

Insights on Car Relocation Operations in One-Way Carsharing Systems

Rabih Zakaria

Université de Technologie et de Sciences Appliquées Libano-Française
Tripoli, Lebanon

Mohammad Dib

Engie
Paris, France

Laurent Moalic

Université de Haute-Alsace
Mulhouse, France

Alexandre Caminada

Université Bourgogne Franche-Comté Belfort, France

Abstract—One-way carsharing system is a mobility service that offers short-time car rental service for its users in an urban area. This kind of service is attractive since users can pick up a car from a station and return it to any other station unlike round-trip carsharing systems where users have to return the car to the same station of departure. Nevertheless, uneven users' demands for cars and for parking places throughout the day poses a challenge on the carsharing operator to rebalance the cars in stations to satisfy the maximum number of users' requests. We refer to a rebalancing operation by car relocation. These operations increase the cost of operating the carsharing system. As a result, optimizing these operations is crucial in order to reduce the cost of the operator. In this paper, the problem is modeled as an Integer Linear Programming model (ILP). Then we present three different car relocation policies that we implement in a greedy search algorithm. The comparison between the three policies shows that car relocation operations that do not consider future demands do not effectively decrease rejected demands. On the contrary, they can generate more rejected demands. Results prove that solutions provided by our greedy algorithm when using a good policy, are competitive with CPLEX solutions. Furthermore, adding stochastic modification on the input data proves that the results of the two presented approaches are highly affected by the input demand even after adding threshold values constraints.

Keywords—Carsharing; car relocation; ILP; greedy algorithm; CPLEX; green city

I. INTRODUCTION

It is straightforward that convenient transportation systems are crucial for supporting the economic development of cities [1], [2]. Generally, people in urban areas commute using different modes of transportation, such as public transport buses, trains, taxis, private cars, bikes, etc. Private cars are more attractive to users for their high flexibility and comfort. However, the increasing number of private cars has serious consequences related to environment issues, traffic and parking congestion [3]. Then, numerous efforts have been made to motivate people to use more sustainable modes of transportation like biking, walking or the use of public transportation facilities when possible. In June 2007, Vélib was successfully launched in Paris. 20,000 self-service bikes were deployed over 1500 stations [4]. Within the first year, the number of subscribed members exceeded 200,000 members and the bikes have been used 26 million times. The success of this system

has motivated cities all over the world to adopt this idea of sharing vehicles, which includes carsharing, bikesharing and other vehicle sharing concepts. Carsharing is one of the innovative solutions that can contribute in promoting sustainable car use. Many studies stated that private cars spend most of their time parked, since many car owners use their cars occasionally. Thus, in this case, one shared car can replace many private owned ones. Carsharing offers on demand access for cars distributed in a defined urban area. Therefore, carsharing systems offer the benefits of owning a private car without actually having to buy it. Carsharing is based on the model of Pay As You Go service, so users do not have to afford all the fees of owning a car like insurance and maintenance, they just pay during the time they access the service as an alternative of ownership in a market shift as predicted in “the age of access” [5]. Usually, users of this kind of systems rent cars for short periods of time. It is a complementary solution for the existing public transportation facilities. It offers the comfort and flexibility of private cars and the reduced costs of public transportation. According to Navigant Research, global carsharing service revenues will grow up to US\$6.2 billion by 2020 [6]. This kind of system has been implemented since the end of forties in Europe [7]. However, they were not successful since it was not easy to monitor the system and protect it from vandalism. Thanks to the advances in Information and Communications Technology (ICT), better facilitation, monitoring and management of reservations and payment operations of these systems have become available [8]. System operators and users are able to locate the stations and check the availability of vehicles in real time. In our study, we are dealing with one-way carsharing systems. Unlike round-trip carsharing systems, one-way carsharing systems allow users to take a car from a station and to drop it off in any other one. Although the one-way option makes the system more attractive to users, carsharing operators encounter difficulties in maintaining enough numbers of vehicles in stations to satisfy user demands. If stations are full, users who want to drop off their cars at the destination station cannot find a free parking place. On the other hand, user demands to take a car from empty stations will not be satisfied. If this imbalance problem occurs frequently, system clients lose their enthusiasm for using the service since it is not reliable and available when they need it. Recently, vehicle-sharing systems have generated a great interest of research in its different majors to solve

the problems that arise upon operating these systems. We will focus in this paper on the problem of car relocation in order to meet user demands. In one-way carsharing systems, the relocation problem is technically more difficult than the relocation problem in bikesharing systems. In the latter, we can use a truck to move several bikes at the same time, while we cannot do this in carsharing system because of the size of cars and the difficulty of loading and unloading cars. This paper presents an exact approach for the relocation problem in one-way carsharing system, followed by a heuristic approach using a greedy search algorithm. The focus is on providing different analysis and results to highlight the particular aspects related to this problem. In the literature, papers do not emphasize on the workload and cost of employees recruited to locate cars between the stations. The objective in this paper is to bring the attention to the complexity of these operations and to provide different analysis for this problem. This paper is structured as follows. The next section presents a practical example for the relocation problem. This is followed by the formulation of an Integer Linear Programming model for the problem. Then, the platform and mobility data used for this study are described. After that, a greedy algorithm and three relocation policies are explained. Different results and analysis are presented in Section VI. Finally, conclusion and future works are provided.

II. LITERATURE OVERVIEW

In the literature, we find many papers dealing with the relocation problem in one-way carsharing system. In 1999, Barth et al. developed a simulation model performance analysis of a multiple stations shared-use vehicles [9]. They found that the carsharing system is most sensitive to vehicle to trip ratio, to the relocation algorithm, and to the charging policy used in case of electrical vehicles. Other papers proposed that carsharing operators can involve users in the system to relocate cars [10], [11]; despite the fact that this technique was fruitful in alleviating the imbalance problem, it highly depends on clients participation, which is obviously not always guaranteed. Mitchel et al. proposed dynamic pricing for mobility-on-demand systems which include carsharing [12]. A price incentive strategy is used to motivate users to change their origin or destination stations to other stations near to them based on system needs. In a different study, a decision support system is presented by [13] to help carsharing operators to decide the values of operating parameters in a near-optimal way. Tuning these parameters reduces, between 37.1% and 41.1%, the number of relocation operations, it decreases the staff cost of 50%. It also reduces the zero vehicle-time between 4.6% and 13.0%. The author in [14] presented a multistage stochastic linear integer model with recourse for dynamic decision-making problem of vehicle allocation. They optimize trip selection in a way that the operator accepts or refuses trips reservations that maximize the profit of the carsharing system. Results showed profit increase but the model was not applied on real network and under real conditions. In [15], the author proposed stochastic mixed-integer programming model to minimize the cost of cars relocation operations in a way that satisfy p-proportion of all near-term demand scenarios. The study used historical data originated from the Intelligent Community Vehicle System (ICVS), which is no longer operational. Authors proved the robustness of the solutions through simulations that consider stochasticity in generating

redistribution plans.

III. RELOCATION PROBLEM IN A PRACTICAL EXAMPLE

This paper is dealing with one-way carsharing systems, which consists of many stations scattered in an urban area. A station has a predefined number of parking spots for its users. System users can take a car from a station and return it to any other station. When a user arrives at an empty station to drive a car, his request will be rejected. On the other side when a user wants to return a car to a station that is full, his request will be rejected as well. Users expect that cars are always available in stations when they need it, and they expect to find a free parking place at the destination station when they want to return the rented car as well. However, maintaining this level of service is not an easy task. This should be done by employees recruited to redistribute cars between the stations; in the following, we refer to these employees by “jockeys”. However, when the operator fails to solve this imbalance problem, users tends to abandon the system, which leads to potential system failure. We modeled our one-way carsharing system by a simple time-space network. To simplify the idea, an example of a simple carsharing system is provided below. Table I shows how many vehicles are available $av_{i,t}$ in each station i for each time step t . We have three stations S_1 , S_2 and S_3 . At time $t = 0$, we have the initial number of available vehicles in each station. Table II shows the number of cars $out_{i,t}$ that users would like to rent by station and by time step. Table III shows the number of cars $in_{i,t}$ that users would like to return to each station at each time step. In Table IV, we see the number of rejected user demands to take a car because a station is empty $out_{i,t}^r$ while Table V shows the number of rejected demands to take a car because a station is full $in_{i,t}^r$ for each time step. In this example, we consider that a station can host at most five cars. It is obvious that the values in these five tables must be non-negative. The input for the system consists of the initial number of available vehicles at $t = 0$ and the values in Tables II and III. While all other values are calculated based on the aforementioned input. To get the number of available cars, we use this equation:

$$av_{i,t} = av_{i,t-1} + (in_{i,t} - in_{i,t}^r) - (out_{i,t} - out_{i,t}^r) \quad (1)$$

In (1), available vehicles in station i at time t is equal to the number of available vehicles at the same station in the previous time step added to the number of arriving cars to the same station at time t minus the number of cars that could not be returned because the station is full. Then, we subtract the number of cars that go out of the station minus the number of rejected requests to take a car out of the station because there is a lack of cars.

TABLE I. NUMBER OF AVAILABLE CARS

t	0	1	2	3	4	5	6	...	T
S_1	2	2	3	3	3	4	5
S_2	4	2	2	1	0	0	0
S_3	3	3	4	4	5	3	3

As we can see in Table IV, we have one rejected demand in station S_2 at time $t = 6$. This rejected demand occurs because station S_2 does not have any vehicle at $t = 5$ and there is one request for vehicle at $t = 6$. On the other side, we see in Table

TABLE II. NUMBER OF REQUESTS FOR DEPARTING CARS

t	0	1	2	3	4	5	6	...	T
S_1	0	0	0	0	0	0	0
S_2	0	2	0	1	1	0	1
S_3	0	0	0	0	0	2	0

TABLE III. NUMBER OF REQUESTS FOR ARRIVING CARS

t	0	1	2	3	4	5	6	...	T
S_1	0	0	1	0	0	1	2
S_2	0	0	0	0	0	0	0
S_3	0	0	1	0	1	0	0

TABLE IV. NUMBER OF REJECTED DEMANDS BECAUSE OF AN EMPTY STATION

t	0	1	2	3	4	5	6	...	T
S_1	0	0	0	0	0	0	0
S_2	0	0	0	0	0	0	1
S_3	0	0	0	0	0	0	0

TABLE V. NUMBER OF REJECTED DEMANDS BECAUSE OF A FULL STATION

t	0	1	2	3	4	5	6	...	T
S_1	0	0	0	0	0	0	1
S_2	0	0	0	0	0	0	0
S_3	0	0	0	0	0	0	0

V that we have one rejected demand in station S_1 at time $t = 6$ since this station has four vehicles at $t = 5$ and there are two requests to return cars at $t = 6$.

IV. ILP FOR THE RELOCATION PROBLEM

We adapted the car relocation model presented in [13] to our study. Thus, a two dimensional time-space matrix $N \times T$ is used to model the relocation problem, N stands for the total number of stations $S = \{1, 2, \dots, N\}$ and T is the number of time steps during a day starting from 1 to T . Each element of the matrix represents a station S_i at time t . For each station $s \in S$ we generate T nodes to represent that station at each time t . Then we put all the $S \times T$ nodes in one row vector $V = (1_1, \dots, 1_T, \dots, N_1, \dots, N_T)$. An employee has three sorts of activities:

- 1) Relocating: is the action taken by the jockey to move a car from a station i to another station j .
- 2) Moving: is the action taken by the jockey to move himself from his current station to another station in order to begin a relocation activity.
- 3) Waiting: when the jockey is not involved in relocating or moving activities we say that the jockey is waiting.

Therefore, to represent these activities three sets of arcs are generated in the time-space network. An arc a_1 is constructed for each node $i_t \in V$, to represent a waiting activity between i_t and i_{t+1} ; this set is called $A_1 = \{\dots, a_1(i_t, i_{t+1}), \dots\}$. Also, $N-1$ arcs a_2 are constructed for each node i_t in V , to represent the move activity from station i and station j , $\forall i, j \in S, i \neq j$, from time t to time $t + t_{ij}$ where t_{ij} stands for the number of time steps required to move from station i to station j ; this set is named $A_2 = \{\dots, a_2(i_t, j_{t+t_{ij}}), \dots\}$. likewise, $N-1$ arcs a_3 are built to represent relocation activities, this set is denoted

$A_3 = \{\dots, a_3(i_t, j_{t+t_{ij}}), \dots\}$. The staff that is responsible for these activities is denoted by a set $E = \{1, \dots, e, \dots, W\}$, W represents the number of recruited employees. An ILP Model is formulated for the relocation problem. Six different decision variables are declared:

- u^e : When an employee e is used at least once during the day, its associated binary variable takes the value 1, while it takes the value 0 otherwise.
- $wait_{i_t i_{t+1}}^e$: When an employee e is involved in a waiting activity at station i from time t to $t + 1$, the associated binary variable is assigned the value 1, while it remains 0 otherwise.
- $move_{i_t j_{t+t_{ij}}}^e$: When an employee e is involved in a moving activity from the set A_2 , the associated binary variable is assigned with the value 1, while it remains 0 otherwise.
- $rel_{i_t j_{t+t_{ij}}}^e$: When an employee e is involved in one of the relocation activities within the set A_3 , the associated binary variable is assigned the value 1, while it remains 0 otherwise.
- $out_{i_t}^r$: This variable can be assigned with integer values. It represents the number of rejected user demands to rent a car from station i at time t .
- $in_{i_t}^r$: This variable can be assigned with integer values. It represents the number of rejected user demands to give back the rented car to a station i at time t .

On other side, the input parameters for the ILP are listed below:

- av_{i_0} : Represents the number of available cars in station i at time 0.
- out_{i_t} : Represents the number of requests to rent a car at time t from station i .
- in_{i_t} : Represents the number of requests to give back a car at time t to a station i .
- p_i : Represents the number of parking spots in a station i .
- c_{ij} : Denotes the estimated cost of a relocation or moving activity from a station i to station j .
- c_e : Denotes the estimated cost of an employee during a day.
- c_{in} : Stands for the estimated cost of the rejection of a demand to give back a vehicle to a station.
- c_{out} : Stands for the estimated cost of the rejection of a demand to rent a vehicle from a station.

Also, one dependent variable is used:

- av_{i_t} : Denotes the remaining available cars at station i at time t .

The relocation problem can be modeled by the ILP model below:

$$\begin{aligned} \text{Min } Z = & c_{ij} \left(\sum_{(i_t, j_{t+t_{ij}}) \in A_3} \sum_{e \in E} \text{move}_{i_t j_{t+t_{ij}}}^e \right) \\ & + \sum_{(i_t, j_{t+t_{ij}}) \in A_4} \sum_{e \in E} \text{rel}_{i_t j_{t+t_{ij}}}^e + c_{out} \sum_{i_t \in V} \text{out}_{i_t}^r \\ & + c_{in} \sum_{i_t \in V} \text{in}_{i_t}^r + c_e \sum_{e \in E} u^e \end{aligned} \quad (2)$$

Subject to:

$$\begin{aligned} & \sum_{i \in S} \text{wait}_{i_1 i_2}^e + \sum_{\substack{i, j \in S \\ i \neq j}} \text{move}_{i_1 j_1 + t_{ij}}^e + \sum_{\substack{i, j \in S \\ i \neq j}} \text{rel}_{i_1 j_1 + t_{ij}}^e \\ & = u^e \quad \forall e \in E \end{aligned} \quad (3)$$

$$\begin{aligned} & \text{wait}_{i_t - 1 i_t}^e + \sum_{(j_{t-t_{ij}}, i_t) \in A_3} \text{move}_{j_{t-t_{ij}} i_t}^e \\ & + \sum_{(j_{t-t_{ij}}, i_t) \in A_4} \text{rel}_{j_{t-t_{ij}} i_t}^e - \text{wait}_{i_t i_{t+1}}^e \\ & - \sum_{(i_t, j_{t+t_{ij}}) \in A_3} \text{move}_{i_t j_{t+t_{ij}}}^e - \sum_{(i_t, j_{t+t_{ij}}) \in A_4} \text{rel}_{i_t j_{t+t_{ij}}}^e \\ & = 0 \quad \forall i_t \in V, e \in E, t > 1 \end{aligned} \quad (4)$$

$$\begin{aligned} & \text{av}_{i_t} = \text{av}_{i_{t-1}} + (\text{in}_{i_t} - \text{in}_{i_t}^r) - (\text{out}_{i_t} - \text{out}_{i_t}^r) \\ & + \sum_{(j_{t-t_{ij}}, i_t) \in A_4} \sum_{e \in E} \text{rel}_{j_{t-t_{ij}} i_t}^e - \\ & \sum_{(i_t, j_{t+t_{ij}}) \in A_4} \sum_{e \in E} \text{rel}_{i_t j_{t+t_{ij}}}^e \quad \forall i_t \in V \end{aligned} \quad (5)$$

$$\text{av}_{i_t} \leq p_i \quad \forall i_t \in V \quad (6)$$

$$\text{in}_{i_t}^r \leq \text{in}_{i_t} \quad \forall i_t \in V \quad (7)$$

$$\text{out}_{i_t}^r \leq \text{out}_{i_t} \quad \forall i_t \in V \quad (8)$$

$$u^e = (0, 1) \quad \forall e \in E \quad (9)$$

$$\text{wait}_{i_t i_{t+1}}^e \in \{0, 1\} \quad \forall (i_t, i_{t+1}) \in A_1, e \in E \quad (10)$$

$$\text{move}_{i_t j_{t+t_{ij}}}^e \in \{0, 1\} \quad \forall (i_t, j_{t+t_{ij}}) \in A_2, e \in E \quad (11)$$

$$\text{rel}_{i_t j_{t+t_{ij}}}^e \in \{0, 1\} \quad \forall (i_t, j_{t+t_{ij}}) \in A_3, e \in E \quad (12)$$

$$\text{in}_{i_t}^r \geq 0 \quad \forall i_t \in V \quad (13)$$

$$\text{out}_{i_t}^r \geq 0 \quad \forall i_t \in V \quad (14)$$

$$\text{av}_{i_t} \geq 0 \quad \forall i_t \in V \quad (15)$$

Equation (2) represents the objective function. It minimizes the weighted aggregation of the number of rejected requests to rent or to give back a car, the number of needed staff and the required moving and relocating operations needed to decrease the number of rejected demands. Constraint (3) serves to ensure that an employee cannot perform more than one task at a time and to assign the value 1 to the variable u^e when the associated employee e is engaged in an activity at $t = 1$. Constraint (4) ensures that an employee cannot be engaged with a new activity before he completed the last one and to assure the continuity of activities for an employee if he is engaged at $t = 1$. Constraint (5) is used to get the remaining available cars at each station at each time step. It is calculated based on the number of remaining cars in the previous time step, the number of cars entering and leaving the station by the users and the number of cars moved in/out of the station by the staff. Constraint (6) ensures that the number of available cars at a station will not exceed its capacity. Constraints (7) and (8) are used to make sure that the number of rejected demands will not be greater than the number of demands. Constraints (9)-(12) are used to impose binary values to the associated variables, and constraints (13)-(15) ensure that the associated variables are non-negative.

V. GREEDY ALGORITHM TO SOLVE THE CAR RELOCATION PROBLEM

A. Motivation Behind the Greedy Algorithm

As described earlier in Section IV, the car relocation problem is modeled as an ILP model. The model is solved using CPLEX. The model was tested using different configurations, it was evident that the running time gets significantly bigger when the number of jockeys increased. It was also noted that stations number, the average number of trips per car and the maximum number of parking spots in each station highly affect the running time. These parameters highly influence number of rejected demands. After running the model with CPLEX with some complex configurations, CPLEX could not give a solution before two days of execution. While the solver could not give any solution for other complex configurations. Fig. 1 shows the variation of CPLEX's running time with respect to the number of jockeys used. This experiment uses a configuration of a simple carsharing system which has 10 parking spaces per station, 18 stations and 83 cars. The average trips per car for this experiment was 12. It was clear that the running time of CPLEX grows significantly when the number of jockeys is increased. This observation was a good motivation to look for another approach to get a result more quickly. For this reason, a simple greedy algorithm is developed to minimize the number of rejected demands while reducing the number of required relocation operations. A relocation operation is performed through two steps: first, a station is selected to take a car from it, then the destination station is chosen to move the car for it by the jockey to rebalance the system.

In this paper, different relocation policies are proposed which are implemented later using a greedy algorithm. This algorithm uses a policy pattern to assess the impact of each policy on resulting rejected demands. When the greedy algorithm is executed, one second of running time was enough to build a good non-optimal solution for configurations regardless of their complexity, and the used policy.

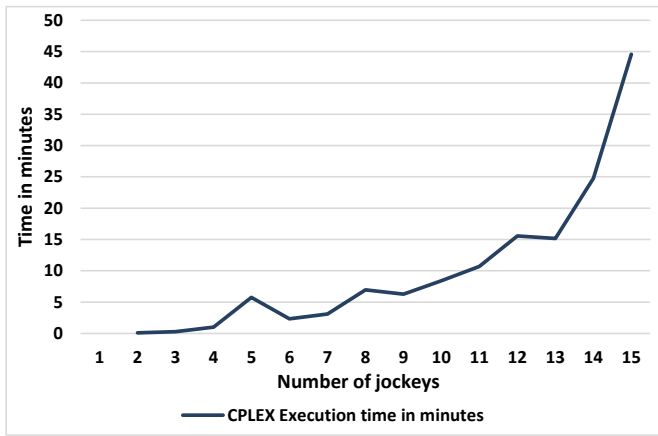


Fig. 1. CPLEX execution time when solving the relocation problem (10 parking places per station, 18 stations, 12 trips per car, 83 cars).

B. Relocation Policies

To increase client satisfaction, it is important to choose a good relocation policy in order to reduce the total number of rejected demands. Different approaches were tested using the greedy algorithm:

- Policy 1: The jockey moves one car from the nearest station to his current station, and if several, he chooses the one that has the highest number of available vehicles, to the nearest station, and if there are several ones, to the station that has the lowest number of available vehicles.
- Policy 2: The jockey moves one car from the station that has the biggest number of available vehicles, and if several, the nearest station, to the station that has the lowest number of available vehicles, and if several, at the closest station.
- Policy 3: The jockey moves one car from the station having the soonest rejected demand because it is full to the station having the soonest rejected demand because it is empty.

NB: In our examples (Fig. 2 and 3), we consider that the only car movements are done by jockeys for relocation purposes.

1) *Policy 1*: In this policy, the priority is given to the time needed to move between the stations. For each operation decision, the jockey chooses the operation that takes the shortest possible time with the objective of having enough time to do the maximum number of relocation operations that can be done during the day. During each relocation operation, the jockey starts by determining the closest station to his present location in order to take a car from this station. If more than one station found on same distance, he selects randomly a station having the biggest number of available vehicles. Then, the jockey chooses the closest station again and if he finds many, he selects the one that has the minimum number of available vehicles among them. For example, Fig. 2 shows a representation of this policy with four stations. The circles represent the stations. The name of the station and the number of available cars at a specified time are displayed in each circle.

In our example, the jockey starts the relocation operation by going to the nearest station from his location which is S_3 . This moving activity is done without a car from the system. Then, in a second step, he drives a car from the selected station S_3 to the station S_1 , because, first, it is the closest station and, second, because it presents the minimum number of available cars.

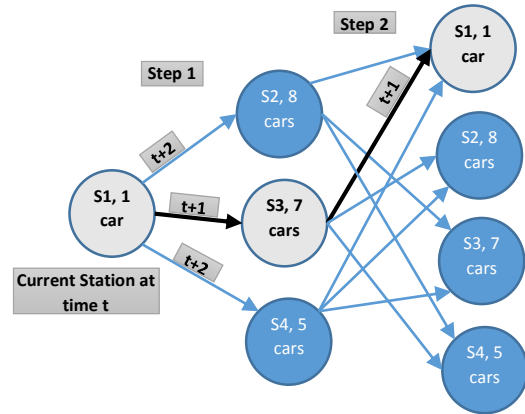


Fig. 2. Simple relocation operation using Policy 1.

2) *Policy 2*: Policy 2 prioritizes the balancing of the cars over the station, aiming to rebalance the number of available cars in each station. Policy 1 has some similarities with Policy 2, but here, the order of selecting stations is reversed. During each relocation operation, the jockey starts by looking for stations having the highest number of available vehicles, and he selects the closest station in the list. After that, stations having the minimum number of cars are selected, and the jockey chooses the closest. As we can see in Fig. 3, the jockey chooses station S_2 in the first step since it presents the highest number of cars, while the choice in the later step remains the same since station S_1 has the lowest number of cars.

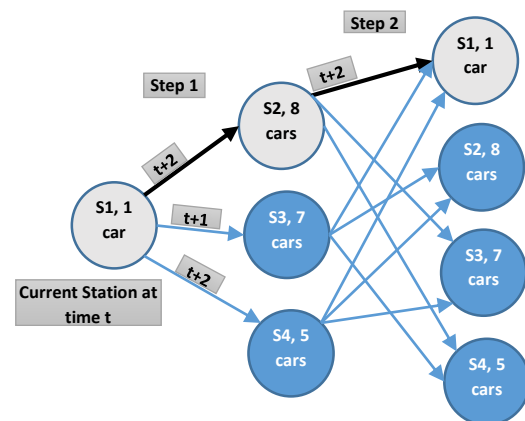


Fig. 3. Simple relocation operation using Policy 2.

3) *Policy 3*: Policy 3 considers that an estimation of what will happen in the future is known by the jockey, so he can foresee the rejected demands even if they occur after several time steps. In addition, in this policy the jockey can see the effect of each relocation operation on the overall system, so the jockey will not remove or add cars when this

may cause a station to be empty or full respectively. Here, the aim is to get rid of the maximum number of rejected demands during a single relocation operation. The relocation operation starts at step 1 by looking for the list of stations where the soonest rejected demands will occur as a result of stations filling up ($inR_{i_t}^r > 0$) and the list of stations that have cars that can be delivered to other stations that will need cars for future demands. In step 2, the jockey tries to find the list of stations where the soonest rejected demands will occur because empty stations ($outR_{i_t}^r > 0$) and the list of stations that may be in shortage for future demands. From these lists, the jockey chooses the target station for the relocation operation in order to reduce the maximum number of rejected demands while preventing to generate future rejected demands. When choosing the best relocation operation, if we have many possibilities with the same effect on rejected demands, we privilege the operation that reduces rejected demands in the closest stations and the soonest possible. In Fig. 4 we propose the flow chart to implement Policy 3 in a greedy algorithm. A greedy algorithm makes the optimal choice at each iteration up to the local optimum.

VI. EXPERIMENTATION AND RESULTS

A. Mobility Data

The mobility data used for this study consists of socio-economical information and survey data that are collected by professional for the objective of regional planning. This data describes people mobility flows in a region of 20 km x 10 km in Paris. The region of the study is plotted into a grid of cells having the same size. A cell has two characteristics:

- The type of the terrain: it describes structure types that are dominant in the area associated to the cell (commercial center, business center, buildings, roads, houses, etc.).
- Attraction weight: based on the terrain type and survey data, this information attributes a dynamic attraction weight to each cell for each 15 minutes of the day.

A 3D matrix $F = (f_{i,j,t})$ represents the people mobility between different cells, where $f_{i,j,t}$ stands for the number of persons who want to move from cell i to cell j at time t . We consider t to be a period of 15 minutes during the day, which makes 96 time periods. Then the flow mobility data is plotted on a map using GIS shapefiles. As a result, 400 cells have been detected as a potential origin or destination point knowing that some cells are eliminated because of their geographical nature e.g. lakes, plains, etc. The final flow mobility data consists of 400 x 400 x 96 elements, which makes 15,360,000 records to represent how people move during the day.

B. Platform for Locating Stations

In order to locate station for a carsharing system in the region on the study, [16] developed a dedicated platform. The platform uses the mobility data that are described earlier in this study. To locate the stations, a multiobjective memetic algorithm has been implemented in the platform. The algorithm optimizes three objective functions:

- Objective 1: The location of the stations should maximize the mobility flow between the cells.

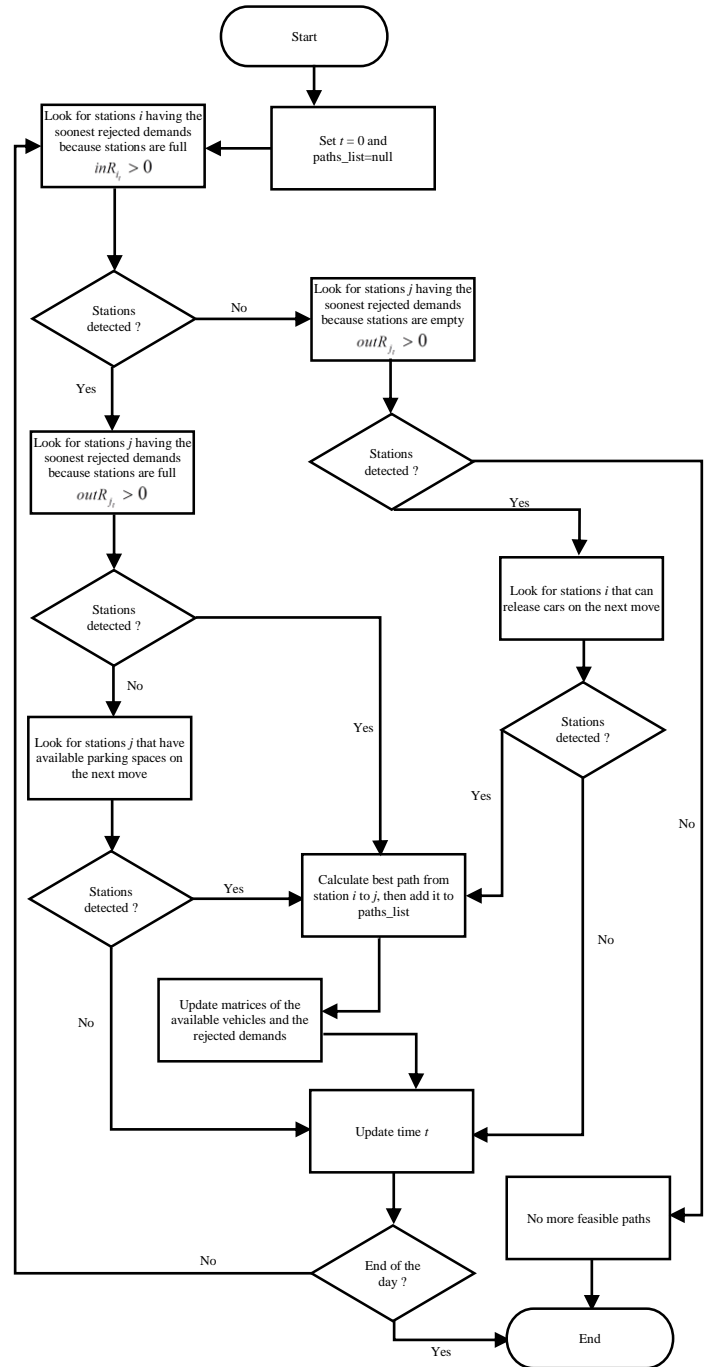


Fig. 4. Flowchart of the Relocation Algorithm using Policy 3.

- Objective 2: The location of the stations should maximize the balance between the ingoing and outgoing flows in each station.
- Objective 3: The location of the stations should minimize the standard deviation of the flows in order to obtain a uniform flow during the day.

Each cell is considered to cover the demand in a radius of 300 meters. Special filters and probability distribution are applied on the mobility data to forecast the potential users for

the service. A study has been carried out with the carsharing operator to set the desired system parameters. We used this platform to generate the data for this study. The generated dataset consists of the four matrices described in Section III. For each generated dataset, we use four parameters:

- 1) Total number of cars in the system.
- 2) Total number of stations in the system.
- 3) Average trips number by car.
- 4) Parking spots number for each station.

C. Relocation Policies Comparison

Fig. 5 shows a comparison of three policies described earlier, these results concern a generated dataset for a car-sharing system that consists of 20 stations having 10 parking places each and 150 cars that have an 9 average trips. As we can see, the performance of policy 1 and policy 2 is rather similar in the beginning. After that, when the number of jockeys is increased to be more than 19, policy 1 and policy 2 start to generate new rejected demands rather than reducing them. Policy 1 is worse than policy 2 in minimizing the rejected demands number. This result is logical, since policy 2 gives the priority to the relocation operations which aims to redistribute the cars in order to rebalance the system. Bad relocation operations may lead to an augmentation in the number of remaining rejected demands in the future. On the other hand, policy 3 performs much better than the other two policies. This can be explained by the fact that the jockey has an estimated knowledge of the future rejected demands. With this knowledge, the jockey is able to take better relocation decisions in order to reduce the maximum number of rejected demands, keeping in mind not to generate future rejected demands. This policy is better from the other two policies since relocation decisions are taken only when needed. When relocation operations are not advantageous to the system, the jockey does not relocate cars but he waits until the appropriate moment for better relocation operations. Fig. 6 clearly shows that the number of relocation operations is somehow constant using policy 1 and policy 2. While in policy 3, the number of relocation operations is decreasing, likewise the number of remaining rejected demands which also decreases.

D. Comparison of CPLEX and Greedy Algorithm

As we can see in the subsection VI-C, the comparison of the performance of the three proposed policies shows that Policy 3 is the best approach for the relocation problem. In the remaining part of this paper, our greedy algorithm implements the policy 3, exclusively. In order to assess the performance of our greedy algorithm we solved the same problem with the same data with CPLEX. Fig. 7 shows that the results of the greedy algorithm are competent with the results obtained by CPLEX; especially the greedy algorithm takes less than one second to deliver a solution while CPLEX may take a long time before delivering a solution as shown in Section V.

E. Stochastic Data Results

After solving the relocation problem using our greedy algorithm and CPLEX, each jockey is affected to a path that should be followed in order to cut down rejected demands. The path is constituted of a series of relocation operations

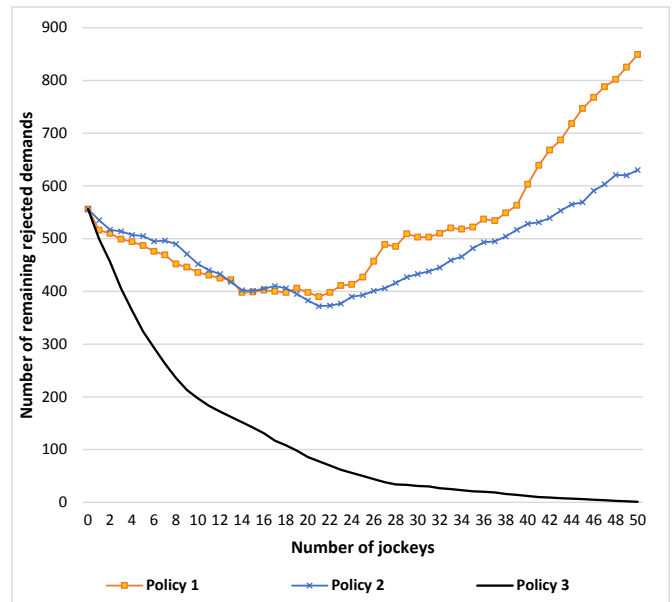


Fig. 5. Comparison of the performance of the three relocation policies.

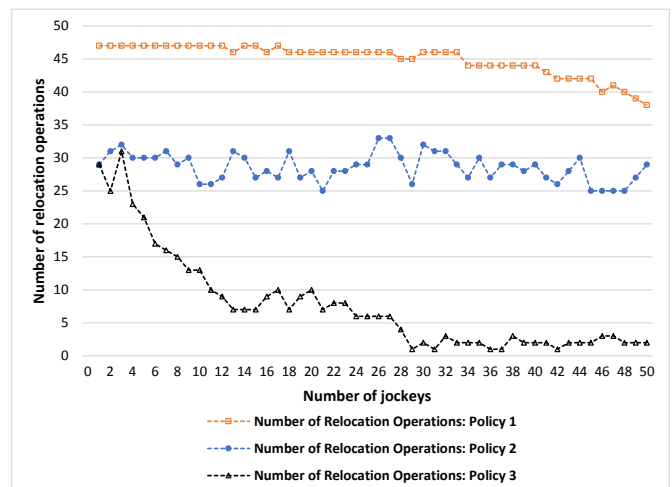


Fig. 6. Number of relocation operations with the three relocation policies.

to be done during the day. A relocation operation tells the jockey from which station and when, a car should be moved, and to which station and when, it should be dropped off. When the number of jockeys is increased in Policy 3, the number of remaining rejected demands decreases as well as the number of needed relocation operations as shown in Fig. 5 and 6. In order to measure the robustness of the resulted relocation operations, we used a special Gaussian method to add stochastic noise to the input data for the incoming and outgoing cars; knowing that the added stochastic noise does not exceed 10% of the original data. In Fig. 8 we see an example of stochastic input data modification on the number of incoming cars at a station in our carsharing system. After that, using the original data, the resulted relocation operation plan is applied, then these operations were evaluated on the input data that was modified in a stochastic manner regardless of the number of available cars; we call this step blind relocation. As shown in Fig. 9, when the number of jockeys is increased the number

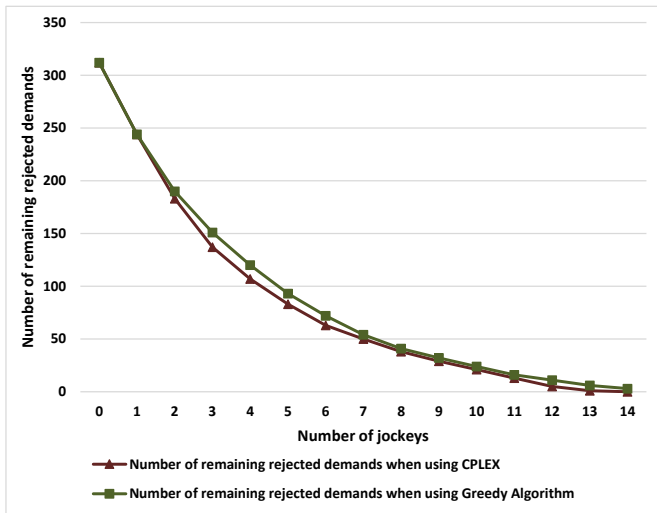


Fig. 7. Comparison results between the greedy algorithm and CPLEX (18 stations, 10 parking spots by station, 88 cars with the average of 12 trips by car).

of remaining rejected demands is decreased with stochastic input data. However, as much as the number of jockeys is increased, the difference between remaining rejected demands increases when using the original data as well as when using stochastic modified data (see both curves with triangles). This is due to the fact that each stochastic modification on the input data, in any station at any time step, will be aggregated and propagated to all the following time steps that concern this station. Thus, since the number of available vehicles in each station at each time step is used to make the decision of the relocation operation, the resulted relocation operations can lose its efficiency drastically when the input data of user demands is changing.

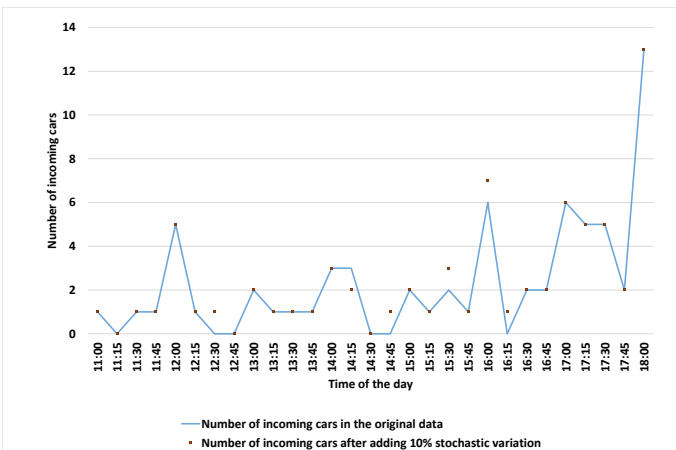


Fig. 8. Stochastic data variation on the number of incoming cars from 11:00 to 18:00 in one station.

F. Integrating Threshold Values in our Greedy Algorithm

In another step, the greedy algorithm is changed by integrating lower and upper threshold values in order to measure the effect of threshold values on the relocation operations when using stochastic modified data. The lower threshold is used

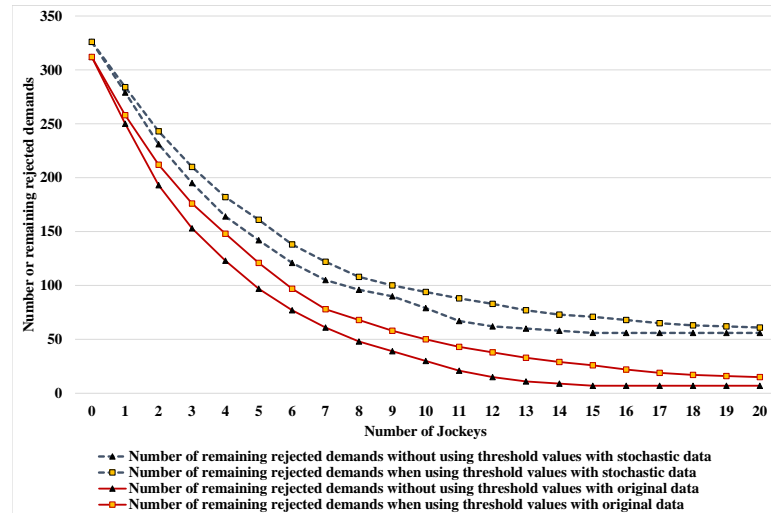


Fig. 9. Effects of threshold values and stochastic data on the total number of remaining rejected demands.

to avoid relocating a car from a station when the number of available vehicles in the station before the relocation operation is less than or equal to the lower critical threshold. In this example, this value is set to one. That is if the station has only one car before the relocation operation, then the algorithm does not apply the relocation in that station. The upper threshold value is used to avoid relocating a car to a station when the number of available cars in the destination station is greater than or equal to the upper critical threshold. In this example, it is set to the maximum number of places in the station minus one. As shown in the chart below, the performance of the threshold strategy in terms of reducing the number of rejected demands (curves with squares) is worse than without threshold constraints. In addition, shown in Fig. 9 the threshold values does not bring improvement for the blind relocation on stochastic data compared to blind relocation without using threshold values. In both cases, the difference in the number of reduced rejected demands starts to be small when the number of jockeys is small, but gets bigger as the number of jockeys increases. On the other side, it is clear that the number of relocation operations when using threshold values, is less than the number of relocation operations without using threshold values since threshold adds a constraint on the decision of a relocation operation until all rejected demands problems are solved as shown in Fig. 10.

G. Identification of Mobility Patterns During the Day

As described earlier, when the number of jockeys is increased, the number of rejected demands decreases as well. However, the cost of relocation operations increases as well. In this study, we consider that we are using each jockey for the whole day, which is impractical. However, in real life there will be staff shifts that depend directly on the demands and needs of relocation operations. In the literature, relocation operations that are carried out at night are called static relocation since user demands for cars is considered negligible during this period. Static relocation is necessary to provide the stations with the appropriate number of cars for the next morning. In static relocation, there are no time window constraints to

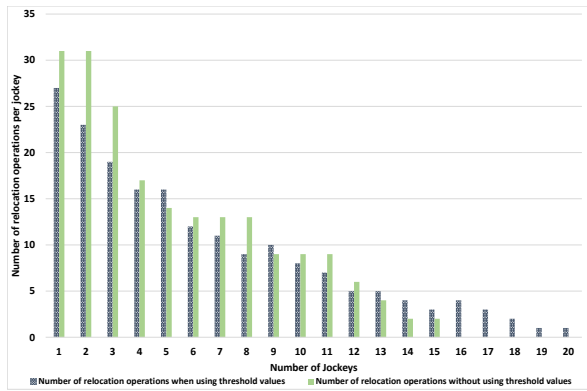


Fig. 10. Number of relocation operations when using threshold values and without using them.

deliver cars to stations at specific times, unlike cars relocation during the day where some stations have urgent needs for cars to satisfy user demands on time. In our approach, relocation operations are carried out during the day. When analyzing the time at which the rejected demands are solved using our greedy algorithm with policy 3, we get the histogram in Fig. 11. This histogram compares the total number of reduced rejected demands per hour of the day when using 15 jockeys for the whole day and when using them from 7:00 to 19:00. When analyzing the histogram in Fig. 11 we can detect some relocation patterns during the day. There are some periods of high activity such as the period from 8:00 to 10:00 and from 17:00 to 19:00. There are also periods of low activity such as period from 11:00 to 16:00. These patterns can be explained by the fact that these intervals correspond to periods of high mobility of customers in the morning when they go to work and in the evening when they come back home. On the other side, we notice that when we limit the working time until 19:00, the number of reduced rejected demands in the late hours (17:00 to 19:00) increases. This can be explained by the fact that the jockey can anticipate rejected demands and reduce them even before their occurrences. Knowing that the number of reduced rejected demands at any time t of the day, does not only represent the number of reduced rejected demands that occur at time t , but it also includes the anticipated rejected demands that occur in the future but reduced by relocation operations performed at time t .

Thereby, the required effort for the jockeying operations, will not be the same during the day; likewise, the number of jockeys should vary as well. Thus choosing the appropriate number of jockeys per time interval is a key factor to reduce the cost of jockeying operations.

In another experience, we divided the working time of jockeys into three periods with an interruption of work between them:

- 1) From 7:00 to 9:00
- 2) From 11:00 to 13:00
- 3) From 17:00 to 19:00

Then we compared the performance of the jockeys in this case with their performance when they work from 7:00 to 19:00. As we can see in the chart below in Fig. 12, even when we divide the working time into three periods, the number

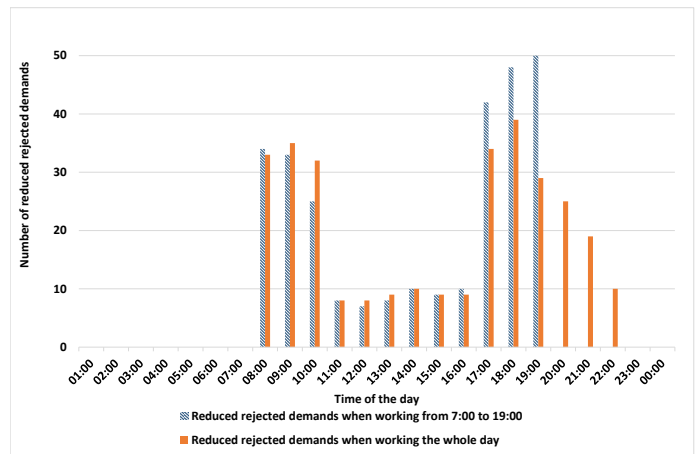


Fig. 11. Number of solved rejected demands in each hour of the day using 15 jockeys.

of reduced rejected demands decreases. However, the slope is smaller since the number of working hours is smaller. We conclude that the company must evaluate the cost of rejected demands in regard to the cost of the jockeying hour.

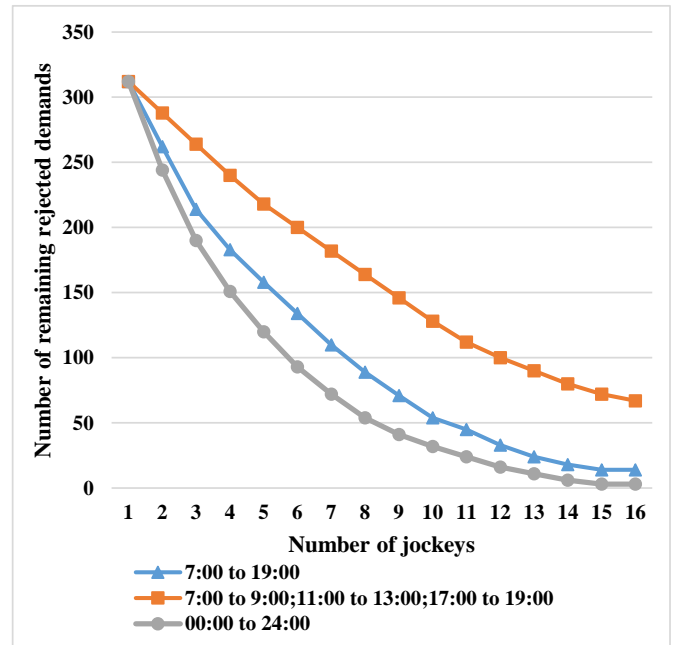


Fig. 12. Number of remaining rejected demands when varying working hours.

VII. CONCLUSION AND PERSPECTIVE WORKS

The one-way car sharing service is appealing to users since they are not required to return the car of departure station and for its flexibility. Nevertheless, this flexibility leads to an imbalance in cars distribution. The imbalance problem affects the image of the service and makes it less attractive to users. To cope with problem, relocation operations are vital to increase the satisfaction of the clients. In this study, three different policies of car relocation are compared. The performance of Policy 3, where the jockey has information on the future state of the system based on historical data and predictions, is much

better than the two other policies that do not consider any future information. It was proven that the implementation of intuitive policies that are based on basic decisions such as the total distance covered by jockeys and the available cars at stations without considering the propagation of the effect of these relocation operations on the future, which may influence the overall service, will not have a great impact in minimizing the number of rejected demands. On the contrary, applying these policies may require more relocation operations, which will eventually rise the total cost of the system. Taking into consideration the historical data to make future estimation is crucial in order to minimize the number of rejected demands. From another side, we see that jockeys pass by inactivity periods when there is no need for relocation operations. Analyzing these periods, suggests that working hours of each jockey can be reduced and so, we can decrease the car sharing operation cost. In addition, we found that the effectiveness of the resulted relocation operations is highly dependent on the input data even when we use threshold values for the relocation operations. As perspective, it is possible to implement a new heuristic approach based on stochastic model using historical data, in the aim of solving the relocation problem. This can be modeled in a simulation environment that considers real life parameters.

REFERENCES

- [1] A. Reno and G. Weisbrod, "Economic impact of public transportation investment," 2009.
- [2] J.-P. Rodrigue, C. Comtois, and B. Slack, *The geography of transport systems*. Routledge, 2013.
- [3] R. Katzev, "Car sharing: A new approach to urban transportation problems," *Analyses of Social Issues and Public Policy*, vol. 3, no. 1, pp. 65–86, 2003.
- [4] F. Meunier, "Véhicules partagés: des défis pour la ro."
- [5] J. Rifkin, *The Age of Access. How the Shift from Ownership to Access is Transforming Capitalisms*. Putnam, New York et al, 2000.
- [6] [Online]. Available: <http://www.navigantresearch.com/research/carsharing-programs>
- [7] S. A. Shaheen, D. Sperling, and C. Wagner, "A short history of carsharing in the 90's," *Institute of Transportation Studies*, 1999.
- [8] D. Jorge and G. Correia, "Carsharing systems demand estimation and defined operations: a literature review," *EJTIR*, vol. 13, no. 3, pp. 201–220, 2013.
- [9] M. Barth and M. Todd, "Simulation model performance analysis of a multiple station shared vehicle system," *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 4, pp. 237–259, 1999.
- [10] M. Barth, M. Todd, and L. Xue, "User-based vehicle relocation techniques for multiple-station shared-use vehicle systems," *Transportation Research Record*, vol. 1887, pp. 137–144, 2004.
- [11] K. Uesugi, N. Mukai, and T. Watanabe, "Optimization of vehicle assignment for car sharing system," in *Knowledge-Based Intelligent Information and Engineering Systems*. Springer, 2007, pp. 1105–1111.
- [12] D. Jorge, G. H. Correia, and C. Barnhart, "Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems," 2014.
- [13] A. G. Kek, R. L. Cheu, Q. Meng, and C. H. Fung, "A decision support system for vehicle relocation operations in carsharing systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 1, pp. 149–158, 2009.
- [14] W. D. Fan, R. B. Machemehl, and N. E. Lowmes, "Carsharing: Dynamic decision-making problem for vehicle allocation," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2063, no. 1, pp. 97–104, 2008.
- [15] R. Nair and E. Miller-Hooks, "Fleet management for vehicle sharing operations," *Transportation Science*, vol. 45, no. 4, pp. 524–540, 2011.
- [16] L. Moalic, S. Lamrous, and A. Caminada, "A multiobjective memetic algorithm for solving the carsharing problem," in *WORLD COMP'13-The 2013 World Congress in Computer Science, Computer Engineering, and Applied Computing*, vol. 1, 2013, pp. 877–883.