voitures hybrides ou électriques) et Daimler (le projet Car2go : voitures Smart à essence). Le véhicule doit être rapporté à l'intérieur d'un périmètre précis dans le cas d'Automobile/Communauto tandis que l'entreprise Car2go a pour caractéristique d'offrir la possibilité de rapporter le véhicule à l'intérieur des territoires desservis. » (Commission sur le transport et les travaux publics, 2014)

Le service de VLS se distingue de l'autopartage de la façon suivante : « L'autopartage permet de réserver le véhicule à l'avance et nécessite de le rapporter au même endroit (point A - point A); cependant, le stationnement se fait majoritairement hors rue. Le véhicule libre-service (VLS) permet aux usagers de faire des allers simples d'un point d'origine à différents endroits (point A - points B, C, ou D); il peut être stationné sur rue où la réglementation le permet, à l'intérieur du périmètre desservi. Toutefois, ce mode ne permet pas de réservation à l'avance et mise davantage sur une utilisation en courte durée par une tarification compétitive à la minute. » (Commission sur le transport et les travaux publics, 2014). L'offre actuelle d'autopartage traditionnel est assurée par Communauto sur le territoire de la Ville de Montréal.

UberX et Téo Taxi compétitionnent plus directement avec l'industrie du taxi. Au sens de la loi, ce sont des projets pilotes. Ils se distinguent du service traditionnel par une infrastructure technologique moderne et une nouvelle gestion des ressources humaines (chauffeurs amateurs pour UberX versus chauffeurs salariés pour Téo Taxi). Ils visent à expérimenter ou à innover en matière de service de transport par taxi et d'étudier, améliorer ou définir des normes applicables (Gouvernement du Québec, 2015a).

Le défi de Taxcelco est le suivant : Comment bonifier son offre de service de manière à rester pertinent malgré la compétition accrue et rester profitable? Ce projet de recherche s'inscrit dans cette initiative. Le moyen utilisé est l'optimisation de la gestion de la flotte de taxis chez Téo Taxi. Nous décomposons le problème de gestion de flotte en trois sous-problèmes : un problème de répartition, un problème de gestion de l'autonomie et un problème de localisation.

Problème de répartition (PR) : Il constitue l'élément principal de cette étude. La solution au problème répond à la question : Comment assigner les clients aux taxis? Tout d'abord, il y a un processus

de sélection des clients. Ensuite, la solution choisit un taxi pour chaque client. Enfin, elle établit le moment où la requête sera satisfaite. En d'autres mots, la solution détermine comment desservir la demande : l'ensemble des mouvements des taxis qui résulte en la satisfaction des besoins clients. L'objectif est de maximiser la rentabilité tout en assurant un niveau de service de qualité.

Problème de gestion de l'autonomie (PGA) : Les véhicules électriques ont une autonomie qui est définie par la capacité de leur batterie. La solution au problème répond à la question : Quand et où un véhicule doit aller se recharger ? Il s'agit d'un problème de gestion de la capacité. L'autonomie est une contrainte et nous cherchons à la satisfaire tout en minimisant les effets négatifs sur la rentabilité. À cela s'ajoute une contrainte de capacité aux stations de recharge qui doit être prise en compte pour éviter les files d'attente aux bornes.

Problème de localisation (PL) : Pour minimiser les temps morts, les taxis libres doivent être repositionnés vers les zones de forte demande. La solution au problème répond à la question : Comment distribuer les taxis libres sur le territoire à desservir ? Ce problème est complémentaire au problème de répartition. Il s'attarde à satisfaire la demande future le mieux possible en déplaçant proactivement les taxis.

1.2 Objectifs

Les objectifs de ce projet de recherche sont les suivants :

1. Concevoir des solutions aux trois problèmes énoncés. Ces solutions doivent être faciles à opérationnaliser.
2. Caractériser et comparer des solutions en fonction de deux indicateurs de performance : le revenu horaire et le temps d'attente moyen des clients.
3. Offrir des pistes d'amélioration aux solutions présentées.

Ce projet a été réalisé en partenariat avec Taxelco. Les intentions de recherche ont été alignées avec les intérêts de l'entreprise. En ce sens, les algorithmes doivent pouvoir être intégrés dans un système de gestion de flotte opérationnel. À terme, le système devra pouvoir gérer plusieurs milliers de taxis. Nous privilégions donc les algorithmes avec une structure légère et facile à maintenir. C'est pourquoi

nous avons opté pour un système modulaire : il a été séparé en trois sous-problèmes. Cela donne beaucoup de flexibilité et de transparence au système. En contrepartie, la modularité implique qu'il n'est pas possible de faire une optimisation globale qui donnerait de meilleurs résultats en termes de rentabilité. C'est un compromis que nous sommes prêts à faire. Par ailleurs, la particularité du problème de gestion de flotte de taxis est que l'horizon de planification est quasi nul. L'information sur la demande est connue en temps réel. Le temps de calcul doit être de quelques secondes. Nous voulons des algorithmes faciles à appliquer ; l'utilisation d'heuristiques est à privilégier. Pour les valider et les caractériser, nous avons conçu un simulateur à événements discrets construit sur mesure pour les besoins du problème. Il reproduit des journées types de demande grâce aux données historiques de Téo Taxi. Les déplacements et les états des taxis sont modélisés individuellement. Le simulateur évalue les indicateurs de performance pour différents scénarios pour pouvoir ensuite les comparer.

1.3 Organisation du mémoire

Ce mémoire est divisé en cinq sections. La section 2 fait une revue de la littérature sur les modèles de gestion de flotte de taxis. Quelques articles sont détaillés et nous notons leurs contributions. Nous mentionnons l'état de la littérature en ce qui concerne les problèmes PR, PGA et PL. La section 3 présente la méthodologie employée pour atteindre les objectifs. Elle explique la procédure de traitement des données, la structure du simulateur, les hypothèses du modèle, et la forme explicite des solutions pour les trois sous-problèmes. La section 4 contient les résultats expérimentaux et leur analyse. Nous comparons et caractérisons les stratégies de répartition et nous étudions les impacts des stratégies de localisation et de gestion de l'autonomie. La section 5 conclut ce document. Elle énonce ses limites et les perspectives de recherche qui en découlent.

2 Revue de la littérature

La modélisation de services de taxi est encore peu présente dans la littérature. Salanova et al. (2011) présentent une revue de la littérature sur le sujet. L'article met l'accent sur les enjeux organisationnels et légaux. Il organise les modèles en deux catégories : les modèles agrégés et équilibrés. Dans les deux cas, le service de taxi est observé de manière macroscopique. La plupart des études ont comme objectif d'évaluer certaines décisions stratégiques comme la taille de la flotte (Yang et Wong, 1998) (modèle par réseau) ou les impacts de la réglementation (De Vany, 1975) (modèle analytique). Ce dernier compare des marchés organisés sous un monopole franchisé, un système de permis et un système sans barrière à l'entrée. De Vany (1975) utilise un modèle analytique basé sur le travail de Douglas (1972) qui est un précurseur dans le domaine des modèles agrégés. Il évalue le prix optimal au taximètre d'un marché en relation avec le taux d'utilisation de la flotte et le bien-être des clients (le temps d'attente). La limite principale de ces modèles est qu'il est difficile d'y intégrer des considérations opérationnelles comme la stratégie de répartition. Il est tout de même possible de le faire. Daganzo (1978) présente un modèle analytique pour comparer différentes stratégies de répartition pour les autobus. Le modèle s'applique aux taxis ; il suffit de fixer la capacité des véhicules à 1. Le problème est formulé de la façon suivante. D'une part, nous supposons un système à l'équilibre : $n = N$ où N est le nombre d'appels en attente et n le nombre de clients à bord d'un taxi. Soit λ_b le taux de cueillette d'un véhicule et $\lambda = M \cdot \lambda_b$ le taux de cueillette total, où M est le nombre de véhicules. Le taux λ_b se calcule de la façon suivante :

$$\lambda_b = \frac{1}{b_1 + b_2 + 2 \cdot d_{2N}/v}. \quad (1)$$

Le dénominateur est le temps total pour réaliser une course. Les paramètres b_1 et b_2 sont les temps d'embarquement et de dépôt. Le dernier terme représente les temps de déplacement (d étant la distance et v la vitesse). Nous présumons que les temps pour atteindre le client et le servir sont les mêmes, car nous sommes à l'équilibre (d'où le facteur 2). La distance à parcourir dépend du nombre de points potentiels ($n + N = N + N = 2N$ dans ce cas) et de la forme du réseau routier modélisé par le facteur r . L'approximation de d_n estime la distance moyenne entre un point aléatoire et le point le plus proche

parmi n points distribués aléatoirement dans une région ayant une aire A :

$$d_n \approx \frac{r}{2} \sqrt{\frac{A}{n}}. \quad (2)$$

Après avoir fait les substitutions appropriées, nous trouvons

$$\frac{M}{\lambda} = b_1 + b_2 + \frac{r\sqrt{A}}{v} \cdot \frac{1}{\sqrt{2N}}. \quad (3)$$

Cette équation est résolue pour N , ce qui nous permet de déduire les indicateurs de performance comme le temps attente ω et le temps de déplacement u :

$$\omega = \frac{N}{\lambda} \quad u = \frac{NM}{\lambda}. \quad (4)$$

La précision et la portée d'un tel modèle sont discutables. L'auteur explique que le modèle n'est pas précis lorsque le système est peu congestionné (quand N est petit). Le système doit absolument être à l'équilibre. De plus, il n'est pas possible d'évaluer la distribution statistique des indicateurs ou de prendre en compte la distribution spatiale de la demande et de l'offre de taxi (Grau et Romeu, 2015). Malgré les limitations des modèles analytiques, ils sont encore utilisés aujourd'hui pour étudier les implications des systèmes de taxis électriques. Sathaye (2014) présente un modèle exhaustif basé sur celui de Daganzo (1978) pour comprendre les compromis entre différents types de véhicules (hybride, électrique, à essence). Son modèle permet d'accéder au coût optimal associé à chaque option. Les résultats montrent que le système de taxis électriques engendre des coûts comparables au système conventionnel.

La modélisation par simulation est une autre approche utilisée pour représenter un système de taxi. Cette méthode se démarque par sa popularité (voir par exemple Kümmel et al. (2016), Lee et al. (2004), Lu et al. (2012), Maciejewski et Nagel (2013) Hasheminezhad et Bahreininejad (2010), Seow et al. (2010)). La grande majorité des articles qui traitent la question des règles de répartition utilise un simulateur pour caractériser les stratégies. Cette pratique est légitime, car la simulation est beaucoup plus flexible que les modèles analytiques et permet de modéliser chaque taxi individuellement. Dans ce qui suit, nous résumons l'article de Grau et Romeu (2015) qui propose un cadre de référence pour la modélisation de services de taxi par simulation. Il constitue à notre avis un bon point de référence à notre

recherche.

Grau et Romeu (2015) présentent une approche par simulation à événements discrets. Chaque taxi est modélisé individuellement ; c'est un modèle par agents. Les comportements, les buts et les règles sont définis pour chaque taxi. Le simulateur utilise un réseau pour modéliser les routes et les déplacements effectués. Les liens entre les noeuds sont caractérisés par une distance et le flux de véhicules. L'intensité du flux de véhicules permet de déduire la vitesse du trafic et par conséquent celle des taxis. Le calcul des vitesses se base sur la fonction de performance de Sheffi (Sheffi, 1985). L'algorithme de Dijkstra (Dijkstra, 1959) (algorithme du plus court chemin) est l'outil employé pour évaluer la distance entre une paire de noeuds dans une structure en réseau. Sachant la vitesse et la distance, il est possible d'évaluer le coût d'une course avec la logique utilisée sur le marché (Commission des transports du Québec, 2013). Le coût est calculé de façon incrémentale. À chaque pas de temps, sa valeur est mise à jour en fonction des deux paramètres considérés : le temps et la distance.

Le modèle inclut un module de décision pour simuler le comportement des chauffeurs de taxi. Le module suppose que les taxis possèdent de l'information sur la demande sur chacune des intersections. La probabilité de choisir une intersection parmi celles disponibles dépend de son attrait. Le niveau d'attrait est égal au nombre de déplacements sur le lien pendant les N dernières simulations. Cette manière de calculer l'attrait suppose un processus d'apprentissage chez les chauffeurs de taxi. Le lecteur peut se référer à la thèse de Song (2006) s'il désire en savoir davantage sur la modélisation du processus de décision et d'apprentissage des taxis.

Le modèle a été appliqué à un cas concret, soit une flotte de taxis à Barcelone. Les résultats montrent que le nombre de taxis par heure et par kilomètre carré doit être plus grand que 28. Sinon, le temps d'attente moyen devient trop important. Au-dessus de 30 taxis/h/km², les chauffeurs devront assumer des pertes. Pourtant, la densité de taxis optimale est de 33-34 taxis/h/km². C'est à ce moment que le coût par course est le plus petit pour l'ensemble du système. L'auteur propose d'utiliser un mécanisme de compensation pour atteindre le point optimal.

2.1 Stratégies de répartition

La répartition ou l'assignation de taxis est un cas particulier du problème général de tournées de véhicules ou *vehicle routing problem* (VRP). Le problème consiste à déterminer les tournées que doivent emprunter les véhicules (l'offre) pour servir les clients (la demande). C'est un domaine très vaste qui comprend un très grand nombre de cas particuliers. Il peut inclure par exemple des contraintes sur la capacité des véhicules, le délai de livraison, la taille de lot ou le type de produit. Les véhicules peuvent provenir d'un ou de plusieurs dépôts d'où provient la marchandise. Parfois, des points de cueillette sont associés à des points de livraison spécifiques. Dans ce cas, nous parlons d'un problème de cueillette et de livraison (Savelsbergh et Sol, 1995). Si la demande est révélée de manière progressive, alors le problème est dynamique (Pureza et Laporte, 2008). La plupart des problèmes étudiés dans la littérature sont statiques : la demande est connue à l'avance. Ce type de problème est généralement modélisé à l'aide d'un réseau et il est résolu à l'aide d'une méthode d'optimisation exacte si le problème est suffisamment petit. Cette approche a été utilisée pour modéliser des flottes de taxis (Wang et al. (2014), Wang et Cheu (2013)). Ces modèles sont utiles pour évaluer les décisions stratégiques comme la taille de la flotte ou la position des stations de recharge. En ce sens, ils sont semblables aux modèles analytiques présentés précédemment. Le désavantage de ces modèles est qu'il est plutôt difficile d'y implémenter une stratégie de répartition sophistiquée à cause de la lourdeur du modèle. Il est possible de le faire (Wang et al., 2014), mais la stratégie n'est pas pertinente si le service de taxi n'inclut pas de réservations, ce qui est souvent le cas.

La plupart des stratégies de répartition que nous retrouvons dans la littérature supposent un système de taxi conventionnel où les taxis sont autonomes et travaillent à leur propre compte. Elles cherchent un compromis entre les intérêts des clients et ceux des chauffeurs (Kümmel et al., 2016). Alshamsi et al. (2009) divisent le territoire en zones. Les taxis potentiels sont ceux dans les zones voisines de la zone du client. Le taxi sélectionné est celui qui était en attente depuis le plus longtemps. Hasheminezhad et Bahreinejad (2010) attribuent une «énergie» à chaque taxi. Cette énergie influence la probabilité d'obtenir un client. Elle augmente lorsque le taxi est en attente et diminue lorsqu'on lui assigne une course. Ce mécanisme permet de répartir équitablement la demande de manière à ce que chaque chauffeur réalise le même revenu. Kümmel et al. (2016) utilise un algorithme de couplage pour trouver l'équilibre

de Nash du problème. L'avantage de cette méthode est qu'elle garantit une solution stable. En d'autres mots, le point d'équilibre assure que personne n'a intérêt à changer son état. La méthode se distingue des autres approches qui cherchent plutôt le point optimal qui est moins stable en général.

Ces trois stratégies ne sont pas applicables à notre problème. Les chauffeurs chez Téo Taxi sont des employés. Leur revenu est donc garanti. Nous nous intéressons plutôt aux algorithmes qui minimisent le temps d'attente pour les clients ou maximisent le revenu global pour l'entreprise. Les articles de Maciejewski mettent en évidence plusieurs stratégies qui sont plus adaptées à notre situation. À travers ces articles, nous pouvons identifier au moins quatre différentes stratégies de répartition (Maciejewski et Nagel (2013), Maciejewski et al. (2016)) :

1. La première est la plus simple. Elle consiste à assigner le client au taxi libre le plus proche au moment de la requête. La notion de plus proche est définie de différentes façons dans la littérature : certains utilisent la norme euclidienne, d'autres utilisent l'algorithme du plus court chemin de Dijkstra. Lee et al. (2004) considèrent le chemin qui permet d'atteindre le client le plus rapidement possible. Pour ce faire, ils considèrent l'état du trafic en temps réel. Cette première méthode a l'avantage d'être simple (peu de temps de calcul). Par contre, sa performance se détériore lorsque le taux d'utilisation augmente. Elle ne prend pas en compte la disponibilité prévue des taxis occupés.
2. Cela nous amène à la deuxième stratégie qui assigne le taxi (libre ou non) le plus près en temps. Pour appliquer cette stratégie, la destination des courses doit être connue à l'avance.
3. La troisième stratégie autorise les réaffectations. En cours de route vers un client, il est possible d'observer un retard. Si c'est le cas, le client peut être affecté à un autre taxi qui serait plus près que le premier. Jusqu'à maintenant, le principe du premier arrivé premier servi s'applique. Les demandes sont traitées individuellement et dans l'ordre d'arrivée. Cette approche est sous-optimale en période de forte demande. Le principe d'équité doit être violé pour diminuer le temps d'attente moyen pour l'ensemble des clients.

4. C'est ce que fait la quatrième stratégie. Elle cumule les demandes et résout un problème d'affectation pour allouer simultanément les taxis aux clients qui attendent d'être servis. Cette idée n'est pas nouvelle. Il s'agit de la stratégie à privilégier pour les problèmes de cueillette et livraison avec fenêtres de temps (Pureza et Laporte, 2008). Le problème d'association est équivalent au problème de minimisation des flots dans un réseau où les poids des arcs sont les temps d'attentes. Ainsi, un modèle en nombres entiers est résolu chaque fois qu'une nouvelle information est connue (nouvelle requête, retard d'un taxi, etc.), mais ce problème se résout très facilement.

2.2 Stratégies d'assignation aux stations de recharge

La différence majeure entre les véhicules à essence et électriques est l'autonomie. Les modèles dans la littérature négligent généralement la contrainte d'autonomie des véhicules et pour de bonnes raisons. Il est facile de démontrer que ce n'est pas une préoccupation importante. Nous supposons un quart de travail de 8h pour un chauffeur de taxi. S'il se déplace à la vitesse de 30 km/h en moyenne, il parcourra 240 km au cours de sa journée de travail. Cette valeur est bien en dessous de l'autonomie d'une voiture à essence conventionnelle (entre 400 et 500 km) et comparable à l'autonomie d'une voiture électrique (entre 150 et 350 km selon le modèle). Le retour vers la station d'essence ou la borne de recharge peut raisonnablement être ignoré. Il est vrai que le délai nécessaire pour recharger un véhicule est plus grand que le délai pour faire un plein d'essence. En pratique, ce n'est pas un problème. La plupart des compagnies de taxis électriques utilisent un système de permutation. À la station, le chauffeur change de véhicule ou, si possible, une nouvelle batterie y est installée. Ainsi, sur un quart de 8h, le temps perdu pour recharger un véhicule peut être estimé à 30 minutes (15 minutes pour le temps de déplacement et 15 minutes pour la permutation). Il est facile de voir que les économies de temps potentielles pouvant être obtenues par l'optimisation du processus de retour vers les stations de recharge sont mineures (15 minutes sur 8h). Tout de même, il est important d'inclure dans le système de répartition une règle fiable pour gérer efficacement l'autonomie des taxis.

Peu de littérature existe sur le sujet. Lu et al. (2012) proposent une heuristique pour sélectionner le taxi en fonction de la demande, de la disponibilité des stations et du niveau de charge. La première étape de leur processus est d'énumérer les véhicules potentiels. Ce sont les taxis dont l'autonomie est suffisante pour se déplacer vers le client, le servir et retourner vers une borne de recharge. Ensuite,

l'heuristique évalue l'intensité de la demande et la disponibilité des stations dans la région L au temps t . Les paramètres L et t décrivent respectivement l'endroit et le moment du dépôt du client. Le taxi avec le niveau de charge le plus faible est assigné à la course si la demande est faible et la disponibilité des stations est grande. Sinon, c'est le taxi avec le niveau de charge le plus élevé qui est assigné. D'autres règles s'appliquent si des stations de permutation existent en L . Pour éviter les pannes, un taxi avec un niveau de charge plus petit que 10% est assigné à la station la plus proche. Pour caractériser l'efficacité de cette heuristique, les auteurs comparent la méthode avec une stratégie de répartition aléatoire : la demande et la disponibilité des stations sont ignorées. Les indicateurs de performance sont le nombre de courses et le temps d'attente à la station. Une diminution du temps d'attente de 31% et une augmentation du nombre de courses 8% sont observées grâce à leur heuristique. La faiblesse de cette stratégie est qu'un taxi ne peut être rechargé que lorsqu'il atteint 10% d'autonomie. Si une station est disponible à proximité du taxi, il devrait aller se recharger si la demande est faible dans sa zone. D'autre part, le choix des indicateurs est critiquable. L'étude ne considère pas le temps d'attente pour le client ni le revenu. Ces deux indices de performance sont souvent ceux que les entreprises préfèrent optimiser. Le temps d'attente aux stations n'est pas critique si la capacité excédentaire est importante.

Zhou et al. (2015) suggèrent un modèle de Stackelberg basé sur la théorie des jeux pour trouver une solution au problème d'assignation des taxis aux stations de recharge. Le modèle se divise en deux étapes. Tout d'abord, les taxis envoient une demande aux stations lorsqu'ils considèrent qu'une recharge est nécessaire. Ensuite, les stations leur retournent un prix. C'est le seul mécanisme de communication entre les deux parties. Le prix permet d'ajuster la distribution de taxis dans la ville. Sa valeur considère trois éléments : le temps de transport, le paiement pour l'électricité et le temps d'attente à la station. Ils supposent que les acteurs sont rationnels et que chaque taxi choisira la station la moins coûteuse. Ils démontrent que leur stratégie permet d'atteindre la distribution de taxis désirée après une dizaine d'itérations. Cependant, la méthode d'assignation n'a pas été comparée à des solutions de recharge. Plus de recherche serait nécessaire pour évaluer les impacts de la méthode sur les indicateurs d'intérêt.

Un enjeu important dans la modélisation d'une flotte de taxis électriques est l'estimation de l'état de charge des véhicules. C'est un défi important en soi. Les modèles actuels étudient les impacts d'un paramètre spécifique comme la vitesse (Vaz et al., 2015) ou la température (Panday et Bansal, 2015). En

pratique, ces modèles sont trop lourds pour être inclus dans les simulateurs de taxis. C'est pourquoi la plupart des articles supposent un taux de décharge constant en fonction de la distance ou du temps. Le modèle de Carpenter et al. (2014) propose une approche plus sophistiquée basée sur un réseau bayésien. Les pertes causées par l'usage de la climatisation sont également prises en compte.

2.3 Stratégies de localisation

Les problèmes de répartition et de localisation sont complémentaires. Une stratégie de répartition attribue un taxi à un client au moment de l'appel tandis qu'une stratégie de localisation assigne un taxi à une demande prévue. L'objectif est de réduire le temps d'attente en déplaçant de façon proactive le taxi vers le point d'origine du client. Selon nos recherches, aucune étude n'a été réalisée sur la localisation dynamique des taxis. Ce mémoire contribue à combler cette lacune de la littérature. Il y a par contre de la littérature sur la localisation des ambulances qui est présentée dans ce qui suit.

Le problème de localisation possède une littérature imposante. Nous ne ferons qu'un survol du sujet. Le lecteur peut se référer à Daskin (2008) pour une taxonomie plus détaillée des modèles de localisation. Avant tout, il y a les problèmes de localisation statique. Leur objectif est de trouver les emplacements optimaux pour des installations fixes (usines, entrepôts, hôpitaux, etc.). Pour ce type de problème, les modèles discrets sont privilégiés, car la plupart du temps il y a un nombre fini de positions possibles pour les installations. Aussi, il existe un très grand nombre d'outils (heuristiques, solveurs génériques) pour les résoudre efficacement. Nous dénotons au moins deux types de modèles discrets : le modèle de p -médiane et le modèle de couverture.

Les modèles de p -médiane cherchent à minimiser la distance pondérée entre la demande et les installations. Il y a deux types de variables de décision : le premier type exprime l'affectation de la demande aux installations et le deuxième indique la position des installations. Cette approche à l'avantage d'avoir une fonction objectif qui peut être exprimée en termes de coûts. Cela donne beaucoup de flexibilité au modèle, car la plupart des enjeux peuvent être exprimés en termes monétaires. Une extension possible à ce modèle serait de considérer le coût d'installation associé à chaque site. Alors, l'objectif devient de minimiser le coût total de transport et d'installation. Nous pouvons également pousser le modèle à privilégier certaines décisions en leur associant une contribution positive (ou un coût négatif). C'est

une façon de modéliser des considérations plus qualitatives.

Le problème de couverture attribue un rayon d'action aux installations. Par exemple, le cercle autour d'une installation peut représenter la région pouvant être desservie par celle-ci. L'objectif peut être de trouver le nombre d'installations nécessaire pour couvrir toute la demande ou de couvrir le plus de points de demande possible avec un nombre fixe d'installations. Ce type de modèle est pertinent dans les industries qui offrent des services d'urgence ou de très grande qualité. Le problème de localisation des ambulances en est un bon exemple (voir Brotcorne et al. (2003)). Gendreau et al. (2001) présentent une méthode pour la gestion dynamique des flottes d'ambulances. Elle comprend une stratégie pour le problème d'allocation (équivalent à la répartition des taxis) et le problème de localisation. Ce dernier est résolu à l'aide d'un modèle statique de couverture. La contrainte absolue de couverture impose que toute la demande soit satisfaite par une ambulance à l'intérieur d'un rayon de r_2 minutes. À cela s'ajoute une contrainte relative qui dicte qu'une proportion α de la demande totale doit aussi être à l'intérieur d'un rayon de r_1 minutes ($r_2 > r_1$). Ce modèle de couverture double a été conçu pour respecter les standards fixés par l'organisation Urgence Santé de Montréal. D'autres contraintes contrôlent le nombre d'ambulances dans chaque site et assurent que les temps de déplacement ne soient pas trop longs.

Le modèle proposé inclut une dimension dynamique dans le sens que la solution produite dépend de l'état actuel du système. La contribution de l'article est l'heuristique utilisée pour accélérer la résolution du problème statique. Il a été démontré que ce problème peut être résolu efficacement (environ 3 minutes) par une recherche avec tabous. Cependant, ce délai est trop important pour faire de la relocalisation en temps réel. De ce fait, les chercheurs proposent une stratégie de parallélisation basée sur un schéma maître-esclave. Pour améliorer l'efficacité de résolution, le problème est divisé en tâches qui sont assignées à des esclaves. La combinaison des sous-problèmes est réalisée par le processus maître. Leur heuristique obtient une solution avec un écart moyen de 2% par rapport à celle trouvée par CPLEX.

L'approche de Gendreau et al. (2001) offre une performance notable. Elle demande toutefois une grande expertise et une infrastructure informatique imposante. Elle ne cadre pas avec nos objectifs. Par ailleurs, l'adaptation de la méthode au problème de localisation des taxis exige un travail qui ne convient

pas à notre projet. Nous valorisons les heuristiques accessible pour les opérateurs de flotte comme celle proposée par Andersson et Värbrand (2007). Les auteurs quantifient tout d’abord le niveau de couverture P_j de chaque zone j . Ce paramètre détermine si le niveau de service est adéquat. Il dépend de la contribution c_j de la zone j qui peut être par exemple proportionnelle au nombre d’appels dans la zone pendant une période spécifique. Il tient compte également de la contribution γ^l de chaque ambulance l dans l’ensemble L_j qui contient les ambulances qui peuvent servir la zone j . Sa valeur est pondérée par l’inverse du temps de déplacement t_j^l nécessaire pour atteindre la zone j . Nous obtenons ainsi la formule suivante :

$$P_j = \frac{1}{c_j} \sum_{l \in L_j} \frac{\gamma^l}{t_j^l}. \quad (5)$$

L’impact de l’ambulance l sur la zone j est faible si le temps pour l’atteindre t_j^l est grand. La méthode identifie rapidement les zones critiques où $P_j \leq P_{min}$. Les auteurs proposent le modèle suivant pour résoudre le problème de relocalisation :

$$\min z \quad (6)$$

$$z \geq \sum_{j \in N^k} t_j^k x_j^k \quad \forall k \quad (7)$$

$$\sum_{j \in N^k} x_j^k \leq 1 \quad \forall k \quad (8)$$

$$\sum_k \sum_{j \in N^k} t_j^k \leq M \quad (9)$$

$$\frac{1}{c_j} \sum_{l \in L_j} \frac{\gamma^l}{t_j^l(x)} \geq P_{min} \quad \forall j. \quad (10)$$

Il possède une contrainte qui garantit qu’aucune zone ne soit dans un état critique (10). Son objectif est de minimiser le temps de déplacement maximal (7). Les contraintes (8) limitent le nombre de relocalisations dans l’ensemble de zones N_k et la contrainte (9) limite le nombre d’ambulances à relocaliser à M . L’affectation de l’ambulance j à la zone k est modélisée par la variable de décision x_j^k . Le modèle est suffisamment petit pour être résolu en moins de 3 secondes avec une heuristique d’énumération.

Le problème de localisation des taxis partage aussi des similarités avec le problème de localisation de véhicules pour des livraisons en charge complète. En effet, dans les deux cas, la demande est de déplacer un lot unitaire d'un point à un autre. Nous mentionnons les travaux de Powell qui ont apporté une contribution importante au domaine. L'article sur l'allocation dynamique de véhicules (Powell, 1987) propose une stratégie qui prend la décision de déplacer un véhicule chargé ou non d'une région à une autre. Les conséquences d'une telle décision sont incertaines. Pour prendre en compte la variabilité dans la demande, le modèle en réseaux utilise deux types d'arcs. Le premier type représente l'information déterministe qui correspond à la demande connue. Le deuxième type d'arcs modélise la dimension stochastique du problème. Il permet d'estimer la valeur marginale d'un véhicule additionnel dans chaque région dans le futur.

Cette stratégie ne peut être appliquée directement au problème de localisation des taxis. Elle possède plusieurs différences notables. D'une part, une partie importante de la demande est connue à l'avance pour le problème de transport de marchandises, soit 40% selon l'article de Powell. Un service de taxis sans réservations ne possède aucune information certaine sur la demande future. D'autre part, le modèle de Powell est tactique. Les variables de décision sont les flux de véhicules d'une région à une autre à un moment donné. Les véhicules ne sont pas modélisés individuellement. La gestion de taxis nécessite un modèle opérationnel qui indique à chaque véhicule un ordre précis.

3 Méthodologie

La méthodologie présente l'approche utilisée pour atteindre nos objectifs. Nous présentons tout d'abord l'outil de simulation que nous avons développé pour modéliser le comportement d'une flotte de taxis électriques. Le simulateur est un système relativement complexe qui comprend plusieurs modules. Chaque module possède une fonction qui lui est propre. Nous avons, entre autres, un module pour l'importation et le traitement des données. Il y a un module de routage qui calcule les temps de déplacement. Une autre partie du code s'occupe de la modélisation du comportement client. La section méthodologie se termine par la description détaillée des algorithmes pour les problèmes de répartition, de gestion de l'autonomie et de localisation.

3.1 Conception d'un simulateur

Après notre revue de la littérature, le choix de la simulation comme outil est naturel. Tous les articles qui traitent de notre problématique utilisent la simulation pour modéliser leur flotte de taxis. En fait, la simulation est la seule méthode qui offre la modularité nécessaire pour caractériser des stratégies de gestion de flotte. Elle permet d'avoir une résolution temporelle et spatiale des variables d'état, ce qui n'est pas possible avec les modèles analytiques. La simulation permet une gestion microscopique de la flotte; les taxis sont modélisés individuellement. Un avantage majeur de la simulation est qu'elle reproduit le processus décisionnel des systèmes réels. En pratique, un système de gestion de flotte prend une décision en fonction de l'état de chaque taxi à un point donné dans le temps. Ensuite, il assigne des directives aux taxis en changeant leur état. Ce processus est reproduit exactement dans le simulateur. Ainsi, les stratégies testées avec le simulateur peuvent être transposées quasi directement dans un système opérationnel.

Plusieurs options s'offrent à nous pour la plateforme du simulateur. La solution la plus utilisée par la littérature est MATSim (Horni et Axhausen, 2016). Ce logiciel libre de droit permet l'implémentation de simulations par agent dans le domaine du transport. Il est applicable à notre problème; Maciejewski et al. (2016) l'ont utilisé pour caractériser différentes règles de répartitions. Malgré tout, nous avons décidé de développer notre propre solution comme Kümmel et al. (2016). D'une part, nous avons

les compétences en programmation pour le faire. D'autre part, une solution sur mesure a le mérite d'être plus flexible et adaptée au problème. Nous avons plus de contrôle sur les hypothèses de base. Par exemple, nous pouvons juger des compromis à faire entre la précision et le temps de calcul. Enfin, la programmation C++ donne accès à une large collection de bibliothèques scientifiques qui sont pratiques pour la résolution de problèmes d'optimisation.

Selon les termes de Bratley et al. (1987), notre simulateur est un simulateur synchrone à événements discrets. Il est synchrone, car la taille des pas de temps est constante dans le temps. Un simulateur non synchrone met à jour ses variables d'état seulement lorsqu'un événement commence ou se termine. Cette approche est à éviter dans notre cas, elle ne permet pas d'identifier facilement les événements du type «attendre jusqu'à ce que» (Bratley et al., 1987) qui constituent la majorité de nos événements. Aussi, la programmation des simulateurs synchrones est plus simple. Ils ont par contre le désavantage d'être plus exigeants au niveau du temps de calcul.

Pour mieux conceptualiser notre simulateur, nous choisissons de le diviser en trois blocs associés à des opérations distinctes : l'importation et le traitement des données collectées, le calcul des variables d'état pour chaque pas de temps (le coeur du simulateur) et le traitement des résultats de simulation. Dans les sections qui suivent, chaque bloc de notre simulateur est expliqué en détail.

3.1.1 Importation et traitement des données

Les données proviennent essentiellement de la base de données *postgres* de Téo Taxi. Deux tables ont été utilisées : *rides* et *vehicle statistics*. La première table contient l'historique des courses réalisées depuis le début des opérations en novembre 2015. Nos analyses ont été faites sur les données recueillies jusqu'à septembre 2016. Pour chaque course, nous savons entre autres la position d'origine du client, sa destination et le montant au taximètre. L'énumération des champs exploités est dans le tableau 1. La deuxième table conserve l'état de la flotte, c'est-à-dire le nombre de taxis dans chacun des états possibles : "libre", "en pause", "en route vers un client", "client à bord", "hors service". Cette information est recueillie à toutes les 5 minutes.

Pour accéder aux données, une sauvegarde de la base de données *postgres* est téléchargée localement.

Tableau 1 – Description des champs de la table *rides*

Temps de la requête	Le moment où le client commande une course sur l'application mobile.
Montant au taximètre	Le montant en dollars facturé au client à la fin de la course.
Position d'origine	Coordonnées géographiques du client lors de la requête.
Temps à l'origine	Moment où le taxi est au point d'origine.
Position de la destination	Coordonnées géographiques de la destination spécifiée par le client.
Temps à la destination	Moment où la course se termine.
Statut de la course	Spécifie si la course a été annulée après la requête.

Cela permet d'éviter de surcharger le serveur en production lors des simulations. La librairie `libpq++` a été utilisée pour effectuer les appels SQL. Les données sont traitées à l'aide d'un script en C++ qui vérifie la validité des données. Il filtre les courses qui contiennent des données aberrantes. Voici une liste des vérifications réalisées :

1. Vérifie si le champ est vide.
2. Vérifie si la course est à l'extérieur du territoire desservi par Téo à l'été 2016. Celui-ci couvre principalement le centre-ville de Montréal.
3. Vérifie si le temps pour servir et pour atteindre le client est plus petit que 30 secondes et plus grand que 30 minutes. Nous avons établi ces bornes arbitrairement.
4. Vérifie si le montant au taximètre est plus petit que 3.5\$ et plus grand que 39\$ (3.45\$ est le minimum permis par la loi et les courses à l'aéroport sont 40\$).

Les données nettoyées ont ensuite été converties dans un format propre à la librairie Armadillo (Sanderson et Curtin, 2016). À la base, Armadillo est un outil pour la résolution des problèmes d'algèbre linéaire. Les objets `arma::mat` (ce sont essentiellement des matrices) supportent plusieurs opérations pour le traitement des données (fonctions statistiques, factorisation, ordonnancement, etc.). Ces objets possèdent une fonction `save` qui enregistre les nombres qu'elle contient dans un fichier `.data`. Ce sont ces fichiers que le simulateur importe au début de chaque simulation. Les objets d'Armdillo constituent les fondations de notre simulateur.

3.1.2 Initialisation des objets

Le coeur du simulateur comprend deux types d'objets : `client` et `taxi`. L'objet `client` contient toutes les courses qui pourront potentiellement être desservies lors de la simulation (analogue à la table `rides`). L'objet possède deux fonctions distinctes pour initialiser ses membres : `import` et `import_rng`. La première fonction importe les courses réalisées dans un intervalle donné, par exemple de 8h à 18h le 26 juin 2016. L'objectif est de reproduire le déroulement d'une journée et, ainsi, comparer la simulation avec la réalité. La deuxième fonction génère une demande aléatoire. Le nombre de demandes par heure est spécifié par l'utilisateur du simulateur. Les courses sont pigées au hasard dans la base de données.

L'objet `taxi` est associé à la table `vehicle statistics`. Il tient compte de la position et de l'état de chaque taxi, et ce en fonction du temps. L'initialisation de cet objet se résume à définir la position et l'état initial de chaque taxi. Puisque notre simulateur est stationnaire, l'impact de cette décision est négligeable si l'horizon de temps est suffisamment grand. Les taxis commencent à l'état "libre" et leur position initiale est un point quelconque sur le territoire Téo Taxi.

En plus d'initialiser certains membres, la fonction `import` de `taxi` alloue la mémoire nécessaire à la simulation. Connaissant le temps total de la simulation et le nombre de taxis, la fonction initialise les matrices (les objets `arma::mat`) qui contiennent les variables d'état. Les mécanismes qui agissent sur ces variables sont expliqués dans la section qui suit.

3.1.3 La simulation

À la base, la simulation est une boucle `for` qui itère sur tous les pas de temps. La discrétisation temporelle est gérée par le paramètre `stodt` (*seconds to dt*). Il spécifie le nombre de secondes par pas de temps `dt` et sa valeur dicte le niveau de précision temporel du simulateur. À chaque itération, nous vérifions les conditions qui déclenchent les algorithmes de gestion de flotte. Par exemple, s'il y a une requête client, l'algorithme de répartition est appelé pour lui affecter un taxi. Par la suite, une deuxième boucle balaye sur chaque taxi pour mettre à jour leurs variables d'état. L'algorithme 3.1 résume ces étapes à l'aide d'une formulation en pseudo-code.

Algorithme 3.1 Algorithme qui décrit le coeur du simulateur

```
1: for all Pas de temps do  
2:   if Requête client then  
3:     Répartition  
4:   end if  
5:   Affection aux stations de recharge  
6:   Localisation  
7:   for all Taxis do  
8:     Mise à jour des variables d'état  
9:   end for  
10: end for
```

3.1.4 Processus de répartition

Le processus de répartition décrit ci-dessous est propre à Téo Taxi qui possède un canal de commande par application mobile. Nous modélisons ce processus à l'aide de trois fonctions. Les liens entre elles sont illustrés à la figure 1.

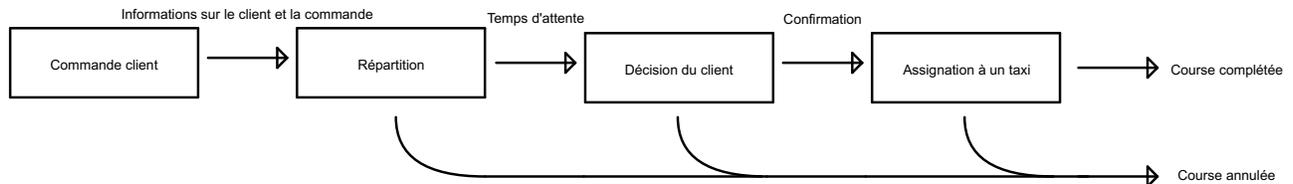


Figure 1 – Processus de répartition

Nous recevons tout d'abord une requête de la part d'un client. Elle contient de l'information sur son point d'origine et de destination. La première fonction est l'algorithme de répartition en tant que tel. Il décide comment servir le client. Il lui retourne un temps d'attente estimé. Il peut décider d'annuler la course s'il n'est pas rentable de la satisfaire. Après, le client prend une décision en fonction du temps d'attente qui lui est affiché. La section 3.2 explique comment nous avons modélisé le processus décisionnel du client. Si le client décide de commander, il sera assigné au taxi choisi par l'algorithme de répartition. C'est à ce moment que le simulateur fait appel au moteur de routage (voir section 3.3) qui calcule les deux temps de déplacement associés à la course (les délais pour atteindre et servir le client). Il est à noter que le système de Téo Taxi permet au client d'abandonner après l'assignation d'où

Tableau 2 – Description des indices de performance

Indicateur de performance	Description
Le temps d'attente ou <i>ETA</i>	La différence de temps entre le moment de la requête et le moment où le taxi arrive au point d'origine du client. Nous considérons seulement les clients qui ont été servis.
Le revenu horaire par taxi ou <i>Rh</i>	La somme des montants au taximètre pour chaque course réalisée divisée par le temps total de simulation en heures et le nombre de taxis.
Le taux d'utilisation ou <i>TU</i>	La proportion du temps où les taxis sont en route vers un client ou avec un client à bord.

la flèche entre «Assignment à un taxi» et «Course annulée».

3.1.5 Indices de performance

La simulation est maintenant terminée. La prochaine étape est d'extraire l'information pertinente sous forme d'indices de performance. Pour notre analyse nous nous sommes restreints à deux indicateurs de performance : Le temps d'attente moyen et le revenu moyen horaire par taxi. Le tableau 2 contient notre définition de ceux-ci. Nous avons choisi ces indicateurs, car ils sont les plus pertinents pour Téo taxi. Le revenu horaire mesure la rentabilité qui est en opposition avec le niveau de service : le temps d'attente. Nous prenons note toutefois du taux d'utilisation qui sera utilisé pour faire des validations.

3.1.6 Interface du simulateur

L'outil développé est une application en soi. Elle possède une interface graphique qui permet d'ajuster les différents paramètres comme le temps total de la simulation, le nombre de taxis ou le type d'algorithme utilisé pour répartir les véhicules (voir figure 2). Dans ce qui suit, nous présentons les différents éléments de l'interface et leurs fonctionnalités. De la gauche vers la droite, la première section contient un journal qui nous informe sur le déroulement de la simulation, soit le temps nécessaire pour réaliser la simulation ou certains indices de performance qu'on ne retrouve pas dans la section *KPI*. La deuxième section contient cinq onglets :

1. *Dispatch* : Ensemble des paramètres qui définissent les règles de répartition et de localisation.
2. *SOC* : Ensemble des paramètres qui définissent les règles pour la gestion de l'autonomie.

3. *Optimize* : Une interface pour produire plusieurs simulations et déduire la valeur optimale d'un paramètre.
4. *Plot* : Une interface pour produire plusieurs simulations et représenter à l'aide d'un graphique l'évolution de paramètres.
5. *UI* : Une interface qui permet de choisir le contenu des graphiques.

La troisième section, nommée *KPI*, contient les indices de performance les plus pertinents soit le temps d'attente moyen et le revenu horaire par taxi. En dessous de la liste d'indicateurs, il y a un tableau qui décrit l'état de la flotte à un temps donné ; c'est-à-dire les variables d'état pour chaque taxi. Enfin, la dernière section présente l'évolution des indices de performance au courant de la simulation à l'aide de graphiques. Par exemple, le graphique *Map* représente le positionnement géographique des véhicules et des clients qui leur sont affectés.

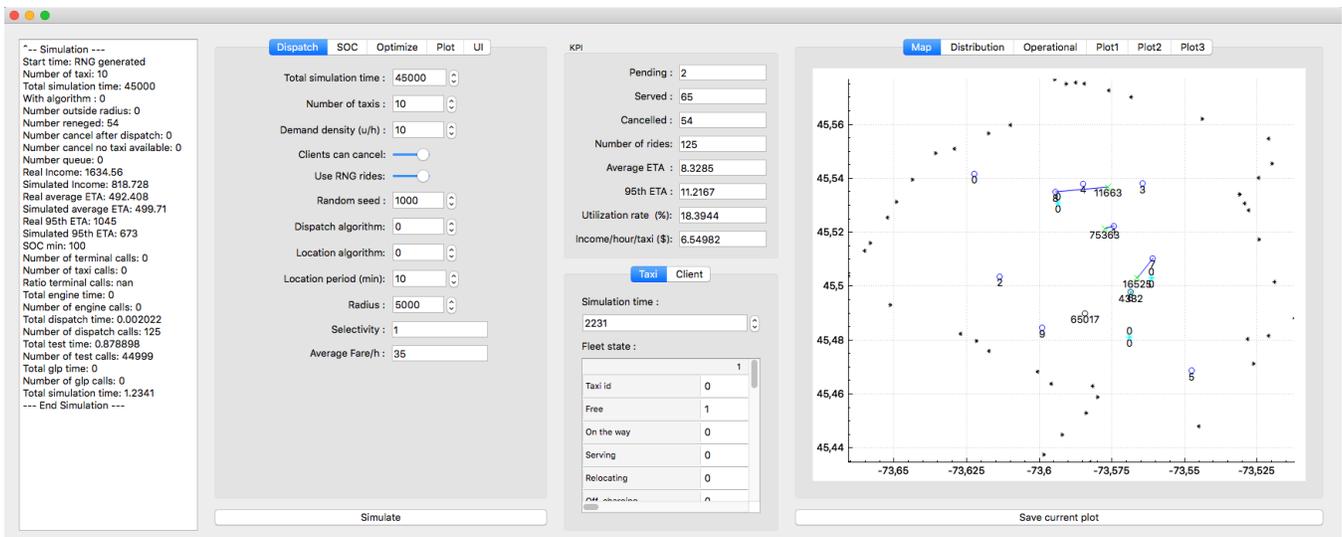


Figure 2 – Interface du simulateur

3.2 Modélisation du comportement client

La modélisation du comportement client n'est pas une tâche simple. En général, nous cherchons à comprendre comment un client réagit face à différents niveaux de service. La théorie des files d'attente explique que le comportement client peut être modélisé par deux taux d'abandon (Ross, 2015) : (1) le client abandonne au moment où il entre dans la file d'attente parce que le temps d'attente est trop

grand (*Balking*); (2) le client quitte la file parce qu'il est fatigué d'attendre (*Reneging*).

Le comportement des clients est généralement ignoré dans les modèles. Nous avons fait un effort pour l'inclure dans le nôtre, car l'information sur le nombre d'annulations est connue. Téo Taxi prend note de cette métrique. Notre modèle n'inclut que le *Balking*. L'entrée dans la file d'attente correspond à l'ouverture de l'application mobile. Ensuite, un temps d'attente est affiché à l'utilisateur. Il prend alors la décision de commander ou non un taxi. Notre objectif est de comprendre comment le temps d'attente affecte le taux d'abandon.

Par souci de confidentialité, nous ne dévoilerons pas comment cette information est déduite de la base de données de Téo Taxi. Par contre, nous pouvons divulguer la courbe utilisée dans notre simulateur (voir figure 3). Elle a été déduite à partir de données agrégées. Le niveau de précision n'est donc pas optimal. À notre avis, le modèle est suffisamment proche de la réalité pour capturer l'idée du phénomène.

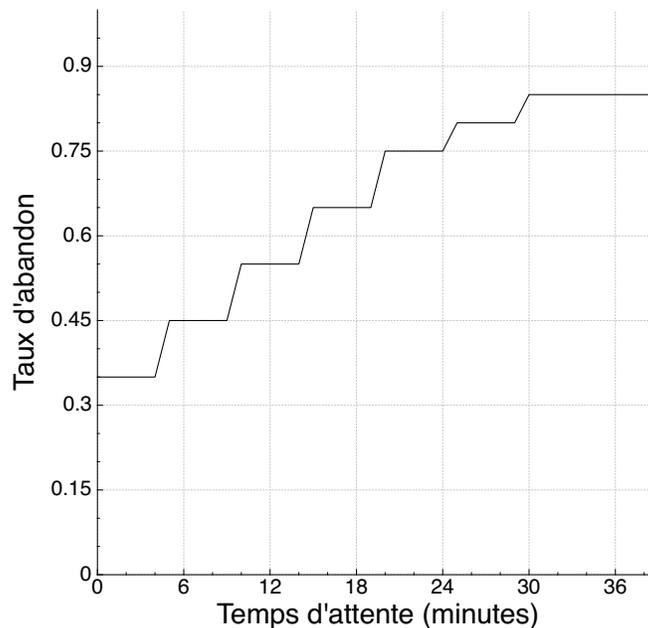


Figure 3 – Le taux de *Balking* des clients en fonction du temps d'attente affiché

Nous ignorons le *Reneging* dans notre modèle, car le phénomène est encore mal compris. Il correspond aux abandons après la requête de taxi. D'après nos études exploratoires, la corrélation avec le temps

d'attente n'est pas significative. Nous faisons l'hypothèse que ces abandons sont causés principalement par les retards des taxis, ce qui augmente le temps d'attente affiché. Cette information n'est pas suivie par le système actuel. D'ailleurs, la notion de retards n'existe pas dans notre simulateur, car les temps de déplacement sont déterministes.

3.3 Conception d'un moteur de routage

Cette section cherche à répondre à la question suivante : Quelle est la *meilleure* façon d'estimer le temps nécessaire pour se déplacer d'un point à un autre dans une ville ? Nous utilisons deux critères pour qualifier les méthodes : le temps de calcul par appel et la précision. La première partie de cette section explore les solutions existantes et la deuxième présente les solutions développées dans le cadre de ce mémoire.

Aujourd'hui, Téo Taxi appelle l'interface de *Google Map* pour obtenir des estimations du temps de déplacement. Elle prend en compte l'état de la circulation en temps réel. L'estimation est donc relativement précise. Cependant, Google ne donne pas accès aux données historiques. Un appel de l'interface nous donne une estimation au moment où il a été fait et les conditions d'utilisation ne nous permettent pas d'enregistrer les données récoltées à l'aide de l'interface (Google, 2015). Il faut dire également que le service est payant et que son utilisation dans un simulateur entraînerait des frais très importants. Pour ces raisons, cette solution doit être rejetée. Par contre, l'interface peut être utilisée en production, c'est-à-dire sur le système de gestion de flotte de Téo Taxi.

Il existe des solutions de rechange gratuites comme OSRM. Ce dernier est un moteur de routage en C++ libre de droits. Il permet de calculer un trajet à partir des données de OpenStreetMap. Le calcul est fait localement ; il ne prend pas en compte les changements récents de la route ou du trafic. D'après notre expérience avec l'outil, OSRM sous-estime systématiquement le temps de déplacement. Nous avons comparé les temps estimés par OSRM avec les données historiques ; nous obtenons une erreur systématique moyenne de 350 secondes sur un déplacement d'une durée moyenne de 755 secondes. L'erreur relative est considérable. Au final, nous avons dû rejeter cette solution à cause du temps de calcul nécessaire par appel (quelques centaines de millisecondes). L'expérience a montré que ce niveau de performance n'est pas acceptable pour faire des simulations. En effet, des centaines de simulations

sont souvent nécessaires pour obtenir des résultats statistiquement représentatifs. C'est ce qui nous amène à développer notre propre solution.

3.3.1 Méthode par discrétisation spatiale

La première méthode discrétise la zone de Téo Taxi en régions et estime le temps de déplacement entre chacune des paires de régions à l'aide des données historiques. Cette approche est en théorie très rapide, car les temps de déplacement sont calculés une seule fois et ils sont ensuite enregistrés dans une matrice. Pendant la simulation, l'estimation de l'*ETA* se résume à aller lire dans la matrice l'élément associé au trajet. Cet avantage est une motivation suffisante pour tester la technique. Ce qui suit décrit la méthode utilisée pour construire la matrice appelée `ETA_mat`.

La première étape consiste à construire le maillage qui subdivise la zone en régions. L'objet *Constrained_Delaunay_triangulation_2* de la librairie CGAL a été utilisé pour réaliser cette tâche. Pour générer un maillage, nous spécifions le polygone qui encadre la zone et la taille des mailles (ou des régions). La deuxième étape est le raffinement du maillage. Elle consiste à ajouter des mailles dans les régions où le nombre de points de demande est grand. Cela permet d'augmenter la précision de la méthode dans les régions où la densité de données historique est grande. Le processus de raffinement est décrit dans l'algorithme 3.2. Il est itératif; il ajoute des régions notées `faces` jusqu'à ce que le nombre de points dans chaque région (`faces.number`) soit plus petit que `crit`. La fonction `add_vertex()` ajoute un point au centre de la région. CGAL ajuste alors automatiquement le maillage et crée trois autres régions plus petites que la première (voir figure 4). Les figures 5 et 6 représentent respectivement le maillage avant et après le processus de raffinement. Notre algorithme donne les résultats attendus; c'est au centre-ville que la densité de régions est la plus grande.

La troisième étape est la construction de la matrice `ETA_mat`. Son type est `arma::mat` et elle possède une taille $nf \times nf$, où nf est le nombre de régions du maillage raffiné. La valeur de l'élément i, j est la moyenne des temps de parcours qui ont comme origine la région i et comme destination la région j . Pour les paires de régions vides (sans données historiques), l'élément correspondant est estimé par interpolation. La valeur de l'élément vide est la moyenne des éléments voisins non vides. L'algorithme ne considère que les voisins les plus proches $n1$ à $n6$ (voir figure 7). Le processus d'interpolation itère

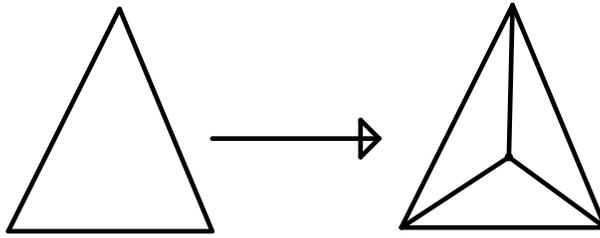


Figure 4 - Illustration de la fonction `add_vertex()`

Algorithme 3.2 Algorithme de raffinement du maillage

```

1: while au moins une région possède plus que crit points do
2:   for all faces do
3:     if faces.number > crit then
4:       faces.add_vertex();
5:     end if
6:   end for
7: end while

```

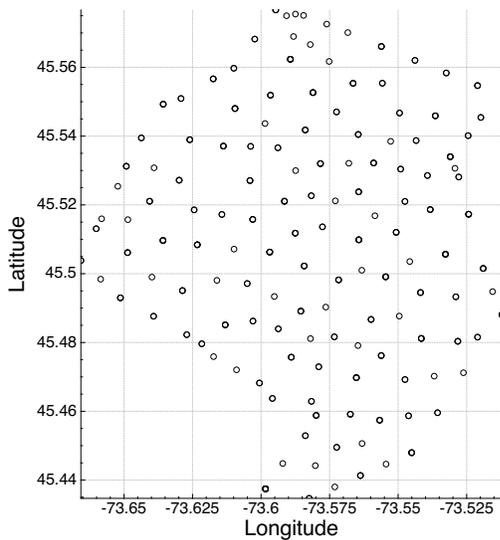


Figure 5 - Maillage de la zone généré par CGAL avec un critère de 2.7 km pour la taille des régions

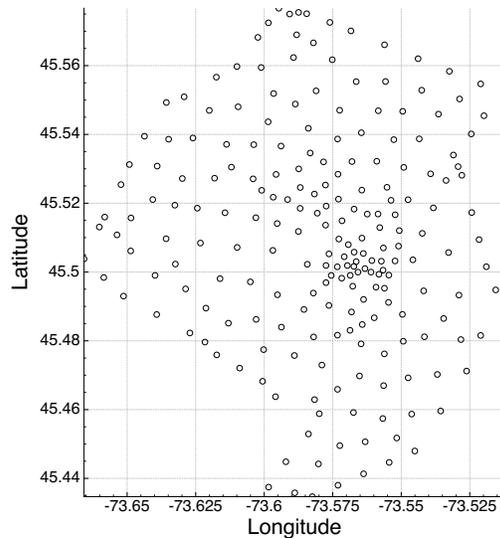


Figure 6 - Maillage de la zone raffiné avec `crit=1000`

jusqu'à ce tous les éléments de `ETA_mat` soient non vides.

Caractérisation

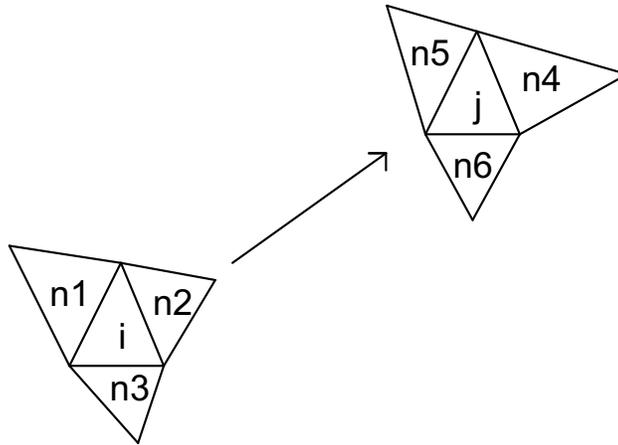


Figure 7 – Illustration des voisins utilisés pour faire l'interpolation

Pour évaluer la qualité de la méthode, la matrice `ETA_mat` a été utilisée pour estimer le temps nécessaire pour réaliser 5000 trajets. La différence entre l'estimation et le délai réel est enregistrée dans le vecteur `err`. Les données utilisées pour faire ce test sont différentes de celles qui ont été utilisées pour construire `ETA_mat`. Cette considération permet d'éviter de biaiser la méthode. Les résultats de cette étude sont dans le tableau 3.

Tableau 3 – Résultats de l'évaluation de la précision de la méthode par discrétisation spatiale

Moyenne de <code>err</code>	8.45 s
Écart-type de <code>err</code>	341.1 s
Erreur absolue moyenne	254.2 s

L'intervalle de confiance à 95% possède une étendue de 22.74 minutes. La méthode est clairement inadéquate. Une analyse est nécessaire pour comprendre pourquoi. Nous prenons par exemple un trajet quelconque A . Ensuite, nous prélevons les 10 courses de la base de données de Téo Taxi qui ressemblent le plus au trajet A . La ressemblance entre la course i et A est estimée à l'aide de s_{iA} :

$$s_{iA} = \Delta(p_i^o, p_A^o) + \Delta(p_i^d, p_A^d) \quad (11)$$

où $\Delta(a, b)$ est la distance entre les points a et b , p_i^o est le point d'origine de i et p_i^d est son point de destination. Les 10 courses associées aux s_{iA} les plus petits sont utilisées pour estimer le temps

Tableau 4 – Temps de déplacement en secondes pour un ensemble de courses semblables

Course 1	792
Course 2	1021
Course 3	563
Course 4	663
Course 5	404
Course 6	740
Course 7	808
Course 8	962
Course 9	670
Course 10	802
Moyenne	772
Écart-type	317

de déplacement du trajet *A*. Le tableau 4 expose le problème de la méthode. Nous observons que la variabilité est très grande pour un ensemble de courses semblables. Cela s’explique, car la méthode ignore l’état du trafic et le trajet parcouru (le taxi ne choisit pas nécessairement la route optimale). Le problème n’est pas notre méthode, mais la variabilité intrinsèque au phénomène. Pour améliorer la qualité de la prédiction, une stratégie à considérer est la segmentation des courses en fonction de l’heure de la journée. Nous avons tenté de le faire, mais sans succès. La base de données de Téo Taxi contient trop peu de courses. La méthode n’est plus statistiquement valide après la segmentation. Dans la section qui suit, nous proposons une autre méthode beaucoup plus simple que la première qui donne des estimations aussi précises.

3.3.2 Méthode par régression linéaire

Cette méthode est la plus simple que l’on puisse imaginer ; le temps de déplacement est estimé par une fonction linéaire qui ne prend comme argument que la distance à vol d’oiseau entre le point d’origine et le point de destination. Les paramètres de la fonction (l’ordonnée à l’origine et la pente) ont été évalués à l’aide de la fonction `gsl_fit_linear` de la librairie *gsl* (*GNU Scientific Library*). La régression linéaire a été faite sur l’ensemble des données historiques. La figure 8 compare le nuage de points expérimentaux avec la droite optimale qui l’estime.

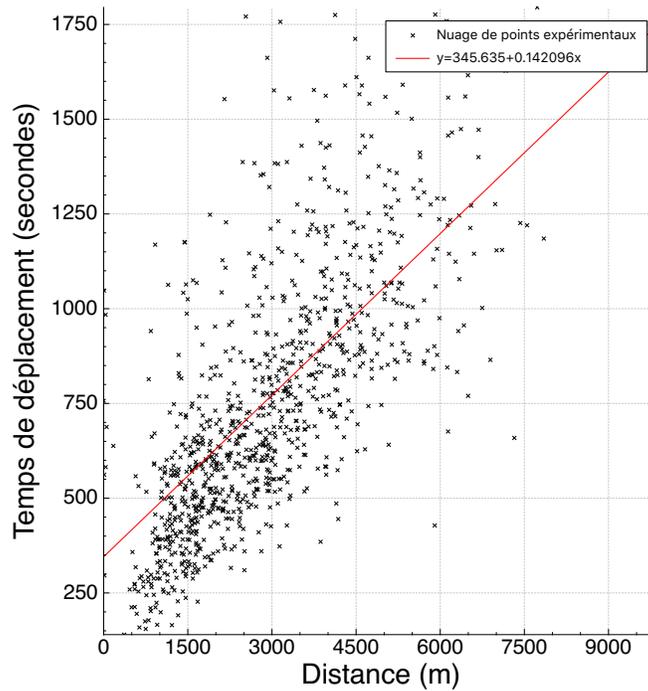


Figure 8 – Régression linéaire du temps de déplacement en fonction de la distance

La même technique a été utilisée pour évaluer la précision de cette méthode. Les caractéristiques du vecteur `err` sont dans le tableau 5. Cette méthode est 25% plus précise que la première et le temps de calcul est semblable. Somme toute, le choix de la méthode a peu d'importance puisque dans les deux cas le niveau de précision est plutôt faible. Pour sa simplicité, nous optons pour la méthode par régression linéaire. C'est elle qui est utilisée dans le simulateur.

Tableau 5 – Résultats de l'évaluation de la précision de la méthode par régression linéaire

Moyenne de <code>err</code>	-0.001 s
Écart-type de <code>err</code>	253.9 s
Erreur absolue moyenne	188.1 s

3.4 Algorithmes de répartition

Les algorithmes de répartition sont des heuristiques qui choisissent les taxis à assigner aux clients. Leur objectif est de maximiser le revenu horaire par taxi tout en garantissant un certain niveau de

service. Les quatre algorithmes étudiés sont énumérés ci-dessous. Les descriptions détaillées de chaque algorithme sont dans les sections qui suivent. Avant cela, nous élaborons un cadre d'analyse pour structurer notre démarche. Tous les algorithmes sont caractérisés en fonction de ce standard.

1. **L'algorithme de répartition élémentaire** : Le client est assigné au taxi libre le plus proche au moment de l'appel.
2. **L'algorithme de répartition avec rayon** : Le client est assigné au taxi libre le plus proche au moment de l'appel si celui-ci est à l'intérieur du cercle de rayon r autour du client. Autrement dit, les taxis libres à l'extérieur du cercle sont ignorés.
3. **L'algorithme de répartition avec file d'attente** : Le client est assigné au taxi (libre ou avec un client à bord) le plus proche au moment de l'appel. Le client assigné à un taxi occupé entre dans une file d'attente.
4. **L'algorithme de répartition par affectation** : Les clients sont affectés aux taxis de manière à minimiser le temps d'attente total.

Note : le taxi le plus proche est celui qui peut atteindre le client le plus rapidement.

3.4.1 Méthode de caractérisation des algorithmes

Le premier algorithme (ou algorithme élémentaire) est notre base de comparaison. Nous cherchons à comprendre comment les solutions 2 à 4 diffèrent de la première. La méthode pour le faire doit être standard. Nous proposons ici un cadre d'analyse qui définit les paramètres de simulation.

Le choix des paramètres aura des conséquences sur la portée des résultats. Il est important de pouvoir les justifier. C'est ce qui est fait dans ce qui suit.

1. **Le temps de simulation** : Bratley et al. (1987) posent la question : "How do we know when we have reached steady-state? When can we act as if we have reached steady state?". Ils expliquent que pour plusieurs la réponse est « jamais ». La méthode d'analyse adéquate pour les simulations stationnaires reste encore à être définie. La règle intuitive est : la simulation doit être suffisamment longue pour que les effets des conditions initiales soient négligeables. Pour vérifier l'atteinte de l'état d'équilibre, nous avons tracé un indicateur de performance en fonction du

temps de simulation. Nous pouvons voir à la figure 9 l'évolution du revenu horaire. Nous posons qu'une simulation de 12.5 heures permet d'atteindre l'état stationnaire.

2. **Le pas de temps (discrétisation temporelle)** : Ce paramètre dépend du niveau de précision voulu. Nous le fixons à une seconde. Cette discrétisation nous semble raisonnable. La plupart de nos événements durent plusieurs minutes.
3. **Le nombre de taxis** : Nous le fixons à 50 : le nombre de taxis que possédait Téo Taxi à l'été 2016.
4. **La demande** : Nous utilisons les données historiques. Les courses sont choisies aléatoirement dans la base de données.
5. **La densité de demande** : Elle constitue la variable indépendante de notre étude. Nous mesurons les indicateurs de performance pour des valeurs entre 20 et 400 demandes par heure.
6. **La zone** : Nous utilisons la zone de service de Téo Taxi à l'été 2016.

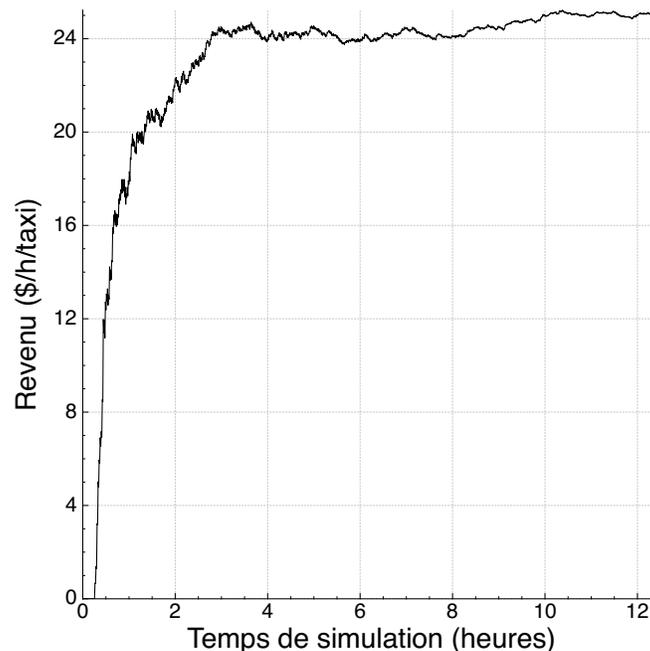


Figure 9 – Évolution d'un indice de performance dans le temps

La méthode de caractérisation (voir algorithme 3.3 pour accéder à sa forme pseudo-code) s'inspire des recommandations de Bratley et al. (1987) dans la section 3.2 du livre. Nous itérons tout d'abord sur le vecteur de demande et pour chaque densité de demande nous faisons N simulations avec des nombres

aléatoires différents. Les N simulations permettent de déduire les intervalles de confiance des indicateurs de performance. Nous fixons notre attention sur le revenu horaire et le temps d'attente moyen.

Algorithme 3.3 Algorithme pour la caractérisation de l'algorithme X de répartition

```

1: for all demand do
2:   for all rng do
3:     Simulation avec la demande demand, les nombres aléatoires rng et l'algorithme 0.
4:     Simulation avec la demande demand, les nombres aléatoires rng et l'algorithme X.
5:   end for
6:   Calcul de la moyenne/variance de la différence relative des indicateurs entre les deux simulations.
7: end for

```

L'intervalle de confiance pour un indicateur X est

$$\bar{X} \pm k\sqrt{s^2/N}, \quad (12)$$

où s est l'écart-type, k est l'ordonnée pour une probabilité de $(1 - \alpha)/2$ dans chaque queue de la distribution de Student avec $N - 1$ degrés de liberté. Nous ne savons pas si notre distribution est normale. Dans ces conditions, il est recommandé d'utiliser un $N > 6$ et un $\alpha \leq 0.8$. Dans ce qui suit, nous employons $\alpha = 0.8$ et $N = 10$.

3.4.2 Algorithme de répartition avec rayon

L'idée de l'algorithme est simple. Un cercle de rayon donné est associé à chaque client. Seulement les taxis à l'intérieur de ce cercle peuvent le servir. La taille du cercle dépend du revenu horaire de la course à répartir : le ratio entre le prix au taximètre et le temps total nécessaire pour la réaliser. Cela inclut les temps de déplacement pour la cueillette et le dépôt du client. La valeur d'une course au taximètre est, par convention, une fonction de la distance parcourue et du temps où le client est à bord. Pour la déduire, nous devons connaître le chemin parcouru par le taxi et la vitesse sur chacun des segments. Or, notre simulateur ne permet pas ce niveau de détail. La trajectoire entre deux points est toujours une ligne droite. La distance et le temps de déplacement sont directement proportionnels. Ceci implique de concevoir une fonction qui évalue le revenu d'une course en fonction de la distance entre deux points. Nous utilisons une interpolation linéaire simple comme à la figure 8. L'ajustement de la courbe a été fait avec les données historiques; nous avons accès à la valeur au taximètre des courses

réalisées par Téo Taxi.

La prochaine étape est de définir une relation qui nous donne le rayon à utiliser en fonction du revenu. L'objectif de cette fonction est de privilégier les courses les plus intéressantes en leur associant un rayon plus important. Par simplicité, nous avons choisi une relation linéaire. Elle est illustrée à la figure 10. À 27\$/h, le rayon est nul ce qui implique qu'aucune course générant un revenu en dessous de ce seuil ne pourra être servie. La position du seuil et la valeur de la pente de la courbe (650 m/(\$/h/taxi) ont été déterminées par un processus d'essais et d'erreurs. Selon nos expériences, ce sont ces paramètres qui offraient les résultats les plus intéressants.

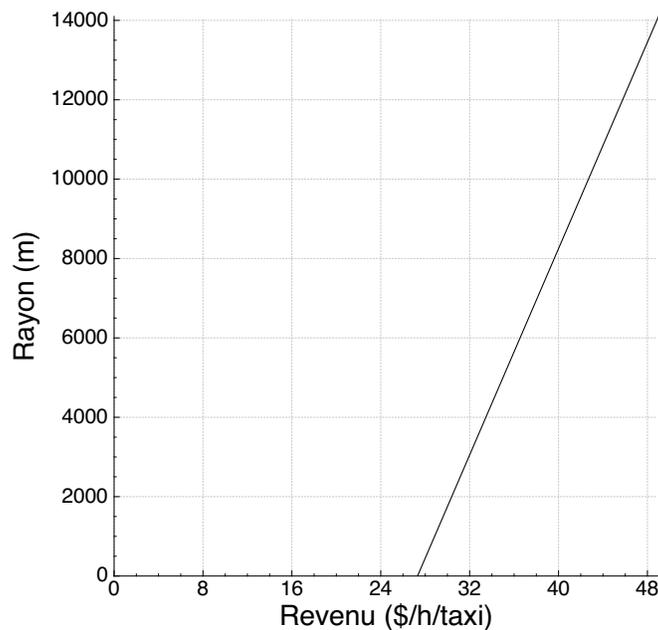


Figure 10 – Fonction de sélectivité

3.4.3 Algorithme de répartition avec file d'attente

Le dernier algorithme ne considère que les taxis libres à l'intérieur d'un cercle de rayon r autour du client. Nous décrivons maintenant un algorithme qui prend en compte la disponibilité future des taxis occupés. Il est possible de le faire parce que la destination des clients est connue à l'avance. Le taxi assigné au client est celui qui peut le servir à l'intérieur du délai le plus court, qu'il soit occupé ou non. L'implémentation de cette règle de répartition est simple; seulement deux ajustements doivent être

faits. Premièrement, lors du calcul de l'*ETA*, il faut ajouter le temps nécessaire pour terminer la course en cours au temps de déplacement. Deuxièmement, la notion de file d'attente est ajoutée à chaque taxi. Seulement les taxis avec une file d'attente vide sont considérés lors du processus de répartition. Si un taxi possède une file non vide, il est assigné automatiquement à la prochaine tâche dans sa file lorsqu'il devient libre.

3.4.4 Algorithme de répartition par affectation

L'algorithme par affectation brise la règle du premier arrivé, premier servi. Au lieu d'assigner un taxi au moment de l'appel, elle se fait par la résolution d'un problème d'affectation. Chaque fois qu'une nouvelle information est connue, le simulateur appelle l'outil CBC de COIN-OR qui résout le problème d'affectation. Les files d'attente et les réaffectations sont autorisées. Notre objectif est de maximiser le revenu horaire.

L'algorithme diffère significativement de l'algorithme élémentaire. Certains paramètres du simulateur doivent être modifiés. Notre modèle du taux d'abandon par *Balking* ne s'applique plus parce que nous ne connaissons pas l'*ETA* à l'avance (car les réaffectations sont autorisées). Nous faisons l'hypothèse qu'aucun *ETA* n'est affiché à l'utilisateur avec ce nouveau système. Les annulations clients n'existent plus. C'est maintenant le système qui décide d'annuler la requête des utilisateurs en fonction de leur rentabilité.

Si les clients ne peuvent plus annuler, il faut inclure dans le modèle un mécanisme pour préserver un certain niveau de service. Pour ce faire, nous proposons d'ajouter un facteur de pondération au revenu horaire en fonction du temps d'attente : si l'*ETA* est plus grand que 20 minutes, le revenu horaire pour cette course est divisé par 10. Ce mécanisme amène naturellement vers une solution où tous les clients sont servis à l'intérieur de ce délai si possible. La solution de rechange est d'ajouter une contrainte qui force les *ETA* à être plus petits que 20 minutes. Cependant, une contrainte fixe amène son lot de désavantages. D'une part, l'ajout d'une contrainte implique qu'il n'est pas toujours possible de trouver une solution au problème. Nous devons alors modifier le problème (réduire le nombre de clients par exemple) de manière itérative jusqu'à en trouver une. D'autre part, la contrainte de 20 minutes n'est pas toujours pertinente. Lorsque la demande est faible, le temps de déplacement nous importe peu. Il

est préférable de servir un client à 21 minutes que de ne rien faire.

Pour obtenir un problème d'affectation classique, le nombre de taxis doit être égal au nombre de clients en attente. Ce n'est pas nécessairement le cas. Nous utilisons la stratégie de Maciejewski et al. (2016) pour contourner cette problématique. Un certain nombre de taxis ou clients fantômes sont ajoutés artificiellement pour atteindre l'équilibre. Le revenu pour l'affectation avec une entité fantôme est de 0. Les taxis fantômes seront affectés aux clients les moins rentables et vice versa. Si un client est affecté à un taxi fantôme, le simulateur annule sa requête. Un taxi affecté à client fantôme reste immobile.

Le lecteur trouvera ci-dessous le détail mathématique du modèle d'affectation.

Ensembles :

- $j \in J$: ensemble de taxis (libres ou avec client à bord, tous les autres sont ignorés)
- $i \in I$: ensemble des clients

Paramètres :

- R_{ij} : revenu horaire si le taxi j sert le client i
- F_{ij} : facteur de pondération pour l'affectation du client i au taxi j

Variables de décision :

- A_{ij} : variable binaire prenant la valeur 1 si et seulement si le client i est affecté au taxi j

Objectif :

$$\max \sum_{i,j} F_{ij} R_{ij} A_{ij} \tag{13}$$

Contraintes :

$$\sum_j A_{ij} = 1 \quad \forall i \tag{14}$$

$$\sum_i A_{ij} = 1 \quad \forall j \tag{15}$$

Éventuellement, nous aimerions prendre en compte l'autonomie des véhicules. Il suffit pour ce faire

de ne définir les variables A_{ij} que pour les affectations qui sont possibles; c'est-à-dire celles pour lesquelles la relation $C_{ij} + e \leq C_{0j}$ est satisfaite, où C_{ij} est la charge nécessaire pour servir le client i avec le taxi j , C_{0j} est le niveau de charge du taxi j au moment de l'affectation et e est un facteur de sécurité.

3.5 Algorithme de gestion de l'autonomie

La modélisation de l'état de charge est un problème complexe. Le comportement des batteries dépend d'un très grand nombre de facteurs comme la vitesse, l'accélération, le type de véhicule et la température. Un effort important doit être fait pour développer un modèle rigoureux. Une telle démarche sort du cadre de notre projet. Nous optons pour un modèle modeste qui utilise un taux de décharge constant. Ce qui suit résume l'ensemble de nos hypothèses.

Hypothèses :

1. Le taux de décharge est constant dans le temps. Nous ignorons donc l'effet du mouvement sur le taux de décharge. Nous fixons sa valeur à 15%/h. Les véhicules ont une autonomie de 210 km et se déplacent à 30 km/h en moyenne : $100\% / (210/30) \approx 15\%$.
2. Le temps perdu à la borne est toujours de 15 minutes. Téo Taxi permute les véhicules aux stations, donc le délai ne dépend pas du niveau de charge. Le 15 minutes est le temps nécessaire pour réaliser les activités opérationnelles comme l'inspection du véhicule. Par hypothèse, il y a toujours un véhicule disponible à la station pour faire une permutation.
3. Téo Taxi possède trois stations. Le nombre de bornes de chaque type diffère d'une station à l'autre. Chaque station possède plusieurs types de bornes avec des spécifications différentes. Nous faisons l'hypothèse que la capacité de recharge d'une station est la somme des capacités des bornes (voir tableau 6 pour les capacités des stations). Une station est une borne dont le taux de recharge est linéaire en fonction de la capacité. Une station de 50 kW prend par exemple 12 minutes pour recharger un taxi avec un niveau de charge de 50% : $50\% \times 20 \text{ kWh} / 50\text{kW} \times 60 \text{ minutes/h} = 12 \text{ minutes}$. Lorsque les 12 minutes sont écoulées, un nouvel appel à la borne est fait. Le taxi reste 15 minutes à la borne même si seulement 12 minutes sont nécessaires pour le recharger.

Tableau 6 – Capacités des stations de recharge

Station	Puissance (W)
S1	100000
S2	364000
S3	50000

L’heuristique la plus simple pour gérer l’autonomie est l’emploi d’un seuil de recharge noté $T1$. Lorsqu’un taxi possède un niveau de charge en dessous de ce seuil, il est affecté à la station la plus proche. Cette approche est raisonnable, car elle permet de maintenir le niveau de charge de la flotte. Par contre, elle est loin d’être optimale. Elle est problématique si une portion importante de la flotte entre en service en même temps. Plusieurs taxis atteindront le seuil de recharge en même temps. Cela génère de l’achalandage important à la station. Par ailleurs, lorsque le taxi atteint le seuil, il peut être loin de la station et il perdra du temps précieux pour se déplacer. Enfin, l’heuristique ne prend pas en compte l’état de la flotte. Par exemple, lors des temps morts (taux d’utilisation faible), il peut être avantageux d’affecter les taxis disponibles aux bornes libres pour maximiser leur utilisation sans affecter le niveau de service ou la rentabilité.

Notre heuristique est une extension de l’heuristique à un seuil ($T1$). Elle comprend la notion de l’appel à la borne qui se produit lorsqu’une station est disponible pour accueillir un taxi. L’appel à la borne se limite aux taxis (libres ou non) avec un niveau de charge sous un certain seuil ($T2$) et qui sont situés à l’intérieur d’un cercle autour de la station. Le rayon du cercle est fixé à 10 minutes. Le seuil d’appel à la borne est variable, il dépend du taux d’utilisation de la flotte. Nous testerons différents seuils pour évaluer l’impact sur les indicateurs de performance.

Cette heuristique possède une faiblesse. Elle ne permet pas de distribuer les efforts uniformément entre les stations. Si plusieurs taxis au centre-ville atteignent le seuil $T1$ en même temps, ils seront tous assignés à la même station : la plus proche du centre-ville. Assurément, ce scénario est plausible, car une grande partie de la demande se trouve dans cette région. Nous ajoutons une contrainte à l’heuristique qui limite le nombre de taxis en attente à une station. La contrainte utilise la notion de

dette qui est le temps nécessaire pour recharger tous les taxis assignés à la borne. Cette dette ne doit pas être plus grande que 30 minutes. Prenons un exemple concret : une borne est libre et appelle un taxi. La dette de la station augmente alors de 12 minutes (si SOC=50%). Une minute plus tard, un deuxième taxi atteint le seuil $T1$ près de cette station, la dette augmente de 21 minutes (si SOC=12%). La dette totale est maintenant de 32 minutes (12-1+21). De ce fait, aucun taxi ne pourra être assigné à cette borne pour les deux prochaines minutes.

3.6 Algorithmes de localisation

Un algorithme de localisation veut repositionner dynamiquement la flotte de taxis de manière à pouvoir servir les clients dans un plus court délai. L'objectif est de positionner les taxis là où la demande est importante. Par conséquent, il dépend d'un modèle de prévision de la demande pour prendre ses décisions. Bien que la prévision de la demande est un enjeu important pour la gestion de la flotte de taxis, elle sort du cadre de cette étude. La demande de Téo Taxi est particulièrement difficile à prédire. D'une part, la demande de taxi est très volatile en général : elle dépend fortement de la température. D'autre part, Téo Taxi est en opération depuis moins d'un an. Ainsi, peu de données historiques sont disponibles. Il n'est pas encore possible de distinguer les effets de la saisonnalité et de la croissance sur la demande. Pour notre analyse, cela importe peu, car le simulateur est stationnaire. La demande est tirée aléatoirement dans la base de données. Il n'y a pas de discrimination en fonction du jour de la semaine ou de l'heure de la journée. Grâce à ces simplifications, il est plutôt facile d'élaborer un modèle de prévision de la demande. Nous avons utilisé le maillage conçu à la section 3.3 qui divise le territoire en zones à desservir. Pour déterminer l'importance relative des zones, nous comptons le nombre de courses associées à chaque zone que nous divisons ensuite par le nombre total de courses. C'est ce ratio que nous employons dans notre modèle pour pondérer chacune des zones. La figure 11 illustre la distribution de la demande en pourcentage. Les valeurs sont arrondies à l'unité pour faciliter la visualisation.

3.6.1 Modèles exacts

Notre modèle en nombres entiers utilise une stratégie par double affectation. Tout d'abord, nous définissons un ensemble de zones K_j qui sont accessibles pour le taxi j . Un taxi est affecté à exactement

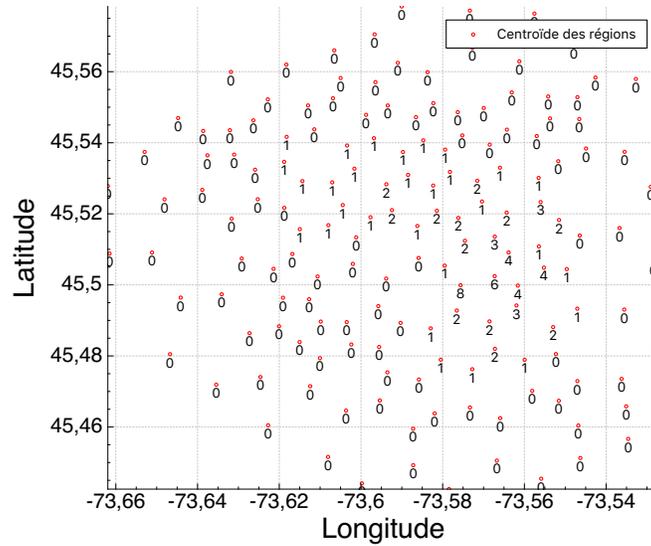


Figure 11 – Distribution de la demande en pourcentage arrondi à l'unité

une zone. Ensuite, nous attribuons un client i au centre de chacune des zones avec un facteur d'importance w_i . La deuxième affectation se fait entre les clients et les zones. Pour assurer une cohérence, nous ajoutons les contraintes (19) qui spécifient que nous ne pouvons affecter le client à la zone k que si un taxi a été affecté à cette même zone. Le facteur α contrôle le nombre de clients pouvant être affectés à un taxi. L'objectif est de minimiser la distance moyenne entre chaque client et chaque taxi après avoir été relocalisé. La formulation mathématique du modèle est présentée ci-dessous. Pour faciliter sa résolution, nous choisissons d'utiliser des variables continues x pour l'affectation client-zone. Un client peut être affecté à plusieurs zones, ce qui est raisonnable. Seulement les variables y doivent être binaires.

Ensembles :

- K : ensemble des points de relocalisation
- I : ensemble des clients
- J : ensemble des taxis (libres ou avec un client à bord, tous les autres taxis sont ignorés)
- K_j : ensemble des points de relocalisation possibles pour le taxi j

Paramètres :

- d_{ik} : distance entre le client i et le point k
- w_i : importance relative du client i

Variables de décision :

- x_{ik} : variable continue représentant la fraction du client i affectée au point k
- y_{jk} : variable binaire prenant la valeur 1 si et seulement si le taxi j est affecté au point k

Objectif :

$$\min \sum_{k,i} w_i x_{ik} d_{ik} \quad (16)$$

Contraintes :

$$\sum_{k \in K_j} y_{jk} = 1 \quad \forall j \quad (17)$$

$$\sum_{k \in K} x_{ik} = 1 \quad \forall i \quad (18)$$

$$\sum_i x_{ik} \leq \alpha \sum_j y_{jk} \quad \forall k \quad (19)$$

Pour les taxis occupés, la relocalisation n'est pas possible. Par contre, ils font partie de l'ensemble j , car nous voulons prendre en compte leur position éventuelle. Les y_{jk} sont égales à un pour les taxis avec client à bord ; le point k est la zone où se trouve la destination du client.

Le désavantage de ce modèle est qu'il est difficile à résoudre : il est NP-difficile. Nous avons $nt \cdot nd$ variables binaires et $nd \cdot nd$ variables continues (nt : nombre de taxis et nd nombre de zones de demande). Jusqu'à maintenant, nous avons considéré un grillage de 160 zones et une flotte de 50 taxis. La taille du problème est encore raisonnable. Par contre, ce ne sera plus le cas pour une flotte de plusieurs milliers de taxis. C'est pourquoi nous proposons un deuxième modèle dont le nombre de variables binaire ne dépend que de nd .

Ensembles :

- K : ensemble des points de relocalisation
- I : ensemble des clients

Paramètres :

- d_{ik} : distance entre le client i et le point k
- w_i : importance relative du client i
- L : nombre total de taxis

Variables de décision :

- x_{ik} : variable continue représentant la fraction du client i affectée au point k
- y_k : variable binaire prenant la valeur 1 si et seulement si un taxi est affecté au point k

Objectif :

$$\min \sum_{k,i} w_i x_{ik} d_{ik} \quad (20)$$

Contraintes :

$$\sum_{k \in K} y_k = L \quad (21)$$

$$\sum_k x_{ik} = 1 \quad \forall i \quad (22)$$

$$\sum_i x_{ik} \leq \alpha y_k \quad \forall k \quad (23)$$

Dans ce modèle, nous identifions les régions optimales c'est-à-dire là où les taxis devraient être. Ensuite, nous résolvons un problème d'affectation pour affecter les taxis aux régions identifiées. Le modèle d'affectation est pratiquement identique à celui présenté à la section 3.4.4. Notons que les taxis occupés sont également pris en compte dans ce modèle. Nous fixons à 1 les y_k des zones qui contiennent la destination des taxis occupés.

Le désavantage de cette stratégie est qu'elle ne vérifie pas si l'affectation taxi-zone est raisonnable. En effet, en réduisant le nombre de variables, nous avons retiré du modèle la possibilité de tenir compte de la position actuelle des taxis. Il n'est donc pas possible de mettre une contrainte pour nous assurer

que les zones y_k de relocalisation sont suffisamment proches des taxis libres.

3.6.2 Heuristique

Nous suggérons une dernière stratégie pour le problème de localisation. Elle utilise une heuristique pour déterminer les zones à cibler par la relocalisation. Ensuite, un problème d'affectation est résolu pour associer les taxis libres aux zones sélectionnées.

Tout d'abord, nous définissons un mécanisme pour prioriser des zones à desservir. Nous posons que la priorité P d'une zone dépend de deux paramètres : le niveau de service NS et la demande D . Une zone difficile à desservir (niveau de service bas) aura une priorité importante. Ce paramètre est pondéré par la demande prévue pour cette zone :

$$P = \frac{D}{NS} = D \cdot \sum_{j=1}^{nt} ETA_j. \quad (24)$$

Nous supposons que le niveau de service est l'inverse de la somme des temps de déplacement pour les nt taxis. Autrement dit, nous voulons prioriser les zones qui sont loin de tous les taxis. Les taxis occupés sont inclus dans cette somme. L' ETA pour un taxi occupé est le temps pour terminer la course courante plus le temps de déplacement vers la zone. La valeur de D est équivalente au facteur de pondération w_i de la section précédente.

Lorsque la valeur de P est connue pour chaque zone, nous sélectionnons les n zones avec les P les plus grands, où n est le nombre de taxis libres. Par la suite, il ne reste qu'à affecter ces taxis aux zones prioritaires de manière à minimiser le temps de déplacement total.

4 Présentation des résultats

Ce chapitre présente les résultats de nos recherches. Dans un premier temps, nous démontrons la validité du simulateur avec des tests de validations théoriques et expérimentaux. Dans un deuxième temps, nous caractérisons chacun des algorithmes définis dans le chapitre précédent, c'est-à-dire pour chacun des problèmes étudiés.

4.1 Validation du simulateur

4.1.1 Validation théorique

La méthode la plus utilisée pour valider un modèle est la comparaison. Nous vérifions si les résultats obtenus par simulation concordent avec les résultats connus issus de modèles théoriques. Un modèle théorique est élémentaire et ne décrit qu'une partie de la réalité. Le comportement d'un système peut être décrit exactement seulement s'il est suffisamment simple. Une façon simple de construire un modèle théorique est de réduire le nombre de variables. Il est alors possible de calculer ses propriétés à l'aide d'équations analytiques exactes. Dans ce qui suit, nous nous servons des indices de performance comme base de comparaison.

La première validation imaginée consiste à déterminer le revenu horaire maximal théorique par taxi. En d'autres mots, nous aimerions connaître le revenu qu'il est possible de faire si les taxis ne font que servir des clients (aucun temps mort). Cette valeur peut être calculée à l'aide de la distribution des montants aux taximètres et des temps de service présents dans la base de données. La moyenne du montant d'une course est de 13.4\$ et en moyenne un client est 755 secondes à bord d'un taxi. Nous obtenons donc 63.9 \$/h. Pour reproduire ce résultat avec le simulateur, les temps d'attente et les taux d'abandons ont été fixés à 0. Nous posons que $nt = 1$ et $dd = 300$ pour obtenir un taux d'utilisation très élevé. Les indicateurs de performance sont $TU = 98.2\%$ et $Rh = 62.3\$/h$. La concordance entre la valeur théorique et simulée est vérifiée.

En pratique, le temps d'attente n'est pas nul. Selon les données historiques, le temps d'attente moyen est de 496 secondes. L'ajout de ce délai amène le revenu horaire à 38.6 \$/h. Ce résultat n'a pas pu

être reproduit par simulation, parce que le temps d'attente moyen augmente avec le taux d'utilisation. Historiquement, la flotte de Téo Taxi a très rarement atteint des taux d'utilisation plus grands que 80%. Téo Taxi choisit de mettre de la capacité supplémentaire justement pour assurer un meilleur niveau de service. Les résultats de simulations montrent que le temps d'attente moyen est comparable au temps de service moyen lorsque le taux d'utilisation est près 100%. Cela est logique. Il n'y a pas de raison de croire que la distance entre l'origine et la destination d'un client soit significativement différente de la distance entre la destination et l'origine de deux clients différents. Le revenu horaire avec un taux d'utilisation de 100% est plus près : $63.9/2=32.0\$/h$. Il s'agit du maximum théorique. Cette estimation est correcte si la demande et l'offre sont synchronisées, c'est-à-dire qu'il y a 2.38 requêtes clients à l'heure par taxi (chaque client prend en moyenne 1510 secondes à être servi). Aussi, elle suppose qu'il n'est pas possible de faire une prévision de la demande. Les paramètres de simulation utilisés pour valider cette estimation sont les suivants : $nt = 1$, $dd = 300$ avec un taux d'abandon nul. Le revenu horaire simulé est $30.19\$/h$, le taux d'utilisation est 98.674% et l'*ETA* moyen est 790 secondes. Les résultats de simulations coïncident avec nos estimations théoriques.

Nous savons qu'il est possible d'augmenter davantage Rh si la demande est connue à l'avance ou si la demande est plus grande que 2.38 clients à l'heure par taxi. En effet, dans ces conditions, il est possible de sélectionner les courses les plus payantes. Nous avons conçu une méthode exacte pour vérifier les bénéfices potentiels. La solution exacte est obtenue par énumération de toutes les possibilités. Chaque client peut être assigné à un des taxis ou il peut être ignoré. Donc, au total, il y a $(nt + 1)^{nc}$ possibilités pour le problème d'affectation, car théoriquement tous les clients peuvent être affectés au même taxi. Le revenu total est calculé pour chacune d'elles. Le scénario qui possède le revenu le plus élevé est la solution optimale. Nous ajoutons une contrainte pour assurer un certain niveau de service : le temps d'attente pour un client ne peut pas être plus grand que 20 minutes.

Nous utilisons la solution exacte dans premier temps pour valider notre simulateur. Étant donné que le temps de calcul est proportionnel à $(nt + 1)^{nc}$, nous devons réduire la taille du problème : $nt = 3$, $dd = 7$. En période de faible demande, nous savons intuitivement que la solution optimale est d'assigner le taxi le plus proche au client. Nos résultats confirment cette intuition. En effet, une simulation avec l'algorithme élémentaire donne le même revenu que la solution exacte. Cette adéquation suggère

que le modèle exact et le simulateur ont été bien conçus.

Les avantages de la méthode exacte sont apparents seulement lorsque le ratio dd/nt est grand. Avec les paramètres $nt = 1$, $dd = 12$, le Rh est de 29.9\$/h et l' ETA moyen est 14.1 minutes avec l'algorithme élémentaire et la solution exacte donne un Rh de 36.0\$/h et un ETA de 10.9 minutes. Ces résultats sont importants pour la suite, car ils encadrent les gains potentiels des algorithmes de répartition. Le 36 \$/h est en quelque sorte la limite théorique de rentabilité du système. Notre objectif est de trouver une heuristique dont la rentabilité est la plus près possible du 36\$/h. Elle doit en même temps garantir un niveau de service acceptable.

4.1.2 Validation expérimentale

Nous savons que le simulateur donne des résultats conformes à la théorie. Maintenant, nous aimerions savoir si le simulateur est capable de reproduire le comportement de la flotte de Téo Taxi. Nous faisons l'hypothèse que les deux seuls paramètres qui varient dans le temps sont le nombre de taxis et la demande horaire. Sachant ces paramètres, le simulateur devrait pouvoir reproduire les indicateurs de performance de Téo Taxi pour une journée donnée. Dans notre modèle, la demande horaire correspond au nombre de personnes à l'heure qui considèrent faire une requête. La meilleure façon disponible d'estimer sa valeur est la métrique du nombre d'ouvertures de l'application mobile par les clients. Cette information est connue avec une granularité horaire. La figure 12 montre comment la demande évolue pour différentes journées types. Pour estimer le nombre de taxis disponibles à un moment donné, nous utilisons la table *vehicle statistics* de la base de données *postgres*. Sa valeur est la somme du nombre de taxis dans les états suivants : "en route vers un client", "avec client à bord" et "libre".

Pour reconstituer les 20 et 22 juin (choisis arbitrairement), nous avons fait une simulation pour chaque heure de la journée. Nous avons utilisé le même algorithme de répartition que Téo Taxi. Les résultats pour le revenu horaire sont aux figures 13 et 14. De façon générale, les tendances sont respectées. Les similarités sont satisfaisantes. Cela conclut notre démarche de validation.

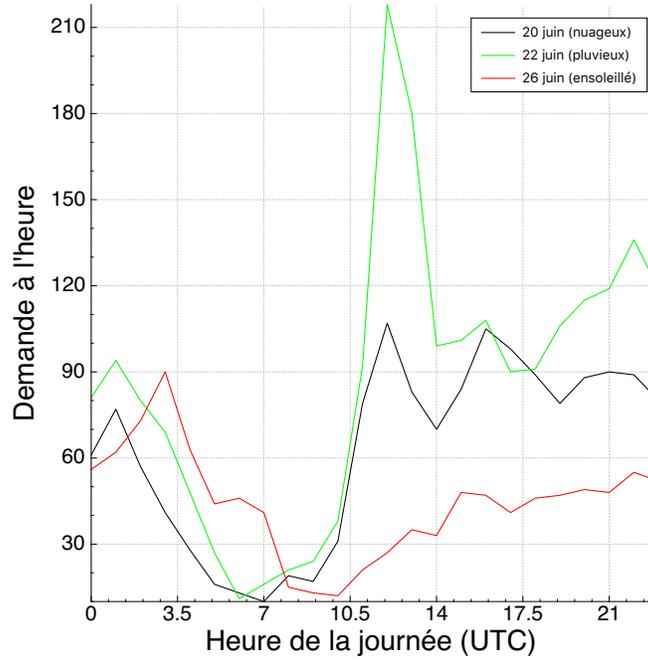


Figure 12 – Journées types et leur demande

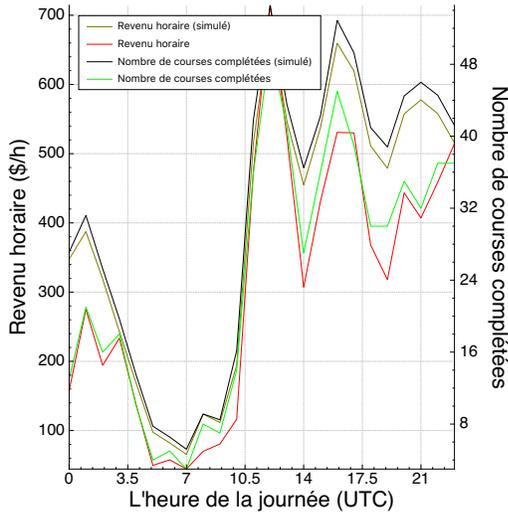


Figure 13 – Revenus horaires réels et simulés pour la journée du 20 juin

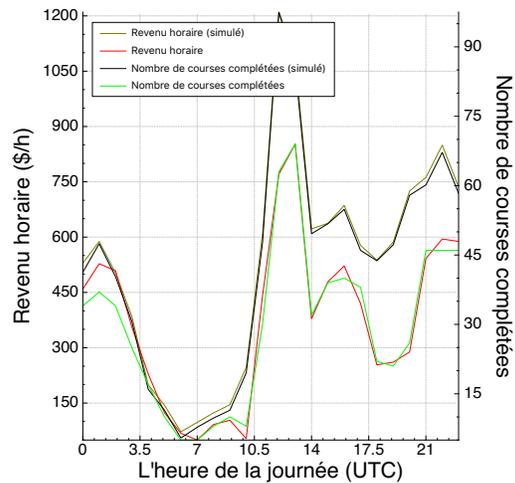


Figure 14 – Revenus horaires réels et simulés pour la journée du 22 juin

4.2 Caractérisation des algorithmes de répartition

Dans les sections qui suivent, nous employons la méthode décrite à la section 3.4.1 pour évaluer la performance des algorithmes de répartition. En quelques mots, la méthode consiste à mesurer les indices de performance pour différents profils de demande.

4.2.1 L'algorithme de répartition avec rayon

Les figures 15 à 18 présentent les résultats pour l'algorithme avec rayon. Deux cas de figure ont été étudiés : avec un rayon fixe ou avec la fonction de sélectivité. Nous avons utilisé la relation tracée à la figure 10 avec deux pentes différentes : 600 et 1000 m/(\$/h/taxi). L'algorithme possède un comportement particulier. En période de faible demande, sa performance est comparable à l'algorithme élémentaire. Par contre, elle diminue au fur et à mesure que la demande augmente. La stratégie est intéressante seulement en période de forte demande : les revenus sont plus importants et le temps d'attente est plus faible.

Ces résultats s'expliquent simplement. La différence du temps d'attente moyen diminue avec la demande, car l'algorithme élémentaire se comporte mal en période de forte demande. Étant donné le nombre limité de véhicules disponibles, le taxi le plus proche peut être loin du client. L'algorithme avec rayon n'a pas ce problème, car le client est toujours proche du taxi. Par contre, lorsque la demande n'est ni forte ni faible (autour de 240 demandes à l'heure), l'algorithme avec rayon va rejeter certaines requêtes alors qu'il y a encore de la capacité non utilisée dans le système, d'où la diminution relative du revenu.

Une autre façon de comprendre la piètre performance de cette stratégie est d'observer les situations où elle devrait offrir une bonne performance. L'idée initiale de la stratégie est d'éliminer les courses éloignées pour privilégier les courses près des taxis. Le problème est que cela présuppose qu'il existe des courses plus intéressantes que celles éliminées. Prenons un cas concret pour illustrer le problème. Disons qu'un client est à 3600 m du taxi le plus proche, nous ignorons sa requête ; il est à l'extérieur des cercles. Le taxi le plus proche est maintenant libre au lieu de se diriger vers ce client. Il attend le prochain client qui peut prendre plusieurs minutes à se manifester. Le gain apparent en temps de déplacement est converti en temps d'attente du prochain client. D'ailleurs, rien ne garantit que le prochain client sera dans le cercle de ce taxi. Manifestement, la probabilité d'avoir une requête immédiatement après la première et dans le cercle du taxi libéré est très faible. Cela se produit seulement en période de très forte demande comme nous l'avons démontré.

Parmi les quatre cas de figure, aucune stratégie ne sort du lot. Le rayon fixe de 3500 m est la solution la plus conservatrice avec une différence relative entre -4% et 3% pour le revenu horaire. La fonction de sélectivité est plus agressive et permet une augmentation de 4% en période de forte demande.

Somme toute, la stratégie avec rayon est à éviter, car les périodes de forte demande sont très rares (300 demandes par heure ou plus). Elle pourrait être utilisée, mais elle devrait être activée seulement lors des périodes de forte demande. Selon notre expérience, cela correspond à un taux d'utilisation de 80% ou plus.

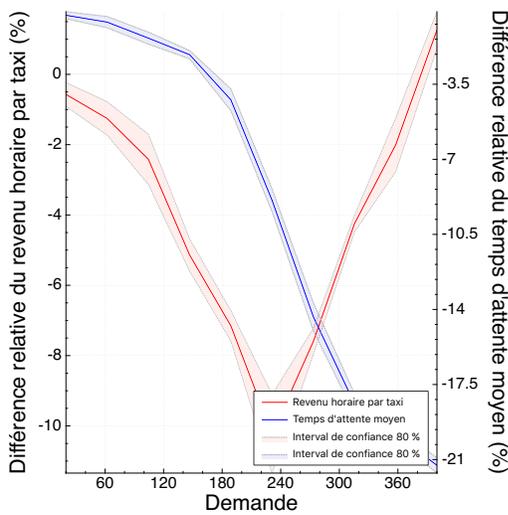


Figure 15 – Performance relative de l’algorithme de répartition avec rayon fixe de 2500 m

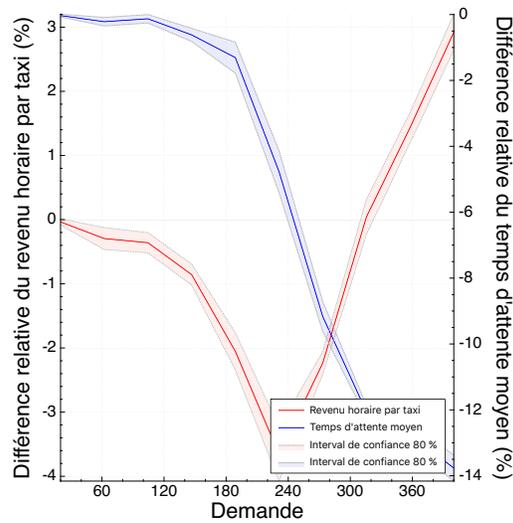


Figure 16 – Performance relative de l’algorithme de répartition avec rayon fixe de 3500 m

4.2.2 L’algorithme de répartition avec file d’attente

La figure 19 présente les résultats pour l’algorithme avec file d’attente. Nous observons une augmentation des deux indicateurs par rapport à l’algorithme élémentaire. Le temps d’attente augmente parce qu’en période de forte demande, la plupart des taxis sont occupés. Ils doivent terminer leur course en cours avant de se déplacer vers le prochain client. Le revenu augmente parce qu’il y a moins d’abandons causés par un manque de disponibilité. Dans notre modèle, les clients abandonnent immédiatement si aucun *ETA* n’est affiché. Cela se produit lorsqu’aucun taxi n’est disponible au moment de l’appel.

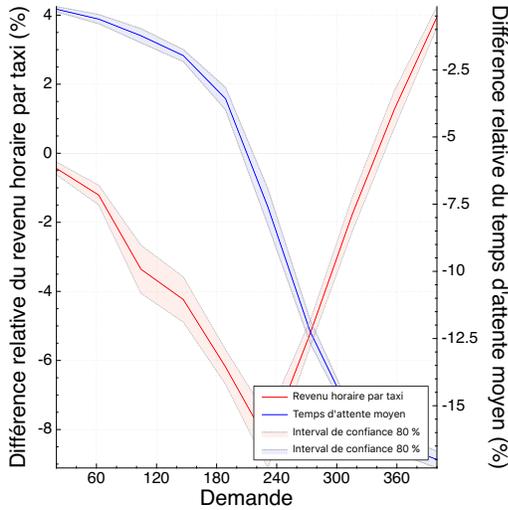


Figure 17 – Performance relative de l'algorithme de répartition avec fonction de sélectivité. Pente = 650 m/(\$/h/taxi)

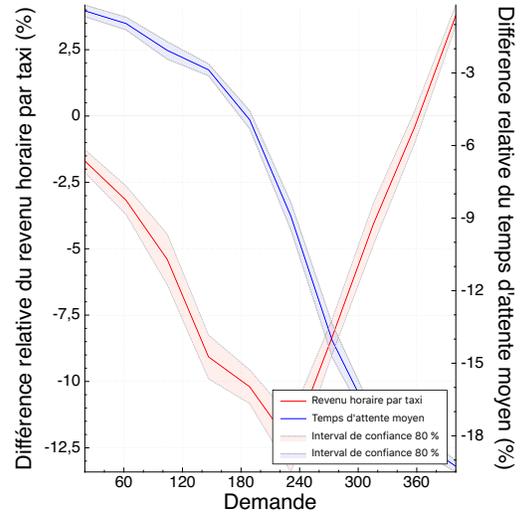


Figure 18 – Performance relative de l'algorithme de répartition avec fonction de sélectivité. Pente = 1000 m/(\$/h/taxi)

L'ajout d'une file d'attente augmente la capacité du système (d'où l'augmentation du revenu), mais diminue le niveau de service (augmentation du temps d'attente moyen). En revanche, nous observons des gains légers sur le temps d'attente lorsque la demande est basse. Ce résultat coïncide avec notre intuition. Cependant, le gain est pratiquement négligeable. La probabilité que le taxi le plus près soit occupé est très faible en période de faible demande (le nombre de taxis disponibles est grand).

4.2.3 L'algorithme de répartition par affectation

Étant donnée la forme particulière de la stratégie, la comparaison directe avec l'algorithme élémentaire est discutable. Nous avons donc ajusté légèrement le cadre d'analyse pour obtenir une comparaison plus juste. Le module de *Balking* a été désactivé : les abandons clients n'existent plus. De cette façon, nous mesurons la performance maximale du système sans faire intervenir l'opinion du client. Il est encore possible d'annuler certaines courses ; cela se produit lorsque la demande est au-dessus de la capacité du système. Selon l'algorithme utilisé, cela se manifeste de différentes façons.

- **Algorithme élémentaire** : une course est annulée lorsqu'aucun taxi n'est disponible au moment de l'appel.
- **Algorithme par affectation** : une course est annulée lorsqu'elle est affectée à un taxi fantôme.

Avec ces modifications, nous obtenons les résultats présentés à la figure 20. À première vue, il a le

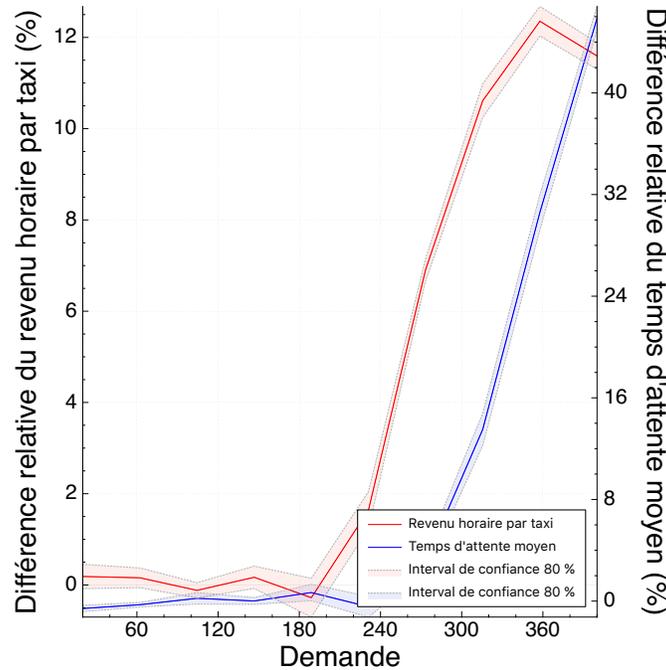


Figure 19 – Performance relative de l'algorithme de répartition avec file d'attente

même comportement que l'algorithme avec file d'attente. Il permet d'augmenter significativement le revenu lorsque la demande est importante et en contrepartie le niveau de service est affecté à la baisse. Cela concorde avec notre intuition. Évidemment, nous ne pouvons comparer ces courbes à celles de la figure 19, car elle inclut le *Balking*.

Tout de même, les résultats sont encourageants. Ils démontrent qu'il est possible d'augmenter de 6\$/h/taxi (ou 20%) la rentabilité du système avec cet algorithme si la clientèle est prête à attendre. Il est intéressant de constater que sa performance est comparable à celle de l'algorithme de répartition exacte (voir section 4.1.1). Le revenu horaire est 37.15\$/h et l'*ETA* moyen est 21.3 minutes pour une demande horaire de 400. Le revenu est légèrement supérieur à notre limite théorique de 36\$/h, car il n'y a pas de contrainte sur le niveau de service. Contrairement à notre heuristique, la méthode exacte impose que tous les clients soient servis en moins de 20 minutes.

4.3 Algorithme de gestion de l'autonomie

Nous employons le cadre d'analyse de la section 3.4.1. L'algorithme de base prend maintenant en compte la gestion de l'autonomie à l'aide de l'heuristique à un seuil $T1$. Dans tous les cas, ce seuil est

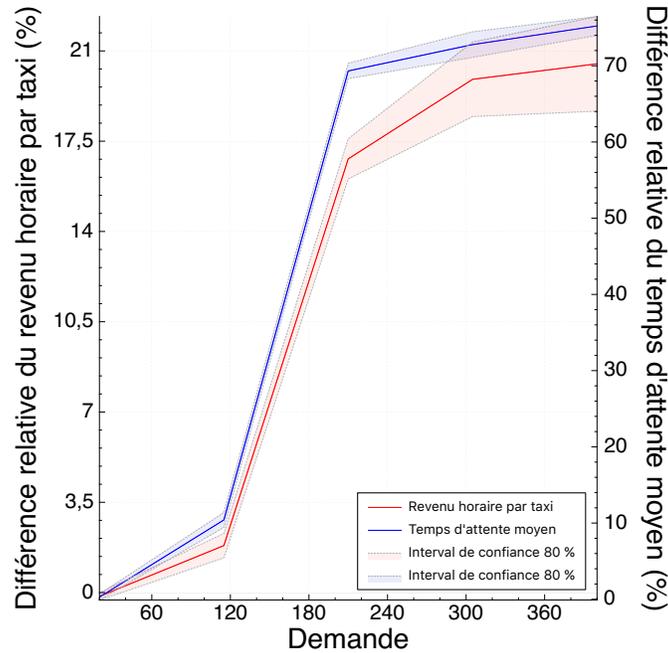


Figure 20 – Performance relative de l'algorithme de répartition avec affectation

de 20% (ou 80 minutes). Ce niveau de charge est amplement suffisant pour servir une course (environ 30 minutes) et retourner vers une station (environ 15 minutes). Idéalement, il faut vérifier si le taxi a la capacité de servir un client avant de l'affecter. Dans notre modèle, la vérification n'est pas faite, car cela implique de modifier la règle de répartition élémentaire. Nous préférons limiter les modifications au cadre d'analyse. Il est plus facile de comparer les résultats de cette façon. D'ailleurs, pendant la simulation, nous pouvons nous permettre d'avoir des niveaux de charge négatifs et cela est utile pour déterminer la fiabilité de l'heuristique de gestion de l'autonomie.

Nous avons comparé deux niveaux $T2$ (50% et 75%) en fonction de la demande. Les courbes des indicateurs de performance sont aux figures 21 et 22. Elles montrent une diminution de la performance. Pour un $T2$ de 75%, le revenu horaire subit une baisse de 10 % et le temps d'attente moyen augmente de 7%. Ce comportement était attendu, car notre heuristique se veut plus sécuritaire que l'heuristique de base. Les appels à la borne augmentent les déplacements vers les stations, ce qui diminue notre capacité de servir la demande. Avec un seuil de 50%, la diminution de la performance est moins importante puisque les déplacements à la borne sont réduits.

Nous proposons maintenant une justification pour la forme de la courbe du temps d'attente moyen. Sa croissance entre 0 et 200 demandes est logique. Les véhicules sont moins bien distribués dans l'espace, car les appels à la borne créent des agrégats de véhicules aux stations de recharge (il n'y a aucune relocalisation de faite, donc, après la recharge, le véhicule reste sur place jusqu'à ce qu'il soit affecté à un client). Lorsque la demande est forte (200 et plus), la taille des agrégats de taxis aux stations diminue. Les taxis sont plus souvent occupés et la différence avec l'heuristique de base se fait moins sentir.

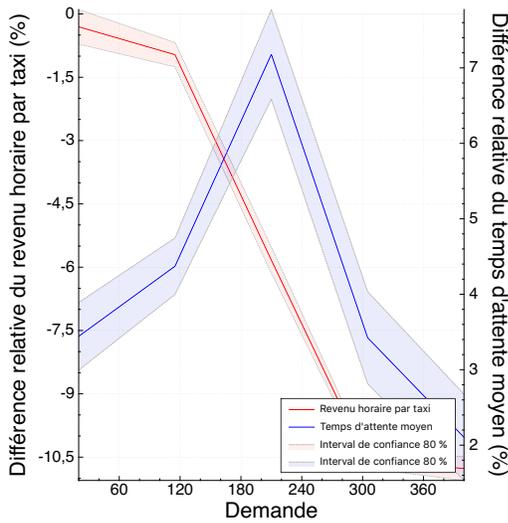


Figure 21 – Performance relative de l'heuristique pour gestion de l'autonomie avec un seuil $T2 = 75\%$

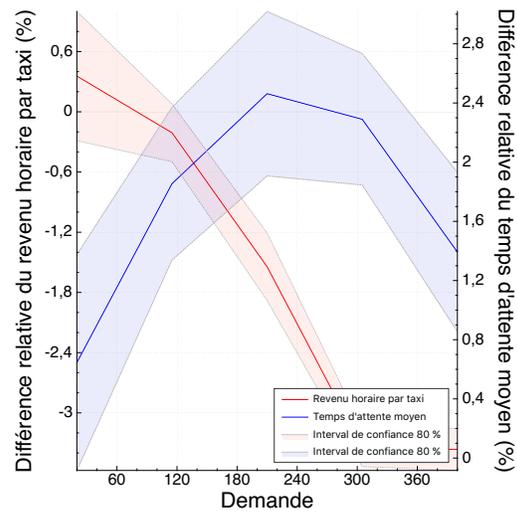


Figure 22 – Performance relative de l'heuristique pour gestion de l'autonomie avec un seuil $T2 = 50\%$

La valeur ajoutée de notre heuristique n'est pas apparente à travers ces indicateurs de performance. Pour comprendre pourquoi la méthode est intéressante, il faut plutôt observer comment évolue l'état de charge de la flotte dans le temps. À cet égard, nous avons conceptualisé trois autres indicateurs : le niveau de charge moyen, le niveau de charge minimal et la dette aux stations. Leur valeur au cours de la simulation sont aux figures 23 et 24. Sans aller dans les détails, l'élément à faire ressortir est la stabilité qu'offre notre heuristique. D'ailleurs, cette stabilité est apparente chez chacun des indicateurs. Le niveau de charge moyen oscille entre 75% et 80% avec notre heuristique et fluctue abruptement entre 45% à 75% avec l'heuristique de base. Sans le seuil $T2$, il y a un achalandage aux stations entre les heures 2 et 6 (et entre 8 et 12), car les dettes sont importantes à ce moment-là. Du point de vue opérationnel, il est plus facile de gérer un flux constant de taxis aux stations. Il faut dire aussi que la

taille des terrains des stations est fixe et, par conséquent, le nombre de taxis qu'elle peut accueillir est limité. Compte tenu de ce qui précède, pour connaître la valeur optimale du seuil $T2$, les coûts et les contraintes opérationnelles aux stations doivent être prises en compte. Une telle étude est un projet en soi. Inclure un modèle opérationnel dans notre simulateur est un défi considérable. Il s'agit d'une piste de recherche future intéressante.

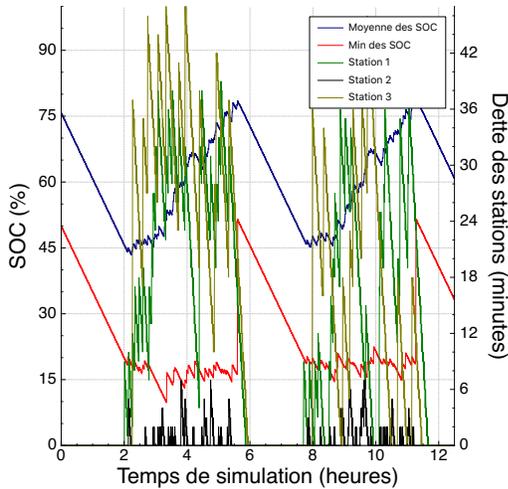


Figure 23 - Évolution temporelle du niveau de charge avec un seuil $T2 = 0\%$

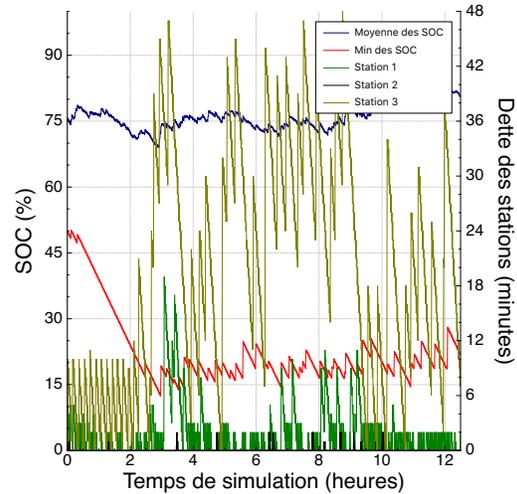


Figure 24 - Évolution temporelle du niveau de charge avec un seuil $T2 = 75\%$

La figure 23 expose deux phénomènes intéressants. D'une part, le niveau de charge minimal reste toujours positif tout au long de la simulation. En fait, pour toutes les simulations réalisées, il n'a jamais atteint un niveau en dessous de 10%. Notre simulateur est déterministe donc nous pouvons nous attendre à ce résultat. Le niveau de charge tend à se maintenir au-dessus de zéro étant donné que la somme des capacités des stations est plus grande que le taux de décharge de la flotte :

$$100\text{kW} + 364\text{kW} + 50\text{kW} > 50 \text{ taxi} \times 15\%/h \times 20\text{kWh/taxi}$$

$$514\text{kW} > 150\text{kW}.$$

En apparence, l'heuristique de base est sans risques. Par contre, si le nombre de taxis augmente, il n'est pas certain que le niveau de charge restera toujours au-dessus de zéro. Le niveau de charge minimum est aussi fonction de la variabilité dans le temps de déplacement. Une grande variabilité implique un seuil plus élevé pour garantir le même niveau de service. À cet égard, la fixation du seuil $T1$ est une

règle d'affaires avant tout, car elle dépend du risque tolérable pour l'entreprise. La notion de risque est complexe et comprend une grande part de subjectivité. À notre avis, nous avons ici un problème qui mérite d'être appréhendé.

Le choix de $T1$ n'est pas un enjeu critique dans notre cas, car la plupart (plus de 90% selon nos simulations) des permutations sont le résultat d'un appel à la borne. Ceci s'explique par la capacité excédentaire que nous avons; la somme des taux de recharge de chaque borne est environ trois fois plus grande que le taux de décharge de la flotte. Ainsi, notre analyse de sensibilité sur ce paramètre n'a pas généré de conclusions notables.

Le deuxième phénomène est la disparité dans l'utilisation des stations de recharge. La station 3 possède une dette de près de 30 minutes pendant les heures d'achalandage versus la station 1 qui a au plus quelques minutes de dette. Si nous nous référons au tableau 6, la disparité s'explique par la différence de capacité des stations. Aussi, la station 3 est plus près des zones où la demande est importante. Naturellement, le taxi se trouve là où la demande est forte. Les courbes de dette sont de bons outils pour guider les décisions d'investissement aux stations de recharge. Nos résultats montrent qu'il pourrait être judicieux d'ajouter de la capacité à la station 3.

4.4 Caractérisation de l'algorithme de localisation

D'après nos expériences préliminaires, le temps de calcul est un enjeu important pour la résolution du problème de localisation. Par conséquent, les résultats présentés dans cette section ont été produits avec notre modèle simplifié : celui où le nombre de variables binaires est moindre. Nous avons dans un premier temps exploré l'impact de la taille du problème sur le temps de calcul (voir figures 25 et 26). Nous employons le solveur CBC pour résoudre notre modèle. La taille du grillage correspond au nombre de zones. Puisque le nombre de taxis doit être plus petit que le nombre de zones, nous avons fait nos tests avec 2 et 5 taxis.

Les courbes nous offrent une information précieuse. Elles nous indiquent le niveau de raffinement maximal pour obtenir des simulations d'au plus quelques minutes. Nous posons qu'une relocalisation doit être faite toutes les 10 minutes. Une simulation estime le comportement de la flotte pour 12 heures.

De ce fait, nous avons 72 problèmes de localisation à résoudre par simulation. Alors, le temps de résolution ne peut être plus grand qu'une seconde, ce qui correspond à un grillage de 30 zones pour deux taxis et une dizaine de zones pour cinq taxis. La contrainte de temps limite grandement la taille des problèmes pouvant être étudiés. Nous rejetons donc cette approche, car elle ne permet pas d'atteindre nos objectifs. Nos solutions devraient être applicables à une flotte de 1000 taxis, ce qui n'est pas possible avec les modèles exacts. Une heuristique devra être conçue pour déterminer s'il est possible de résoudre un problème de cette taille en quelques secondes.

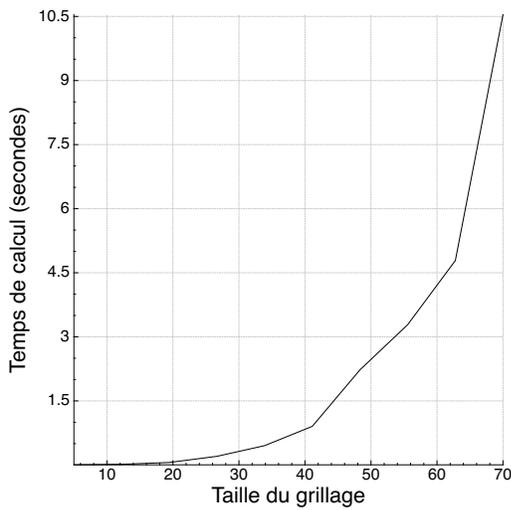


Figure 25 – Temps de calcul pour la résolution du problème de localisation avec deux taxis

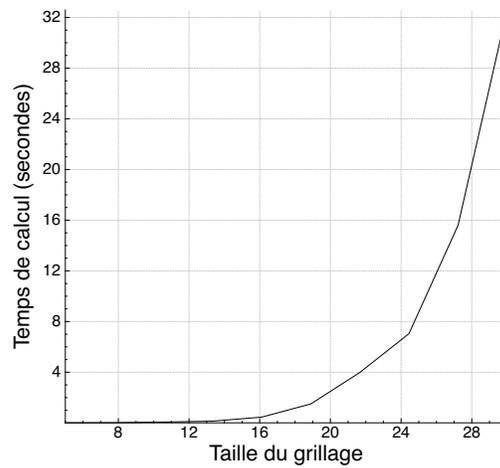


Figure 26 – Temps de calcul pour la résolution du problème de localisation avec cinq taxis

À la lumière de ces résultats, nous avons opté pour notre heuristique qui possède un temps de calcul négligeable par rapport aux méthodes exactes. Les gains de la stratégie de localisation sont dévoilés à la figure 27. Encore une fois, la stratégie affecte relativement peu les indicateurs à faible demande. La relocalisation des taxis est avantageuse lorsque la demande est autour de 200. Alors, le revenu horaire subit une augmentation de 1.6% et le temps d'attente diminue de 7%. Aucun gain n'est apparent avec un grand taux d'utilisation, car tous les taxis sont occupés : aucune relocalisation des taxis libres n'est possible.

Pour justifier notre choix de la fréquence de relocalisation, nous avons produit les graphiques aux figures 28 et 29. Ils démontrent qu'une fréquence plus élevée (ou une période plus courte) permet

d'améliorer les deux indicateurs. Tout de même, l'écart de performance est d'au plus quelques pourcents. Nous nous sommes limités à une période de 10 minutes à cause des enjeux humains.

Comme nous pouvions l'espérer, la stratégie de localisation affecte principalement le temps d'attente. En moyenne, le taxi est plus près du client lors de la requête d'une course. Par contre, l'augmentation de revenu reste négligeable, car la diminution du temps d'attente ne permet pas nécessairement de faire plus de courses. En effet, quand nous avons de la capacité excédentaire, le revenu maximal dépend de la demande totale et non de notre gestion de la flotte. L'augmentation du revenu est principalement causée par la réduction du taux d'abandon qui dépend pour sa part du temps d'attente.

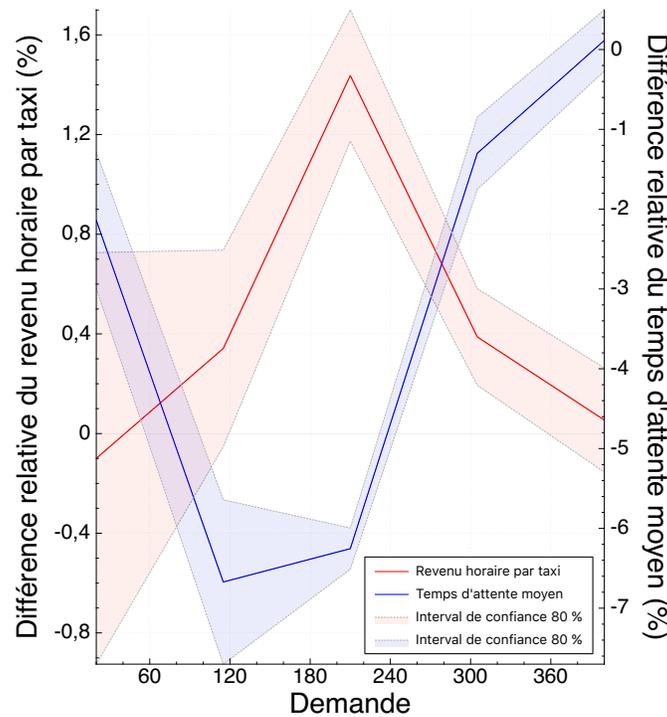


Figure 27 – Performance relative de l'heuristique de localisation avec une période de 10 minutes

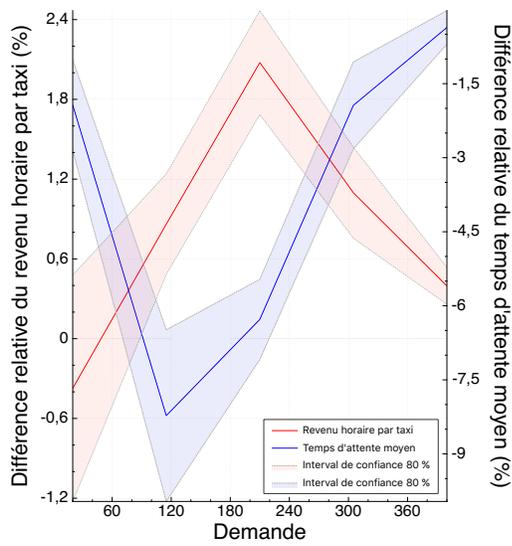


Figure 28 – Performance relative de l'heuristique de localisation avec une période de 5 minutes

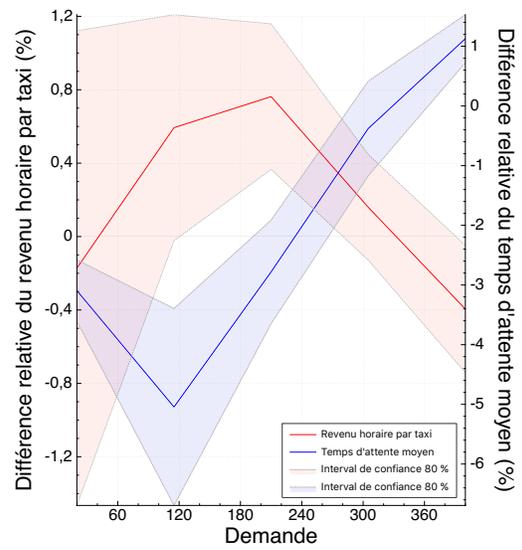


Figure 29 – Performance relative de l'heuristique de localisation avec une période de 60 minutes

5 Conclusion

5.1 Synthèse des résultats

Nous avons dans un premier temps démontré la validité de notre méthodologie. Le simulateur a passé deux types de tests : théoriques et expérimentaux. Nos premiers tests théoriques consistaient à comparer le revenu horaire simulé et théorique dans des conditions idéales ; c'est-à-dire lorsque les temps morts sont nuls. Ensuite, pour vérifier les assignations faites par le simulateur, nous avons conçu un modèle exact qui trouve la solution optimale au problème de répartition par énumération. Par ailleurs, le modèle exact a permis d'estimer le revenu horaire par taxi maximal théorique à 36\$/h/taxi. Les tests expérimentaux comparaient les indicateurs mesurés par Téo Taxi à ceux obtenus par simulation. Tous les tests étaient concluants.

Par la suite, les algorithmes pour les trois sous-problèmes ont été caractérisés grâce au simulateur. Les conclusions qui ressortent sont les suivantes :

Problème de répartition (PR) :

1. **L'algorithme de répartition avec rayon** : L'avantage de cette stratégie est la diminution du temps d'attente moyen pour les clients. Toutefois, le fait d'ignorer les requêtes des clients trop loin entraîne une diminution du revenu si la demande n'est pas assez grande. Cette stratégie ne devrait être employée qu'en période de forte demande.
2. **L'algorithme de répartition avec file d'attente** : La file d'attente accroît la capacité du système et hausse le revenu (au plus 13%) lorsque la demande augmente. Il y a peu d'effets sur le temps d'attente lorsque la demande est faible et il augmente en période de forte demande (au plus 50%).
3. **L'algorithme de répartition par affectation** : Cette stratégie possède une performance dont l'allure ressemble à celle de la précédente. Cependant, ses effets sur les indicateurs sont plus prononcés. Elle génère à la fois une hausse du revenu (au plus 21%) et du temps d'attente (au plus 75%).

Problème de gestion de l'autonomie (PGA) : La valeur ajoutée de notre heuristique n'est pas ap-

parente à travers les indicateurs de performance définis initialement. Le fait d'ajouter les appels à la borne (associés au seuil $T2$) diminue le revenu horaire et accroît le temps d'attente, car ils réduisent la disponibilité des véhicules. Notre heuristique se veut plus sécuritaire et stable que l'heuristique de base (qui utilise seulement le seuil $T1$). Elle permet une utilisation des stations beaucoup plus uniforme dans le temps.

Problème de localisation (PL) : La stratégie de localisation est la seule qui permet à la fois une augmentation du revenu et une diminution du temps d'attente. Les gains sont modestes, mais ils peuvent être amplifiés avec une réduction de la période de relocalisation.

5.2 Contribution

Ce mémoire dresse les premières avancées dans le domaine de la gestion centralisée de flottes de taxis électriques. À notre connaissance, le modèle d'affaire de Téo Taxi est unique dans le monde. Jusqu'à maintenant, aucune recherche ne s'était attardée à concevoir des stratégies pour la gestion d'une flotte composée de chauffeurs de taxi salariés. Nous sommes les premiers à présenter un système qui inclut à la fois la répartition, la localisation et la gestion de l'autonomie pour une flotte de taxis électriques.

Nous avons conçu un outil d'aide à la décision : un simulateur à événements discrets qui reproduit le comportement d'une flotte de taxis électriques. Dans le cadre du projet, le simulateur était notre outil de caractérisation : une méthode standard pour comparer et évaluer la performance des différentes stratégies opérationnelles. L'étendue de l'utilité d'un tel simulateur ne s'arrête pas là. L'impact des décisions stratégiques peut également être chiffré par simulation : la taille et la forme du territoire, la taille de la flotte, la composition de la flotte, le nombre et la position des stations de recharge, le système de gestion des pauses des chauffeurs, etc.

5.3 Limites

Lors de la conception de nos modèles et de notre simulateur, nous avons la nécessité d'établir certaines hypothèses. Elles sont indispensables pour obtenir des problèmes compréhensibles et raisonnables. Les nombreuses hypothèses que nous énumérons ici sont, à notre avis, les plus significatives :

1. **Modélisation des déplacements des taxis :**

- Une régression linéaire est utilisée pour estimer le temps de déplacement.
- Les taxis suivent une ligne droite lors de leurs déplacements.
- L'effet du trafic est ignoré.
- Les temps de déplacements sont déterministes.
- Les temps pour l'embarquement et le dépôt du client sont nuls.

2. **Modélisation de la demande :**

- La densité horaire de la demande est constante dans le temps et l'espace.
- L'abandon des clients ne dépend que du temps d'attente. Nous négligeons les abandons après la commande de taxi.

5.4 Perspectives

Comme nous l'avons démontré dans la revue de la littérature, les perspectives de recherche sont nombreuses pour le problème de gestion de flotte de taxis électriques. L'utilisation de véhicules électriques et d'un système centralisé provoque des défis nouveaux. Nous présentons dans ce qui suit les perspectives prometteuses qui complètent nos avancées.

La première amélioration possible consiste à ajouter une dimension stochastique aux algorithmes. Celle-ci a été négligée dans le cadre de cette recherche pour trois raisons :

1. **Modèle de prévision de la demande :** Pour employer un algorithme stochastique, un modèle fiable de prévision de la demande est indispensable. Présentement, un tel modèle n'existe pas pour la demande de Téo Taxi. Sa conception demande un effort considérable. Elle constitue un projet en soi.
2. **Temps de calcul :** les effets stochastiques compliquent nécessairement la résolution du problème. Nous avons le mandat de concevoir des algorithmes efficaces. Une attention particulière devrait être donnée à chaque sous-problème pour trouver des méthodes plus sophistiquées, mais tout aussi rapides à résoudre.
3. **Accès aux données :** Téo Taxi est un nouveau projet et possède peu de données historiques. Pour le moment, il est difficile d'établir avec confiance l'étendue de la variabilité des paramètres

comme le temps de déplacement ou le temps de permutation.

La deuxième perspective consiste en une analyse de sensibilité plus poussée. Dans le contexte de ce projet, nous avons fixé plusieurs paramètres comme la taille de la flotte (50 taxis), la taille du territoire, le rayon d'action des stations de recharge (10 minutes), etc. Il pourrait être intéressant d'évaluer les interactions entre ceux-ci. Par exemple, une question à se poser : Est-ce que nos heuristiques se comportent adéquatement lorsque le taux de décharge global s'approche de la capacité des stations? Sinon, comment les modifier pour améliorer leur fiabilité?

Au niveau algorithmique, les pistes de recherche les plus prometteuses pointent vers les modèles génériques qui permettent la résolution des trois sous-problèmes simultanément. Powell (2007) proposent un modèle orienté objet qui généralise les notions d'offre, de demande et de décision. La méthodologie est appliquée au problème de répartition dynamique de transporteurs routiers. L'offre est l'ensemble A de véhicules; leurs attributs sont conservés dans un vecteur $a \in A$. De la même façon, les caractéristiques des requêtes sont dans les vecteurs b qui forment l'ensemble B . Enfin, un ensemble de décisions D est défini; l'élément $d \in D$ est souvent associé à une requête b , mais ce n'est pas toujours le cas. Le contenu de d est très général, il peut représenter un repositionnement, un retour vers un dépôt, un appel vers une borne de recharge, etc. Le vecteur de variables décisionnelles $x_t = (x_{tad})_{a \in A, d \in D}$ définit le nombre de véhicules du type a affecté à une décision du type d à un temps t . Cette formulation permet de prendre en compte tous les types de décisions possibles et génère la solution optimale théorique du problème. L'outil de choix pour résoudre ce type de problème est l'ADP (*Approximate Dynamique Programming*). Cette approche se démarque par sa complexité, mais également par les gains substantiels qu'elle apporte en termes de réduction de coûts.

À notre avis, la piste de recherche la plus intéressante concerne la gestion de l'offre, c'est-à-dire l'évaluation du nombre de taxis optimale. Nos simulations suggèrent que la décision qui affecte le plus la rentabilité est le nombre d'effectifs en place pour desservir la demande. Ce choix est toutefois complexe, car il influence le niveau de service; les impacts sur la clientèle sont difficiles à prédire. Selon nos hypothèses, le nombre optimal de taxis est toujours égal à un, car nous voulons maximiser le revenu unitaire (par taxi). Évidemment, cette solution est absurde, car nous négligeons l'effet du temps d'attente sur la demande. Comment le niveau de service aujourd'hui affecte-t-il l'intention de commander

un taxi demain? Une analyse comportementale doit être faite pour pouvoir chiffrer la valeur d'une minute d'attente et ainsi l'inclure dans le processus de prise de décision.

La taille de flotte optimale dépend également des coûts d'opération qui ont été négligés jusqu'à présent. Initialement, nous avons fait l'hypothèse que les coûts étaient directement proportionnels au nombre de taxis. Si c'est le cas, la maximisation des revenus par taxi est équivalente à la maximisation des profits. Assurément, ce n'est pas le cas en pratique. Le nombre de taxis en service a des effets non linéaires sur le coût. D'ailleurs, un modèle de gestion de l'offre doit également prendre en compte les contraintes humaines associées aux quarts de travail et la disponibilité des chauffeurs. À cela s'ajoute la gestion des pauses qui constitue un problème exhaustif en soi.

Éventuellement, les coûts opérationnels pourraient être comparés aux coûts marketing pour la gestion de la demande. Comme nous l'avons vu, la demande de taxi est très volatile; elle varie de manière importante au cours de la journée. Il peut être trop complexe ou coûteux d'ajuster le nombre de taxis en conséquence. À ce moment, nous pouvons privilégier l'utilisation de la tarification différenciée, par exemple. L'idée est de stabiliser la demande pour faciliter la synchronisation avec l'offre.

La gestion de l'offre doit être faite adéquatement pour obtenir des bénéfices significatifs avec les algorithmes que nous avons développés. En particulier, pour les règles de répartitions, nous avons montré que les gains en performance sont les plus importants en période de forte demande. C'est à ce moment également que le revenu horaire est le plus grand. Toutefois, le niveau de service est affecté à la baisse. En résumé, les mécanismes qui permettent d'augmenter la rentabilité sont assez bien compris. Il suffit de maximiser le ratio entre la demande et le nombre de taxis. La difficulté provient de la gestion des attentes clients. Comment faire pour augmenter leur seuil de tolérance au temps d'attente?

Comme nous l'avons démontré, il reste encore beaucoup d'opportunités de recherche dans ce domaine. Notre contribution reste modeste dans chacun des axes que nous avons explorés. Notre intention était d'avoir une vue d'ensemble sur la question. Nous voulions proposer des stratégies accessibles et pertinentes dans un contexte appliqué. À notre avis, ce travail constitue un bon point de départ à l'atteinte de cet objectif.

Bibliographie

- Alshamsi, A., Abdallah, S., et Rahwan, I. (2009). Multiagent self-organization for a taxi dispatch system. *8th International Conference on Autonomous Agents and Multiagent Systems*, Budapest, 10-15 mai, p. 21-28.
- Andersson, T. et Värbrand, P. (2007). Decision support tools for ambulance dispatch and relocation. *Journal of the Operational Research Society*, 58(2) :195–201.
- Bratley, P., Fox, B. L., et Schrage, L. E. (1987). *A guide to simulation*. 2ème éd., New York, Springer-Verlag, 397 p.
- Brotcorne, L., Laporte, G., et Semet, F. (2003). Ambulance location and relocation models. *European Journal of Operational Research*, 147(3) :451–463.
- Carpenter, T., Curtis, A. R., et Keshav, S. (2014). The return on investment for taxi companies transitioning to electric vehicles. *Transportation*, 41(4) :785–818.
- Commission des transports du Québec (2013). Mécanisme de fixation des tarifs de transport par taxi. Récupéré de http://www.ctq.gouv.qc.ca/taxi/tarifs_de_transport_par_taxi/mecanisme_de_fixation_des_tarifs.html.
- Commission sur le transport et les travaux publics (2014). Véhicules en libre-service (VLS) : une nouvelle offre de mobilité à encadrer.
- Daganzo, C. F. (1978). An approximate analytic model of many-to-many demand responsive transportation systems. *Transportation Research*, 12(5) :325–333.
- Daskin, M. S. (2008). What you should know about location modeling. *Naval Research Logistics*, 55(4) :283–294.
- De Vany, A. S. (1975). Capacity utilization under alternative regulatory restraints : an analysis of taxi markets. *The Journal of Political Economy*, 83(1) :83–94.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1) :269–271.

- Douglas, G. W. (1972). Price regulation and optimal service standards : The taxicab industry. *Journal of Transport Economics and Policy*, 6 :116–127.
- Gendreau, M., Laporte, G., et Semet, F. (2001). A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Computing*, 27(12) :1641–1653.
- Google (2015). Google Maps/Google Earth APIs terms of service. Récupéré de <https://developers.google.com/maps/terms>.
- Gouvernement du Québec (2015a). Projets pilotes dans le secteur du transport par taxi. Récupéré de <https://www.transports.gouv.qc.ca/fr/entreprises-partenaires/entreprises-taxi/Pages/projets-pilotes-taxi.aspx>.
- Gouvernement du Québec (2015b). *Loi concernant les services de transport par taxi*. c. S-6.01. Récupéré de <http://legisquebec.gouv.qc.ca/fr/ShowDoc/cs/S-6.01>.
- Grau, J. M. S. et Romeu, M. A. E. (2015). Agent based modelling for simulating taxi services. *Procedia Computer Science*, 52 :902–907.
- Hasheminezhad, M. et Bahreininejad, A. (2010). A multi-agent taxi dispatching system. *International Journal of Agent Technologies and Systems*, 2(2) :1–10.
- Horni, A., N. K. et Axhausen, K. (2016). The Multi-Agent Transport Simulation MATSim. London : Ubiquity Press. DOI : <http://dx.doi.org/10.5334/baw>. License : CC-BY 4.0.
- Kümmel, M., Busch, F., et Wang, D. Z. (2016). Taxi dispatching and stable marriage. *Procedia Computer Science*, 83 :163–170.
- Lee, D.-H., Wang, H., Cheu, R., et Teo, S. (2004). Taxi dispatch system based on current demands and real-time traffic conditions. *Transportation Research Record*, 1882 :193–200.
- Lu, J.-L., Yeh, M.-Y., Hsu, Y.-C., Yang, S.-N., Gan, C.-H., et Chen, M.-S. (2012). Operating electric taxi fleets : A new dispatching strategy with charging plans. *2012 IEEE International Electric Vehicle Conference*, Greenville, 4-8 mars, p. 1-8.
- Maciejewski, M., Bischoff, J., et Nagel, K. (2016). An assignment-based approach to efficient real-time city-scale taxi dispatching. *IEEE Intelligent Systems*, 31(1) :68–77.

- Maciejewski, M. et Nagel, K. (2013). Simulation and dynamic optimization of taxi services in MATSim. VSP Working Paper 13-05, TU Berlin, Transport Systems Planning and Transport Telematics.
- Panday, A. et Bansal, H. O. (2015). Hybrid electric vehicle performance analysis under various temperature conditions. *Energy Procedia*, 75 :1962–1967.
- Powell, W. B. (1987). An operational planning model for the dynamic vehicle allocation problem with uncertain demands. *Transportation Research Part B : Methodological*, 21(3) :217–232.
- Powell, W. B. (2007). Real-time dispatching for truckload motor carriers. Dans G. Don Taylor (dir.), *Logistics Engineering Handbook*, CRC Press, p. 323-342.
- Pureza, V. et Laporte, G. (2008). Waiting and buffering strategies for the dynamic pickup and delivery problem with time windows. *INFOR : Information Systems and Operational Research*, 46(3) :165–176.
- Ross, A. (2015). Queues : Balking, reneging, and retrials. Récupéré de <https://people.emich.edu/aross15/q/brr.html>.
- Salanova, J. M., Estrada, M., Aifadopoulou, G., et Mitsakis, E. (2011). A review of the modeling of taxi services. *Procedia-Social and Behavioral Sciences*, 20 :150–161.
- Sanderson, C. et Curtin, R. (2016). Armadillo : a template-based C++ library for linear algebra. *Journal of Open Source Software*, 1 :1–26.
- Sathaye, N. (2014). The optimal design and cost implications of electric vehicle taxi systems. *Transportation Research Part B : Methodological*, 67 :264–283.
- Savelsbergh, M. W. et Sol, M. (1995). The general pickup and delivery problem. *Transportation Science*, 29(1) :17–29.
- Seow, K. T., Dang, N. H., et Lee, D.-H. (2010). A collaborative multiagent taxi-dispatch system. *IEEE Transactions on Automation Science and Engineering*, 7(3) :607–616.
- Sheffi, Y. (1985). *Urban Transportation Networks : Equilibrium Analysis with Mathematical Programming Methods*. Englewood Cliffs, Prentice-Hall, 416 p.

- Song, Z. (2006). *A simulation based dynamic model of taxi service*. [thèse de mémoire], Hong Kong, The University of Hong Kong, 54 p.
- Vaz, W., Nandi, A. K., Landers, R. G., et Koylu, U. O. (2015). Electric vehicle range prediction for constant speed trip using multi-objective optimization. *Journal of Power Sources*, 275 :435–446.
- Wang, H. et Cheu, R. (2013). Operations of a taxi fleet for advance reservations using electric vehicles and charging stations. *Transportation Research Record*, 2352(1) :1–10.
- Wang, H., Cheu, R. L., et Lee, D.-H. (2014). Intelligent taxi dispatch system for advance reservations. *Journal of Public Transportation*, 17(3) :115–128.
- Yang, H. et Wong, S. (1998). A network model of urban taxi services. *Transportation Research Part B : Methodological*, 32(4) :235–246.
- Zhou, H., Liu, C., Yang, B., et Guan, X. (2015). Optimal dispatch of electric taxis and price making of charging stations using stackelberg game. *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, Yokohama, 9-12 novembre, p. 4929-4934.