Long-term vehicle reservations in one-way free-floating carsharing systems
A variable quality of service model

Molnar, Goran; Correia, Gonçalo Homem de Almeida

DOI
10.1016/j.trc.2018.11.017

Publication date
2019

Document Version
Accepted author manuscript

Published in
Transportation Research Part C: Emerging Technologies

Citation (APA)
Molnar, G., & Correia, G. H. D. A. (2019). Long-term vehicle reservations in one-way free-floating carsharing systems: A variable quality of service model. *Transportation Research Part C: Emerging Technologies*, *98*, 298-322. https://doi.org/10.1016/j.trc.2018.11.017

Long-term vehicle reservations in one-way free-floating carsharing systems: a variable quality of service model

Goran Molnar (corresponding author)
Faculty of Electrical Engineering and Computing
University of Zagreb,
Unska 3, 10000 Zagreb,
Croatia
E-mail: goran.molnar2@fer.hr
Phone: (+385) 1 6129 967

Gonçalo Homem de Almeida Correia
Department of Transport & Planning
Delft University of Technology
P.O. Box 5048 - 2600 GA Delft
The Netherlands
Email: G.Correia@tudelft.nl
Phone: (+31)15 27 81384

Department of Civil Engineering,
University of Coimbra,
Rua Luís Reis Santos - Pólo II
3030-788 Coimbra,
Portugal

ABSTRACT

Reservations in daily services can improve user satisfaction, and give additional information about the demand patterns to the operators. However, providing reservations to carsharing clients is difficult. While carsharing is especially convenient if it is allowing one-way trips and vehicle drop-off anywhere in the service area (called free-floating), this flexibility increases management complexity because of vehicle stock imbalance. Most of the commercial providers of free-floating carsharing offer reservations under highly restrictive terms, for example only up to 30 minutes in advance. In this paper, we propose an innovative reservation enforcement technique that allows substantially longer reservation times while keeping the system profitable and achieving high service quality. A simple way to enforce reservations is locking vehicles until the departure time of a client. However, it comes at the cost of idling vehicles that could be used by other users and decreasing the revenue. Our approach, called relocations-based reservation enforcement method (R-BR) combines vehicle locking and relocation movements. It locks vehicles only a short time before the trip departure if a suitable vehicle is close enough due to the natural trip patterns. If no such vehicle is available, a car is relocated from another place. Further, we propose a variable quality of service (QoS) model in which the guaranteed radius around the user within which the reserved vehicle will be placed, and the maximum allowed reservation time before the departure depends on the zone of trip departure. A simulation-based optimization is used whereby the carsharing operation is simulated and optimized using an iterated local search (ILS) metaheuristic for adjustment of service level parameters. The proposed technique is tested on a set of artificial problem examples and a case study of a simulated working day in the Lisbon Municipality, Portugal. Results show that the proposed R-BR method is substantially better than the simple vehicle locking when the constant QoS approach is used and that the devised ILS metaheuristic can further increase the system performance, especially with high trip volumes.

Keywords: *free-floating carsharing, reservations, relocations, iterated local search, quality of service, sustainable transport*

1. Introduction

Carsharing is a type of mobility service that provides short-term car rental to its users (Shaheen et al., 1999, Correia and Antunes, 2012). Such services involve a fleet of vehicles distributed across the city that can be accessed and used by their members. Unlike traditional rent-a-car services, the typical rental durations are very short and charged by the minute or the hour. They are typically privately owned and marketed as a membership-based service.

Carsharing systems provide the flexibility and accessibility of a private car, without, however, the costs and responsibilities of owning one. To the user, they are an alternative to both private vehicle ownership and public transport (Namazu and Dowlatabadi, 2018.). To policymakers, they are interesting due to their potential to reduce pollutant emissions as well as the need for parking spaces and costly expansions of the public transport service coverage (Litman, 2000; Schuster et al., 2005). Carsharing systems can be divided into round-trip and one-way trip systems, the latter allowing clients more flexibility as they do not require the vehicle to be returned to the original location. Furthermore, they can be divided into station-based and free-floating systems. In station-based carsharing, users can return the vehicles only to a set of specific locations (stations), while free-floating carsharing allows users to park the vehicle in any legal parking space in the service area. In both cases, allowing more flexibility to the user also creates added management complexity due to vehicle stock imbalance, well documented in the literature (e.g., Correia and Antunes, 2012, Huang et al., 2018).

While carsharing has the potential for lowering the environmental footprint of the city commute (Vasconcelos et al., 2017), an important obstacle to the broader adoption is the fact that the service is still more difficult to access than for example a taxi. Aside from being dispersed at attractive locations around the city to allow walk-ins, the taxi service typically offers the dial-a-ride, e-hail, and booking services which add additional value and increase the suitability of the service for different purposes.

A possible way to increase availability and user satisfaction in one-way carsharing systems could be providing vehicle *reservations*. Reservations are available in a wide range of services and industries: reserving a table at a restaurant, seats in a theatre or booking hotel rooms are nowadays ubiquitous everyday actions. Reservations are available in other transportation services as well: virtually all of the air traffic is reserved ahead, and most taxi providers allow their users to reserve a ride (Copeland and McKenney, 1988, Wang and Cheu, 2013, An and Lo, 2014, Hu and Liu, 2016; Lu et al., 2018). Reservations can give the providers additional useful information, such as daily, weekly and seasonal demand patterns, and the way users respond to various campaigns. Knowing the demand ahead helps these services to plan their operations and organize the resources to improve efficiency. Therefore, the operators commonly encourage users

to perform reservations as soon as possible. Pricing incentives are a frequently used way to achieve early user response – booking a hotel room or a flight just one day ahead is almost always much more expensive than doing it some months in advance.

Providing vehicle reservations in carsharing can be a highly challenging issue though, and has hardly been addressed in the literature. The topic of using resource reservation as a management strategy in carsharing has been mainly explored for parking at the destination when there is a shortage of parking spaces (Kaspi et al. 2014, Kaspi et al. 2016). Unlike the airline and hospitality industries, where the reserved resources are under complete control by the provider, this is not the case in carsharing. The shared fleet movements are dynamic and difficult to predict, due to varying demand. For a carsharing service provider, knowing reservations a few days ahead, i.e., where a vehicle is going to be picked-up, does not help much in running the enterprise as relying on daily user trips is not enough to provide the guarantee that a vehicle will be available at the reserved location and time. Instead, some other mechanism needs to be used to enforce the reservation service and ensure that the user will have the reserved vehicle at the place and time he/she desires.

A simple and effective strategy that can be used to enforce reservations is *vehicle locking*. In this approach, the user selects a vehicle close to the desired location and the departure time. After this, the vehicle is considered *locked* and inaccessible for use by any other member (similar to a waiter in a restaurant putting a "reserved" label on a table). A prominent drawback of such approach is that it lowers the vehicle utilization rates and the revenue produced by the locked vehicle. This is such a notable issue that many one-way carsharing providers do not have reservation services at all, or if they do, they offer it under highly restrictive conditions. For example, the global operators Car2Go and ZipCar allow reservations for one-way trips, but only up to 30 minutes before the trip start (car2Go, 2017; Zipcar OneWay, 2017). Some other services allow longer reservations, however, charge for them by the minute (DriveNow, 2018; Enjoy, 2017). The utility of such service is therefore highly limited as reserving a vehicle for a trip to the airport a week ahead or a trip to work tomorrow morning is not possible or at best, is expensive. These restrictions substantially decrease the quality of service being provided by a mode that is supposed to serve a higher share of demand in the future.

Relocation operations are vehicle movements initiated by the service provider and performed by a team of employees. So far, relocations have been used mainly to solve the vehicle stock imbalance problem, both in the station based and free-floating carsharing systems. Relocation trips do not generate revenue and represent a cost for the company due to the fuel and staff expenses. However, research has shown that such investment can lead to higher overall profits by providing the ability to fulfill more demand. It is possible to find in the literature several

optimization and simulation methods dedicated to this problem (Jorge et al., 2014; Weikl and Bogenberger, 2013; Weikl et al., 2016; Boyaci et al., 2015; Deng and Cardin, 2018; Huang et al., 2018).

To the best of our knowledge, no research has been done to demonstrate the drawbacks of the vehicle locking method for providing carsharing reservations, nor in providing a more efficient alternative that can cope with the disadvantages. We propose an innovative reservation enforcement method named Relocations-Based Reservations (R-BR) that complements vehicle locking with relocations operations in a free-floating one-way carsharing system. We hypothesize this approach will allow longer reservation times while keeping the vehicle utilization rates and revenues reasonably high.

Methodologically, this work is based on a simulation-optimization approach. We built a custom microsimulation environment that allows insight into user-operator interactions under different conditions related to reservations. A carsharing company might not support reservations at all or might use various strategies to ensure that reserved vehicles will be at the requested location at the required time. Companies might also sometimes reject reservations and users will not use the service unless an available car is close enough to reach it by walking. The developed model has similar properties to others that have been proposed in the literature to study the management of carsharing systems such as (Di Febraro, et al., 2012; Jorge et al., 2014; Nourinejad and Roorda, 2014; Kek et al., 2009).

We assess the reservation quality of service (QoS) using two parameters: (1) time in advance allowed for making a reservation, denoted *h*, and (2) radius around the trip origin, denoted *r*, where the reserved vehicle is guaranteed to be available at the time of the client departure. We assume that users would like to be able to reserve a car anytime they want, therefore longer *h* means better user satisfaction. Conversely, we assume that users would like to walk the shortest possible distance to the reserved vehicle (Correia et al., 2014). More formally, to improve user satisfaction, it is desirable to maximize *h* and minimize *r*.

Applying an equal setup everywhere in the service area might not be optimal. Tactically increasing the service quality in certain zones of the city and decreasing it in others has the potential to improve the profitability of the service and accepted demand, without impacting the service quality too much. Based on this idea, we define the Variable Reservation Service Quality Problem (VRSQP): given a set of zones in a city, with the possibility to choose a separate service quality level in each zone, we want to find the best set of (*radius*, *time ahead*) parameters in order to maximize the objective function (denoted Z). The objective function is defined as a weighted sum of individual goals: profit (maximized), satisfied demand (maximized), allowed time between the moment of reservation and trip start (maximized) and the radius around the user

(minimized). These goals can be contradictory in some cases, which makes the VRSQP a multi-objective optimization problem. By choosing the appropriate weight for each of the four individual goals, it is possible to model the operator preferences: some businesses might be entirely profit-oriented and set to ignore all other goals, others might prefer a more balanced approach where profit is not improved if it causes large drops in service quality.

Choosing a setup of the geographically varying pairs of *r* and *h* is a complex problem. We propose to use an Iterated Local Search (ILS) metaheuristic (Lourenço *et al.*, 2001; Lourenço *et al.*, 2003) in a simulation-based optimization approach for finding good and realistic solutions to the VRSQP. In this setup, the simulator acts as an evaluator for the variable service quality layouts proposed by the ILS algorithm. Based on the evaluation from the simulator, the algorithm creates increasingly better solutions and discards those that produced bad results in the simulation.

The methodology is applied on several problem instances: two extreme hypothetical cities (small town and large major city) and a case study of Lisbon Municipality, Portugal, with four different demand levels. Two key experiments are performed:

1) We compare the vehicle locking and the R-BR method under a constant QoS in the entire service area,
2) The R-BR method is further optimized under a variable QoS with the ILS.

The remainder of this paper is organized as follows: Section 2 brings a detailed description of using relocations to enhance the reservations system. Section 3 presents the variable reservations quality of service concept by which reservations are offered in a different way across the city to maximize an objective function. The paper continues in Section 4 with the description of the ILS heuristic proposed to solve the problem in a simulation-based optimization approach. Section 5 presents the numerical experiments and application to the Lisbon case study which is followed by the results in Section 6. Finally, Section 7 gives the main conclusions of the paper and perspectives for future work.

2. The Relocations-Based Reservations (R-BR) method

Let us imagine that a user calls at 17:00 and wants to reserve a vehicle to be available the same day at 21:00 at a specific location of the city. In the vehicle locking approach, the operator searches for the closest vehicle to the desired location. If the closest vehicle is within the acceptable radius (r) from the location, the reservation is accepted, otherwise, it is rejected. If the reservation is accepted, the current closest vehicle is marked as locked and in that way, reserved for the user. In our example (Figure 1), the closest vehicle that was found will be locked at 17:00 and will remain in its location until the desired departure time (21:00) when the user picks it up.

Notice that this reservation process would be the same had the user searched a specific vehicle himself by using a smartphone or a laptop with internet access.

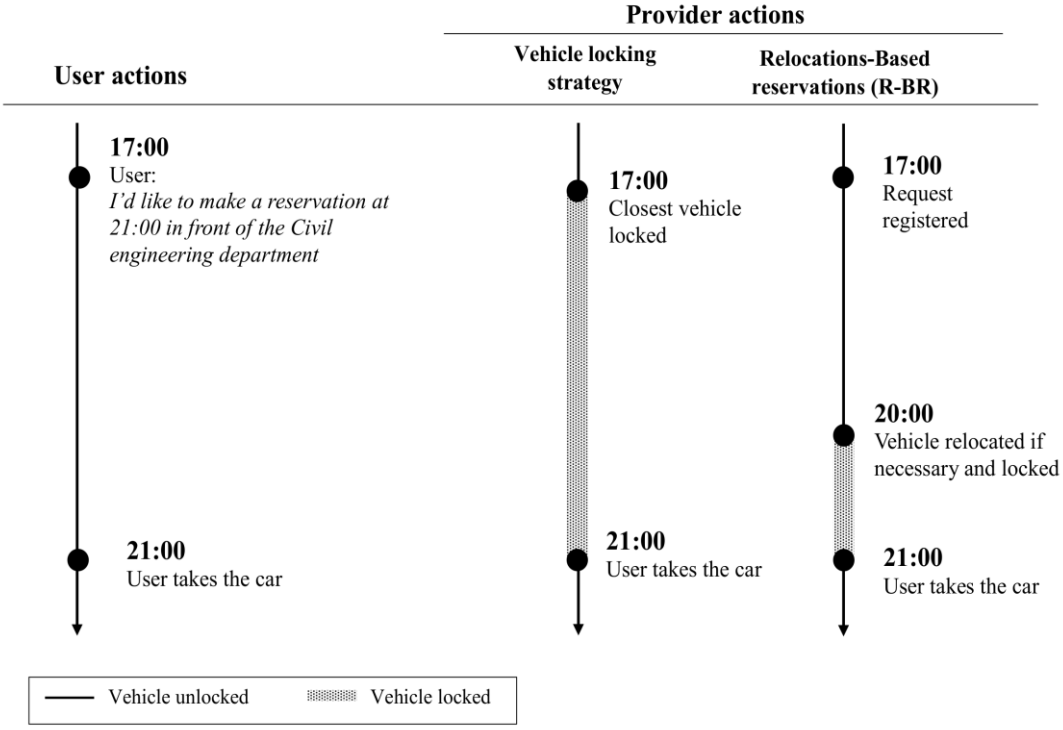


Figure 1: Vehicle locking and relocation enforcement strategies

Using the Relocations-Based Reservations (R-BR) method that we propose, when a client makes a reservation, no action is taken immediately. At that moment, the reservation is checked for feasibility, as there exists the QoS limit of accepting reservations no more than h minutes ahead. After the reservation is accepted, all vehicles in the network continue to be available as if no reservation has taken place until the *response time* moment, denoted as t_a. Response time moment is the time before the desired departure at which the system starts processing the reservation and activates the relocations enforcement mechanism. At that point a decision needs to be made: lock some nearby vehicle or use a relocation movement. This decision is made based on the location of the closest available vehicle to the client trip origin. If the nearest vehicle is within the acceptable QoS radius (denoted as r), that vehicle is locked until the user takes it at the desired departure time.

When the vehicle is relocated, it will be locked until the user takes it. If there are no vehicles available for relocation a taxi must be provided to the client since he/she was expecting a vehicle. In our example shown in Figure 1, the vehicle locking system caused the vehicle to stay idle for 4 hours whilst using the R-BR method we propose, the car would be idle much shorter

(only up to 1 hour in the example). The flowchart of this approach is presented in Figure 2. Note that the values of *r* and *h* can be set globally, equal for the entire service area or they can vary depending on the origin zone as in the VRSQP problem.

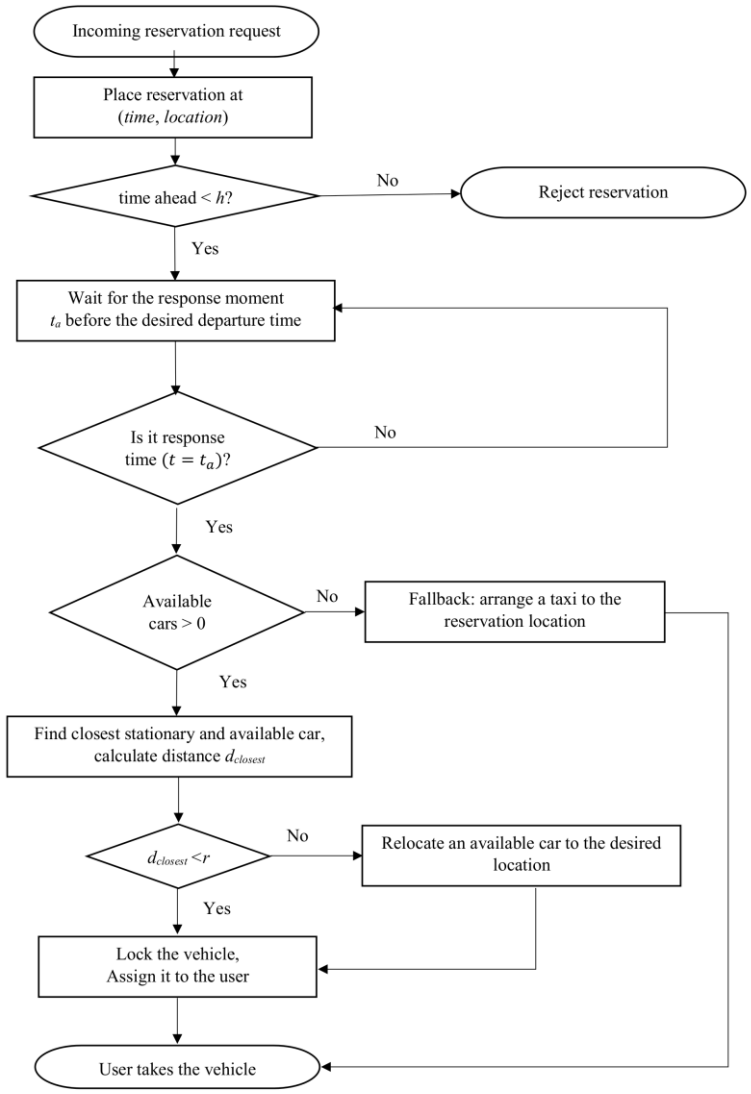


Figure 2: Flowchart of the proposed relocations based reservation enforcement strategy

An important aspect that needs to be decided is how long before the reservation does the system need to respond. If the response time is too long, we expose the system to the risk of having low vehicle utilization rates, similar to the ones obtained with vehicle locking, if it is too short, we risk having unreliable service where delays can happen. In this paper, we propose that this parameter should be set in such a way that the system still has enough time for a relocation,

even under the most pessimistic traffic conditions for the particular case-study city. While a more realistic value could be used in the function of actual traffic conditions, we decided to use the most conservative estimates due to the fact that such forecasting could be unreliable. Last-minute cancellations of accepted reservations would undermine the user trust, and therefore we selected higher reliability instead of slightly better profit.

A key issue when applying R-BR method is choosing a right balance of QoS parameters *r* and *h* and other performance indicators such as profit and satisfied demand. Any change of these will affect users who in general want to be able to reserve as early as possible and want their cars to be as close to them as possible. Achieving a high quality of service can require more effort from the provider, and it has the potential to cause drops in profitability. In this paper, we propose two algorithms that can help set these parameters: (1) a simple QoS-sweep algorithm to choose the best global service quality (equal in the entire service area regardless of the origin location) and (2) an ILS metaheuristic to choose these parameters when they can vary, depending on the origin zone.

2.1.Vehicle stock balancing

The simulation environment we designed supports two different types of relocations:

1. Reservation support relocation movements,
2. Balancing relocation movements.

The key application of relocations in this work is the first one: relocations are used to bring a vehicle to the location of a reservation if there are no nearby cars. The second type is a traditional application in carsharing, used to improve the vehicle stock balance and increase the probability that a vehicle will be close to the average user, including the users doing walk-ins (Jorge et al. 2014). Even though the focus of this work is on the first type, in the simulator, both can be used independently or complementary. In both cases, the relocation decisions are based on the current state of the simulated fleet (reserved, available and occupied vehicles), and the demand forecast data in the rectangular grid across the city, during several time periods.

While the first type of movements has a reservation support purpose, they can nevertheless be used to improve balance. Consider the situation where a type 1 movement is needed because there are no vehicles close to the user for a reserved a ride. A relocation movement will be performed to move a vehicle to the departure location. Depending on the choice of the car to relocate, we could increase or decrease system balance. Always choosing the closest car is the myopic cheapest move, however, it does not take the vehicle stock balance into account and has the potential to worsen it. Conversely, performing relocation trips that are best from the balancing

point of view could lead to a large number of long and expensive relocation trips. For the reservation support movements, we use the middle-ground approach, where cars are relocated from the closest zone with a vehicle stock surplus. In the case where there are no surplus zones in the system, the closest available car is relocated.

For the second type, we implemented a simple strategy where a number of balancing trips is periodically dispatched. Balancing effort intensity is parametrized by two parameters: (1) balancing trips per period, denoted b_n and (2) balancing period duration, denoted b_p. All the balancing trips are started at the same time at the beginning of each balancing period. Increasing the number of balancing trips per period and shortening the period (increasing the dispatch frequency) will improve the balance, however, the costs of operating these relocations will increase.

The balancing algorithm uses the forecasted demand during several days and the city area divided into a rectangular grid to determine which zones have a surplus of vehicles and which have a deficit. Depending on the severity of the deficit/surplus, the zones are prioritized into suppliers and demanders, and relocation movements from suppliers to demanders are produced. Cost of relocation movements is calculated based on the cost per minute driven for relocation trips, denoted C_r.

We note that faithfully modeling different balancing relocation strategies is a separate, complex issue that is still under research on its own (Weikl and Bogenberger, 2013; Jorge et al., 2014; Weikl et al., 2016, Boyaci et al., 2015). Considering that a highly realistic simulation of relocations is not the goal of this paper, and to ensure faster execution of the model, we intentionally omit details related to advanced optimization approaches for relocation movements, as well as the details of running the appropriately sized workforce. We assume that at any given moment, a staff member can immediately be available in any part of the city to start the relocation if needed. This is a simplification of the real systems which often work with their own staff team who require some time to reach the vehicles to be relocated and have a varying number of available staff throughout the day. However, we assume that the relocations are performed by out-sourced people who do each relocation operation one service at a time and are paid by minute of relocation rides. When other staff payment models are used, some approximations are needed. For example, an operator might observe that approximately 35% of their relocation costs are spent traveling to the relocation movement origin and take that into account when calculating the value of the C_r parameter.

3. Variable Reservation Quality of Service (QoS)

Varying the service parameters such as prices according to the local conditions is widely used in transport services as referred before (Yang et al., 2010; Jorge et al., 2015; Angelopoulos et al., 2018; Xu et al., 2018). To further tailor the one-way carsharing operation to the demand, aiming at increasing the profit of the company whilst allowing for reservations and keeping the service quality high, we propose a variable QoS model across a city. Given varying trip patterns across the zones of a city, tailoring the reservation parameters has the potential to improve the system efficiency. In the variable service model, the service area is divided into N zones, with individual QoS parameter values in each trip origin zone:

$$QoS_i = (r_i, h_i),$$

where r_i is the maximum allowed distance of the reserved vehicle (radius around the user) in the i-th zone and h_i is the maximum allowed reservation time ahead of the trip start in the same zone.

The Variable Reservation Service Quality Problem (VRSQP) is defined in general terms as the problem of finding the optimal set of *QoS* parameters for the zones in the city, for which an objective function that describes the operator preferences is maximized. While the profit of a carsharing company is a good foundation for comparing the solution quality from the perspective of the operator, it is clearly not enough to guarantee that service of good quality is being provided to the travelers. Large r might lead to savings in relocation trips and higher profits, nevertheless, the users prefer it to be as small as possible. For many users, walking half a kilometer to reach the vehicle makes little sense, especially if their trip is going to be short. Likewise, users prefer to be able to place reservations with as few restrictions as possible, therefore the longer the allowed reservation time, the better the service quality offered to the clients. Finally, the satisfied demand is the fourth important factor which determines the solution quality, higher demand acceptance levels mean that a better user coverage was achieved.

Therefore we define the problem of finding the optimal reservation parameters as a combination of all or some of the following individual objectives 1) maximizing the profit, 2) maximizing the reservation times, 3) minimizing the radius around the user where the reserved vehicle will be available, and 4) maximizing the satisfied demand. All of those can be combined into a multiobjective function as follows:

$$Max(Z) = w_P \frac{P - P_{min}}{P_{max} - P_{min}} + w_h \frac{\bar{h}}{h_{max}} + w_r \left(1 - \frac{\bar{r}}{r_{max}}\right) + w_d \frac{d_{sat}}{d_{tot}} \quad (1)$$

where P is the profit for a given solution, P_{min} and P_{max} are the lower and upper profit bound estimates, $\bar{h}$ is the average reservation time across all zones of the city, h_{max} is the maximum

reservation time, $\bar{r}$ is the average radius across all zones of the city, r_{max} is the maximum radius, d_{sat} is the number of satisfied trips, $d_{tot}$is the total demand (maximum potential number of carsharing trips) and w_P, w_h, $w_r$and w_d are the weight factors determining the relative priority of each function component during optimization. Since the radius is to be minimized, in the objective function it is converted to a maximization objective by subtracting $\frac{\bar{r}}{r_{max}}$ from 1.

The operator has complete freedom to choose the relative importance of each performance indicator and even to entirely exclude them from consideration. For example, a profit-oriented business might optimize only profit ($w_P = 1, w_h = w_r = w_d = 0$), not caring at all about satisfied demand or cars being close to the users. Some other operator might be in a middle of a marketing campaign during which they want to increase satisfied demand and brand exposure, even at the cost of slightly lower profit. A third provider might choose a balanced approach where service quality drops are acceptable, but only if they are justified by a high profit increase.

Note that the components of the objective function are normalized to an interval of [0, 1]. Therefore, if the weight factors are chosen in a way that their sum is equal to one and the profit limit estimates are correct, the entire objective function will be normalized to that interval as well, thus being possible to be represented by a percentage. The bound values h_{max}, r_{max} and d_{tot} are known in advance as they are the input to the optimization. However, the upper and lower bounds of the profit, P_{max} and P_{min} are not known in advance and need to be estimated.

In the variable QoS solutions generated by the algorithm, there might be considerable variance in radiuses and times across the service area. While a constant service quality is certainly easier for users to understand, we argue that implementing and communicating the variable QoS to users is not an insurmountable challenge. In commercial applications, a large part of the trips is requested using smartphone applications or online, and similar information can easily be communicated via phone as well. E-hail services such as Uber or Lyft, use dynamic pricing, where prices change during the day, depending on the demand and the number of cars on the road. Despite using nontransparent and variable pricing model, such services are well accepted by users.

In our simulation model, we use a fixed trip database and assume that the input demand is constant. Giving the optimizer too much freedom when choosing the service quality would certainly break this, e.g., frequently placing a car 3 km away from the user who reserved it would give a very bad impression and discourage users from using the service, this way also lowering the demand and invalidating the profit calculation based on the constant demand assumption. To achieve *ceteris paribus* conditions in the optimization process, especially related to the input

demand volume, the simulation framework provides two ways to control the algorithm's freedom to modify the solution: (1) hard QoS limits $h_{max}, r_{max}, h_{min}, r_{min}$ and (2) soft objective function weights.

By imposing a hard limit using the algorithm parameters, it is guaranteed that service quality will never reach nonsensical values that would impose significant changes in the demand: the operator might request the cars always to be placed 500m or less from the user. The soft configuration of the objective function further adjusts the degree of the algorithm's freedom. Unlike the hard limits, using these soft weight parameters defines only the tendency to give higher importance to some performance indicators over the others, without guarantees on the final values. Combining both the hard limits and precisely defining the tendency to prioritize certain parameters, it is possible to ensure that the optimization objectives of the operator are met and that the service quality variations are sufficiently small to prevent having a notable impact on the demand.

4. Solution Algorithm for the Variable Reservation Service Quality Problem (VRSQP)

In the past decades, various metaheuristic techniques have shown to be successful on difficult problems characterized by a large search space and complex constraints. They are generic algorithmic frameworks that can be applied to a diverse range of optimization problems, and while they do not guarantee to find the optimum, they are providing results of great practical utility. For many problems, they are yielding state-of-the-art results (Glover et al., 2006; Luke, 2013).

In our simulation-based optimization approach, we combine simulation with the ILS metaheuristic (Lourenço *et al.*, 2001; Lourenço *et al.*, 2003) to solve the Variable Reservation Service Quality Problem (VRSQP). The metaheuristic is used to devise a set of QoS parameters that are provided to the simulator as an input, and the simulator is used to evaluate profit and satisfied demand. Based on this feedback, the algorithm iteratively decides which changes in the QoS across the city to perform to enhance the objective function further. This way, the simulator is acting as an evaluator for the solutions suggested by an algorithm to solve the VRSQP.

4.1. Reservation Simulator

For this research, we devised a custom discrete-event microscopic simulator to investigate various reservations-related decisions and reproduce the user/operator interactions under different conditions. The main design goals for the simulator were:

- Ability to model user decisions while performing walk-ins or reserving a vehicle while taking into account the spatial effects of moving vehicles in a minute-to-minute simulation environment;
- Ability to model the provider behavior when deciding whether to accept or reject incoming trips;
- Ability to choose reservation demand as a percentage of walk-ins vs. reservation requests;
- Ability to estimate performance indicators such as the revenue, operating costs, profit, and percentage of satisfied and rejected demand.

Our simulation-based methodology assumes the existence of a carsharing trip database with origin/destination coordinates, start times and trip durations for each trip. Trip estimation can be performed by surveying users or specialized mode choice simulations. Further, the demand forecast can be based on historical data on fleet utilization and individual trips. Given that most contemporary carsharing providers use sophisticated fleet tracking systems that record precise movements of vehicles during the time, we believe that acquiring trip datasets should not be an issue for potential customers. Since there is ample research on this separate subject (Ortúzar and Willumsen, 2011; Sinha and Labi, 2007), detailed elaboration of the techniques for demand modeling is out of the scope of this paper.

The simulation is based on a list of walk-in and reservation trips synthesized from the initial trip database by a component called *mode divider*. The number of reservations is parametrized by the reservations percentage parameter $\rho = \frac{number\ of\ reservation\ trips}{total\ number\ of\ trips}$, defined as a ratio of the number of trips that are reserved ahead and the total number of trips. The mode divider uses the Monte Carlo method to divide the input demand into walk-ins and reservations and set up the reservation times.

Summarized, the inputs to the simulator are 1) walk-in trips 2) reservation trips, 3) initial vehicle locations, 4) QoS parameters (arrays of r and h across the service area), 5) maximum comfortable walk-in distance c_{wd}, 6) forecasted ideal vehicle stocks in the service area during periods of time B_{ideal}, and 7) balancing relocations dispatch period b_p, 8) balancing relocations trip number per period limit b_n. The pseudocode of the simulator is available in Algorithm 1. The key component of the simulator is the *vehicle location record* data structure, denoted as VLR. It is a dictionary which contains the vehicle status and locations for each minute in the simulated period. Supported values of the status variables for a vehicle are: 1) “stationary and available”, 2) “stationary and locked”, 3) “moving by user”, 4) “moving by a staff member”. Vehicles are available to start new trips and to be reserved only when in status 1) “stationary and available”. In all other cases, they are already assigned to a user or in use by a staff member. The simulation

starts by loading the initial vehicle locations and initializing the vehicle time record. Results of a simulation run are estimates of the carsharing provider profit and satisfied demand as well as the complete record of all vehicle movements (OD locations and trip start and end time for each performed trip).

Algorithm 1: Reservation simulator pseudocode

*Procedure Reservation Simulator (**walk-ins**, **reservations**, **initial vehicle locations, QoS,** c_{wd}, B_{ideal}, b_p, b_n)*

```
VLR = initialize vehicle time record(initial vehicle locations)
for each t in the simulation period
        initialize set WID_t containing all walk-in demand starting at t
        initialize set RD_t, containing all reservations to respond to at t
        for each walk-in trip w_i in WID:
                get the closest stationary and available vehicle c_sa
                if (closest vehicle distance > c_wd)
                        reject walk-in
                else accept walk-in:
                        calculate walking duration t_w
                        lock vehicle c_sa from t until t_start = t+ t_w
                        Update VLR: set status of the vehicle c_sa to "moving by user"
                                from t_start until t_end = t_start + duration of the trip
                        Update VRL: set status of the vehicle c_sa as "stationary and available"
                                at the destination location of w_i from t_end onwards
        next i
        for each reservation trip res_i in RD:
                processReservation(res_i, B_ideal)
        next i
        if(t mod b_p = 0)
                dispatchBalancingTrips(B_ideal, b_n)
        end-if

next t
calculate profit
```

The simulation model filters the trips from the trip database for each minute t sequentially. Walk-ins are accepted or rejected by the user, as defined by the comfortable walk-in distance (c_{wd}) parameter. If a walk-in is accepted, it is assumed that the user will reach the

vehicle by walking from his current location (trip origin) to the closest vehicle and that he will start walking immediately after sending the request. Walking duration t_w is estimated under the assumption that the walking speed is 5 km/h and that the walking distance is the Euclidean distance multiplied by a random number in the interval [1, 2] to take the impact of the street layout into consideration. Note that c_{wd} is a parameter used to define user behavior with regard to walk-ins and that it is not related to reservation service quality parameter *r* which applies only to the reservations.

Reservation enforcement (Type 1 relocations) are handled by the *processReservation*(res_i , B_{ideal}) function in Algorithm 1. There are two possible implementations this function can be redirected to: locking and relocations. The locking version is simple and straightforward: if close enough, lock the closest vehicle from the moment the reservation is made, until the departure, otherwise reject the reservation. The relocations strategy is implemented as detailed previously in Figure 2, and uses the ideal vehicle stock B_{ideal} to choose vehicles to relocate. The carsharing operator resorts to a taxi service as a backup to ensure the reservations are satisfied even in cases where the fleet is overloaded and there are no free vehicles. The user will be charged the standard service price and the carsharing company will pay the taxi. This way, the service is paying the difference between normal carsharing fees and taxi rides. These trips are considered to be satisfied demand as the service ensured the trip can be performed under the same pricing conditions. Such trips are undesirable as a taxi is typically more expensive and these outsourced trips generate losses.

The balancing trips (type 2 relocations), if balancing is used (i.e., if b_n>0) are dispatched in regular time intervals which is denoted as *dispatchBalancingTrips*(B_{ideal}, b_n) function in Algorithm 1. Balancing trip assignment is performed based on comparisons of the vehicle stock in the currently running instance and their ideal distribution. Vehicles are relocated from zones with the highest surplus to the zones with the highest deficit (Jorge et al, 2014), where a set of all zones in a time interval *t* is denoted $\mathbf{W_t}$. In case more relocations than b_n are needed to fully balance the system, the simulator will have to choose the distribution according to probabilistic priorities. For each cell with a surplus (supplier) and for each cell with a deficit (demander) origin and destination probabilities are calculated according to the following equations:

$$Prob\_O_{i_t} = \frac{St_{i_t} - B_{ideal_{i_t}}}{\sum_{j \in N, St_{j_t} - B_{ideal_{j_t}} > 0} St_{j_t} - B_{ideal_{j_t}}}, \forall i_t \in \mathbf{W_t}, St_{i_t} - B_{ideal_{i_t}} > 0, \quad (2)$$

$$Prob\_D_{i_t} = \frac{B_{ideal_{i_t}} - St_{i_t}}{\sum_{j \in \mathbf{N}, B_{ideal_{j_t}} - St_{j_t} < 0} B_{ideal_{j_t}} - St_{j_t}}, \forall \mathrm{i_t} \in \mathbf{W_t}, B_{ideal_{i_t}} - St_{i_t} < 0, \quad (3)$$

where $Prob\_O_{i_t}$ is the probability that cell i will be an origin for the balancing trip at time t, $Prob\_D_{i_t}$ is the probability that cell i will be a destination for the balancing trip at time t, St_{i_t} is the vehicle stock in zone i during time t. Based on these probabilities, the origin and destination is assigned in each trip, using a random proportional rule (Dorigo and Birattari, 2011; Hancock, 1994).

While real systems would most likely include undesirable effects, such as no-shows and late cancellations, in this work, we do not take them into account. Each area is likely to have slightly different no-show patterns, depending on the local culture and user habits and such data is difficult to obtain. Nevertheless, we believe that each provider will devise some type of a penalty strategy (e.g., charging a no-show fee or forbidding the user to re-book the same vehicle after not taking it on time) to discourage such behavior and compensate for the financial losses, similar to practices in taxi reservations (He et al., 2018). For this reason, we argue that the effects of no-shows can be neglected for the purpose of this work.

Several decisions in the simulator are based on random behavior, for example, user-car walking time estimation and dividing the demand into walk-ins and reservations. The simulator has two modes: non-deterministic and deterministic. In the deterministic mode, random value generators are always initialized with the same seed thus producing the same list of random numbers.

The profit (denoted *P*) calculation is performed by going through the VLR and calculating the revenue (R) and costs (C) of vehicle operations:

$$P = R - C.$$

The revenue component is calculated as a sum of individual revenues of user trips, based on the service price per minute, denoted as π, and the given trip duration estimates, denoted $duration(trip_i)$:

$$R = \sum_{i \in \boldsymbol{L}} duration(trip_i) \cdot \pi\,,$$

where $\boldsymbol{L}$ is the set of all user trips. The costs are calculated as a sum of fixed costs (C_f) and variable costs (C_v):

$$C = C_f + C_v.$$

Fixed costs do not depend on the number of performed trips and are calculated as a sum of daily parking costs and daily vehicle depreciation costs for all the vehicles in the fleet:

$$C_f = A \cdot C_{park} + A \cdot C_{veh},$$

where A is the fleet size (number of cars), C_{park} is the cost of parking per vehicle per day in the city and C_{veh} denotes the costs of depreciation per vehicle per day as defined before. Variable costs are calculated as the sum of costs of vehicle maintenance, relocation and taxi:

$$C_v = \sum_{i \in \boldsymbol{L}} duration(trip_i) \cdot C_{mv} + \sum_{i \in L'} duration(trip_i) \cdot C_r + \sum_{i \in G} \left(\mathrm{C}_{taxi_{start}} + duration(trip_i) \cdot \mathrm{C}_{taxi_{km}} \right),$$

where $\boldsymbol{L}$ is the set of all user trips, $\boldsymbol{L}'$ is the set of all relocation trips, C_{mv} is the cost of vehicle maintenance per minute driven, C_r is the cost of a relocation operation per minute driven, $\boldsymbol{G}$ is the set of all trips redirected to taxi, $distance(trip_i)$ is the estimated distance of trip i, $\mathrm{C}_{taxi_{start}}$ is the taxi start price and $\mathrm{C}_{taxi_{km}}$ is the price of driving 1 km in a taxi.

4.2. Iterated Local Search

We apply the *iterated local search* (ILS) metaheuristic (Stützle and Hoos, 1999; Lourenço *et al.*, 2001; Lourenço *et al.*, 2003; Luke, 2013) to solve the VRSQP. Iterated local search is a simple, but an effective method based on repeated applications of the *local search* and *perturbation* operators to perform exploration of the neighborhoods of known good solutions while avoiding being stuck in local optima. There are numerous implementations of this metaheuristic and it has been successfully applied for solving classical optimization problems such as the travelling salesman problem (TSP) and graph coloring problems (GCPs) (Stützle and Hoos, 1999; Katayama and Narihisa, 1999, Chiarandini and Stutzle, 2002), as well as more specific ones, e.g. scheduling and vehicle routing (Lourenço et al, 2003; Carlier, 1982; Hashimoto et al, 2008).

The key component of the algorithm is the local search operator (LSO). Local search is based on a definition of a *neighborhood structure* and *proximity measures* for our solutions. The operator is based on searching for the best solution in a close neighborhood of a given solution. (Johnson et al., 1988). It enumerates all solutions from the neighborhood and results in a local optimum which is the element of the neighborhood with the best evaluation of the objective function. The quality of local optima greatly depends on the definition of the neighborhood structure and the initial solution.

While local optima are as good as or better than the initial solution, in the context of all possible solutions to the problem, there is no guarantee of their quality. Unless the choice of the

initial solution is incredibly lucky or the problem is very simple to solve, local search results tend to be globally suboptimal. To expand the search from the initial solution neighborhood, the perturbation operator (PO) is used in the ILS. The PO introduces the modifications to the current solution, and in the ILS algorithm, it is used to give a "kick" to the results of the local search. The intensity of the modifications of the PO should be high enough to ensure that the LSO does not converge back to the same solution in the next iteration, but also low enough to prevent the algorithm from degrading to random restart local search.

The set of feasible solutions Q is defined by a tuple $Q(N, r_{min}, r_{max}, h_{min}, h_{max})$. It contains all possible values of the r and h parameters within the allowed radius $[r_{min}, r_{max}]$ and time $[h_{min}, h_{max}]$ intervals for each zone $n \in N$. For the purpose of using a heuristic to solve the problem, we discretize both radius and reservation time values, with minimum resolution steps of r_{resol} and h_{resol} as parameters which, along with the allowed intervals for *r* and *h*, define the solution space:

$$|Q| = \left(\left(\frac{r_{max} - r_{min}}{r_{resol}} + 1\right) \cdot \left(\frac{h_{max} - h_{min}}{h_{resol}} + 1\right)\right)^N$$

Note that the size of the feasible solution space grows as the radius and time resolution increases and especially quickly as spatial resolution is increased (number of zones *N*). Further, we emphasize that $r_{min}, r_{max}, h_{min}$and h_{max} are algorithm parameters for establishing possible ranges of variation, not necessarily corresponding to actual values the algorithm will produce in the solutions. They are defined in order to allow users control over the values of the QoS – allowing the radius to be larger than 500m does not make much sense as this has a potential to place cars too far from the users to be practically accessible. The algorithm guarantees that solutions will not have radius *r* larger than r_{max} in any of the city zones.

For each *QoS* set, it is possible to calculate the profit and the accepted demand by running the simulator with these specific reservation parameters in the city zones. To ensure that the evaluation of different solutions can be compared we use the deterministic mode of the simulator. The pseudo-code of the algorithm is given in Algorithm 2. The overall algorithm has five parameters: execution time and the allowed QoS radius and time intervals. The LSO and PO operators have more detailed parameters as described below. The algorithm begins by generating an initial solution $QoS_{initial}$, with h and r across zones initialized to random values in the allowed intervals as set in the input, $r \in [r_{min}, r_{max}], h \in [h_{min}, h_{max}]$. The local search is then applied to this solution, resulting in the first local optimum QoS^*. The main algorithm loop then runs until the allowed time passes ($time$). The loop consists of the perturbation operator generating the current perturbed solution QoS' and then applying the local search to the perturbed solution to create a new local optimum QoS'^*. The best found solution is kept at all times and the *random*

ILS movement strategy is used, meaning that the next initial solution for the LSO is always the current perturbation result.

Algorithm 2: Pseudocode of the implemented iterated local search (ILS) metaheuristic

Procedure Iterated Local Search (time, $r_{min}, r_{max}, h_{min}, h_{max}$)

$QoS_{initial}$=generate initial solution($r_{min}, r_{max}, h_{min}, h_{max}$)
QoS^* = local search($QoS_{initial}$)
repeat until time expired
    QoS' = perturb(QoS^*)
    QoS'^* = localSearch(QoS')
    if the evaluation for QoS'^* is greater than the evaluation for QoS^* then
        $QoS_{best} = QoS'^*$
        $QoS^* = QoS'^*$
    end-if
loop

4.3. Local Search Operator (LSO)

The LSO used in our approach is a simple method that tries to increase and then decrease both the reservation distance and reservation time in the solution. After trying all the options, it chooses the change that caused the biggest improvement in the objective function value or retains the original value if no improvements have been produced. It has eight parameters: the initial solution *QoS*; the distance and time steps r_{step} and h_{step} define the increments of the variables that the local search will perform; the *partToSearch* value needs to be in the interval [0, 1] and it determines the approximate percentage of the *QoS* table which can be changed by the operator; the parameters $r_{min}, r_{max}, h_{min}, h_{max}$ define the allowed interval for the reservation distance and reservation time. The pseudocode of the operator can be found in Algorithm 3.

Algorithm 3: Pseudo-code of the local search operator (LSO)

Procedure local search (QoS, r_{step}, h_{step},$r_{min}, r_{max}, h_{min}, h_{max}$, **partToSearch***)*

for each element i in the service quality table QoS_i = (r_i,h_i)
    initialize random number e $\in [0, 1]$
    if e > partToSearch
        continue with next element QoS_{i+1}
    else
        while(improvement achieved)
            $r_{down} = r_i - r_{step}$ *, unless this would make* $r_{down} < r_{min}$

$QoSR_{down} = (r_{down}, h_i)$

$r_{up} = r_i + r_{step}$, *unless this would make* $r_{up} > r_{max}$

$QoSR_{up} = (r_{up}, h_i)$

update QoS to the element of $\{QoS, QoSR_{down}, QoSR_{up}\}$ *for which Z is maximal*

loop

while(improvement achieved)

$h_{down} = h_i - h_{step}$, *unless this would make* $h_{down} < h_{min}$

$QoSH_{down} = (r_i, h_{down})$

$h_{up} = h_i + h_{step}$, *unless this would make* $h_{up} > h_{max}$

$QoSH_{up} = (r_i, h_{up})$

update QoS to the element of $\{QoS, QoSH_{down}, QoSH_{up}\}$ *for which Z is maximal*

loop

end-if

next i

The operator visits each element of the *QoS* table in the main loop. The functionality of determining the part of the table to change is implemented by a random number generator which accepts modifications of solution elements with probability equal to the *partToSearch* parameter. The operator is non-deterministic and this way, it can achieve more diversity in the search. For each table element that the local search is modifying, the operator first tries to adjust the reservation radius. It first adds r_{step} to the current distance value, then it subtracts r_{step} and evaluates both modifications. If neither the adding nor the subtracting of the step value improved the solution, the table remains unchanged. If any of these produced an improvement, the table is updated so that the new r value for the current element is either the added or subtracted value, depending on which one caused a greater quality increase in the objective. The analogous procedure is performed with the reservation time-ahead parameter h.

The procedure ends when the main loop has iterated through all table elements. By systematically investigating the effects of distance and time variation and combining the effects of small changes, the local search can produce notable improvements to the initial solutions, especially when iterated with the perturbation operator in the ILS metaheuristic.

4.4. Perturbation operator (PO)

The perturbation operator introduces random changes in the part of the *QoS* table elements. The operator has eight parameters: the input *QoS* to modify, number of changes to make c, distance and time change steps Δr and Δh and maximum allowed distance and time, denoted

d_{max} and h_{max}. In total, the algorithm performs n change attempts of the randomly selected cells in the *QoS* table. It might change up to c cells or less if some cells are changed multiple times or left unchanged due to reaching the allowed limits of their values.

After choosing an element to change, the distance for the current element is modified. The algorithm first decides on the direction of the change: increase or decrease. To perform this choice, a random Boolean value ω_r is produced by the random value generator. If ω_r is true, then the distance in the current element will be increased for the distance step Δr, if it is false it performs the decrease, unless the change would take r out of the $[r_{min}, r_{max}]$ interval. The analogous operation is performed for the reservation time-ahead using the Boolean variable ω_h.

The fact that both the perturbation as well as the local search operator act only on a randomly selected subset of the table, combined with various possibilities for parameter selection, provides possibilities to adjust the algorithm to work as needed: so that the perturbation is low enough in order to keep the algorithm focused but still high enough not to prevent the algorithm from being stuck in local optima.

Algorithm 4: Perturbation operator

Procedure perturb (QoS, c, $\Delta r, \Delta h, r_{min}, r_{max}, h_{min}, h_{max}$,)

repeat c times

choose a random element of the QoS table $QoS_i = (r_i, h_i)$

choose two random Boolean variables ω_r *and* ω_h

if ω_r *is* true

$r_i = r_i + \Delta r$, *unless it would make* $r_i > r_{max}$

else

$r_i = r_i - \Delta r$, *unless it would make* $r_i < r_{min}$

end-if

if ω_h *is* true

$h_i = h_i + \Delta h$, *unless it would make* $h_i > h_{max}$

else

$h_i = h_i - \Delta h$, *unless it would make* $h_i < h_{min}$

end-if

end-repeat

5. Computational experiments

To test the performance of the proposed methodology we use two sets of benchmark problem instances: *hypothetical cities* and *case-study city*. The demand for the hypothetical cities is generated using a custom built trip generator with trip patterns based on typical features for two extreme cases: town and metropolis. The case-study city in this paper is the Lisbon municipality in Portugal, whose data was obtained from an agent-based model of Lisbon carsharing mobility (Martinez et al., 2017). Both categories include trips from a single working day of carsharing with reservation-ahead times assumed to be less than 16 hours. Even though we simulate only one day, reservations from "the night before" and any time before t=0 in the simulation are allowed as long as there is enough time for a system response during the simulation period. In real systems that do not have simulation limitations, periods much longer than 16 hours might be feasible. Trips that start at $t < t_a$ require response too early during the simulation start, and are rejected by the system.

While we ran several experiments that simulated up to three days of trips, it has shown that the results tend to be approximately proportional to the simulated number of days. Since the current limit of 16 hours allows simulating overnight reservations, we believe that simulating only one day is sufficiently representative for the purpose of this work. When simulating largest datasets, scalability issues start to occur as simulation times get longer. Due to the analyses above that indicate that there is no loss of generality when running one day and given the speed benefits of it, in this work, we simulate one day of trips.

5.1. Hypothetical cities

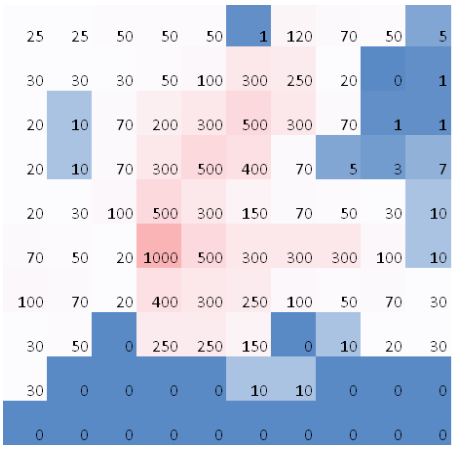
Problem instances for hypothetical cities are generated using a custom generator, built specifically for this study to enable generating carsharing trips in different environments. Two different hypothetical cities are used: town and metropolis: the town is representative of a small urban area with several tens of thousands of people, while the hypothetical *metropolis* is a large, densely populated urban area, typical of some of the largest cities in the world. Throughout this paper, we use a convention on naming our problem instance based on the city name followed by the number of trips in the parentheses, e.g., Town (500). The detailed features of the hypothetical cities used in this work are shown in Table 1, where geographical size, trip number, average trip duration and distance, number of cars and response time t_a are defined. In the case of a small town, response time is much shorter than in the metropolis since even in the most conservative estimates, it takes up to 40 minutes to reach any location in the service area. In the metropolis, this is estimated to take up to three hours.

Table 1: Hypothetical city features

Problem instance	**Total size**	**Trip number**	**Average trip time (min)**	**Average trip distance (km)**	**Fleet size A (cars)**	t_a **(min)**
Town (500)	5 x 5 km	500	5	2.6 km	10	40
Metropolis (40,000)	50 x 50 km	40,000	51	24.6 km	2000	180

The generation process involves both trip generation and distribution. It takes into account the total volume of trips and the temporal and spatial demand variations, all of which can be specified in the input. Generating trips based on an input distribution is implemented using a technique inspired by the fitness proportional selection operator used in genetic algorithm and the pseudorandom proportional rule in the ant colony optimization metaheuristic (Dorigo and Stutzle, 2006; Hancock, 1994; Luke, 2013).

Modeling trip intensity is done analogously: for each time interval, an individual value can be set up for the relative probability that an individual cell will be an origin and a separate value for the probability that the same area will be a destination. This allows modeling of the demand variations, as can be seen in Figure 3 which shows the origin/destination probabilities during the morning rush hour. The city consists of a highly populated central business district and broad residential areas on the outskirts. The southern part of the service area is water surface, therefore all relative chances of being an origin or destination are zero in these zones. As typical in the mornings, residential areas have more outgoing trips than the business areas. Business areas have more incoming trips during the morning. Initial vehicle locations are devised based on the trip origin probabilities during morning rush hour – this way the initial number of vehicles is proportional to the number of origins across zones to match the morning demand.



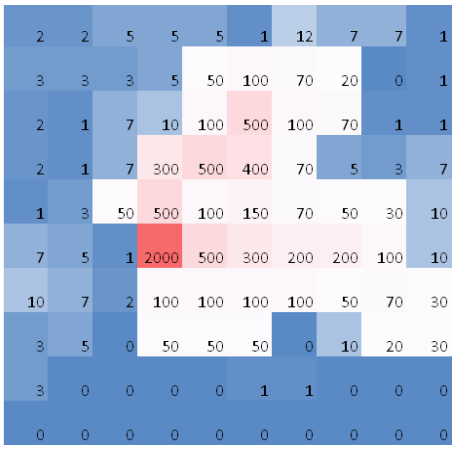


Figure 3: Relative chances of being an origin (left) and destination (right) in the "Metropolis (40,000)" problem instance during the morning rush hour

5.2. Case-study city: Lisbon

Realistic problem instances are based on the carsharing trip forecasting case study of the Lisbon municipality in Portugal (Lopes et al., 2014; Martinez et al., 2017) and an extensive mobility survey, performed by the Lisbon Municipality (Câmara Municipal de Lisboa, 2005; Jorge and Correia, 2013). Results from these studies were also used to decide on the fleet size and initial vehicle distribution. Unlike the rough estimates included in the hypothetical city models, this model takes into account the precise transportation habits of the local population and microscopic effects occurring on a high-resolution network of nodes.

The Lisbon area has been dealing with several mobility issues, including congestions and lack of parking space. Innovative transport solutions, including carsharing, are one of the alternatives that could help reduce mobility problems in the city. Details of the Lisbon problem instances are reported in Table 2.

Table 2: Lisbon problem instances features

	Geographical size (km)	**Trip number**	**Average trip duration (min)**	**Average trip distance (km)**	**Fleet size A (cars)**	t_a **(min)**
Lisbon (3,000)	11.6 x 11.0	3,000	14.3	4.3	80	60
Lisbon (6,000)	11.6 x 11.0	6,000	14.5	4.5	159	60
Lisbon (12,000)	11.6 x 11.0	12,000	14.5	4.4	318	60
Lisbon (25,000)	11.6 x 11.0	25,000	14,5	4,4	664	60

5.3. Simulation parameters for all runs

All of the following parameters are the same for all problem instances defined above. This can be a limitation since for instance parking price should be different in a small city when compared to a big city. Nevertheless maintaining these parameters equal allows for a better comparison between the scenarios.

The costs of vehicle ownership are estimated using the Interfile tool for car ownership costs estimation (INTERFILE, 2016). As a reference vehicle, we use an average city vehicle with the initial cost of 20,000€, under assumptions that the company financed the entire initial cost using a loan with an interest rate of 12% and the vehicle's residual value after three years equal to 5,000€. For such a vehicle, the cost of depreciation (C_v) is 17€ per day, with expected use

duration of 3 years. The cost of maintaining the vehicle (C_{mv}) is estimated to be 0.007 € per minute, taking into account insurance, fees, taxes, fuel, maintenance and wear of the vehicle. The relocation cost C_r is estimated to 0.20€ per minute, and it includes fuel, vehicle maintenance and staff costs, as well as the compensation for the cost of reaching the relocation trip origin. We estimate the parking cost C_p to be 1.2€ per hour **(**Emel Lisboa, 2018). The service fee per minute is set to 0.30€ per minute, based on the rates of the global operator Car2Go (Car2Go, 2017). Taxi start price is set to 3.5€ and price of driving 1 km is 0.47€ (Numbeo, 2018). User walking speed is set to 5 km/h and the vehicle walk-in distance c_{wd} is set to 250 m (Correia, 2009).

5.4. Running the experiments

The simulator and the optimization algorithm are implemented in Java 1.8 programming language. The experiments were performed on a computer equipped with a 2.4 GHz Intel Core i7-4700HQ processor and 16 GB of RAM, using Java 1.8 runtime environment under Windows 10 operating system. The running time to simulate one full day is less than a second in the smallest instances and around 30 seconds for the ones with the largest number of cars and trips.

The objective function parameters are chosen to give half of the weight to the profit: $w_p = 0.5$ and the rest is equally distributed to other three service quality components: $w_r = 0.167$, $w_h = 0.167$ and $w_d = 0.167$. The sum of all factors is equal to one, to ensure that the resulting values will be normalized to the [0, 1] interval. To optimize metaheuristic performance, tuning experiments were performed on the Lisbon (6,000) dataset. The best performing configuration found during tuning is shown in Table 3, and this set of algorithm parameters was used in all subsequent experiments in this work. Note that all of these parameters are algorithm input parameters, as explained in Section 4.2 and that the distance and time limits $[r_{min}, r_{max}]$, $[h_{min}, h_{max}]$ refer to allowed intervals, not the realized values in the solutions, although they might coincide.

Table 3: Iterated local search metaheuristic parameters

Parameter	Value
Comfortable walk-in distance c_{wd}	250 m
Local search distance step $\boldsymbol{r_{step}}$	200 m
Local search time step $\boldsymbol{h_{step}}$	480 min
Local search percentage to explore **partToSearch**	100 %
Perturbation distance change $\Delta \boldsymbol{r}$	100 m
Perturbation time change $\Delta \boldsymbol{h}$	300 min

Perturbation number of elements to consider $\boldsymbol{c}$	50 (50%)
Minimum allowed distance $\boldsymbol{r_{min}}$	50 m
Maximum allowed distance $\boldsymbol{r_{max}}$	500 m
Minimum allowed time $\boldsymbol{h_{min}}$	60 min
Maximum allowed time $\boldsymbol{h_{max}}$	1080 min
Radius resolution $\boldsymbol{r_{resol}}$	1 m
Time resolution $\boldsymbol{h_{resol}}$	1 min

6. Results

6.1. R-BR Method under Constant *QoS*

In the first round of the experiments, we wanted to assess the differences in the described reservation enforcement strategies under constant service quality in the entire city area. The reservation service quality for all experiments in this round was set to $r = 200\ m$, $h = 600\ min$ (10 h). In the input trip volume, which consists of spontaneous walk-ins and trips reserved ahead, we varied the reservations percentage (denoted ρ), while keeping all other conditions constant. Note that $\rho = 10\%$ does not mean 10% more trips in total, it means that 10% of trips that were walk-ins in the original dataset are now long-term reservations.

The results are shown in Figure 4. The x-axis of each graph shows the reservation percentage, and the y-axis shows the daily profit in euros for that day. A detailed breakdown of profit into its components: revenue (R), fixed, variable and total cost (denoted C_f, C_v and C respectively) as well as the percentage of demand satisfied and outsourced demand (taxis) are specifically presented for the Town (500) and the Lisbon (25,000) problem instances in Tables 4 – 7.

When the reservation service is not offered ($\rho = 0$), Town (500), Lisbon (3,000) and Lisbon (6,000) instances are generating losses and all others are profitable. As seen in Table 4, in our Town (500), only 9.80% of the demand is satisfied for $\rho = 0$. Despite the fact that there are 500 potential trips, most of them are not done because the closest vehicle is too far for a comfortable walk-in at the moment of the request (c_{wd}= 250m). A small fleet with only 10 vehicles is not enough for sufficient coverage and to ensure that, on average, vehicles are close enough to interested users. The revenues generated by such a low number of trips are not sufficient even to cover the fixed costs of fleet ownership and parking. Similar results are observed in the smallest Lisbon instances. These results indicate that carsharing can hardly be profitable below a certain threshold of a minimum trip volume and confirm similar findings in other studies (Klintman, 1998; Celsor and Millard-Ball, 2007; Rotaris and Danielis, 2018).

When reservations are allowed ($\rho > 0$) and vehicle locking is applied as the enforcement method, the profit steeply drops. At $\rho = 20\%$, only Lisbon (25,000) is profitable and when ρ reaches 30%, vehicle locking is not profitable in any of our problem instances. This method causes long waiting periods during which vehicles are idle and therefore it reduces the overall service availability for potential trips. These effects are clearly visible in Table 4 and Table 6: more reservations bring less satisfied demand and less revenue. Very large losses result from high reservation percentages and high trip volume, e.g., more than 20,000 € per day in Lisbon (25,000).

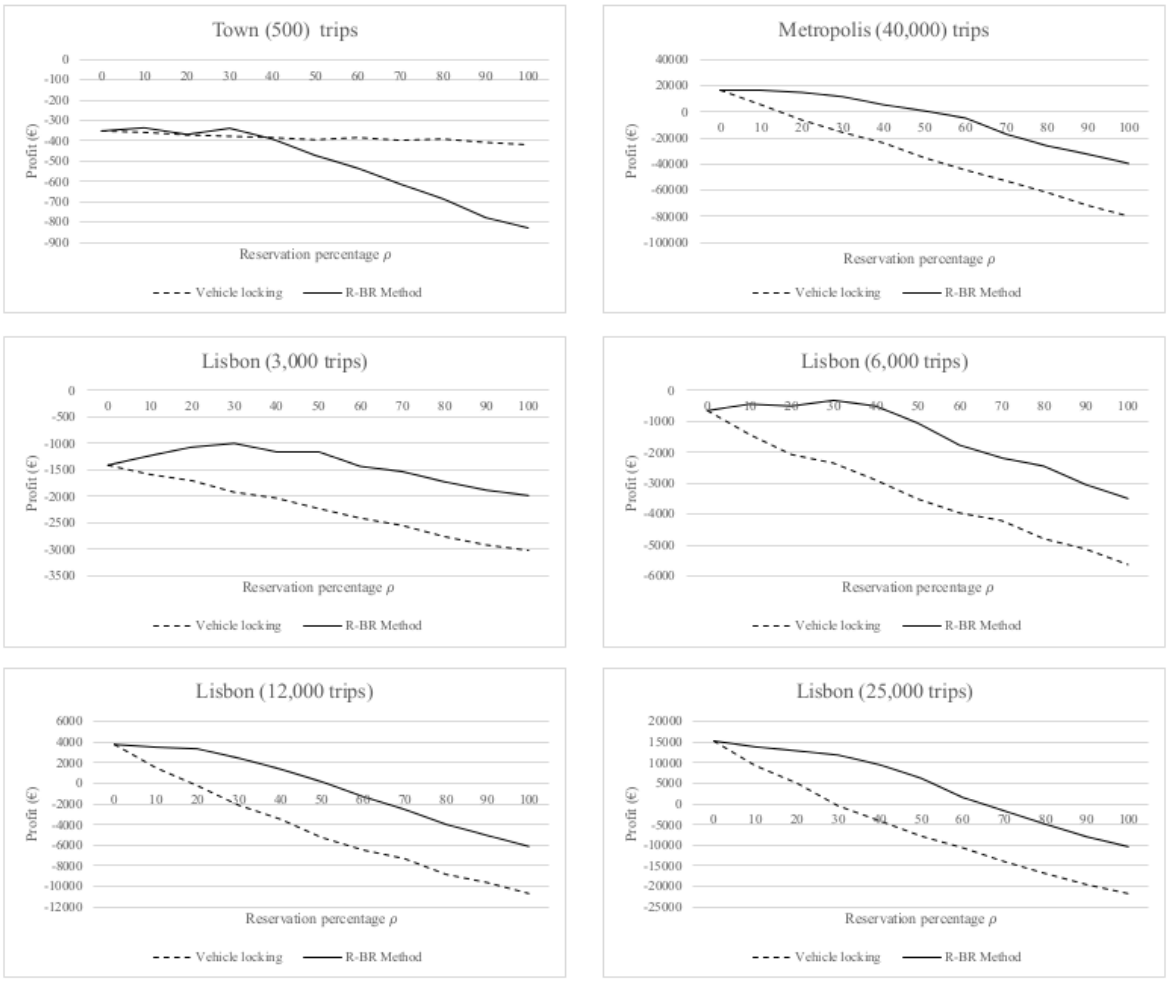


Figure 4: Comparison of vehicle locking and reservation strategies under constant QoS

The proposed RB-R method also brings profit drops when reservations are present. However, in all problem instances except in Town (500), R-BR considerably outperforms vehicle locking. In Lisbon (3,000) and Lisbon (6,000), R-BR brings more revenue and allows more demand to be satisfied. Nevertheless, the improvement is not sufficient to turn around these losses. In the remaining four instances, R-BR can maintain the profitability of the operations with two to three times the reservation volume than the vehicle locking. While R-BR imposes additional relocation and taxi costs, the method leads to less rejected trips which brings higher revenues

(Table 5 and Table 7). Whether the benefits of the added revenue will outweigh the costs depends on the number of performed trips. Note that even with $\rho = 100\%$ all trips are not accepted since the QoS settings allow the system to reject reservations more than 10 hours ahead (h=600min).

In the Town (500) instance, the profit of vehicle locking is similar to the RB-R for low ρ, however, with more than 50% of reservations, the RB-R quickly starts to be worse than locking (Figure 4). Part of this is due to the increased costs brought by the relocation movements, however, the key reason for poor performance with high ρ is the increased level of trip outsourcing to taxi. With $\rho > 50\%$, the outsourcing rate quickly starts rising, adding large extra costs (Table 5).

The performance of the newly proposed R-BR method outperforms the simple vehicle locking, in all cases with sufficiently high demand. While R-BR method has a high potential to improve the profit, it should only be considered for the systems that are profitable with no reservations as the added operational cost of enforcing reservations can lead to even more losses when the number of trips is small.

Table 4: Vehicle locking method performance in Town (500 trips) problem instance

Reservations percentage (ρ)	**Profit (P) (€/day)**	**Costs (€/day)**			**Revenue (R) (€/day)**	**Demand satisfied**
		C	**C_f**	**C_v**		
0 %	-351.27	460.55	458.00	2.55	109.28	9.80%
20 %	-370.68	460.09	458.00	2.09	89.40	8.03%
40 %	-383.93	459.77	458.00	1.77	75.83	6.48%
60 %	-384.66	459.75	458.00	1.75	75.09	5.89%
80 %	-391.00	459.60	458.00	1.60	68.60	5.28%
100 %	-419.37	458.92	458.00	0.92	39.56	4.10%

Table 5: R-BR method performance in Town (500 trips) problem instance

Reservations percentage (ρ)	**Profit (P) (€/day)**	**Costs (€/day)**			**Revenue (R) (€/day)**	**Outsourced (taxi) demand**	**Demand satisfied**
		C	**C_f**	**C_v**			
0 %	-351.27	460.55	458.00	2.55	109.28	0.00%	9.80%
20 %	-368.91	605.95	458.00	147.95	237.04	2.21%	21.29%
40 %	-391.47	813.42	458.00	355.42	421.95	7.44%	35.81%
60 %	-535.02	1,102.37	458.00	644.37	567.35	16.53%	51.61%
80 %	-686.17	1,405.30	458.00	947.30	719.12	28.43%	64.72%
100 %	-829.65	1,744.10	458.00	1,286.10	914.46	40.69%	80.57%

Table 6: Vehicle locking method performance in Lisbon (25,000 trips) problem instance

Reservations percentage (ρ)	**Profit (P) (€/day)**	**Costs (€/day)**			**Revenue (R) (€/day)**	**Demand satisfied**
		C	**C_f**	**(C_v)**		
0 %	15,249.54	31,502.07	30,411.20	1,090.87	46,751.61	42.92%
20 %	12,870.18	36,080.55	30,411.20	5,669.35	48,950.73	34.41%
40 %	9,497.43	42,789.81	30,411.20	12,378.61	52,287.24	26.61%
60 %	1,558.51	54,054.44	30,411.20	23,643.24	55,612.95	20.61%

80 %	-4,965.13	68,583.61	30,411.20	38,172.41	63,618.48	14.75%
100 %	-10,402.24	84,073.84	30,411.20	53,662.64	73,671.60	9.86%

Table 7: R-BR method performance in Lisbon (25,000 trips) problem instance

Reservations percentage (ρ)	**Profit (P) (€/day)**	**Costs (€/day)**			**Revenue (R) (€/day)**	**Outsourced (taxi) demand**	**Demand satisfied**
		C	**C_f**	**C_v**			
0 %	15,249.54	31,502.07	30,411.20	1,090.87	46,751.61	0.00%	42.92%
20 %	12,870.18	36,080.55	30,411.20	5,669.35	48,950.73	0.00%	45.62%
40 %	9,497.43	42,789.81	30,411.20	12,378.61	52,287.24	1.15%	49.91%
60 %	1,558.51	54,054.44	30,411.20	23,643.24	55,612.95	4.35%	54.96%
80 %	-4,965.13	68,583.61	30,411.20	38,172.41	63,618.48	12.33%	64.68%
100 %	-10,402.24	84,073.84	30,411.20	53,662.64	73,671.60	23.97%	77.29%

Given that the *h* limit of ten hours allows very long vehicle idle periods, we performed an additional investigation of the vehicle locking performance with various durations of the allowed reservation time. The results for Lisbon (25,000) problem instance, shown in Figure 5, demonstrate that decreases in *h* allow higher profits. However, these profit improvements come at the cost of accepting less demand and less user satisfaction. With very short h=120 min, only up to 8% of reservations is accepted, the number of accepted trips is approximately equal to the number of walk-ins and reservations hold only a very small share of total accepted trips. As the number of walk-ins drops linearly as we increase the ρ, the profit drop is roughly linear as well when *h* is low. These results show that even with high trip volume, vehicle locking is not a viable strategy to achieve long reservation times, hence the current carsharing operators' practice of very short reservation periods makes sense.

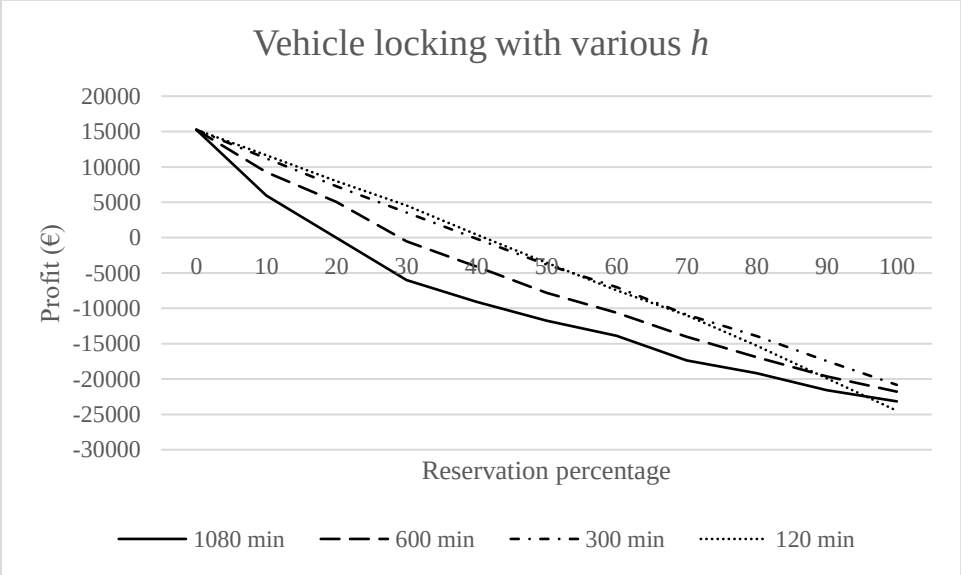


Figure 5: Vehicle locking performance under various reservation time limits

To assess the sensitivity of the profit and accepted demand on variations in the service level, we used a simple *QoS sweep* algorithm which examines all (r, h) combinations with the (200 m, 240 min) steps under various reservation percentages. The algorithm was applied to Lisbon (12,000) problem instance with the R-BR method. The reservation QoS variations do not influence the system when there are no reservations, and in this case, the profit is 3,767.44€, and 36.20% of trips are accepted. For the same problem instance, Figure 6 shows the results of the QoS sweep when reservations are present. Increasing the reservation percentage, in general, lowers the profit and increases the accepted demand. A notable exception is the case with very small $h = 120\ min$ (2 hours), for which the demand decreases due to highly limited capability to accept reservations.

Variations in r change the level of accepted demand only slightly since neither walk-ins nor reservations depend directly on it. Walk-in acceptance depends on the separate, comfortable walk-in distance parameter (c_{wd}) and the vehicle stock around the user. As r increases, there is less need for relocations as reservation support. These trips help balance the system, and therefore r increases cause less balanced vehicle stock, leading to less performed walk-ins. Reservations acceptance depends only and directly on h. Therefore, a higher reservation percentage causes higher sensitivity of demand on h changes. At $\rho = 100\%$ and with $h = 0$, no trips are accepted. With $h = 18h$ (1080 min), all trips are accepted as the longest time-ahead in our problem instances is 18h before departure. The rate of the reservation acceptance increase as a function of h depends on the distribution of time-ahead among these trips.

Profit

ρ = 25%

	h=120	*h=360*	*h=600*	*h=840*	*h=1080*
r=100	881,98 €	1 896,25 €	2 460,28 €	2 745,96 €	2 900,96 €
r=300	1 055,77 €	2 498,51 €	3 217,56 €	3 539,37 €	3 794,39 €
r=500	1 062,55 €	2 879,88 €	3 984,59 €	4 462,82 €	4 741,66 €

ρ = 50%

	h=120	*h=360*	*h=600*	*h=840*	*h=1080*
r=100	-2 258,59 €	-880,27 €	-285,18 €	-436,63 €	-1 045,51 €
r=300	-1 989,63 €	67,55 €	955,59 €	841,38 €	255,28 €
r=500	-1 673,07 €	1 147,89 €	2 198,28 €	2 289,45 €	1 749,98 €

ρ = 100%

	h = 120	*h = 360*	*h = 600*	*h = 840*	*h = 1080*
r = 100	-10 665,24 €	-6 336,37 €	-6 692,25 €	-8 755,45 €	-10 501,32 €
r = 300	-9 992,71 €	-5 048,76 €	-5 449,73 €	-8 041,41 €	-9 865,39 €
r = 500	-9 489,47 €	-3 822,79 €	-4 261,53 €	-7 197,86 €	-9 120,72 €

Accepted demand

ρ = 25%

	h=120	*h=360*	*h=600*	*h=840*	*h=1080*
r=100	32,78%	37,79%	40,77%	43,05%	44,48%
r=300	32,55%	37,73%	40,61%	42,57%	44,11%
r=500	32,18%	36,94%	40,02%	42,09%	43,50%

ρ = 50%

	h=120	*h=360*	*h=600*	*h=840*	*h=1080*
r=100	29,35%	41,35%	48,44%	52,68%	55,16%
r=300	29,13%	40,70%	47,74%	51,97%	54,84%
r=500	28,98%	40,33%	47,18%	51,51%	54,51%

ρ = 100%

	h = 120	*h = 360*	*h = 600*	*h = 840*	*h = 1080*
r = 100	19,10%	58,49%	80,24%	92,89%	100,00%
r = 300	19,10%	58,49%	80,24%	92,89%	100,00%
r = 500	19,10%	58,49%	80,24%	92,89%	100,00%

Figure 6: Profit and accepted demand under various constant QoS levels

Profit is sensitive to both h and r. A small radius means the lower probability that an available car will be close enough to the requested departure location at the system response moment. This causes the costly relocators to do more work, thus decreasing the profit. Increasing h increases the profit until the levels at which the fleet becomes saturated and a large degree of

trip outsourcing is needed. At $\rho = 100\%$, this effect is clearly visible: the profit first grows as h increases from 120 to 360, however further increases to $h \geq 600\,min$ do not help and start decreasing the profit.

6.2. R-BR and Fleet Size

Fleet size planning is an important issue any potential carsharing service faces, especially when the operator is starting its business. Too many vehicles will lead to costs that are too high to reach a profit. Conversely, an insufficient vehicle number will reduce service coverage and customer satisfaction. We conducted experiments in the Metropolis instance with constant QoS r=200 m and h=600 min to investigate the influence of fleet size on the carsharing profit when reservations are allowed. As can be seen in Figure 7, the best profit with no reservations ($\rho = 0$) is achieved with 3,000 vehicles (approximately equal to the profit with 2,000 vehicles). Having 5,000 vehicles causes a prominent profit drop. From the chart on the left, it is visible that with $\rho > 0$ and vehicle locking, small fleets give the best results.

When the R-BR method is applied, the decision is more complex as fleet sizes that work great with no reservations can perform poorly with a high number of reserved trips. The key issue with R-BR and small fleets is that many expensive outsourced trips are required as the number of reservations increases. The chart on the right in Figure 7 illustrates this: having 1,000 vehicles gives results close to the best configurations with no reservations, however, with more than 20% the profit quickly drops. Using 3,000 vehicles gives the best profit, especially with high reservation percentages. Adding more does not help as is visible in the 5,000 vehicles case that causes a fixed cost increase that is not justified by the savings in outsourced taxi trips.

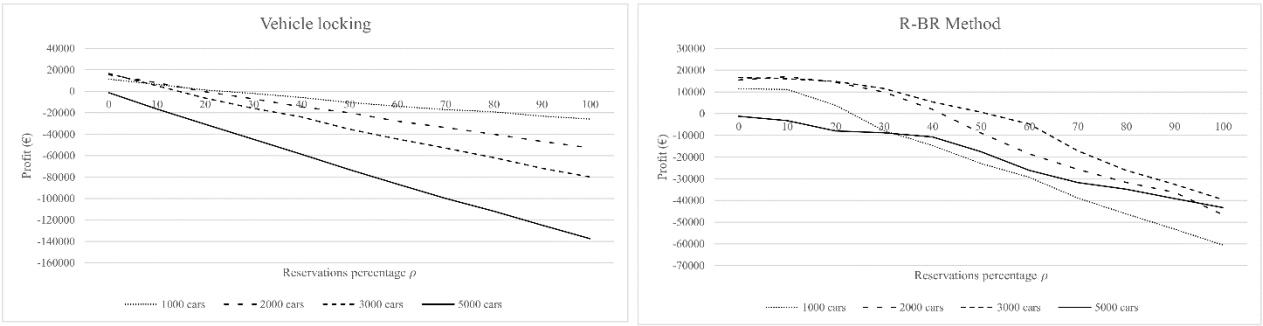


Figure 7: Profit with different relocation enforcement methods and number of vehicles

6.3. Balancing relocations

Relocation movements are a frequently used technique to ensure vehicle stock balance across different areas of a city. We wanted to investigate the effect of balancing relocations when used with and without reservations. Without reservations, the effect of relocations on profit is directly a result of better vehicle stock balance and better coverage of the areas with the most demand. We ran a series of experiments in Lisbon (6,000), Lisbon (12,000), Lisbon (25,000), and Metropolis, where a varying balancing intensity was used with walk-ins only allowed (no reservations). The impact of the movement limit b_n is investigated with balancing movements starting at each hour (b_p = 1h).

The charts in Figure 8 show the changes in the profit (y-axis) resulting from various relocation intensities (trips per hour, x-axis). In the left chart, it is evident that balancing cannot improve the profit with the low demand in Lisbon (6,000) problem instance. Without balancing (0 balancing trips per hour), the system is generating losses of 653 € per day and adding balancing trips only makes it worse as cost of relocations is not compensated with increased revenue from additional trips. With a sufficient number of daily trips, however, the balancing helps, as the demand density is higher and a significant revenue increase is possible with the vehicle stock that better follows the demand. As seen in the right chart in Figure 8, using 50 relocation trips per hour produces the best results in Lisbon (25,000). However, even in this instance with high demand, insisting on a perfectly balanced system can be too costly. As seen in the example with 300 balancing trips per hour in Figure 8, too much effort to balance the system can significantly undermine the profit, even cause the profitable system to start producing losses.

Detailed results of applying balancing relocations are provided in Table 8, where profit for the best found balancing intensity is shown for four problem instances with sufficient demand. While in Lisbon instance with moderate and high demand balancing helps, in Lisbon (6,000) and in Metropolis, the simple balancing approach was not able to improve profit. In Lisbon (6,000), it is most likely due to low demand, while a possible reason for inefficiency in Metropolis is a big service area, in which the standard 10 by 10 grid might not be detailed enough to provide precise information about hot-spot locations.

Table 8: Balancing relocations performance

Problem instance	Initial profit (€)	Best found balancing intensity (trips per hour)	Profit with balancing (€)
Lisbon (6,000)	-652.96	0	-652.96
Lisbon (12,000)	3,767.44	5	3,936.03

Lisbon (25,000)	15,249.54	50	17,447.98
Metropolis (40,000)	16,573.85	0	16,573,85

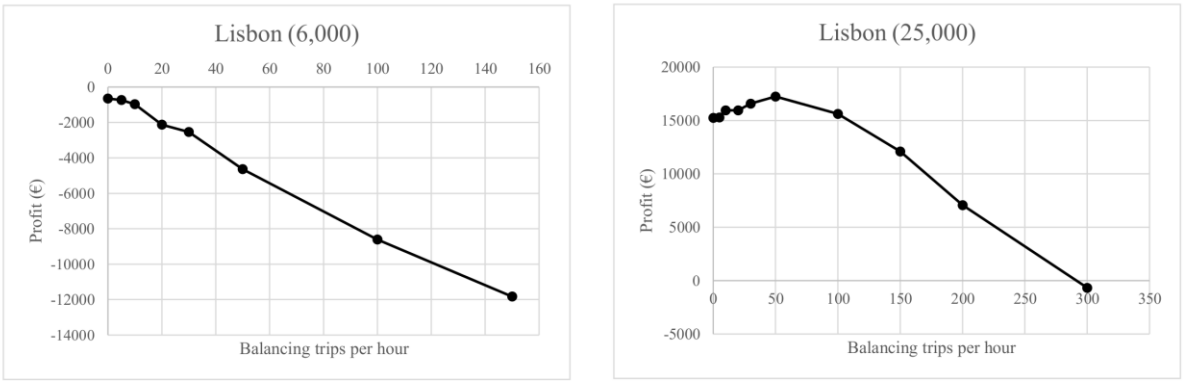


Figure 8: Effect of balancing relocations with no reservations (walk-ins only)

When balancing is used complementary to the R-BR method of enforcing reservations, both the reservation (type 1) and the balancing movements (type 2 relocations) help improve the vehicle stock balance. In Figure 9, the impact of balancing intensity on the profit (y-axis) is shown for varying reservation percentage (x-axis). Moderate intensity of balancing trips helps improve profit when the number of relocation trips is low, as seen with 50 balancing trips per hour. However, as the reservations percentage becomes higher than 20%, balancing starts causing profit declines. As we use free-floating carsharing, the balancing does not help much with reservations when the radius around the user is low (200 m in our reference example) – except in the unlikely case where the relocator moved the vehicle to a position inside the allowed radius around an unknown future reservation. The balancing algorithm balances the vehicle stock across zones without knowing the future reservation demand hot-spots. Since a reserved trip requires an additional relocation if the previous balancing trip did not bring the vehicle close enough, an overall number of relocation trips might unnecessarily increase when combining reservations and balancing.

While the results with reservations produce lower profit than best results with no relocations, due to the different nature of the demand involved, they are not directly comparable. This experiment assumes a constant number of trips, therefore not accounting for the potential of reservation service to attract more customers and generate more trips. Such increases in trip volume depend on the market and user preferences, and if substantial, they would allow carsharing providers to further benefit from all positive effects of increased trip volume. Finally, implementing more sophisticated relocations, e.g., better integration between relocation support

and balancing has the potential to improve performance further when balancing is used along with R-BR method.

Given the fact that simple balancing relocations combined with reservations appear detrimental even with low reservation percentages, and to ensure the ability to compare the results with vehicle locking, the variable QoS experiments presented in the following sections were done without balancing relocations.

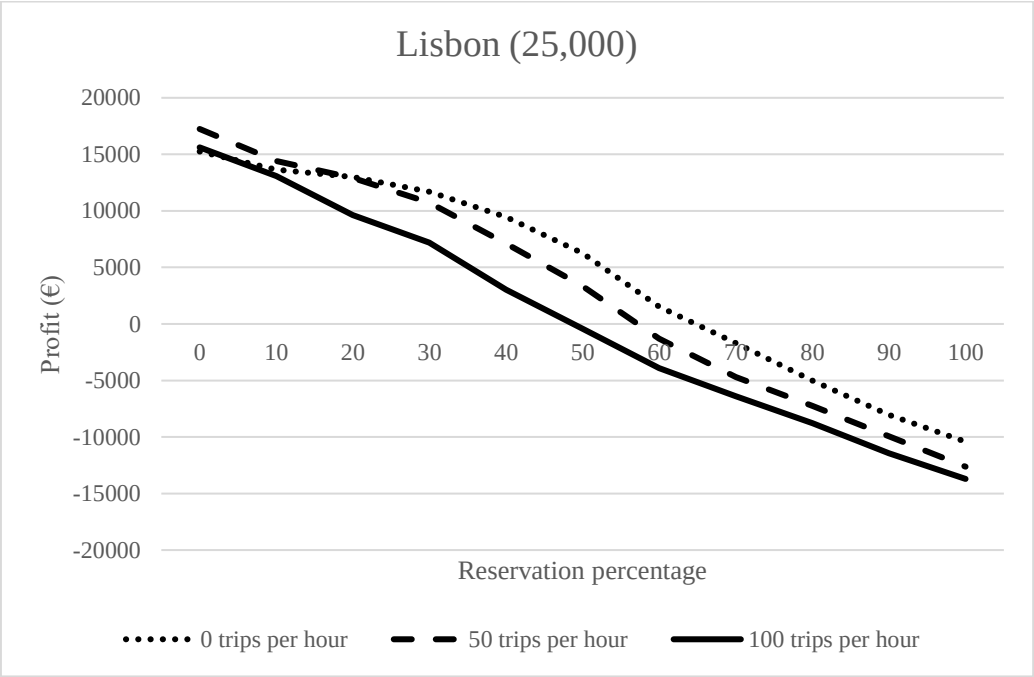


Figure 9: Balancing relocations used with the R-BR method

6.4. Variable *QoS*

To further improve the performance of the reservations, we ran the devised ILS metaheuristic on our problem instances in a setting with high reservation load: $\rho = 50\%$. Profit bounds for all problem instances P_{min} and P_{max}were estimated by running the *QoS sweep* algorithm, however with a finer resolution of (50 m, 60 min). Further, an additional run of the ILS was done for highly profitable problem instances. After the profit bounds have been established, the best constant QoS was found by calculating the entire objective function in the QoS sweep algorithm.

We performed five runs of the ILS metaheuristic on each problem instance with the initial solution being a random solution with $r \in [50, 500]$ and $h \in [60, 1080]$. The other algorithm parameters were used as in Table 3. For smaller problem instances, the algorithm is able to produce good solutions more quickly than for those with more trips and cars. Therefore, the time limit of 2 hours was used for problem instances with less trips, while up to 10h was used for those

with more trips and longer time needed for solution evaluation. The results are given in detail in Table 9, where average, median, best and worst Z in the performed runs, as well as the standard deviation of the objective function values. The results show that ILS was always able to find a better solution than the best known with constant QoS.

Table 9: ILS performance in 5 runs

Problem instance	Constant QoS	Variable QoS					
	Best known Z	ILS time limit (h)	Z in 5 ILS runs				
			Average	Median	Best	Worst	Std. dev
Town (500)	53.81%	2	62.74%	62.00%	71.54%	57.72%	5.66%
Metropolis (40,000)	76.49%	10	80.79%	81.02%	82.75%	79.07%	1.38%
Lisbon (3,000)	68.24%	2	74.14%	73.93%	74.97%	73.21%	0.75%
Lisbon (6,000)	66.59%	2	77.14%	77.05%	78.35%	75.92%	1.05%
Lisbon (12,000)	68.08%	2	76.58%	76.03%	79.87%	73.09%	2.53%
Lisbon (25,000)	70.42%	5	76.92%	78.73%	78.96%	72.18%	2.95%

A detailed comparison of best known constant and variable QoS solutions is provided in Table 10, where the elements of the objective function are given: average time $\bar{h}$, and radius $\bar{r}$, over zones, as determined by the heuristic as well as the satisfied demand, profit and the overall objective function value Z. These results show that the ILS metaheuristic was able to improve the profit in all problem instances, in some of them substantially. In most examples, the satisfied demand is slightly lower than with the constant QoS. Further, average r and h are very comfortable for all variable QoS solutions: less than 200m and more than 12h in all problem instances. Further, Lisbon (6,000) is not profitable even with the best constant QoS. However, ILS was able to achieve high increases of the profit, turning the service generating losses into a profitable one. Attaining profitability with the large reservations pressure of 50% used in these experiments is very difficult, as shown in the constant QoS experiments. Nevertheless, using ILS, it was possible to optimize the profit to the levels which are better than the profits with no reservations. The especially good QoS with high profits for Metropolis (40,000) and Lisbon (25,000) problem instances show the potential of this method with large trip volume. For example, the Metropolis (40,000) with no reservations has a daily profit of 16,573.85 € (Figure 4), while the profit of the best variable QoS solution with 50% reservations is a slightly higher value of 17,905.17€. At the same reservation level, the constant QoS solution with the best Z is barely

profitable (less than 300€ per day). The highest known constant QoS profit is 11,122.17€, achieved with much higher r and similar h QoS (r=500m, h=840 min). For an additional comparison, vehicle locking with 50% of reservations causes losses of more than 35,000 € per day.

We additionally compared the performance of both ILS and best constant solutions with the basic random restart search algorithm in which the best out of randomly generated solutions is selected. With run-times equal to those given to the ILS, the random restart algorithm was not able to outperform the best found constant solution and the solutions found by ILS outperform it. A very simple algorithm such as random restart is not able to refine the solutions well enough to give good results and produces a very irregular distribution of radiuses and times that does not reflect the shape of the central business district as ILS does.

Table 10: Comparison of best known constant and variable QoS solutions

Problem instance	**Best found constant QoS**					**Best found variable QoS**				
	h (min)	r **(m)**	$\frac{d_{sat}}{d_{tot}}$	**Profit (€)**	Z	$\bar{h}$ **(min)**	$\bar{r}$ **(m)**	$\frac{d_{sat}}{d_{tot}}$	**Profit (€)**	Z
Town (500 trips)	180	50	19%	-368.59	53.81%	751.39	129.0	38%	-335.44	71.54%
Metropolis (40,000 trips)	1080	50	53%	293.15	76.49%	924.71	139.15	47%	17,905.17	82.75%
Lisbon (3,000 trips)	1080	50	49%	-1,368.32	68.24%	764.65	135.56	40%	-725.82	74.97%
Lisbon (6,000 trips)	900	50	50%	-1,255.04	65.39%	786.83	153.35	44%	275.90	78.35%
Lisbon (12,000 trips)	900	500	52%	2,267.86	68.08%	821.83	196.36	48%	2,832.93	79.87%
Lisbon (25,000 trips)	960	500	57%	10,668.61	70.42%	785.12	186.32	51%	11,552.52	79.88%

Detailed values of QoS parameters across the zones for the best-known solution for Metropolis (40,000) are shown in Figure 10, where the radiuses (left) and reservation times ahead (right) are displayed in a heat map. Comparing these figures with O-D probabilities in Figure 3, indicate that the ILS metaheuristic was able to capture the general behavior of the system. In general, the algorithm lowered the service quality in zones with many trips and kept it very high in areas with fewer trips. Increasing the radius in the central business district has the benefit of lowering the relocation costs. Likewise, it is in general kept low in zones with fewer trips where

due to the low concentration of vehicles, relocations will most likely be needed regardless of r. Average r is 139m, however, median r is 50m, equal to the lowest allowed value r_{min} = 50m. Distribution of times ahead is similar: the h heat-maps (Figure 10, right) give an approximate outline of the city center contours. In areas with a lot of trips, the algorithm had a tendency to lower the allowed time-ahead t, most likely due to the fact that many reservations in these areas have the potential to overload the fleet and cause too many relocations and outsourcing costs. The average and median values of r and h are only slightly worse than in the best known constant QoS and even the zones with the lowest QoS by far outperform the reservation time-ahead of 30 minutes that is nowadays allowed by the carsharing operators.

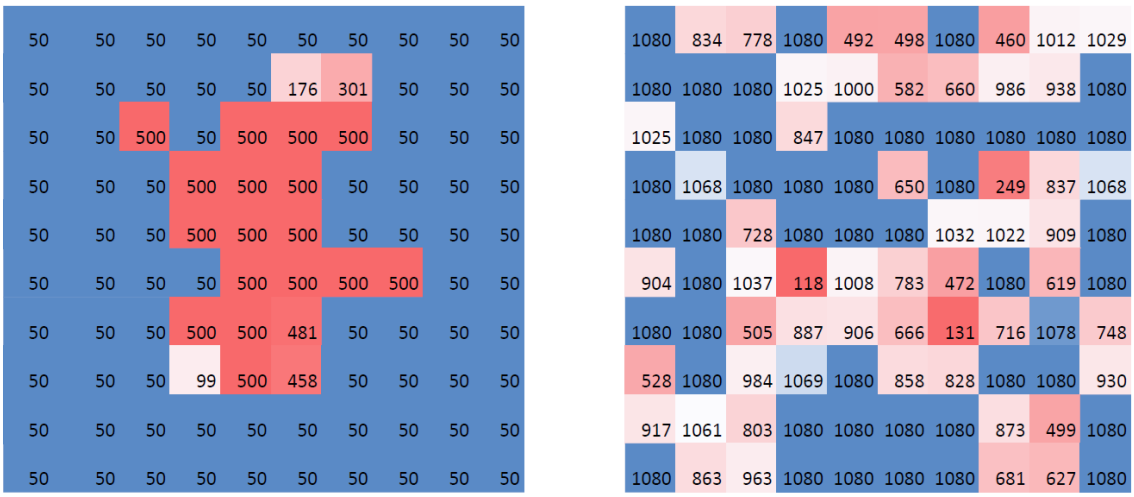


Figure 10: radiuses (left) and times ahead (right) in the best known variable QoS solution for Metropolis (40,000)

6.5. Choosing the optimization objective

The experiments conducted so far used a balanced set of weights that put 50% of the weight into the profit, and the remaining 50% was equally distributed on the satisfied demand, radius and reservation times. To evaluate performance with different goals, we performed additional two experiments: (1) optimising only the profit while not caring about other objectives with weights set to $w_p = 1.0$, $w_r = w_h = w_d = 0$ and (2) optimizing only the demand and not caring about other objectives, using weight factors $w_p = w_r = w_h = 0$, $w_d = 1.0$. Both experiments were performed on Lisbon 12,000 problem instance with very high reservation load ($\rho = 50\%$). The results in Table 11 show the performance of the best-found solution in 5 different runs with the two different objective function setups.

In the first experiment, a highly profitable result of 2,875.64 €/day was found. While this profit oriented solution for Lisbon (12,000) is the best known profit for that problem instance, it is only slightly better than 2,832.93 €/day found from a solution with more balanced optimization criteria (Table 10). In practical terms, those values can be considered equal as the difference between the two values is 42€/day, which is a profit increase of only 1.4%. While both solutions result in a similar profit, the solutions themselves are considerably different, and the other criteria are much worse in the "for-profit-only" result. The average radius around the user in the profit-oriented solution (361m) is almost two times bigger than the radius in the balanced solution (196m). Likewise, the allowed reservation time is on average 25% shorter in the "for-profit-only" solution. Given the fact that 1% profit increase does not give sufficient justification to force users to walk double the distance to reach a reserved car, a reasonable provider will choose the solution from the balanced configuration.

When only the demand is optimized, as seen in the second experiment in Table 11, the result has a high demand. However, the best-found solution from Table 9 outperforms it in all other criteria, low profit being especially notable. While this means that optimizing only the demand is not an attractive option for commercial providers, it is still an effective demonstration of the flexibility that the weighted multiobjective function provides.

The algorithm can prioritize different user preferences and adapt to various business goals. Each carsharing provider has complete freedom to choose the optimization goals that best fit its current moment. Further, it is possible to run multiple algorithm configurations for the same problem and then leave the final decision to the human. Provided with several different solutions, the operator can decide on a good trade-off between various criteria, with the results from Table 10 and Table 11 being a good illustration of that potential.

Table 11: Algorithm results with different optimization goals in Lisbon 12,000

Experiment	**Weights setup**				**Results**				
	Profit (w_p)	**Radius (w_r)**	**Time (w_h)**	**Demand (w_d)**	$\bar{h}$ **(min)**	$\bar{r}$ **(m)**	$\frac{d_{sat}}{d_{tot}}$	**Profit (€)**	Z
1. Optimize profit	1.0	0.0	0.0	0.0	641.17	360.86	46.26%	2,875.64	98.44%
2. Optimize demand	0.0	0.0	0.0	1.0	715.87	251.06	52.79%	93.12	52.79%

6.6. Performance comparisons

In previous experiments, the effects of various carsharing operation alternatives such as adjusting the fleet size and balancing relocations were investigated in a reservation context. The variable QoS experiments in Section 6.4 and 6.5 were performed with very high reservation percentage ($\rho = 50\%$) and they show that the R-BR method is robust and cost-efficient and keeps the operation profitable even under such immense pressure on the reservation support services. We, however, also wanted to evaluate the R-BR performance under low percentage of reserved trips, for example in cases where users are not very interested in reservations, or in early stages of implementation, when promotional activities are still in progress and users are not yet informed about the new service. To assess the performance of the system under low usage of the reservations, we did a series of experiments on four input instances: Lisbon (6,000), Lisbon (12,000), Lisbon (25,000) and Metropolis, and compared the performance of three different reservation enforcement mechanisms: (1) vehicle locking, (2) R-BR with a constant QoS (r=200m, h=600 min) and (3) R-BR with variable QoS. The ILS algorithm parameters were set as in Table 3 and computation times can be checked in Table 9. The results in Figure 11 show profit variation depending on the reservation percentage with each of the operation configurations. It is evident that R-BR always outperforms vehicle locking and that optimized solutions with variable QoS always outperform both other techniques.

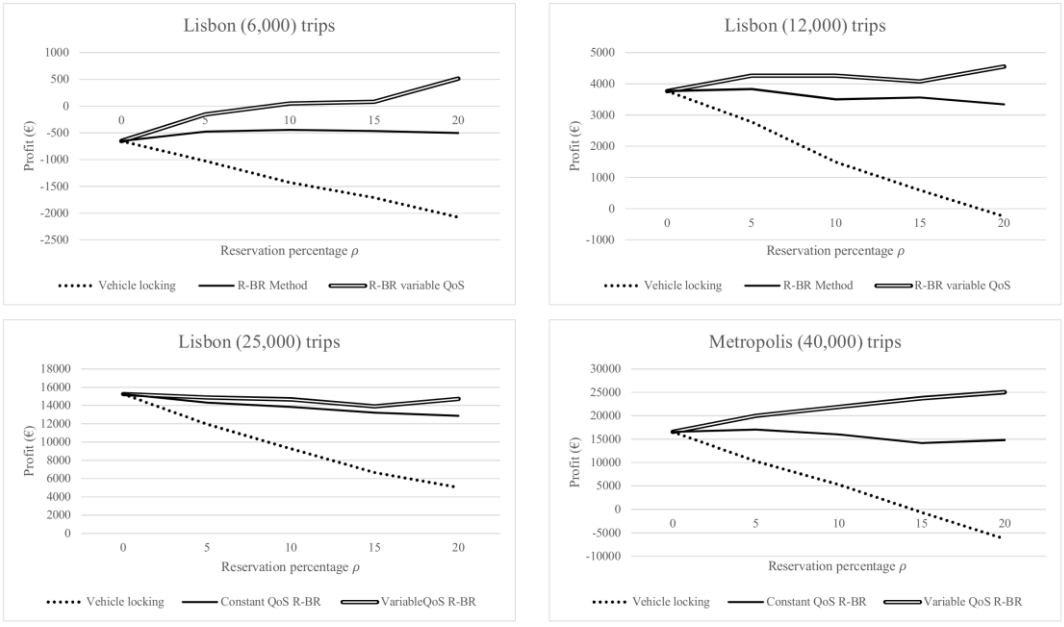


Figure 11: Profit comparison for low reservation percentages

Furthermore, comparison of variable QoS with balancing, given in Table 12 indicates that for each problem instance except Lisbon (25,000), the variable QoS with reservations achieves better profit than having no reservations. When reservation percentages are very low (< 5%), the profits become approximately equal to the base case. The Lisbon (6,000) variable QoS performance case is especially important as it demonstrates that, provided sufficient reservation trips, the optimization method can turn a business struggling with losses into a profitable one.

Table 12: Comparison of R-BR and balancing relocations

	Profit (€) with no reservations		**Reservations profit (€) with variable QoS R-BR**			
			Reservation percentage ρ (%)			
	No optimizations	**Balancing**	**5**	**10**	**15**	**20**
Lisbon (6,000)	-652.96	-652.96	**-155.31**	**51.98**	**82.206**	**513.08**
Lisbon (12,000)	3,767.44	3,936.03	**4,259.42**	**4,260.87**	**4,077.65**	**4,551.93**
Lisbon (25,000)	15,249.54	**17,447.98**	14,875.58	14,673.76	13,904.72	14,721.02
Metropolis	16,573.85	16,573.85	**20,010.56**	**21,895.06**	**23,753.08**	**25,047.84**

6.7. Practical considerations

We recommended gradual implementation in the existing carsharing systems. Before implementation, a rigorous viability assessment of reservations is required, since long-term reservations are not well suited to, e.g., carsharing systems with low demand or small fleets. Reservations have a clearly defined market: airport trips, intermodal trips connecting users to trains, buses, and other modes with a fixed schedule, users who need a temporary replacement car for everyday commute, people who do not own a car and might want to use carsharing to go to work during a week of bad weather etc. Therefore, a survey of user preferences to assess the market size for the long-term reservation service should be performed prior to implementing R-BR to help decide if the revenue increase from such a service would justify the change in operations.

The gradual implementation of the R-BR methodology can be performed in the following four steps:

1. Setting up the constant service quality reservation system,
2. Experimenting with improving the key performance indicators (profit, satisfied demand, user satisfaction) using the ILS metaheuristic,
3. Evaluating the real-world performance and repeating steps 2 and 3.

The optimization in step 2 requires a demand database or a forecast. Further, the ILS algorithm in a simulation-optimization approach is not fast enough for real-time decisions so it would typically be used for long-term iterative planning, on, e.g., weekly or monthly basis. Depending on the yearly and daily variations in demand, it is advisable to create several separate setups for typical working day vs. a weekend, summer months, and other specific circumstances. We recommend experimenting with several different solutions before choosing the final one, while carefully monitoring the service quality KPIs in the real carsharing system.

6.8. Experiments' summary

We summarize the results of all experiments performed in this work in Table 13, where a list of tested reservation enforcement methods and a list of problem instances that were part of the experiments are given. The table further provides a short description of the observed results as well as the Section where the experiment is described in more detail. We use "VL" as an abbreviation for vehicle locking method, "all" for experiments in which all 6 of our problem instances described were tested, and "sufficient demand" for all except Town (500) and Lisbon (3,000).

Table 13: Experiments summary

Experiment	Methods	Problem instances	Section	Result
Varying ρ with constant QoS	VL, R-BR	all	6.1	R-BR achieves higher profit than VL except in Town (500) with high ρ
Varying *h* with constant QoS with vehicle locking	VL	Lisbon (25,000)	6.1	Vehicle locking achieves higher profit with less accepted reservation demand when *h* is small
Varying ρ with different combinations of constant QoS	R-BR	Lisbon (12,000)	6.1	Variations in *r* and *h* have a large impact on profit and demand
R-BR and Fleet size	R-BR	Metropolis (40,000)	6.2	Investments in additional vehicles that are not reasonable with no reservations can cause substantial profit improvement with reservations
Balancing relocations	R-BR	sufficient demand	6.3	With enough trips and low reservation percentage, simple balancing strategies help improve profit

Variable QoS (balanced)	R-BR	all	6.4	ILS can improve the profit without lowering the QoS and the demand too much, according to the desired goals
Variable QoS (profit oriented)	R-BR	Lisbon (12,000)	6.4	ILS can focus on profit only if the operator desires to do so
Performance comparisons	VL, R-BR	sufficient demand	6.6	Variable QoS has the potential to produce a better profit than with no reservations

7. Conclusions and Future Work

Carsharing systems are being classified as a sustainable green mode, especially if provided with electric vehicle fleets. Nevertheless, one-way systems, in which the user may drop-off the vehicle anywhere inside a service area, are still being used by only a small segment of the urban transport demand. As the number of users grows, so do the logistic management problems to be solved if the quality of service and profitability are to be maintained.

Vehicle reservations provided through simple vehicle locking, by which the vehicle must stay idle until the client comes, have a too high impact on the profit to be viable as we showed in this paper. While we are not aware of any research which estimates the potential of reservations to attract more customers, we believe that customers do not favor highly restricted versions of reservations provided by commercial carsharing providers. Services such as restaurants, theatres, and hotels allow flexible reservation options, however, one-way carsharing reservations are currently mostly limited to very short time or are at best, very expensive. As the carsharing services do not control the trips their clients will be doing, additional information about the demand gathered from the reservations is not very valuable without a reliable and sustainable way to ensure that a reserved vehicle will indeed be in the correct place at the proper time.

In this paper, we have proposed an innovative relocations-based reservations (R-BR) method in which vehicles are only locked sometime before the beginning of the trip, and if a vehicle is not naturally available, one will be relocated. We hypothesized that this method could perform well even with much higher reservation times ahead than the commercial carsharing providers offer nowadays (typically 30 min). Furthermore, we proposed optimizing the allowed reservation time ahead and the proximity of reserved vehicles to users in different areas of the city to tailor the system to the demand profile.

To test the performance of both types of reservations (with and without the variable quality of service), we have created a simulator that enables evaluating various service configurations related to reservations. Several test instances have been created with daily carsharing trips in typical cities of various sizes. Our instances include the trips in two artificial

cities and a set of experiments with the case-study city of Lisbon (Portugal) where long reservation times are allowed. Results show that the vehicle locking strategy with long reservation times gives bad results for anything but a very low number of reservations (up to 20% of the total trip number). Furthermore, we show that the relocations method we propose is able to keep the system profitable with up to 60% reservations in the Lisbon (25,000 trips) example. The R-BR method achieved better results than locking in all problem instances except the Town (500) with a very low number of trips.

Unprofitable results for low trip volumes are in line with similar conclusions by other researchers: small towns are not well suited for commercial carsharing services. In such places, the revenues are too low to allow the successful operation of commercial providers, and the successful examples are typically restricted to the volunteer-based community services (European Commission, 2009). Conversely, big cities have shown to be very suitable for carsharing as it is much easier to sustain a company with a high concentration of the demand in areas with high population density and high revenue from a lot of daily trips.

As a guideline to operators interested in implementing reservations, if the system is not a profitable one without reservations, offering them may be risky. While this method is able to increase the customer satisfaction and the profit of already successful carsharing enterprises, in systems that are not profitable, the gains from adding reservations will most likely be very low to none, and with low trip volume, it might even cause profit decreases. The sensitivity to the vehicle fleet size experiments indicates that investments in additional vehicles can be useful when R-BR is applied even in cases when it shows little benefits in traditional systems where no reservations are allowed.

The implemented ILS metaheuristic was able to perform complex trade-offs of increasing the profit without lowering the service quality and the demand too much. It is able to learn from the performed daily trips and successfully identify areas with the highest demand, where QoS adjustments bring the most benefits. In general, increasing the r in the areas with the highest demand has shown to be a very effective tool that ILS used to lower relocation costs and increase profit. Further, lowering h in the areas with the highest demand can help prevent system overload and high costs of reliance on outsourcing. The flexibility of adjusting the relative importance of multiple optimization criteria allows further adjustments to the current goals of the operator.

While the objectives of this work were achieved using the above experiments, several refinements to the methodology are possible. We believe that an interesting research direction would be giving more attention to adding more realism to reservation time distribution and investing more computational resources to simulate longer periods of time, as well as supporting more realistic relocations, e.g., for companies with permanent staff in charge of relocations.

Another interesting direction is investigating the performance of reservations when used with various different balancing mechanisms.

In this paper, we do not consider added demand as a result of introducing the reservation system. A more detailed demand model that adjusts the demand with the QoS changes would further improve the accuracy of the obtained results and extend the applicability of the method to allow larger variance in the QoS. This approach would also allow users of the model to view more precise profit effects of each optimization variable that has been changed. Better integration of the effects of the QoS variation into the profit calculation is especially suitable for businesses, which are typically concerned with profit maximization as their top priority. Such demand models are usually non-linear mode choice models, nevertheless, they are suitable for integration into the simulation model, which is another advantage of the simulation-based optimization approach.

Acknowledgments

The authors would like to thank the Portuguese Science Foundation for financing the scholarship TRA/0528/2012 under which this work has been developed. The work was also supported by the InnoVshare project, financed as well by the Portuguese Science Foundation (Grant PTDC/ECM-TRA/0528/2012).

REFERENCES

An, K., & Lo, H. K. (2014). Ferry service network design with stochastic demand under user equilibrium flows. Transportation Research Part B: Methodological, 66, 70-89.

Angelopoulos, A., Gavalas, D., Konstantopoulos, C., Kypriadis, D., & Pantziou, G. (2018). Incentivized vehicle relocation in vehicle sharing systems. Transportation Research Part C: Emerging Technologies, 97, 175-193.

Boyacı, B., Zografos, K. G., & Geroliminis, N. (2015). An optimization framework for the development of efficient one-way car-sharing systems. European Journal of Operational Research, 240(3), 718-733.

Câmara Municipal de Lisboa, 2005. Lisboa: O desafio da mobilidade. Câmara Municipal de Lisboa - Licenciamento Urbanístico e Planeamento Urbano

car2go, http://www.car2go.com/, Accessed December 28, 2017.

Carlier, J. (1982). The one-machine sequencing problem. European Journal of Operational Research, 11(1), 42-47.

Celsor, C., & Millard-Ball, A. (2007). Where does carsharing work? Using geographic information systems to assess market potential. Transportation Research Record, 1992(1), 61-69.

Copeland, D. G., & McKenney, J. L. (1988). Airline reservations systems: lessons from history. MIS quarterly, 353-370.

Correia, G. (2009). Carpooling and carpool clubs: Clarifying concepts and assessing value enhancement possibilities. PhD in Transportation, Department of Civil Engineering, Instituto Superior Técnico.

Correia, G., & Viegas, J. M. (2011). Carpooling and carpool clubs: Clarifying concepts and assessing value enhancement possibilities through a Stated Preference web survey in Lisbon, Portugal. Transportation Research Part A: Policy and Practice, 45(2), 81-90.

Correia, G., & Antunes, A. P. (2012). Optimization approach to depot location and trip selection in one-way carsharing systems. Transportation Research Part E: Logistics and Transportation Review, 48(1), 233-247.

Correia, G., Jorge, D. R., & Antunes, D. M. (2014). The added value of accounting for users' flexibility and information on the potential of a station-based one-way car-sharing system: An application in Lisbon, Portugal. Journal of Intelligent Transportation Systems, 18(3), 299-308.

Correia, G., Jorge, D. R., & Antunes, D. M. (2014). The added value of accounting for users' flexibility and information on the potential of a station-based one-way car-sharing system: An application in Lisbon, Portugal. Journal of Intelligent Transportation Systems, 18(3), 299-308.

Chiarandini, M., & Stützle, T. (2002, September). An application of iterated local search to graph coloring problem. In Proceedings of the Computational Symposium on Graph Coloring and its Generalizations (pp. 112-125).

Deng, Y., & Cardin, M. A. (2018). Integrating operational decisions into the planning of one-way vehicle-sharing systems under uncertainty. Transportation Research Part C: Emerging Technologies, 86, 407-424.

Di Febbraro, A., Sacco, N., & Saeednia, M. (2012). One-way carsharing: solving the relocation problem. Transportation research record, 2319(1), 113-120.

Dorigo, M., & Birattari, M. (2011). Ant colony optimization. In Encyclopedia of machine learning (pp. 36-39). Springer, Boston, MA.DriveNow, https://de.drive-now.com/, accessed December 28, 2017.

DriveNow, https://www.drive-now.com/de/en/pricing/, accessed October 1, 2018

Emel Lisboa, http://www.emel.pt/pt/onde-estacionar/via-publica/tarifarios/, (in Portuguese), accessed October 1, 2018.

Enjoy, https://enjoy.eni.com/, accessed December 28, 2017.

European Commission, (2009), F09: Car-sharing in Small Cities, fact sheet. More Options for Energy Efficient Mobility through Car-Sharing (MOMO CAR-SHARING) project

Glover, F. W., & Kochenberger, G. A. (Eds.). (2006). Handbook of metaheuristics (Vol. 57). Springer Science & Business Media.

Hancock, P. J. (1994, April). An empirical comparison of selection methods in evolutionary algorithms. In AISB Workshop on Evolutionary Computing (pp. 80-94). Springer, Berlin, Heidelberg.

Hashimoto, H., Yagiura, M., & Ibaraki, T. (2008). An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. Discrete Optimization, 5(2), 434-456.

He, F., Wang, X., Lin, X., & Tang, X. (2018). Pricing and penalty/compensation strategies of a taxi-hailing platform. Transportation Research Part C: Emerging Technologies, 86, 263-279.

Huang, K., Correia, G., & An, K. (2018). Solving the station-based one-way carsharing network planning problem with relocations and non-linear demand. Transportation Research Part C: Emerging Technologies, 90, 1-17.

Hu, L., & Liu, Y. (2016). Joint design of parking capacities and fleet size for one-way station-based carsharing systems with road congestion constraints. Transportation Research Part B: Methodological, 93, 268-299.

INTERFILE, http://excel.interfile.de/Autokostenrechner/autokostenrechner.html, Accessed Jan 15, 2016.

Johnson, D. S., Papadimitriou, C. H., & Yannakakis, M. (1988). How easy is local search?. Journal of computer and system sciences, 37(1), 79-100.

Jorge, D., & Correia, G. (2013). Carsharing systems demand estimation and defined operations: a literature review. European Journal of Transport and Infrastructure Research, 13(3), 201-220.

Jorge, D., Correia, G., & Barnhart, C. (2014). Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. IEEE Transactions on Intelligent Transportation Systems, 15(4), 1667-1675.

Jorge, D., Molnar, G., & Correia, G., (2015). Trip pricing of one-way station-based carsharing networks with zone and time of day price variations. Transportation Research Part B: Methodological, 81, 461-482.

Kaspi, M., Raviv, T., & Tzur, M. (2014). Parking reservation policies in one-way vehicle sharing systems. Transportation Research Part B: Methodological, 62, 35-50.

Kaspi, M., Raviv, T., Tzur, M., & Galili, H. (2016). Regulating vehicle sharing systems through parking reservation policies: Analysis and performance bounds. European Journal of Operational Research, 251(3), 969-987.

Katayama, K., & Narihisa, H. (1999, July). Iterated local search approach using genetic transformation to the traveling salesman problem. In Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1 (pp. 321-328). Morgan Kaufmann Publishers Inc..

Kek, A. G., Cheu, R. L., Meng, Q., & Fung, C. H. (2009). A decision support system for vehicle relocation operations in carsharing systems. Transportation Research Part E: Logistics and Transportation Review, 45(1), 149-158.

Klintman, M. (1998). Between the private and the public: Formal carsharing as part of a sustainable traffic system. KFB-MEDDELANDE, (1998_2).

Litman, T. (2000). Evaluating carsharing benefits. Transportation Research Record: Journal of the Transportation Research Board, (1702), 31-35.

Lopes, M. M., Martinez, L. M., & Correia, G., (2014). Simulating carsharing operations through agent-based modelling: an application to the city of Lisbon, Portugal. Transportation Research Procedia, 3, 828-837.

Lourenço, H. R., Martin, O., & Stützle, T. (2001, July). A beginner's introduction to iterated local search. In Proceedings of MIC (Vol. 2, pp. 1-6).

Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). Iterated local search. In Handbook of metaheuristics (pp. 320-353). Springer, Boston, MA.

Lu, C. C., Yan, S., & Huang, Y. W. (2018). Optimal scheduling of a taxi fleet with mixed electric and gasoline vehicles to service advance reservations. Transportation Research Part C: Emerging Technologies, 93, 479-500.

Luke, S. Essentials of Metaheuristics. Lulu, 2013. Available for free at http://cs. gmu. edu/~ sean/book/metaheuristics.

Martínez, L. M., Correia, G., Moura, F., & Mendes Lopes, M. (2017). Insights into carsharing demand dynamics: outputs of an agent-based model application to Lisbon, Portugal. International Journal of Sustainable Transportation, 11(2), 148-159.

Namazu, M., & Dowlatabadi, H. (2018). Vehicle ownership reduction: A comparison of one-way and two-way carsharing systems. Transport Policy, 64, 38-50.

Nourinejad, M., & Roorda, M. J. (2014). A dynamic carsharing decision support system. Transportation research part E: logistics and transportation review, 66, 36-50.

Numbeo, https://www.numbeo.com/taxi-fare/in/Lisbon, Accessed October 1, 2018.

Ortúzar, J., Willumsen, L.G., 2011, Modelling Transport, John Wiley & Sons, Chichester, United Kingdom

Rotaris, L., & Danielis, R. (2018). The role for carsharing in medium to small-sized towns and in less-densely populated rural areas. Transportation Research Part A: Policy and Practice, 115, 49-62.

Schuster, T., Byrne, J., Corbett, J., & Schreuder, Y. (2005). Assessing the potential extent of carsharing: A new method and its implications. Transportation Research Record: Journal of the Transportation Research Board, (1927), 174-181.

Shaheen, S., Sperling, D., Wagner., C., 1999. A Short History of Carsharing in the 90's. Journal of World Transport Policy and Practice 5(3), 16-37.

Sinha, K., Labi S., 2007, Transportation Decision Making: principles of project evaluation and programming, John Wiley & Sons, Chichester, United Kingdom

Stützle, T. (2006). Iterated local search for the quadratic assignment problem. European Journal of Operational Research, 174(3), 1519-1539.

Stützle, T., & Hoos, H. H. (1999). Analyzing the run-time behaviour of iterated local search for the TSP. In III Metaheuristics International Conference.

Vasconcelos, A. S., Martinez, L. M., Correia, G. H., Guimarães, D. C., & Farias, T. L. (2017). Environmental and financial impacts of adopting alternative vehicle technologies and relocation strategies in station-based one-way carsharing: An application in the city of Lisbon, Portugal. Transportation Research Part D: Transport and Environment, 57, 350-362.

Viegas, J. M., & Martínez, L. M. (2010). Generating the universe of urban trips from a mobility survey sample with minimum recourse to behavioural assumptions. In Proceedings of the 12th World Conference on Transport Research.

Wang, H., & Cheu, R. (2013). Operations of a taxi fleet for advance reservations using electric vehicles and charging stations. Transportation Research Record: Journal of the Transportation Research Board, (2352), 1-10.

Weikl, S., & Bogenberger, K. (2013). Relocation strategies and algorithms for free-floating car sharing systems. IEEE Intelligent Transportation Systems Magazine, 5(4), 100-111.

Weikl, S., Bogenberger, K., & Geroliminis, N. (2016). Simulation Framework for Proactive Relocation Strategies in Free-Floating Carsharing Systems. In Transportation Research Board 95th Annual Meeting (No. 16-2725).

Xu, M., Meng, Q., & Liu, Z. (2018). Electric vehicle fleet size and trip pricing for one-way carsharing services considering vehicle relocation and personnel assignment. Transportation Research Part B: Methodological, 111, 60-82.

Yang, H., Fung, C. S., Wong, K. I., & Wong, S. C. (2010). Nonlinear pricing of taxi services. Transportation Research Part A: Policy and Practice, 44(5), 337-348.

ZipCar OneWay Car Sharing, http://www.zipcar.com/oneway, accessed December 28, 2017.