HAL open science

# Prediction-Based Fleet Relocation for Free Floating Car Sharing Services

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.. Prediction-Based Fleet Relocation for Free Floating Car Sharing Services. ICTAI 2021 - 33rd International Conference on Tools with Artificial Intelligence, Nov 2021, Virtual, United States. pp.1-5. hal-03491955

**HAL Id: hal-03491955**

**https://hal.science/hal-03491955v1**

Submitted on 18 Dec 2021

**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.

# Prediction-Based Fleet Relocation for Free Floating Car Sharing Services

Gregory Martin*†, Matthieu Donain*, Elisa Fromont†‡, Tias Guns§, Laurence Roze¶, Alexandre Termier†
* *Stellantis, 78943 Vélizy, France*
† *Univ Rennes, Inria, CNRS, IRISA, 35000 Rennes, France* ‡ *Institut Universitaire de France (IUF)*
§ *Vrije Universiteit Brussel / KULeuven, Belgium* ¶ *Univ Rennes, INSA, Inria, CNRS, IRISA, 35000 Rennes, France*
{gregory.martin, elisa.fromont, laurence.roze, alexandre.termier}@irisa.fr

***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 predictions is to help computing the best possible car positions for the next day, hence the need to model both the prediction task and the optimisation task in a compatible way. As the allocation optimisation involves reasoning about the number of cars to assign to geographical regions, we propose to predict the expected utilisation 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 integer linear programming method that solves the relocation problem while taking into account the model predictions and relocation distances. We experiment with the datasets from a citywide car sharing company and show how our method can increase the allocation strategies and hence profitability of the service.**

## I. Introduction

Free floating car-sharing has emerged as a new public transportation method within urban areas. It contributes to reducing the cities' carbon footprint by allowing shared cars to be freely taken, used and dropped by customers within the city. Contrarily to one-way car-sharing services, in a free-floating service there is *no predefined station*: the cars can be taken and parked on any public standard parking place, usually within a single city and its surroundings.

Customers of car-sharing services prefer to have available cars in a close vicinity when needed. To this aim, the operators employ staff, called *jockeys*, to relocate vehicles inside the city. This is usually done empirically, mostly during the night and at a small scale, with a rough knowledge of high-demand areas. Several papers have discussed the lack of precise relocation strategies, either by designing methods an operator needs to apply [1] or by designing incentives so that customers may consider ending their trips in areas selected by the operator [2].

In this work, we aim to maximize the use of a service's car fleet in order to make the service sustainable and profitable for the operator. We consider the use of jockeys to relocate cars on a nightly basis, as the concrete daily action taken by car companies to increase shared cars usage. Hence, the goal is to propose a location for all the cars of the fleet at the beginning of the day, so as to maximize expected car use.

There exist multiple strategies to tackle this problem. We choose a two part strategy. First, we predict the expected utilization, in minutes, of cars when placed in defined areas of the city at the beginning of the day. Then, according to these predictions, we design a car relocation strategy to maximize the total expected utilization of the service in one day. The relocation problem is cast as an integer programming problem over the predictions, subject to constraints involved when wishing to re-allocating vehicles in practice.

Section II covers the relevant related work. We propose a general city modeling, a utility prediction strategy and a car assignment methodology in Section III. Both the car utilization prediction and the car relocation are evaluated in Section IV with a proprietary dataset coming from a real service.

## II. Related Work

Several works in the bike-sharing literature [3, 4, 5] are addressing the relocation problem. For example, in [3] the authors jointly optimize the number of trips per day and some relocation costs (the total distance driven by the relocation trucks that can move multiple bikes per trips). In [4, 5] the authors improve this solution by proposing continuous relocations during the day while also taking into account the user demand at all times. Numerous relocation strategies have also been proposed for cars: in [6], the authors solve an integer linear programming (ILP) problem to optimize the utilization of cars in a one-way car-sharing service where relocations are done continuously during the day. Their ILP model aims to reduce the number of unfulfilled demand throughout the day, while minimizing the number of relocations made by the staff. The strategies proposed in [1, 7] were designed for a free-floating car-sharing service, but they only consider the relocation problem *per se*: given as input an initial distribution of cars in the city and an expected final one, they minimize the relocation costs without taking into account the profit provided by the well-located cars. We take inspiration from all these solutions but cannot apply them directly to our free-floating car sharing problem since, in our problem, there is no fixed stations, the fleet contains few cars (compared to bikes), a jockey can only move one car at a time, relocations occur during the night and we are interested in optimizing the number of driven minutes, not the number of trips.

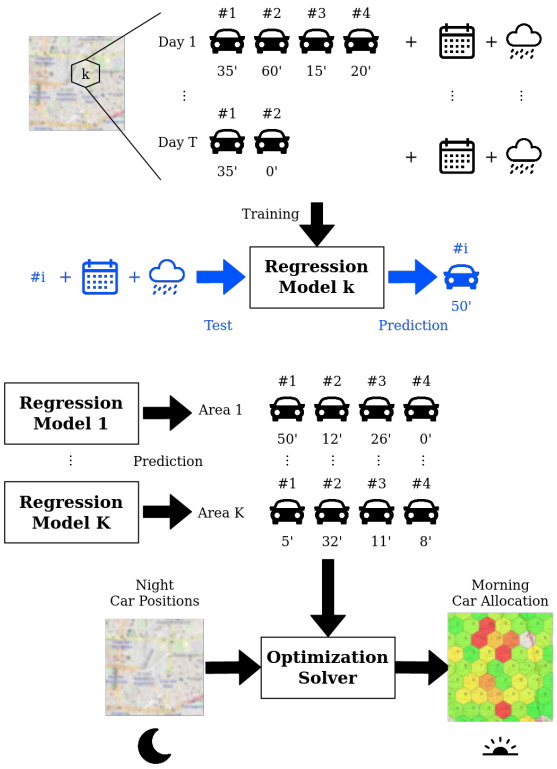


Fig. 1: Overview of the method, a regression model is trained for each cell to predict the driving time of each car, knowing both time and weather-related features. The optimization solver uses both the models learned to predict the driving time of each car and the night car positions to return the best morning car positions.

## III. Method

Our goal is to propose a morning assignment of all cars of a free floating car sharing service such that it maximizes their expected utilization that day. We first discuss the geographic city modeling, how the prediction problem of the expected utility of a car is modeled and finally define a corresponding car assignment problem.

### A. City and Trip Modeling

For efficiency and generality reasons, it is not possible to geographically model a city as the set of all its possible parking spaces. Following a study from [8], we discretized the city surface into an hexagonal grid where each hexagon (also called *cell*) has a circumradius of 500 meters. This distance is the longest one a customer is willing to walk to find an available car of the service [8]. The set of hexagons is noted $K$. All the parking spaces within the surface of the cell are labeled with the index $k$ of this hexagon. Note that in our case, we remove unnecessary hexagons, those where no trip has begun or ended in.

### B. Utility modeling

We assume that the historical data is a set of user *trips*, namely a set of tuples as follows: $trips = \{(id_{car}, ts_{from}, ts_{to}, gps_{from}, gps_{to})\}$. From the timestamp ($ts$) we compute the rent duration. Location information ($gps$) is provided as GPS coordinates, which can be mapped to the geographical cells defined earlier.

*a) Actual utility:* Based on the historical data of a single day, we want to compute the utility of each car for every cell. Since cars can be used for several trips during the day, we compute the actual utility of each car as: $U^*(id_{car}) = \sum_{t \in trips(id_{car})} utility(t)$. Where $trips(id_{car})$ is the set of trips that have this car ID and the utility $utility(t)$ of a trip $t$ is its net revenue (proportional to the trip duration).

Our goal is to predict the potential utility of a car placed in any cell $k$. This can be cast as a regression problem, where the training data are the actual utilities defined above. However the number of cars in each cell $k$ influence the potential utility of an additional car, e.g adding a second or fifteenth car will yield a different utility for the added car. In order to take into account this phenomena, we first order the cars of a cell in a deterministic way, namely by pick-up order ($1^{st}$ car, ..., $n^{th}$ car). This models the temporal relation between cars based on their first pickup. We can now define the added value of an $i^{th}$ car as $U_k^i = utility(id_{car}^i)$ with $id_{car}^i$ the initial car in cell $k$ that was ordered (index $i$) by first pickup time.

*b) Prediction:* We call $\hat{U}_k^i$ the predicted utility of a car in cell $k$ at rank $i$. As cells can behave differently, we will learn a function $f_k$ per cell. As features, we use the rank $i$ and a set $fs$ of features including weather features, temporal features and lagged observations of the previous days. Hence, we wish to learn a regression function $f_k$ per cell such that $\hat{U}_k^i = f_k(i, fs)$. We discuss regression methods to use in Section IV-C.

*c) Clustering cells:* Instead of learning one predictor $f_k$ per cell, we can also use similarities in utility behaviour for different cells. By grouping cells, more data becomes available for every function we wish to learn, and ideally more accurate predictions are made. We will hence investigate in Section IV the effect of clustering the cells and learn one function $f_c$ per cluster $c$.

### C. Cost-based Car Relocation Problem

To maximize the utilization of cars during the day, we use two sets of Boolean decision variables: $F_{ki}$ denotes that the $i^{th}$ car is relocated away *from* cell $k$, while $T_{ki}$ denotes that a car is relocated *to* cell $k$ to be its $i^{th}$ car. Then by $s_{ki} \in \{0, 1\}$ we denote whether there is an $i$th vehicle in cell $k$ at the end of the previous day. Using these variables, choosing whether or not there is a car at the $i^{th}$ position in cell $k$ is computed by $s_{ki} - F_{ki} + T_{ki}$. The decision to add or remove a car from is done knowing the predicted utilization of this car ($\hat{U}_k^i$). Furthermore, since there is a finite number of relocation possible, constrained by the number of jockeys, we limit the number of relocation (constant $\gamma$). Moreover, jockeys cannot teleport hence they must be moved by a so-called 'sweeper' car to the next car after dropping off a vehicle. Operationally, these sweeper trips should be minimized, hence we introduce $q_{kl}$ the cost of a jockey being moved by a sweeper car from cell $k$ to cell $l$; and $p$ the price per minute paid by users

renting a car so that profits/costs can be balanced off. Finally, $E_{kl}$ represents the number of sweeper trips necessary between cell $k$ and $l$, with $q_{kl}$ the cost of one such sweeper trip.

Using the defined variables, the Integer Linear Programming (ILP) formulation of the problem is then:

$$\underset{F,T,E}{maximize} \sum_{k\in K}\sum_{i\in I} p\cdot \hat{U}_{ki}\cdot(s_{ki}-F_{ki}+T_{ki}) - \sum_{k\in K}\sum_{l\in K} q_{kl}\cdot E_{kl} \tag{1}$$

such that:

$$s_{ki}-F_{ki}\geq 0 \quad \forall k\in K, i\in I \tag{2}$$

$$s_{ki}+T_{ki}\leq 1 \quad \forall k\in K, i\in I \tag{3}$$

$$\sum_{k\in K}\sum_{i\in I} T_{ki} = \sum_{k\in K}\sum_{i\in I} F_{ki} \tag{4}$$

$$\sum_{k\in K}\sum_{i\in I} T_{ki}\leq \gamma, \sum_{k\in K}\sum_{i\in I} F_{ki}\leq \gamma \tag{5}$$

$$(s_{ki}-F_{ki}+T_{ki})\geq (s_{ki+1}-F_{ki+1}+T_{ki+1}) \quad \forall k\in K, i\in I\setminus\{m\} \tag{6}$$

$$\sum_{l\in K} E_{kl} = \sum_{j\in I} T_{kj} \quad \forall k\in K \tag{7}$$

$$\sum_{k\in K} E_{kl} = \sum_{i\in I} F_{li} \quad \forall l\in K \tag{8}$$

$$F_{ki}\in\{0,1\}, T_{ki}\in\{0,1\} \quad \forall k\in K, i\in I \tag{9}$$

$$E_{kl}\in\{0,\ldots,|I|\} \quad \forall k,l\in K \tag{10}$$

The objective function in Eq (1) maximizes the total expected revenue from the utility over the computed car placement while taking into account the costs to reposition jockey between cars to relocate. Eq (2) states that only vehicles that are present at the end of the day can be moved away overnight, while Eq (3) states that if there is already a vehicle present then another vehicle can not be moved to the location. These two constraints effectively state that depending on $s_{ki}$ either $F_{ki}$ or $T_{ki}$ will be 0. Eq (4) states that the number of pickups and drop-offs has to be the same, while Eq (5) ensures that both the number of pickups and dropoffs is lower than the relocation budget $\gamma$. Eq (6) ensures that there are no 'gap' in the final assignment. Finally Eq (7) makes sure that each time a car is relocated *to* $k$, then a sweeper car needs to pick up the jockey. Eq (8) assures that there is a sweeper car dropping of a jockey for each car relocated *from* $l$.

## IV. Experiments

### A. Dataset

Our experiments are done on a dataset from a real free-floating car sharing service. This dataset being sensitive for the service, we call it "CarService".

This dataset accounts for 1 138 246 trips made between the mid 2018 and early 2019. It covers 243 days for an average of 4684 daily trips and has been separated into three subsets: training, validation and test set. The training set regroups the trips from the first 145 days, the validation set are the next 48 days and the test set are the last 50 days.

Each trip in the database is tagged with the car identifier, both the origin and destination GPS positions and both the rent and park timestamps. We further tagged the trips with workday/holiday and day-of-week temporal features.

An additional weather dataset is used as exogenous data to train our prediction models. It has been retrieved from the SYNOP hourly weather broadcasts from the nearest weather station. We extract the temperature (°C), the relative humidity (%), the pressure (hPa), the wind speed (km/h), the cloud cover (%) and the amount of rain (mm).

### B. Cell Clustering Algorithms

We compare a strategy without any grouping (*No Clustering*) where each function $f_k$ is estimated independently for all the 151 cells $k$ of the city grid, and compare it to a strategy based on a *K-Medoids* clustering. In the latter, each cell is represented by a *vector* such that the $i^{th}$ coordinate of the vector is the average driven time in minutes of the car of rank $i$. All the cell vectors do not have the same size. Thus the distance between vectors of different sizes $i$ and $i+h$ (with $h>0$) is only computed on the first $i$ common car ranks between both vectors. The Euclidean distance between these two common parts divided by its common length is given to the clustering algorithm. The optimal number (for prediction purpose) of clusters is found using the validation set for each model and feature set. The utility of an cell $k$ becomes $U_k^i = f_c(i, fs)$ where $f_c$ is the predicted utility of the cluster $c$ which contains the cell $k$.

### C. Utility Prediction Algorithms

The regression methods tested are 1) the mean of historical values (baseline), 2) a Gradient Boosting Tree Regressor (GBR) and 3) a Support Vector Machine Regressor (SVR). Different sets of features were used to train the models.

*Features.* The first set of features $FS_1$ contains no features, such that only the rank $i$ of each car is used to train all the regression models. The second set of features $FS_2$ contains exogenous information: the *day of week*, whether this day was a *workday or not* and a *weather information*. The weather information is obtained by averaging the hourly broadcasts for each day, thus creating weather vectors for each day. Overall, $FS_2$ contains 13 attributes. The third set of features $FS_3$ contains two historical features (in addition to $FS_2$) for a given cell $k$ with a given rank $i$: the two last (in time) utility values $f_k(i)$ known (not predicted) in the data.

*Models.* Our baseline model (*Mean*) is the average of the utility values computed for each cell, for a given rank $i$ for all days in the training set. This baseline only uses $FS_1$.

We use the regression variant of Scikit-learn implementations of both the Gradient Boosting Tree (GBR) and the Support Vector Machine (SVR) with the default parameters for each model to directly predict the utility of the car in each cell for the four previously mentioned feature sets and with or without clustering the cells beforehand. Each attribute had been normalized by subtracting the average and dividing by the standard deviation.

### D. Evaluation Metrics

We use the *Mean Absolute Error* (MAE) which is the average number of minutes misestimated by the trained regression models on each cell during one day, defined as: $\mathrm{MAE} = \frac{1}{|K|}\sum_{k\in K}\sum_{i\in I}|obs_{ki} - pred_{ki}|$, with $obs_{ki}$ the real utility of the i$^{th}$ car in cell $k$ and $pred_{ki}$ the predicted utility of the i$^{th}$ car in cell $k$.

We also use a ratio for each day between the MAE and the average real driven time per cell, called here the *Ratio Mean Absolute Error* (RMAE): $\mathrm{RMAE} = \frac{MAE}{(1/|K|)\cdot\sum_{k\in K}\sum_{i\in I} obs_{ki}}$

### E. Relocation Strategies

*a) Strategies:* In order to evaluate our approach to relocate the cars of the fleet, we will compare the expected revenue of four different strategies. We denote by *Historical* the strategy based on the historical fleet position every morning, and its true utilization during the day. The corresponding revenue is the number of driven minutes multiplied by the price paid by users for car renting. *Optim MU* denotes a relocation strategy which maximizes (our utility prediction and ILP formulation) the car utilities without taking into account any costs. *Optim SO* denotes the strategy, described in Section III-C, which uses a joint (Single) Objective. This objective maximizes the service utilization revenue minus the costs to reposition the jockeys. *Optim DO* denotes a two steps (Double) optimization strategy: the first one maximizes the fleet utility (as for Optim MU) while the second one, after fixing the car location found, minimizes the jockeys' repositioning costs.

*b) Utility Estimation:* The ground truth utility used to evaluate the revenue earned by the service with our approach is a mix between the historical utility and the predicted one: when the solver proposes a relocation solution different from the historical one, the predicted utility is used for the cars for which the historical value is unknown.

*c) Utility Conversion and Cost Estimation:* For all the strategies, we assume that the price $p$ paid by the user per driven minutes is constant. Thus, the fleet expected revenue is the total utility multiplied by $p$. The sweeper car cost $q_{kl}$, is computed given the gross hourly salary of two jockeys $2 \cdot c_j$ (a jockey is "repositionned" from $k$ to $l$ by someone driving a sweeper car), the cost $c_s$ of running a sweeper car $s$ per km, the average speed (in km/h) of a sweeper car and the distance $d_{kl}$ (in km) between the cell $k$ where a jockey is picked up by the sweeper car and the cell $l$ where he/she is dropped off : $q_{kl} = \frac{d_{kl}\cdot(2\cdot c_j)}{s} + d_{kl}\cdot c_s$. Furthermore, we make the hypothesis that there are 7 jockeys that can do 10 relocations on average each night, so $\gamma = 70$.

### F. Case Study

We want to evaluate both the prediction part and the relocation part of our approach. For the former, we evaluate how well can $f_k(i, fs)$ be predicted for a cell $k$, a car rank $i$ and a feature set $fs$ with the regression approaches and baseline (Mean). For the later, we compute the daily expected revenue of the car sharing system when the utility is predicted and used to solve the relocation problem.

| Clustering | Features | "CarService" | | |
|---|---|---|---|---|
| | | Mean | GBR | SVR |
| No Clustering | $FS_1$ | 288 (45%) | 288 (45%) | 289 (45%) |
| | $FS_2$ | N/A | 288 (44%) | 283 (44%) |
| | $FS_3$ | | 290 (45%) | 284 (44%) |
| KMedoids | $FS_1$ | 289 (45%) | 289 (45%) | 288 (45%) |
| | $FS_2$ | N/A | 278 (43%) | 278 (43%) |
| | $FS_3$ | | 278 (43%) | **277 (43%)** |

TABLE I: Prediction MAE (lower is better) and RMAE of the car utility per day and per (group of) cell(s), using a historical baseline (Mean) and two regression models (GBR and SVR) with different sets of features ($FS_1$,$FS_2$ and $FS_3$).

*1) Regression Model Performance:* The columns of Table I represent the regression models used (historical *Mean*, *Gradient Boosting Tree* and the *Support Vector Machine*). The lines describe the results with or without clustering using the three feature sets. MAE (in min) and RMAE (in %) are related to the true number of minutes driven in each cell and day and the predicted number of minutes on the test set. The historical Mean performance has not been computed with $FS_2$ and $FS_3$ since its statistical value could be questionable.

One can notice that with $FS_1$, the historical mean gives one of the worst MAE ($\sim$288 mins) showing the potential of the machine learning models and our chosen features.

The addition of the temporal and weather features increases significantly in most cases the performance of the regression models (lines with $FS_2$ exhibit lower MAE and RMAE than lines with $FS_1$) and especially when using the K-Medoids algorithms to group the cells. Adding information about past values in $FS_3$ further benefit the SVR regression model (the GBR regression model keeps stable results when using the K-Medoids). In all cases except the most simple (Mean regressor and/or just $FS_1$ as features), using a clustering over the similar cells allows to reduce the error values, most of the time significantly. The clustering of similar cells allows the regression algorithms to train on more trips with the same features, better generalizing the training set. These results allow to choose for our regression approach the SVR regressor with a K-Medoids clustering of the similar cells while using the feature set $FS_3$: this combination offers a 2% reduction in RMAE compared to the baseline, as well as a reduction of 10 minutes (on average per cell and per day) in the MAE. This will be used for the following experiments.

| Strategy | "CarService" | |
|---|---|---|
| | Revenue (Base 100) | Nb Relocation |
| Historical | 100 | N/A |
| Optim MU | 109 | 53 |
| Optim SO | 107 | 49 |
| Optim DO | 107 | 53 |

TABLE II: Average daily revenue and number of relocations for four relocation strategy in the case of "CarService".

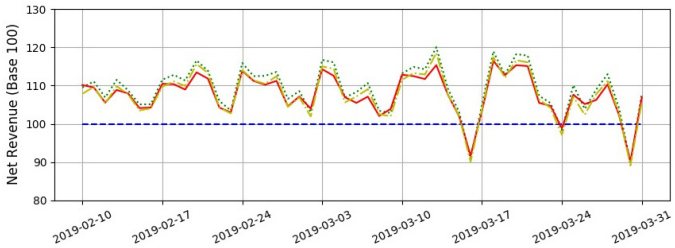


Fig. 2: Daily revenue expected by "CarService" (one x-axis tick is a Sunday), with the Historical strategy (horizontal dashed blue line at 100) as the base index. The red full line, the dashed yellow and green lines are respectively the normalized revenues provided with Optim SO, Optim DO and Optim MU strategies.

*2) Relocation Performance:* From now on we call *solver car assignment* the optimal solution to our optimization problem found by the proprietary solver Gurobi. The utility (number of driven minutes) is estimated using the best regression model found in the previous experiment. Thus, a K-medoids clustering of the city cells is followed by a SVR regressor, using the feature set $FS_3$.

For all approaches, the total daily revenue expected by both services are computed. These are plotted per day in Figure 2 and averages per day are reported for in Table II. We expect the revenue to be higher for *Optim MU* since it does not take into account any relocation cost but this strategy is also less realistic. We also expect *Optim SO* to give a higher revenue than *Optim DO* since the one step strategy can take into account the costs while relocating the car which should help reducing these costs.

Table II shows that for "CarService", all the optimization methods could help (on average) to increase the profit of the company. Further details in Figure 2 show that there are only two weeks where this is not true and the gain is almost 10% compared to the current service (historical) for the rest of the test set. Surprisingly, the different between the relocation strategies is small. This is probably because the service is relatively concentrated and the distance between each cell is low enough to have a very light weight on the relocation costs. This also explains why there is no difference between the two-step (Optim DO) and single step (Optim SO) optimization strategies since the cost does not influence the relocations.

## V. Conclusion

We tackled the problem of optimizing car relocation in a free floating car sharing service in order to increase its daily usage and profitability. We proposed to predict the expected car utilization and showed how such predictions could be used in a ILP solver to optimize the relocations while taking into account the sweeper costs. Although the predicting phase could be further improved by introducing more exogenous data, our approach could already increase the efficiency and profitability of the service to up to 7%.

Our relocation solution could benefit from further operational research formulation to optimize the route of the sweeper car and of all the jockeys (both are pickup and delivery vehicle routing problems [9]). This would be valuable but non trivial: jointly optimizing a vehicle routing and our assigment problem requires adding non-standard constraints and the problems are of a scale unattainable by generic ILP solvers.

**Acknowledgements.** This work has been done in collaboration between Stellantis and Inria through the *OpenLab AI* and was supported by the CIFRE program of ANRT.

## References

[1] S. Weikl and K. 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*, vol. 57, pp. 206–223, 2015.

[2] A. B. Brendel, J. Brennecke, P. Zapadka, and L. Kolbe, "A Decision Support System for Computation of Carsharing Pricing Areas and its Influence on Vehicle Distribution," *ICIS 2017 Proceedings*, Dec. 2017.

[3] T. Raviv, M. Tzur, and I. A. Forma, "Static repositioning in a bike-sharing system: models and solution approaches," *EURO Journal on Transportation and Logistics*, vol. 2, no. 3, pp. 187–229, 2013.

[4] S. Ghosh, P. Varakantham, Y. Adulyasak, and P. Jaillet, "Dynamic repositioning to reduce lost demand in bike sharing systems," *Journal of Artificial Intelligence Research*, vol. 58, pp. 387–430, 2017.

[5] S. Ghosh, J. Y. Koh, and P. Jaillet, "Improving Customer Satisfaction in Bike Sharing Systems through Dynamic Repositioning." in *IJCAI*, 2019, pp. 5864–5870.

[6] R. Zakaria, M. Dib, L. Moalic, and A. Caminada, "Car relocation for carsharing service: Comparison of CPLEX and greedy search," in *2014 IEEE Symp. on Comp. Intel. in Vehicles and Transportation Systems*, Dec. 2014, pp. 51–58.

[7] C. A. Folkestad, N. Hansen, K. Fagerholt, H. Andersson, and G. Pantuso, "Optimal charging and repositioning of electric vehicles in a free-floating carsharing system," *Computers & Operations Research*, vol. 113, p. 104771, Jan. 2020.

[8] R. Seign and K. Bogenberger, "Prescriptions for the successful diffusion of carsharing with electric vehicles," in *Conference on Future Automotive Technology*, 2013.

[9] M. W. P. Savelsbergh and M. Sol, "The General Pickup and Delivery Problem," *Transportation Science*, vol. 29, no. 1, pp. 17–29, Feb. 1995.