



HAL
open science

Optimisation du Positionnement de Voitures en Autopartage basée sur la Prédiction de leur Utilité

Gregory Martin, Matthieu Donain, Elisa Fromont, Tias Guns, Laurence Roze,
Alexandre Termier

► **To cite this version:**

Gregory Martin, Matthieu Donain, Elisa Fromont, Tias Guns, Laurence Roze, et al.. Optimisation du Positionnement de Voitures en Autopartage basée sur la Prédiction de leur Utilité. Conférence Nationale en Intelligence Artificielle 2022 (CNIA 2022), Jun 2022, Saint-Etienne, France. pp.1-10. hal-03866192

HAL Id: hal-03866192

<https://hal.science/hal-03866192>

Submitted on 22 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Optimisation du Positionnement de Voitures en Autopartage basée sur la Prédiction de leur Utilité

G. Martin^{1,2}, M. Donain¹, E. Fromont^{2,3}, T. Guns⁴, L. Roze⁵, A. Termier²

¹ Stellantis

² Univ Rennes, Inria, CNRS, IRISA

³ Institut Universitaire de France (IUF)

⁴ Vrije Universiteit Brussel / KULeuven

⁵ Univ Rennes, INSA, Inria, CNRS, IRISA

{gregory.martin, elisa.fromont, laurence.roze, alexandre.termier}@irisa.fr

Résumé

Le succès d'un service d'autopartage en *free-floating* dépend d'une bonne répartition des véhicules dans la ville, c'est-à-dire où et quand les utilisateurs en ont besoin. Cela nécessite de prédire la demande des utilisateurs, ce qui présente un défi en raison du peu de données disponibles et de la variabilité de ces demandes. L'objectif de ces prédictions est d'aider à calculer la meilleure relocalisation possible de voitures pour le jour suivant. Ainsi il est nécessaire de modéliser à la fois une tâche de prédiction de la demande et une tâche d'optimisation du placement des véhicules. Comme l'optimisation du placement des voitures implique un raisonnement sur le nombre de voitures à affecter dans des zones précises de la ville, nous proposons de convertir le problème de prédiction de la demande en prédiction de l'utilisation attendue d'une voiture lorsqu'elle est placée dans une zone de la ville. Nous abordons les défis liés à la modélisation de cette prédiction et du problème de la relocalisation des véhicules de la flotte du service d'autopartage. Ainsi, une méthode d'optimisation linéaire en nombres entiers est proposée pour résoudre ce problème de relocalisation tout en tenant compte de la prédiction de l'utilisation des véhicules selon leur placement et des distances de relocalisation. Les expériences se basent sur deux ensembles de données provenant de deux services d'autopartage en *free-floating* urbains et nous montrons comment notre méthode peut améliorer les stratégies de relocalisation et donc la rentabilité des services.

Mots-clés

Autopartage, Optimisation Linéaire en Nombres Entiers, Régression.

Abstract

The success of a *free-floating* car-sharing service depends on a good allocation of the vehicles across the city, i.e., where and when they are needed by citizens. This requires predicting the demand across the geographical regions and across time, which is challenging due to the sparsity and variability of the data. Furthermore, the purpose of these pre-

dictions is to help compute the best possible car relocation for the next day, hence the need to model both the prediction task and the optimization task in a compatible way. As the allocation optimization involves reasoning about the number of cars to assign to geographical regions, we propose to convert the prediction problem into predicting the expected utilization of a car when added to a region. We discuss the challenges in modeling both the machine learning and the relocation problem, and we propose a mixed-integer linear programming method that solves the relocation problem while taking into account the model predictions and relocation distances. We experiment with two datasets from citywide car sharing companies and show how our method can increase the allocation strategies and hence profitability of the services.

Keywords

Carsharing, Integer Programming, Regression

1 Introduction

L'autopartage en *free-floating* a émergé comme une nouvelle méthode de transport public dans les zones urbaines. Il contribue à réduire l'empreinte carbone des villes en permettant à des voitures d'être partagées par de multiples utilisateurs en ville. Contrairement aux services d'autopartage *one-way*, il n'y a pas de stations prédéfinies dans un service en *free-floating* : les voitures peuvent être prises et garées sur n'importe quelle place de parking, généralement en ville et dans ses environs.

Quand les utilisateurs d'un service d'autopartage ont besoin d'une voiture, il est indispensable que des véhicules assez proches soient disponibles. À cette fin, les opérateurs emploient du personnel, appelé *jockeys*, pour repositionner les véhicules dans la ville. Cette opération est généralement effectuée de manière empirique, principalement la nuit et à petite échelle, avec une connaissance approximative des zones à forte demande où les véhicules doivent être déplacés. Plusieurs articles ont constaté l'absence de stratégies de relocalisation précise, et ont conçus des méthodes

qu'un opérateur doit appliquer [16] ou des incitations afin que les utilisateurs envisagent de terminer leurs trajets dans des zones sélectionnées par l'opérateur [2].

Dans cet article, nous cherchons à maximiser l'utilisation de la flotte de voitures d'un service afin de rendre le service durable et rentable pour l'opérateur. Nous considérons l'utilisation de jockeys pour relocaliser les voitures durant la nuit, comme seule action quotidienne concrète prise par le service d'autopartage pour augmenter l'utilisation de la flotte de véhicules. Par conséquent, l'objectif est de proposer un emplacement pour toutes les voitures de la flotte au début de la journée, de manière à maximiser l'utilisation durant la journée prédite des voitures.

Il existe plusieurs stratégies pour résoudre ce problème. La plus précise consisterait à prévoir les déplacements exacts des clients pour un jour donné, c'est-à-dire le nombre de déplacements, et pour chaque déplacement, l'origine, la destination et l'heure de départ. En sachant cela, nous pourrions envisager les différentes manières de relocaliser les voitures au début de la journée afin de maximiser la satisfaction du client. Ainsi, cela nous permettrait de modéliser la dynamique de la flotte afin de simuler différentes incitations au stationnement et stratégies de relocalisation des voitures. Cependant cela est en pratique extrêmement difficile à réaliser. À la place, nous proposons une stratégie alternative moins fine, mais efficace qui est décomposée en deux parties. Tout d'abord, nous prédisons l'utilisation prévue des voitures lorsqu'elles sont placées dans des zones bien définies en ville au début de la journée, la prédiction est faite en minutes d'utilisation. Ensuite en fonction de ces prédictions, une stratégie de relocalisation des voitures est conçue afin de maximiser l'utilisation totale attendue du service en une journée. Le problème de relocalisation est présenté sous la forme d'un problème d'optimisation linéaire en nombres entiers sur les prédictions, assujéti à des contraintes pratiques sur la relocation des véhicules.

La Section 2 expose les travaux de l'état de l'art. Une modélisation générale de la ville, une stratégie de prédiction de l'utilité d'une voiture et une méthodologie d'affectation des voitures sont exposées dans la Section 3. La prédiction de l'utilité des voitures et la relocalisation des véhicules sont évaluées dans la Section 4 avec deux ensembles de données propriétaires provenant de services réels. Nous concluons dans la Section 5.

2 État de l'Art

Prédiction des trajets pour les services de partage de véhicules. Dans un contexte de vélopartage, [7] proposent une méthode pour une stratégie de relocalisation par l'opérateur où les vélos peuvent être relocalisés à tout moment de la journée. Leur solution n'est pas directement applicable à notre contexte d'autopartage, car la relocalisation des voitures n'est possible que la nuit dans notre cas. De plus, les données sont beaucoup plus rares dans un contexte d'autopartage. Une autre stratégie pour localiser les voitures tous les matins serait de pouvoir prédire la demande des clients et tous les déplacements effectués au cours d'une

journée. Cette stratégie est liée à la prédiction des matrices d'origine-destination. [8] ont proposé une telle méthode pour un service de vélopartage *one-way*, où les vélos peuvent être pris dans une station et laissés dans une autre. Dans notre contexte d'autopartage, le nombre de points d'origine et de destination à considérer est beaucoup plus élevé que pour les services de vélopartage et le nombre de trajets est beaucoup plus faible : il en résulte des tenseurs binaires origine/destination très creux, en considérant le temps comme troisième dimension. Les propriétés de nos tenseurs empêchent également d'utiliser des méthodes statistiques standard telles que (S)ARIMA [15] pour des prédictions (saisonnnières) de couple origine-destination d'intérêt. Par rapport au vélopartage, une autre difficulté vient de la dépendance entre les déplacements en voiture et la forte influence des conditions de circulation non saisonnières (travaux de construction, événements spéciaux, conditions météorologiques extrêmes) et des particularités de la ville (présence de bâtiments culturels, d'écoles, de supermarchés, etc.).

Dans [13], il est montré que la location des voitures dépend de facteurs comme la géographie ou la condition sociale. Ces facteurs sont difficiles à prendre en compte sans une grande quantité d'informations exogènes. Nous discutons de l'utilité de certains d'entre eux dans la Section 4. Le problème de la prédiction de déplacements binaires dans un tenseur peut être lié à la prédiction de liens dans un graphe/réseau dynamique [1]. Cependant, la définition d'un score qui refléterait l'affinité entre deux zones impliquant l'existence d'un voyage à un moment donné, de manière similaire à ce qui peut être fait pour les personnes et les liens dans un réseau social, est difficile et source d'erreurs dans notre contexte. En particulier parce que dans un contexte de *free-floating*, chaque paire possible de noeuds doit être considérée, ce qui empêche l'utilisation de scores de similarité basés sur la topologie du réseau comme ceux présentés dans [10].

Stratégies de relocalisation pour les services de partage de véhicules. Plusieurs travaux dans la littérature sur le vélopartage [4, 5, 11] abordent le problème de la relocalisation des vélos. Par exemple, dans [11], les auteurs optimisent conjointement le nombre de trajets par jour et certains coûts de relocalisation. Ainsi est prise en compte la distance totale parcourue par les camions relocalisant les vélos, sachant qu'ils peuvent déplacer plusieurs vélos par trajet. Dans [4, 5] les auteurs améliorent cette solution en proposant des relocalisations continues dans la journée tout en prenant en compte la demande des usagers au cours de la journée.

De nombreuses stratégies de relocalisation ont également été proposées pour les voitures : dans [17], les auteurs proposent une optimisation linéaire en nombres entiers (*ILP*) pour optimiser l'utilisation des voitures dans un service d'autopartage *one-way* où les relocalisations se font en continu pendant la journée. Leur modèle d'*ILP* vise à réduire le nombre de demandes non satisfaites durant la journée, tout en minimisant le nombre de trajets de relocalisa-

tions effectués par le personnel. Les stratégies proposées dans [3] et [16] ont été conçues pour un service d'autopartage en *free-floating*, mais elles ne considèrent que le problème de relocalisation en soi : étant donné en entrée une distribution initiale des voitures dans la ville et une distribution finale attendue, elles minimisent les coûts de relocalisation. Cependant il n'y a aucune prise en compte des bénéfices apportés par les voitures bien localisées.

Comme nous l'avons expliqué dans l'introduction, nous prenons ces solutions comme inspiration sans pouvoir les appliquer directement à notre problème d'autopartage en *free-floating*. En effet dans notre problème il n'y a pas de stations fixes, la flotte contient peu de voitures, par rapport à des vélos d'un service de vélopartage, un jockey ne peut déplacer qu'une voiture à la fois, les relocalisations ont lieu pendant la nuit et nous sommes intéressés par l'optimisation du nombre de minutes conduites et non du nombre de trajets.

3 Méthodologie

Notre objectif est de proposer une affectation pour le lendemain matin de toutes les voitures d'un service d'autopartage en *free-floating* de telle sorte à ce qu'elle maximise leur utilisation prédite ce jour-là. Nous détaillons d'abord la modélisation géographique de la ville, puis la manière dont le problème de prédiction de l'utilité d'une voiture est réalisé et enfin nous définissons le modèle de relocalisation des voitures selon la prédiction de leur usage.

3.1 Modélisation de la Ville et des Trajets

Pour des raisons d'efficacité et de généralité, il n'est pas possible de modéliser une ville comme étant avec l'ensemble de toutes les places de stationnement. D'après les résultats de l'étude [14], nous avons discrétisé la surface de la ville en une grille hexagonale où chaque hexagone (aussi appelé *cellule* par la suite) a 500 mètres pour rayon. Cette distance représente ce qu'est prêt à parcourir un client trouver une voiture disponible [14]. L'ensemble des hexagones de cette grille est noté K . Toutes les positions situées dans la surface d'une cellule sont étiquetées avec l'indice k de l'hexagone correspondant. Notez que les hexagones inutiles sont supprimés, ce sont ceux dans lesquels aucun déplacement n'a commencé ou ne s'est terminé. Dans la suite, nous pouvons également regrouper, par clustering, les cellules (hexagones) qui se comportent de manière similaire selon certains critères donnés. Notez que cela revient simplement à avoir un plus petit nombre K de cellules différentes.

3.2 Modélisation de l'Utilité

Il est supposé que les données historiques sont un ensemble de *trajets* utilisateur, à savoir un ensemble de tuples tel que : $trajets = \{(id_{vehicule}, ts_{depart}, ts_{arrivee}, gps_{depart}, gps_{arrivee}, distance, duree)\}$. Les trajets de jockeys, c'est-à-dire les déplacements des employés dans le but de repositionner les véhicules pendant la nuit, sont exclus des données. À partir de l'horodatage (ts), des caractéristiques supplémentaires sont déduites telles que des informations météorologiques sur la température

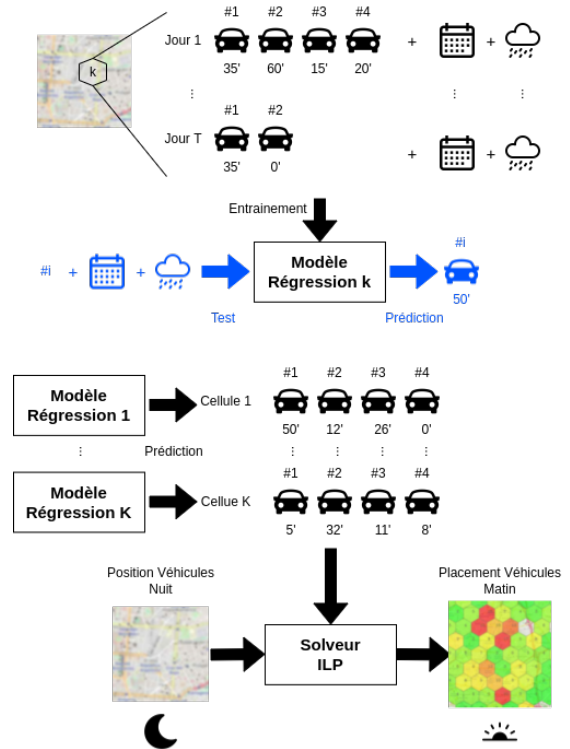


FIGURE 1 – Vue d'ensemble de notre approche : un modèle de régression est entraîné pour chaque cellule afin de prédire, à partir de données historiques, le temps d'utilisation de chaque voiture (par ordre de départ) sachant des attributs temporels et météorologiques. Le solveur d'optimisation utilise à la fois les régresseurs appris pour prédire le temps d'utilisation de chaque voiture et le positionnement de la flotte avant la phase de relocalisation pour obtenir le meilleur positionnement des voitures pour le lendemain matin.

et les précipitations. Des informations de lieux de départ et d'arrivée (*gps*) sont fournies sous forme de coordonnées GPS, qui sont mises en correspondance avec les cellules géographiques précédemment définies. De plus, des informations supplémentaires sur le trajet sont disponibles, comme les kilomètres parcourus (*distance*) ou la durée du trajet en minutes (*duree*). Un résumé de ces informations est présenté en haut de la Figure 1.

Utilité Réelle. À partir des données historiques d'une seule journée, nous voulons calculer le temps d'utilisation total, c'est-à-dire l'utilité totale, $U_k^*(n)$ dans chaque cellule k étant donné la présence de n voitures disponibles dans cette cellule au début de la journée. Il n'est pas suffisant de compter uniquement l'utilité des déplacements de cette cellule vers d'autres cellules, car les voitures peuvent avoir été utilisées pour des trajets ultérieurs au cours de la journée. À partir de ce constat, l'utilité totale est calculée pour chaque véhicule :

$$U^*(id_{vehicule}) = \sum_{t \in trajets(id_{vehicule})} utilite(t)$$

où $trajets(id_{vehicule})$ est l'ensemble des trajets pour une

journée dont l'identifiant du véhicule est $id_{vehicule}$ et l'utilité $utilite(t)$ d'un trajet t correspond à la durée du trajet. Soit $matin_vehicules(k)$ l'ensemble des identifiants des véhicules qui se trouvaient dans la cellule k au début de la journée, avec $|matin_vehicules(k)| = n$. L'utilité réelle d'une cellule k qui a eu n véhicules initiaux est mesurée tel que :

$$U_k^*(n) = \sum_{id_{vehicule} \in matin_vehicules(k)} U^*(id_{vehicule})$$

Notez que cette valeur ne peut être calculée avec les données historiques que pour $n = |matin_vehicules(k)|$.

Utilité Estimée. Notre objectif est maintenant de prédire $U_k(n)$ pour toute cellule k et toute valeur n , sans savoir quels trajets seront effectués et quels véhicules seront utilisés pour ceux-ci. Ainsi, il s'agit d'un problème de régression où les données d'apprentissage sont les utilités réelles définies ci-dessus. Cependant, comme seuls les $U_k^*(n)$ sont observés pour un n par cellule par jour, cela donne un ensemble d'apprentissage creux et peu susceptible de conduire à un modèle de régression correct pour des valeurs arbitraires de n .

Ainsi le problème de régression de $U_k^*(n)$ est décomposé, afin d'avoir plus de données d'entraînement et des prédictions plus fines. Les voitures sont d'abord ordonnées par ordre de première utilisation (première voiture, deuxième voiture, etc.). Le but est de modéliser la relation temporelle entre les véhicules par rapport au moment de leur première utilisation dans la journée. Il est maintenant possible de définir la valeur ajoutée de la i^{eme} voiture comme $U_k^i = utilite(id_{vehicule}^i)$ avec $id_{vehicule}^i$ l'identifiant du véhicule initialement dans la cellule k qui a été ordonné (indice i) par horodatage de première utilisation. Nous avons alors $U_k^*(n) = \sum_{i \in 1..n} U_k^i$. À partir de cette reformulation, nous identifions l'objectif de prédiction du gain d'utilité U_k^i apporté par la i -ème voiture dans une cellule k . La décomposition de la valeur d'utilité et ce problème de régression plus fin permettent d'avoir plus de données d'entraînement, et donc de mieux estimer $U_k^*(n)$ pour différentes valeurs de n (notamment pour $n < |matin_vehicules(k)|$).

Prédiction. L'objectif est de prédire U_k^i pour chaque cellule k et rang i en utilisant les données d'apprentissage comme définies précédemment, cette prédiction est appelée \hat{U}_k^i . Comme les cellules peuvent avoir un comportement différent, un modèle de régression f_k sera appris par cellule. Comme attributs, nous utilisons le rang i et un ensemble f_s d'attributs exogènes comprenant des attributs météorologiques, des attributs temporels et les observations d'utilité des jours précédents (car les données sont séquentielles). Par conséquent, un régresseur f_k est appris par cellule tel que

$$\hat{U}_k^i = f_k(i, f_s).$$

Nous discutons des modèles de régression à utiliser dans la Section 4.3.

Clustering des Cellules. Au lieu d'apprendre un prédicteur f_k par cellule, il est possible d'utiliser les similarités dans le comportement de l'utilité pour différentes cellules.

En regroupant les cellules, davantage de données sont disponibles pour l'entraînement de chaque modèle et, idéalement, les prédictions sont plus précises. Dans la Section 4 est étudié l'effet du *clustering* des cellules dont le comportement est similaire en termes d'utilité, avec un modèle f_c appris par *cluster* c de cellule.

3.3 Optimisation de la Relocalisation des Véhicules

Relocalisation Naïve. Étant donné \hat{U}_k^i , l'utilisation prédite de la i ème voiture située dans la cellule k , et une limite Γ sur le nombre de voitures disponibles, le problème de relocalisation des voitures peut se résumer à déterminer la meilleure assignation de ces Γ voitures dans les différentes cellules. Il faut noter que si \hat{U}_k^i était linéaire en i , par exemple $\hat{U}_k^i = i \cdot w$ pour une certaine valeur w , alors le problème de relocalisation serait un problème de sac à dos non borné de capacité Γ , où chaque cellule représente un élément. Cependant cette simple hypothèse de linéarité ne tient pas, une formulation différente est utilisée pour modéliser le problème de relocalisation. Plus précisément, nous utilisons l'optimisation linéaire en nombres entiers (*ILP*) pour le modéliser comme un problème d'affectation des véhicules aux cellules, où la capacité totale Γ et l'ordre temporel i des véhicules dans chaque cellule doivent être respectés. Des variables booléennes V_{ki} sont utilisées pour indiquer s'il y a un i -ème véhicule dans la cellule k . Le nombre total de véhicules dans une cellule k est donc $\sum_i V_{ki}$. Avec K l'ensemble de tous les identificateurs de cellule définis précédemment, m étant une limite supérieure du nombre de véhicules autorisés dans une cellule et $I = \{1..m\}$. Soit $V = (V_{ki})_{k \in K, i \in I}$, alors la formulation *ILP* du problème est :

$$\underset{V}{argmax} \sum_{k \in K} \sum_{i \in I} \hat{U}_k^i V_{ki} \quad (1)$$

$$s.t. \sum_{k \in K} \sum_{i \in I} V_{ki} \leq \Gamma \quad (2)$$

$$V_{ki} \geq V_{ki+1} \quad \forall k \in K, i \in I \setminus \{m\} \quad (3)$$

$$V_{ki} \in \{0, 1\} \quad \forall k \in K, i \in I \quad (4)$$

L'équation (2) est une contrainte de capacité, tandis que l'équation (3) garantit qu'il n'y a pas d'« écart » dans le rang des variables indicatrices, c'est-à-dire que si $V_{ki} = 0$, alors tout $j > i$ doit être 0 également. Par exemple, s'il n'y a pas de troisième véhicule dans une cellule, il n'y a pas non plus de quatrième véhicule, cinquième véhicule, etc.

Relocalisation basée sur les Jockeys. La formulation précédente calcule le placement optimal, mais ne tient pas compte du fait que 1) à la fin de la journée, les véhicules sont déjà situés dans certaines cellules, et 2) les voitures doivent être relocalisées par des jockeys, d'où la nécessité de prendre en compte leur capacité limitée et les coûts associés à chaque relocalisation.

Pour répondre à la remarque 1), nous désignons par $s_{ki} \in \{0, 1\}$ la présence ou absence d'une i -ème voiture dans la

cellule k à la fin du jour précédent. Pour répondre à la remarque 2), nous incluons une limite γ sur le nombre total de relocalisations possibles de véhicules que les jockeys peuvent effectuer. De plus les jockeys ne pouvant pas se téléporter, ils doivent donc être déplacés par une voiture dite « balai » vers la voiture suivante après avoir relocalisé une voiture précédente. D'un point de vue opérationnel, ces trajets en voiture-balai doivent être minimisés, c'est pourquoi est introduit q_{kl} le coût du déplacement d'un jockey par une voiture-balai de la cellule k à la cellule l ; et p le prix par minute payé par les utilisateurs qui louent une voiture afin que les profits/coûts de la relocalisation puissent être équilibrés. L'objectif est maintenant de trouver la relocalisation qui conduit au plus grand revenu prédit possible moins les coûts de dépose des jockeys, tout en devant effectuer au plus γ relocalisations. Pour modéliser ce problème sous la forme d'un programme linéaire en nombres entiers, nous utilisons deux ensembles de variables de décision booléennes : F_{ki} indique que la i -ème voiture doit être retirée cellule k , tandis que T_{ki} indique qu'une voiture doit être relocalisée dans la cellule k pour être sa i -ème voiture. En utilisant ces variables binaires, la présence ou non d'une voiture à la i -ème position dans la cellule k est calculée par $s_{ki} - F_{ki} + T_{ki}$, qui remplace V_{ki} dans la formulation précédente. Enfin, E_{kl} représente le nombre de trajets de voiture-balai nécessaires entre les cellules k et l , avec q_{kl} le coût d'un de ces trajets de staff.

En intégrant ce résultat dans le problème d'optimisation précédent et en ajoutant les contraintes supplémentaires appropriées, on obtient la formulation suivante :

$$\begin{aligned} \underset{F,T,E}{\operatorname{argmax}} \quad & \sum_{k \in K} \sum_{i \in I} p \cdot \hat{U}_{ki} \cdot (s_{ki} - F_{ki} + T_{ki}) \\ & - \sum_{k \in K} \sum_{l \in K} q_{kl} \cdot E_{kl} \end{aligned} \quad (5)$$

such that :

$$s_{ki} - F_{ki} \geq 0 \quad \forall k \in K, i \in I \quad (6)$$

$$s_{ki} + T_{ki} \leq 1 \quad \forall k \in K, i \in I \quad (7)$$

$$\sum_{k \in K} \sum_{i \in I} T_{ki} = \sum_{k \in K} \sum_{i \in I} F_{ki} \quad (8)$$

$$\sum_{k \in K} \sum_{i \in I} T_{ki} \leq \gamma, \sum_{k \in K} \sum_{i \in I} F_{ki} \leq \gamma \quad (9)$$

$$(s_{ki} - F_{ki} + T_{ki}) \geq (s_{ki+1} - F_{ki+1} + T_{ki+1}) \quad \forall k \in K, i \in I \setminus \{m\} \quad (10)$$

$$\sum_{l \in K} E_{kl} = \sum_{j \in I} T_{kj} \quad \forall k \in K \quad (11)$$

$$\sum_{k \in K} E_{kl} = \sum_{i \in I} F_{li} \quad \forall l \in K \quad (12)$$

$$F_{ki} \in \{0, 1\}, T_{ki} \in \{0, 1\} \quad \forall k \in K, i \in I \quad (13)$$

$$E_{kl} \in \{0, \dots, |I|\} \quad \forall k, l \in K \quad (14)$$

La fonction objectif à l'équation (5) maximise le revenu obtenu grâce à l'utilité prédite par rapport au placement des véhicules tout en prenant en compte les coûts des trajets de voiture-balai pour emmener les jockeys entre les voitures à déplacer. L'équation (6) indique que seuls les véhicules présents à la fin de la journée peuvent être déplacés pendant la nuit, tandis que l'équation (7) indique que si un véhicule est déjà présent à cet emplacement, un autre véhicule ne peut pas être ajouté. Ces deux contraintes indiquent effectivement que, selon s_{ki} , soit F_{ki} , soit T_{ki} sera égal à 0. L'équation (8) indique que le nombre de prises et de déposes de jockeys doivent être les mêmes, tandis que l'équation (9) garantit que le nombre de voitures relocalisées est inférieur au budget de relocalisation γ . L'équation (10) garantit qu'il n'y a pas d'« écart » dans le positionnement final, cette contrainte est obtenue en substituant $s_{ki} - F_{ki} + T_{ki}$ dans l'équation (3) de la première formulation. Enfin, l'équation (11) s'assure qu'à chaque voiture relocalisée en k , alors une voiture-balai doit récupérer le jockey. De la même manière, l'équation (12) assure qu'il y a une voiture-balai qui dépose un jockey pour chaque voiture retirée de l .

4 Expérimentations

4.1 Données

Les expériences sont réalisées sur deux jeux de données provenant de deux services réels d'autopartage en *free-floating*. Ces jeux de données étant sensibles pour l'entreprise dont ils sont issus, nous les appelons « ServiceA » et « ServiceB ».

Le premier jeu de données représente 1138246 trajets effectués entre mi-2018 et début-2019. Les données couvrent 243 jours pour une moyenne de 4684 trajets quotidiens. Les données ont été séparées en trois sous-ensembles : ensemble d'entraînement, de validation et de test. L'ensemble d'entraînement regroupe les trajets des premiers 145 jours, l'ensemble de validation sont les 48 jours suivants et l'ensemble de test sont les derniers 50 jours.

Le second jeu de données est constitué de 130219 trajets effectués entre début-2019 et début-2020. Les données couvrent 306 jours pour une moyenne de 425 trajets quotidiens. Comme pour le premier jeu de données, trois sous-ensembles ont été créés : l'ensemble d'apprentissage correspond aux 204 premiers jours, l'ensemble de validation aux 51 jours suivants et enfin l'ensemble de test aux 51 derniers jours.

Chaque trajet dans les deux jeux de données possède l'identifiant du véhicule, les positions GPS de départ et d'arrivée et les horodatages de départ et d'arrivée. Déduit des trajets, des attributs temporels sont rajoutés, tels que le fait qu'un jour soit ouvert ou non et le jour de la semaine.

Deux ensembles de données météorologiques sont utilisés comme données exogènes pour entraîner les régresseurs. Ils sont récupérés à partir des bulletins météorologiques horaires SYNOP de la station météorologique la plus proche pour chaque service. À partir des bulletins SYNOP, nous extrayons la température (°C), l'humidité relative (%), la

pression (hPa), la vitesse du vent (km/h), la couverture nuageuse (%) et la quantité de pluie (mm). Certains rapports horaires étant manquants, nous avons calculé les valeurs manquantes en considérant un changement linéaire entre les données connues avant et après l'entrée manquante.

4.2 Algorithme pour le Clustering des Cellules

Nous comparons une stratégie sans *clustering* (*No Clustering*) où chaque fonction f_k est entraînée pour toutes les cellules k de la grille de la ville. Une comparaison est faite à la stratégie basée sur un *K-Medoids clustering*. Ainsi, chaque cellule est représentée par un vecteur tel que la i -ème coordonnée du vecteur est le temps d'utilisation moyen en minutes de la voiture de rang i . Comme la taille du vecteur décrivant chaque cellule dépend du rang le plus élevé de la voiture rencontré dans cette cellule particulière, tous les vecteurs de cellule n'ont pas la même taille. Ainsi, la distance entre des vecteurs de tailles différentes, i et $i + h$ (avec $h \geq 0$), n'est calculée que sur les i premiers rangs de voiture communs entre les deux vecteurs. La distance euclidienne entre ces deux parties communes divisée par sa longueur commune est donnée à l'algorithme de clustering. Le nombre optimal de clusters est trouvé en utilisant l'ensemble de validation pour chaque modèle et ensemble d'attributs. L'utilité d'une cellule k devient $U_k^i = f_c(i, f_s)$ où f_c est l'utilité prédite du cluster c qui contient la cellule k (avec f_s les attributs).

La zone desservie par « ServiceA » est couverte par 151 cellules et la zone desservie par « ServiceB » est couverte par 199 cellules. Ce plus grand nombre de cellules dans le second cas (avec moins de voitures dans la flotte) aura des conséquences dans les tâches de prédiction et d'optimisation suivants.

4.3 Algorithme de Prédiction d'Utilité

Les régresseurs testés sont 1) la moyenne des valeurs historiques (référence), 2) un régresseur par *gradient boosting* (*GBR*) et 3) un régresseur par machine à vecteur de support (*SVR*). Différents ensembles d'attributs (FS_1 , FS_2 , FS_3 , FS_4) ont été utilisés pour l'entraîner des modèles.

Attributs. Le premier ensemble d'attributs FS_1 ne contient aucun attribut exogène, de sorte que seul le rang i de chaque voiture est utilisé pour entraîner les régresseurs. Le deuxième ensemble d'attributs FS_2 contient des informations exogènes : le *jour de la semaine*, le fait que ce jour soit un *jour ouvrable ou non* et les *informations météorologiques*. Les informations météorologiques sont la moyenne des bulletins horaires pour chaque attribut météorologique et jour. Ainsi FS_2 contient donc 13 attributs. Le troisième ensemble d'attributs FS_3 en contient deux concernant l'historique en plus de FS_2 pour une cellule donnée k avec un rang donné i : les deux dernières (dans le temps) valeurs d'utilité $f_k(i)$ connues dans les données. Enfin, le quatrième ensemble FS_4 contient l'état de charge des véhicules électriques au début de la journée, en plus de FS_3 . Cet ensemble n'est disponible que pour le jeu de données de « ServiceB ».

Modèles. Notre modèle de référence (*Moy*) est la moyenne des valeurs d'utilité historique pour chaque cellule, pour un rang donné i pour tous les jours de l'ensemble d'apprentissage. Ce modèle de référence utilise uniquement FS_1 .

Les implémentations de Scikit-learn¹ sont utilisés pour le *Gradient Boosting Tree (GBR)* et la machine à vecteur de support (*SVR*) avec des paramètres choisis par validation. Chaque modèle prédit directement l'utilité d'une voiture dans chaque cellule pour les quatre ensembles d'attributs mentionnés précédemment et avec ou sans clustering préalable des cellules. Chaque attribut a été normalisé en soustrayant la moyenne et en divisant par l'écart-type.

À titre d'information, d'autres approches et stratégies d'apprentissage automatique ont été testées, mais ont donné des résultats peu convaincants. Il s'agit notamment d'une approche alternative en deux étapes : un modèle de classification est d'abord utilisé pour prédire si une voiture quitte ou non la cellule et, dans le cas échéant, un deuxième modèle est utilisé pour prédire l'utilité de cette voiture. Comme le premier modèle de classification n'est pas parfait (environ 70% de précision), l'erreur est propagée à la deuxième étape et l'ensemble du processus en deux étapes donne des résultats moins bon que les stratégies de prédiction actuelles. Une deuxième approche testée n'utilise pas i comme attribut et prédit directement l'ensemble du vecteur d'utilité pour chaque cellule, c'est-à-dire qu'elle utilise des modèles de régression multiple. Cette approche a obtenu de moins bons résultats que le modèle de référence *Mean*, et n'est donc pas retenu.

4.4 Métrique d'Évaluation

Nous utilisons le *Mean Absolute Error (MAE)* pour mesurer le nombre moyen de minutes surestimées ou sous-estimées par chaque modèle de régression entraîné pour chaque cellule pendant un jour, défini comme : $MAE = \frac{1}{|K|} \sum_{k \in K} \sum_{i \in I} |obs_{ki} - pred_{ki}|$, avec obs_{ki} l'utilité réelle de la $i^{ème}$ voiture dans la cellule k et $pred_{ki}$ l'utilité prédite de la $i^{ème}$ voiture dans la cellule k .

Nous utilisons aussi un ratio pour chaque jour entre la *MAE* et le temps d'utilisation réel moyen par cellule, appelé ici le *Ratio Mean Absolute Error (RMAE)* :

$$RMAE = \frac{MAE}{(1/|K|) \cdot \sum_{k \in K} \sum_{i \in I} obs_{ki}}$$

4.5 Stratégies de Relocalisation

Stratégies. Afin d'évaluer notre approche *ILP* pour relocaliser les véhicules en ville pour le lendemain matin, comme décrit dans la Section 3.3, nous allons comparer le revenu attendu de quatre stratégies différentes. Nous désignons par *Historique* la stratégie basée sur la position historique de la flotte chaque matin, et son utilisation réelle au cours de la journée. Le revenu correspondant est le nombre de minutes d'utilisation total multiplié par le prix payé par les utilisateurs pour la location d'une voiture. *Optim MU* désigne une stratégie de relocalisation qui maximise (avec notre modèle

1. <https://scikit-learn.org/>

Clustering	Attributs	ServiceA		
		Moy	GBR	SVR
Aucun Clustering	FS_1	288 ± 27 (45%)	288 ± 27 (45%)	289 ± 28 (45%)
	FS_2	N/A	288 ± 25 (44%)	283 ± 21 (44%)
	FS_3		290 ± 24 (45%)	284 ± 21 (44%)
KMedoids	FS_1	289 ± 27 (45%)	289 ± 28 (45%)	288 ± 28 (45%)
	FS_2	N/A	278 ± 25 (43%)	278 ± 25 (43%)
	FS_3		278 ± 24 (43%)	277 ± 24 (43%)
Clustering	Attributs	ServiceB		
		Moy	GBR	SVR
Aucun Clustering	FS_1	90 ± 25 (92%)	90 ± 25 (91%)	92 ± 31 (90%)
	FS_2	N/A	97 ± 26 (99%)	101 ± 32 (99%)
	FS_3		97 ± 25 (98%)	100 ± 32 (98%)
	FS_4		94 ± 25 (95%)	100 ± 32 (98%)
KMedoids	FS_1	90 ± 25 (92%)	91 ± 26 (92%)	91 ± 31 (88%)
	FS_2	N/A	90 ± 26 (91%)	98 ± 31 (95%)
	FS_3		90 ± 25 (90%)	95 ± 31 (93%)
	FS_4		85 ± 26 (84%)	95 ± 31 (92%)

TABLE 1 – Performance *MAE* (la plus faible est la meilleure) avec l'écart-type et *RMAE* de la prédiction d'utilité des voitures par jour et par (groupe de) cellule(s), en utilisant une base historique (*Moy*) et deux modèles de régression (*GBR* et *SVR*) avec différents ensembles d'attributs (FS_1 , FS_2 , FS_3 et FS_4) dans le cas du « ServiceA » puis du « ServiceB ».

d'ILP et notre prédiction d'utilité) les utilités des voitures sans tenir compte des coûts. *Optim SO* désigne la stratégie, décrite dans la Section 3.3, qui utilise un objectif unique. Cet objectif maximise le revenu de l'utilisation du service moins les coûts de trajets des voitures-balais. *Optim DO* désigne une stratégie d'optimisation en deux étapes : la première maximise l'utilité de la flotte, comme pour *Optim MU*, tandis que la seconde, après avoir fixé l'emplacement optimal des véhicules, minimise les coûts des trajets de voiture-balai.

Prédiction de l'Utilité. L'utilité utilisée pour évaluer le revenu gagné par le service avec notre approche est un mélange entre l'utilité historique et l'utilité prédite : lorsque le solveur ILP propose une solution de relocalisation différente de l'historique, l'utilité prédite est utilisée pour les voitures pour lesquelles cette valeur est inconnue.

Conversion d'Utilité et Estimation des Coûts. Pour toutes les stratégies, nous supposons que le prix p payé par l'utilisateur par minutes de conduite est constant. Ainsi le revenu attendu de la flotte est : l'utilité totale multipliée par p . Le coût d'un trajet de la voiture-balai q_{kl} , décrit dans la Section 3.3, est calculé en fonction du salaire horaire brut d'un jockey c_j qui doit être doublé (lors du repositionnement d'un jockey de k à l , il y a deux jockeys dans la voiture-balai : celui qui doit être emmené et le conducteur de la voiture-balai), avec le coût c_v de fonctionnement d'une voiture-balai par km, s la vitesse moyenne (en km/h) d'une voiture à l'intérieur de la ville et la distance d_{kl} (en km) entre la cellule k où un jockey est pris en charge par la voiture-balai et la cellule l où il est déposé :

$$q_{kl} = \frac{d_{kl} \cdot (2 \cdot c_j)}{s} + d_{kl} \cdot c_v$$

De plus après une observation du fonctionnement du Servi-

ceA, une hypothèse est faite sur un maximum de 70 voitures relocalisées chaque nuit, donc $\gamma = 70$, ce qui correspond à 7 jockeys qui peuvent relocaliser 10 voitures chacun.

4.6 Étude de Cas

Nous voulons évaluer à la fois la partie prédiction et la partie relocalisation de notre approche pour les deux services. Dans le premier cas, nous évaluons la qualité de la prédiction de $f_k(i, fs)$ pour une cellule k , un rang de voiture i et un ensemble de caractéristiques fs avec les différents régresseurs décrits dans la Section 4.3. Dans le deuxième cas, nous calculons le revenu quotidien attendu de chaque système d'autopartage lorsque l'utilité est prédite et utilisée pour résoudre le problème de relocalisation.

4.6.1 Performance des Modèles de Régression

Les colonnes du Tableau 1 présentent la performance des régresseurs utilisés, c'est-à-dire la moyenne historique (*Moy*), le *Gradient Boosting Tree* (*GBR*) et la machine à vecteur de support (*SVR*). Les lignes décrivent les résultats avec ou sans clustering en utilisant les trois (ou quatre si disponibles) ensembles d'attributs présentés précédemment. La *MAE* (en min) et *RMAE* (en %) sont calculés en fonction de l'utilité réelle dans chaque cellule et chaque jour et au nombre prédit d'utilité sur l'ensemble de test. Comme indiqué précédemment, la performance de la moyenne historique n'a pas été calculée en utilisant les ensembles d'attributs FS_2 , FS_3 et FS_4 car sa validité statistique serait discutable.

ServiceA. On peut remarquer qu'avec FS_1 , la moyenne historique donne l'une des pires *MAE* (~288 mins), ce qui montre le potentiel des modèles d'apprentissage automatique et des attributs que nous avons choisis lorsque les données sont suffisantes.

L'ajout d'attributs temporels (le jour de la semaine et le

fait qu'il s'agisse d'un jour ouvrable ou non) et météorologiques augmente de manière significative, dans la plupart des cas, les performances des régresseurs (les lignes avec FS_2 présentent des MAE et $RMAE$ plus faibles que les lignes avec FS_1) et en particulier lors de l'utilisation d'un algorithme de clustering *K-Medoids* pour regrouper les cellules. L'ajout d'informations sur les valeurs passées dans FS_3 profite davantage au modèle de régression *SVR* (le modèle de régression *GBR* conserve des résultats stables lors de l'utilisation d'un *K-Medoids*). Dans tous les cas sauf le plus simple (*Moy* et/ou seulement FS_1 comme attributs), l'utilisation d'un clustering sur les cellules similaires permet de réduire, la plupart du temps de manière significative, l'erreur tout en gardant la variance des erreurs stable. Le regroupement de cellules similaires permet aux régresseurs de s'entraîner sur un plus grand nombre de données avec les mêmes caractéristiques, ce qui permet de mieux généraliser l'ensemble d'entraînement. Ces résultats permettent de choisir pour notre approche de régression le régresseur *SVR* avec un clustering *K-Medoids* des cellules similaires tout en utilisant l'ensemble d'attributs FS_3 : cette combinaison offre une réduction de 2% du $RMAE$ par rapport à la référence *Moy*, ainsi qu'une réduction de 10 minutes (en moyenne par cellule et par jour) de la MAE . Cette combinaison sera utilisée pour les expériences suivantes.

ServiceB. Cet ensemble de données pose un défi beaucoup plus important à l'algorithme de prédiction que le premier. La première raison est que le nombre de véhicules est plus faible dans ce service (500 par rapport à 600 dans le ServiceA) alors que, comme nous l'avons déjà dit, le nombre de cellules couvertes par le service est plus grand (199), donc le nombre de cellules où il y a très peu d'activité est beaucoup plus grand que dans le ServiceA ce qui nuit à toutes les performances moyennes par cellule. En outre, et pour les mêmes raisons, le service est également moins actif, car il peut être gênant pour l'utilisateur de parcourir une longue distance à pied pour louer une voiture disponible. De ce fait, le nombre de voitures dans chaque cellule (correspondant à la taille du vecteur de rang à prédire) est beaucoup plus faible et l'utilité de chaque voiture est souvent (à un taux de 55%) nulle dans le jeu de données.

Cela peut expliquer les performances relativement bonnes de la référence (*Moy*) avec une erreur de ~ 90 mins qui correspond à près de 92% de la moyenne par jour et cellule du total des minutes de conduite, par rapport à la prédiction des modèles d'apprentissage automatique du Tableau 1. Pour ce jeu de données, l'algorithme *GBR* est plus performant que le *SVR*. Comme pour l'autre ensemble de données, les résultats sont meilleurs lorsque les cellules sont regroupées et le fait de disposer d'informations sur l'état de charge de la voiture (FS_4) permet à l'algorithme *GBR* d'obtenir les meilleures performances. Les meilleurs résultats de prédiction fournissent une MAE de ~ 85 minutes, ce qui correspond à 84% de la moyenne par jour et à la cellule du total des minutes conduites. Comme nous le verrons dans la section suivante, ces résultats de prédiction loin d'être parfaits, nous permettent tout de même d'améliorer le service global ce qui nous laisse espérer une marge d'amélioration beau-

coup plus importante.

4.6.2 Performance de la Relocalisation

À partir de maintenant, nous appelons « placement optimal » la solution optimale à notre problème d'optimisation trouvée par le solveur propriétaire Gurobi [6]. L'utilité (nombre de minutes d'utilisation) est estimée en utilisant le meilleur modèle de régression trouvé dans l'expérience précédente. Ainsi, dans le cas du ServiceA, un algorithme de clustering *K-medoids* des cellules de la ville est suivi d'un régresseur *SVR*, utilisant l'ensemble d'attributs FS_3 . Dans le cas du ServiceB, un algorithme de clustering *K-medoids* est appliqué suivi d'un régresseur *GBR*, utilisant l'ensemble d'attributs FS_3 car la formulation de la relocalisation ne prend pas en compte l'état de charge.

Pour toutes les stratégies, les revenus quotidiens totaux estimés par les deux services sont calculés. Ils sont représentés par jour dans la Figure 2 et les moyennes journalières sont indiquées pour les deux ensembles de données dans le Tableau 2. Nous nous attendons à ce que le revenu soit plus élevé pour *Optim MU* puisqu'il ne prend pas en compte les coûts de relocalisation, mais cette stratégie est également moins réaliste. Nous nous attendons également à ce que *Optim SO* donne un revenu plus important que *Optim DO* puisque la stratégie en une étape peut prendre en compte les coûts de relocalisation de la voiture, ce qui devrait aider à détecter les relocalisations de véhicules non rentables. Dans ces deux derniers cas, le coût des jockeys est intégré à la valeur de revenu affiché.

ServiceA. Le Tableau 2 montre que pour ce service, toutes les méthodes d'optimisation contribuent (en moyenne) à augmenter les revenus de l'entreprise. Des détails supplémentaires dans la Figure 2 (en haut) montrent qu'il n'y a que trois jours où ce n'est pas le cas. Néanmoins le gain est de presque 10% par rapport au service actuel (*Historique*) pour le reste de l'ensemble de test. Pour ce service, étonnamment, il n'y a presque aucune différence entre les stratégies de relocalisation. Cela est probablement dû au fait que le service est très utilisé et a donc une utilité qui est proportionnée par rapport à de potentiels coûts liés aux relocalisations. Cela explique également pourquoi il n'y a pas de différence entre les stratégies d'optimisation en deux étapes (*Optim DO*) et en une seule étape (*Optim SO*) puisque le coût n'influence pas significativement les relocalisations.

ServiceB. Les résultats pour le ServiceB, présentés dans le Tableau 2 et dans la Figure 2 (en bas), sont en grande partie cohérents avec ce que nous attendions. De manière plus surprenante, nous pouvons constater que la stratégie *Optim DO* donne des résultats inférieurs à la référence *Historique*, tandis que les autres stratégies d'optimisation sont meilleures (+9% pour *Optim MU* et +5% pour *Optim SO*). Cependant les résultats sont moins impressionnants que pour le ServiceA. La raison en est l'inverse de celle de la première ville : comme le service est moins utilisé de manière absolue, la proportion des coûts de relocalisation est plus grande et la variation dans le plan de ramassage des voitures-balais compte plus sur la fonction objectif de la stratégie. Cela montre également l'importance de la straté-

Stratégie	ServiceA		ServiceB	
	Revenu (Base 100)	Nb Relocalisation	Revenu (Base 100)	Nb Relocalisation
Historique	100	N/A	100	N/A
Optim MU	109	53	109	42
Optim SO	107	49	105	42
Optim DO	107	53	99	42

TABLE 2 – Revenu quotidien moyen et nombre de relocalisations pour quatre stratégies de relocalisation dans le cas du ServiceA et du ServiceB.

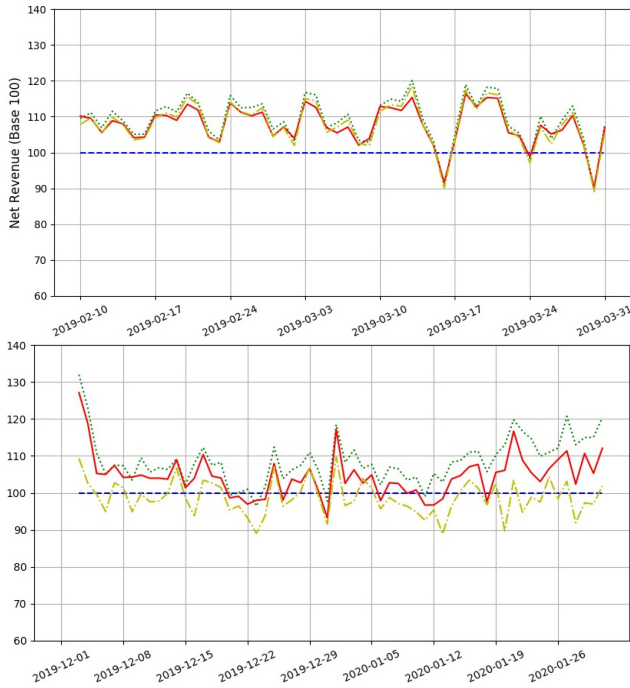


FIGURE 2 – Revenus quotidiens estimés par les services fonctionnant dans le a) ServiceA (en haut) et le b) ServiceB (en bas) (une graduation d’abscisse correspond à un dimanche), avec la stratégie Historique (ligne bleue pointillée horizontale à 100) comme indice de référence (base 100). La ligne pleine rouge, les lignes pointillées jaunes et vertes sont respectivement les revenus normalisés (sur la référence) fournis par les stratégies *Optim SO*, *Optim DO* et *Optim MU*. Le coût des jockeys est pris en compte dans les valeurs de revenu pour *Optim SO* et *Optim DO*.

gie d’optimisation en une étape (*Optim SO*) par rapport à celle en deux étapes (*Optim DO*) lorsque les coûts sont élevés. Pour ServiceB, notre stratégie la plus réaliste (*Optim SO*) donne donc une augmentation de 5% du bénéfice de l’entreprise.

5 Conclusion

Nous avons abordé le problème de l’optimisation de la relocalisation des véhicules dans un service de partage de voitures en *free-floating* afin d’augmenter son utilisation quotidienne et sa rentabilité. Nous avons proposé de prédire l’utilisation des voitures et montré comment ces prédictions pouvaient être utilisées avec un modèle d’optimisation li-

néaire en nombres entiers pour optimiser l’utilisation des véhicules tout en prenant en compte les coûts des trajets de voiture-balai pour les jockeys. Bien que la phase de prédiction puisse être encore améliorée en introduisant davantage de données exogènes, notre approche peut déjà augmenter l’efficacité et la rentabilité du ServiceA et du ServiceB jusqu’à 7% et 5% respectivement.

Notre solution de relocalisation pourrait bénéficier d’une formulation plus poussée pour optimiser l’itinéraire de la voiture-balai et de tous les jockeys (ces deux problèmes sont liés aux problèmes de tournées de véhicules [12]). Cela serait utile, mais non trivial : l’optimisation conjointe des tournées des véhicules et de notre problème de relocalisation nécessite l’ajout de contraintes supplémentaires et les problèmes sont d’une ampleur non triviale pour des solveurs ILP génériques ; et la combinaison avec le routage complet de la voiture-balai et des jockeys serait plus difficile. Une autre direction serait de s’inspirer de [9], et d’utiliser le résultat de l’optimisation comme fonction de perte pour la formation des prédicteurs, plutôt que de former d’abord les prédicteurs et d’optimiser ensuite les prédictions, afin d’obtenir des prédictions plus axées sur la décision au prix de temps de calcul plus élevé.

Remerciements

Ce travail a été réalisé en collaboration entre le Groupe Stelantis et Inria à travers l’*OpenLab Artificial Intelligence* et a été soutenu par le programme CIFRE de l’ANRT.

Références

- [1] Charu C. Aggarwal and Karthik Subbian. Evolutionary network analysis : A survey. *ACM Comput. Surv.*, 47(1) :10 :1–10 :36, 2014.
- [2] Alfred Benedikt Brendel, Julian Brennecke, Patryk Zapadka, and Lutz Kolbe. A Decision Support System for Computation of Carsharing Pricing Areas and its Influence on Vehicle Distribution. *ICIS 2017 Proceedings*, December 2017.
- [3] Carl Axel Folkestad, Nora Hansen, Kjetil Fagerholt, Henrik Andersson, and Giovanni Pantuso. Optimal charging and repositioning of electric vehicles in a free-floating carsharing system. *Computers & Operations Research*, 113 :104771, January 2020.
- [4] Supriyo Ghosh, Jing Yu Koh, and Patrick Jaillet. Improving Customer Satisfaction in Bike Sharing Sys-

- tems through Dynamic Repositioning. In *IJCAI*, pages 5864–5870, 2019.
- [5] Supriyo Ghosh, Pradeep Varakantham, Yossiri Adulyasak, and Patrick Jaillet. Dynamic repositioning to reduce lost demand in bike sharing systems. *Journal of Artificial Intelligence Research*, 58 :387–430, 2017.
- [6] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2020.
- [7] Pierre Hulot, Daniel Aloise, and Sanjay Dominik Jena. Towards Station-Level Demand Prediction for Effective Rebalancing in Bike-Sharing Systems. In *Proc. of the 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, pages 378–386, 2018.
- [8] Yexin Li and Yu Zheng. Citywide bike usage prediction in a bike-sharing system. *IEEE Transactions on Knowledge and Data Engineering*, 32(6) :1079–1091, 2020.
- [9] Jayanta Mandi, Emir Demirovic, Peter J. Stuckey, and Tias Guns. Smart predict-and-optimize for hard combinatorial optimization problems. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 1603–1610, 2020.
- [10] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A Survey of Link Prediction in Complex Networks. *ACM Comput. Surv.*, 49(4) :69 :1–69 :33, December 2016.
- [11] Tal Raviv, Michal Tzur, and Iris A. Forma. Static repositioning in a bike-sharing system : models and solution approaches. *EURO Journal on Transportation and Logistics*, 2(3) :187–229, 2013.
- [12] M. W. P. Savelsbergh and M. Sol. The General Pickup and Delivery Problem. *Transportation Science*, 29(1) :17–29, February 1995.
- [13] Stefan Schmöller, Simone Weikl, Johannes Müller, and Klaus Bogenberger. Empirical analysis of free-floating carsharing usage : The Munich and Berlin case. *Transportation Research Part C : Emerging Technologies*, 56 :34–51, 2015.
- [14] Rene Seign and Klaus Bogenberger. Prescriptions for the successful diffusion of carsharing with electric vehicles. In *Conference on Future Automotive Technology*, 2013.
- [15] W.W.S. Wei. *Time Series Analysis : Univariate and Multivariate Methods*. Pearson Addison Wesley, 2006.
- [16] Simone Weikl and Klaus Bogenberger. A practice-ready relocation model for free-floating carsharing systems with electric vehicles – Mesoscopic approach and field trial results. *Transportation Research Part C : Emerging Technologies*, 57 :206–223, 2015.
- [17] Rabih Zakaria, Mohammad Dib, Laurent Moalic, and Alexandre Caminada. Car relocation for carsharing service : Comparison of CPLEX and greedy search. In *2014 IEEE Symp. on Comp. Intel. in Vehicles and Transportation Systems*, pages 51–58, December 2014.