EURO Journal on Transportation and Logistics 12 (2023) 100110

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.elsevier.com/locate/ejtl

The benefit of complete trip information in free-floating carsharing systems

Claudia Archetti [a,*], Maurizio Bruglieri [b], Gianfranco Guastaroba [c], M. Grazia Speranza [c]

[a] ESSEC Business School, Cergy-Pontoise, France
[b] Department of Design, Politecnico di Milano, Via G. Durando 10, Milan, Italy
[c] Department of Economics and Management, University of Brescia, C.da S. Chiara 50, Brescia, Italy

ARTICLE INFO

Keywords:
Free-floating carsharing
Anticipation of information
Mathematical programming
Simulation study

ABSTRACT

Free-floating and instant-access carsharing systems are two features of the most flexible carsharing systems. In the former, users are allowed to freely park the car in any legal parking spot within the boundaries defined by the service operator. In the latter, users are not requested to make any reservation in advance before picking up the car. This paper aims at evaluating the importance of complete trip information in free-floating instant-access carsharing systems. To this aim, we consider a system, referred to as the look-ahead system, where users can reserve a car in advance and are also allowed to directly pick up a car without any reservation. In both cases, the user is requested to specify complete trip details, including the estimated usage duration and the location where the car will be returned. Taking advantage of a complete knowledge of the trip details, the service operator can assign a reservation request to a car that is in use at the time of reservation, provided it will become available at the time and location the user will need it. In its nature, the operational setting we consider is dynamic, as trip information is revealed to the service operator at the time the user requests a car. We also investigate the possibility of suggesting to the user that makes a reservation a pickup location different from the desired one, provided it is not too distant from the latter. We compare the performance of the look-ahead system with the case no information is anticipated, which resembles the service provided by the main carsharing operators currently active in Milan (Italy). Additionally, we use, as a benchmark, the static case where all the information about the users requests (pickup and return locations, as well as delivery time) is known before the start of the planning horizon. We consider a system with no car relocations performed by ad-hoc operators. This enables us to measure the pure benefit of anticipated information, i.e., the benefit coming only from knowing in advance where and when the vehicles will be returned, purged from potential vehicles availability related to relocation operations. The matching between user requests and cars is obtained, for the look-ahead system, iteratively at fixed time intervals through the solution of a Binary Linear Program (BLP) and, for the benchmark case, through the solution of a single BLP. No optimization model is needed for the case where no information is anticipated. A simulation study, based on real-world data from the city of Milan, shows that the look-ahead system can satisfy a number of requests much greater than the case without anticipated information and close to the benchmark case. Moreover, we perform a sensitivity analysis on different parameters, including the maximum distance a user is available to walk and the minimum amount of anticipation requested to users for booking requests, showing their impact on the performance of the systems.

1. Introduction

Several studies indicate that a large fraction of the population does not use, or uses very rarely, public transport, preferring instead to use a private car instead. As an example, according to a study conducted in 2013 on modal split for passenger transports in Europe (see Eurostat, 2017), private cars accounted for 83.2% of inland passenger transports in the EU, whereas motor coaches, buses and trolley buses (9.2%) and passenger trains (7.6%) both accounting for less than a tenth of all traffic. The motivations of such low usage rates are mainly related to the inability of traditional public transportation systems to satisfy the mobility needs of most users. The limits that are most often raised towards public transportation systems are: lack of flexibility, frequent delays, long detours, and lack of comfort. On the other hand, public transportation systems are known to be inefficient: overcrowded during rush hours and under-used in off-peak hours (see Chapuis et al., 2020). All the above reasons stimulated the design and implementation of

* Corresponding author.
E-mail addresses: archetti@essec.edu (C. Archetti), maurizio.bruglieri@polimi.it (M. Bruglieri), gianfranco.guastaroba@unibs.it (G. Guastaroba), grazia.speranza@unibs.it (M.G. Speranza).

https://doi.org/10.1016/j.ejtl.2023.100110
Received 28 July 2022; Received in revised form 17 March 2023; Accepted 30 May 2023
Available online 5 June 2023
2192-4376/

new mobility systems, which aim at attracting users by offering better and more flexible services. *Carsharing systems* are among those that report a high growing rate (see Shaheen et al., 2015; Golalikhani et al., 2021a). The number of cities where at least one carsharing system is available is constantly trending upwards, as well as the number of users. The success of carsharing systems can be explained by multiple factors. Firstly, they enable users to travel directly to the desired destination, thus avoiding the delays experienced with public transportation systems with fixed itineraries and frequencies. Moreover, users have access to cars without several of the responsibilities and costs associated with car ownership, including purchase and insurance costs. A recent survey conducted in Italy (see YouGov Italy, 2020) shows that only 57% of car owners use their vehicle almost every day, whereas the remaining 43% use it 2–3 times per week or less frequently. Thus, the fixed costs associated with car ownership are hardly amortized. Some important benefits are also frequently offered to carsharing users, such as the opportunity to travel in restricted traffic zones or to park in dedicated areas, often free of charge. In fact, when using private cars, the time required to find a parking spot close to the destination may be remarkable. Several studies have shown that this time is not negligible, and contributes substantially to the overall travel time. An overview regarding this issue is given in Shoup (2006). A more recent survey by Pasaoglu Kilanc et al. (2012) reports that the average driving time was less than one fourth of the average parking time in different European countries.

Modern carsharing systems can be classified according to different criteria (see Chun et al., 2019). The classification in two-way (also called round-trip) and one-way systems is related to the return location. In the former systems, users are required to begin (by picking up the car) and end (by returning the car) their trip at the very same location. In the latter systems, users are allowed to begin and end their trip at two different locations. Typically, this produces an unbalanced distribution of cars over the operational area: at some periods of the day some areas show an excess of cars available, whereas some other areas experience a shortage of cars. This is one of the major drawbacks of one-way systems, due to the costs associated with the operations needed to relocate cars among the different areas according to their demand. A second classification is related to the degree of freedom given to the users in terms of parking the car. In station-based (or stationary) services, users are required to park the car at one of the designated parking stations. In other systems, users are allowed to freely park the car at any legal parking location within the geographical boundaries determined by the operator. One-way systems where cars can be freely parked are called *free-floating carsharing systems*. As far as the service model is concerned, carsharing systems can be classified into two main categories (see Alfian et al., 2015): *reservation-based* and *instant-access* systems. Reservation-based systems require customers to make reservations in advance and within a given deadline (e.g., one day before the planned vehicle usage). This type of service model is considered, among others, in Barth and Todd (2000), Zhao et al. (2018); and Gambella et al. (2018). Conversely, instant-access systems (also called last-minute reservation systems) do not require any reservation. In other words, if the car is available where requested, the user has instant access to it. In between, there are carsharing systems with *real-time reservations* (e.g., addressed in Folkestad et al., 2020), and those with open-ended reservations (i.e., without a specific start or end of the reservation period, as in Cepolina and Farina, 2014). Carsharing systems with real-time reservations are particularly relevant to our research. In these systems, users submit their request (through a mobile app or a web-based application) and receive immediately (i.e., in real-time) or within a very short time the notification of whether it has been accepted or rejected. The recent trend in carsharing systems is to provide users with the maximum service flexibility, and free-floating and instant-access systems are two examples of such trend. However, such flexibility provided to the users entails an increased managerial complexity for the service operators.

In this paper, we consider a carsharing service, which we call *look-ahead carsharing system*, having the following key features. The service type is free-floating, whereas the reservation process allows both instant-access and real-time reservations. In the latter case, a user is allowed to place a reservation even if no car is available at the desired pickup location at the time of booking. The underlying idea is that it is possible to assign to a user a car that is in use at the time of reservation, but will become available later at the time the user will need it. Clearly, this matching is only possible if the service operator knows beforehand when and where the cars will be returned. In more detail, a user is allowed to directly pick up a car if available at the time and location desired (i.e., instant-access). In this case, we assume that, in order to start the engine, the user must indicate where the car will be returned, along with an estimation of the return time. If, instead, a user prefers to reserve a car in advance for future travel needs, a reservation can be made indicating a time window and a desired parking location for the pickup. As for the case of instant-access, the user will also provide the estimated return time and the return location. We observe that the impact on users of providing complete trip information beforehand is that their degree of freedom reduces since, as an example, they cannot change their final destination at the last minute or deliver the car rented later than the time initially scheduled (within a given tolerance). Nevertheless, the reduced flexibility of the service for the users enables to significantly improve the performance of the carsharing system, allowing to satisfy a higher number of requests, as shown in our experimental analysis. Such improvements are motivated by the observation that with complete trip information, a user's request can be satisfied with a car that is not available at the time of reservation, but that the operator knows will become available in the future.

The acceptance or rejection of the reservation is notified to the user in real-time or within a very short time frame (e.g., 5 min) by the service operator. The acceptance/rejection decision is taken based upon the information regarding the cars parked and those currently in use, as well as on other simultaneous reservations. Note that, since we consider a dynamic setting, the information regarding the trip of a user is revealed to the service operator at the time the user requests a car or has instant-access to a car. To increase the likelihood of accepting a reservation, users may be requested to walk for a certain distance to pick up the car at a location different from the desired one but not too distant. Similar to Correia and Antunes (2012), we assume that the repositioning of cars, if any, is carried out by the operator after the end of the planning horizon (i.e., when the carsharing system is not functioning), and is therefore not explicitly included in our analysis. This assumption is introduced to study the pure benefit of anticipated information, that is, to assure that the benefit can be directly attributed to the availability of complete trip information. In order to gauge the benefits of the look-ahead carsharing system, we perform a simulation study based on real-world data from the city of Milan (Italy). Given a set of Key Performance Indicators (KPIs), we compare the performance of the look-ahead carsharing system with a system, which we refer to as the *current system*, that mimicks the behavior of the main carsharing systems currently operating in the city of Milan, i.e., free-floating services with instant-access only. Hence, a user can pick up a car only if it is available at the time and location desired. Moreover, users are allowed to utilize cars without providing any kind of information about the trip. As a benchmark, we also consider the static case, which we call the *perfect info system*, where all the information about the users requests (pickup and return locations and time) is known to the service operator before the start of the planning horizon.

The main contributions of the paper are the following:

- this is the first paper to study the pure benefit of the look-ahead system purged from any additional element that may interfere with this analysis (such as vehicle relocation);

- a mathematical formulation of the problem of matching user requests and cars for the look-ahead system is presented, which turns out to be a Binary Linear Program (BLP), iteratively solved at fixed time intervals;
- a mathematical formulation of the problem of matching user requests and cars for the perfect info system is presented, which is a BLP for which we prove that an integer solution always exists to its continuous relaxation;
- a set of instances, obtained by simulation of a case study (Milan, Italy), are generated;
- a detailed comparison of the performance of the look-ahead system with that of the current system (instant-access) and of the perfect info system is provided.

The paper is structured as follows. In Section 2 we first provide a brief review of the foremost optimization problems studied in the literature on carsharing systems, and then focus on the papers that are closely related to our research. Section 3 details the look-ahead carsharing system and provides the mathematical programming formulations for the static and dynamic optimization problems, needed to simulate the look-ahead and the perfect info systems, respectively. The experimental analysis is presented in Section 4. Finally, in Section 5 we draw some conclusions and outline possible future research directions.

2. Carsharing systems: Literature review

Carsharing, as an organized form of shared use of a fleet of vehicles, can be dated back to 1948 in Zurich (Switzerland), where the service was born motivated by economic reasons (e.g., see Shaheen et al., 1999). Although in the following years other attempts to launch public carsharing systems were not successful, from the '80s several carsharing services were developed, stimulated by a growing eco-sustainable awareness of the citizens and a structurally well-organized offer of services. The ever increasing research attention on carsharing is demonstrated by the number of papers and surveys published on the topic. Among the latter, we mention Jorge and Correia (2013), Illgen and Höck (2019), Ferrero et al. (2018), Golalikhani et al. (2021a), and Golalikhani et al. (2021b). In the following, we focus on the literature that is most closely related to our research, and refer the interested reader to the above-mentioned surveys and the references cited therein for a broader overview of the contributions on the study of carsharing systems.

The survey by Golalikhani et al. (2021b) emphasizes a "gap of understanding" of the scientific community regarding the most common business practices. The authors analyze 34 business-to-consumer carsharing operators and, based on the data collected, they provide a detailed description of the current business practices, e.g., those concerning pricing and reservation policies. In particular, they observe that in some successfully implementations of carsharing systems it is mandatory for users to provide complete trip information upfront. Some examples mentioned in the survey are Cambio (https://www.cambio-carsharing.de/) and Drive-Now (https://www.drive-now.com/) in Berlin, Connectcar (https://www.connectcar.nl/) in Amsterdam, Communauto (https://www.communauto.com/) in Quebec, and CityHop (https://www.cityhop.co.nz/) in Auckland, to name a few.

Ferrero et al. (2018) analyze and classify 137 papers covering the last fifteen years of research on carsharing in the transportation and management literature, and identifies some mainstreams based on the different carsharing services and the research questions that emerged from the papers. Interestingly, the survey highlights an imbalance between the literature related to the operational level, on one hand, and the economic and business development aspects linked to the service models, on the other hand. Indeed, in the operations research literature many authors are concerned with the solution of operational problems, such as the aforementioned vehicle relocation problem in one-way carsharing services to balance demand and availability of cars in different parking locations (e.g., see the survey by Illgen and Höck, 2019). The relocation operations can be carried out by dedicated operators (*operator-based relocation*) as in Bruglieri et al. (2014, 2017, 2019), or by the users themselves (*user-based relocation*), who receive an economic incentive to return or pickup the car to a destination different from the desired one, as in Barth et al. (2004), or at a different time, as in Schiffer et al. (2021). According to Golalikhani et al. (2021a) there is a third type of approach to tackle the vehicle imbalance problem. This third approach aims at addressing the problem through trip selection, i.e., user requests are accepted or rejected considering the price and the impact of the trip on the performance of the system without dynamic relocation procedures. This approach is particularly relevant to our research, as in the considered carsharing service user requests are accepted or rejected by maximizing, through the solution of an optimization model, the number of requests accepted, without any dynamic relocation of cars. This line of research is relatively unexplored. As an example of its application, we mention the paper by Correia and Antunes (2012) where the authors develop an optimization approach to locate the stations of a one-way carsharing system, with a focus on a selection of trips aimed at maximizing the profit yielded by the service provider, while vehicle relocation is only considered at the end of the day when the system is not operating. Some strategic and tactical problems have also been studied, such as fleet sizing (see, among others, Barth and Todd, 1999; George and Xia, 2011; Cocca et al., 2019; Xu and Meng, 2019), optimal location of stations and their size (e.g., see Correia and Antunes, 2012; Nair and Miller-Hooks, 2014; Li et al., 2016), and staff sizing for relocation-maintenance operations (see, for example, Bruglieri et al., 2018).

We now review the literature on reservation-based and instant-access carsharing systems that is most relevant to our research. To the best of our knowledge, only few authors focus their analysis on the service features. Molnar and Correia (2019) study a long-term reservation mechanism for one-way free-floating carsharing systems. They assume that reservations are accepted no later than a given threshold before the pickup time. Such a threshold can be by far larger than the 30 min usually allowed by most carsharing companies. After the reservation is accepted, all vehicles in the system continue to be available as if no reservation has taken place until a given *response time*. In that moment, either a nearby vehicle of the carsharing fleet is locked to be assigned to the request, or a vehicle is relocated, or, if no vehicle is available, a taxi service is provided. We observe that a limit of this approach is that, when a request is accepted, the operator does not know the cost to satisfy such request, since it will become known only in a second phase how the request will be satisfied (a nearby vehicle, or a vehicle relocated, or a taxi service is provided). Conversely, in the current paper the requests are accepted with no need for vehicle relocation, or for using a taxi that imply extra costs for the carsharing operator. Alfian et al. (2015) evaluate the performance of different reservation systems of a one-way carsharing service. To this aim, the authors develop a simulation model of a carsharing service comparing reservation-based and instant-access one-way systems in terms of different patterns of user behavior. A similar study is proposed in Nourinejad and Roorda (2014), where a benchmark and a dynamic optimization-simulation model are defined and the performance of a one-way carsharing system is evaluated. The benchmark model is set up with complete knowledge of user requests, whereas the more realistic dynamic model receives user information only when requests are placed. The dynamic model performs satisfactorily when compared to the benchmark model. The study shows the importance of various policies, such as the extension of the required reservation time, on reducing the fleet size of the one-way carsharing systems considered. Increasing the reservation time from zero (i.e., an instant-access system) to 30 min can reduce the fleet size by 86% by keeping the same service level. However, although any increase in the reservation time can lead to lower fleet size and lower cost, it can also make the system less attractive to the users and lead, potentially, to lower demands. We stress that, compared to our research, none of the latter two papers assumes to know in advance the information on when and where the carsharing vehicle picked up by a user will be returned. In Cepolina and Farina (2014) a one-way station-based carsharing system with automated vehicles and both instant-access and open-ended reservations is considered. The authors determine the fleet size and the relocation strategy that minimize the system cost, taking into account service level and efficiency. Relocation strategy parameters (e.g., a threshold on the minimum number of vehicles in each station) define when and among which stations relocations should be performed. The optimization problem is solved by a simulated annealing algorithm. The authors observe that, since the vehicles considered are automated, the vehicle relocation is easy to be performed as it needs neither operators nor users. In Folkestad et al. (2020) a free-floating carsharing system with electric vehicles and both instant-access and real-time reservations is considered. Since the service is free-floating and the vehicles are electric, operators might need to relocate cars with depleted batteries to charging stations, as this operation is not necessarily carried out by users. This task is additional to car relocations pursuing a better distribution of the fleet. Moreover, unlike reservation-based carsharing systems, where operators have to relocate cars in order to fulfill binding reservations by customers, in this study operators are not responsible to fulfill reservations, but work to ensure a more balanced vehicle distribution.

Although several papers show the beneficial impact of digital communication and mobile technologies on carsharing systems, providing real-time information on the vehicles and helping users in locating the cars available, as in Hayashi et al. (2014) and Hildebrandt et al. (2015), only few papers exploit the real-time information. In this direction, Li and Petering (2018) propose a discrete event simulation model to analyze the operational and financial performance of a one-way station-based carsharing system with complete journey online reservations (i.e., the parking location and the time window are reserved not only for the pickup but also for the delivery). The system accepts the requests only if they can be feasibly accommodated without vehicle relocations. The experiments focus on showing how system performance depends on the total number of vehicles (or parking spaces) when the total number of parking spaces (or vehicles) is fixed, and also on both the allocation of parking spaces among stations and on the distribution of travel demand among different pairs of origins and destinations.

In this paper, we focus on studying the pure benefit of anticipating the information on when and where the vehicles will be returned in a free-floating carsharing system with both real-time reservations and instant-access. Note that with the expression 'pure benefit' we mean that, as we do not consider vehicle relocation, the benefit obtained can be directly attributed to the availability of complete trip information. In contrast with Li and Petering (2018), our analysis compares the performance of the *look-ahead carsharing system* with both an instant-access system, that we call the *current system*, and the *perfect info system*, as explained in the former section.

3. The look-ahead carsharing system

In this section, we present the *look-ahead carsharing system*, where a user is allowed to submit a request for picking up a car in a desired location (e.g., a circle around the GPS position of the user) and a certain time window, even if no car in that area is available at the time of reservation. In the latter case, the request will, hopefully, be satisfied by assigning a car currently in use. In fact, in the system under analysis, when submitting a request, users are requested to anticipate the information on when and where they intend to return the car. In this system, users are also allowed to have instant-access to a car, if it is available at the time and location desired. In this case, to start the engine the user must provide information regarding where the car will be returned, along with an estimate of the return time. Leveraging on this additional information, if a car is, or will become, available in the parking location associated with a user request, the request made is satisfied by assigning the corresponding car. In addition, a user is available to deviate from the desired pickup parking location, if this allows to satisfy the request. In other words, if a car is, or will become, available at a parking location within a maximum walking distance from a user-specified location, the request is satisfied by assigning the corresponding car. In case no such car exists – i.e., no car is, or will become, available within the desired time window and at the desired parking location or nearby – the request is rejected. The main goal is to maximize the number of satisfied requests.

In the following, we formally define the optimization problems solved to find an optimal matching of user requests and cars for the perfect info and for the look-ahead systems. We first consider the *static* case, i.e., the case in which the decision-maker has complete knowledge of the requests received over the entire planning horizon considered: the decision-maker knows beforehand all the booking requests, as well as when and where the users will return the cars rented. This is what is assumed in the perfect info system. The optimization problem is cast as a BLP and an interesting property of the corresponding optimal solution is proven. The mathematical formulation of the static case is subsequently modified to account for a *dynamic* setting, as in the look-ahead system.

3.1. Static problem: Description and mathematical formulation

This section presents a mathematical model for the static case. The formulation is based on the following notation. Let $[0,T]$ be a given planning horizon. We denote as P the set of parking locations considered by the system. The considered system is free-floating, that is, users are allowed to pick up and return a vehicle at any location within the geographical boundaries determined by the service operator. To capture the problem with a mathematical program, we discretized the space into a finite number of parking locations.

Let B be the set of booking requests. For each $b \in B$ the following information is given: the time window within which the user wants to pick up the car (hereafter denoted as $[e_b^{start}, l_b^{start}]$) at a desired parking location $p_b^{start} \in P$, an estimated usage duration d_b, and a chosen parking location for returning the car $p_b^{end} \in P$. Accordingly, a second time window, hereafter denoted as $[e_b^{end}, l_b^{end}]$, is associated with each request $b \in B$ defining an estimate of the interval of time within which the car will be returned if request b is satisfied, and is calculated as $e_b^{end} = e_b^{start} + d_b$ and $l_b^{end} = l_b^{start} + d_b$. We assume that $e_b^{end} \geq l_b^{start}$. Let w_b be the maximum distance that the user who made the booking request $b \in B$ is willing to walk to pick up the car. This means that request $b \in B$ can be satisfied by assigning it to a car parked in a parking location $\tilde{p}_b \in P$ different from p_b^{start}, provided that $dist(p_b^{start}, \tilde{p}_b) \leq w_b$, where $dist(\cdot,\cdot)$ denotes the walking distance between the two parking locations. However, to discourage such assignments when unnecessary, assigning a request $b \in B$ to a car parked in a parking location different from p_b^{start} is penalized in the objective function, and this penalization increases with the distance of the parking location $\tilde{p}_b$ assigned from the desired one.

Let C be the set of cars not in use and available to be picked up. Each car $c \in C$ is associated with parking location $p_c \in P$, where the car is stationing, and a time window of availability, denoted as $[e_c, l_c]$ and where $l_c \leq T$, i.e., the car cannot be picked up before the earliest time e_c and after the latest time l_c. The time window $[e_c, l_c]$ models the availability of a car, which may be due, for instance, to planned operations on the car (e.g., refueling, cleaning or small mechanical repairs).

Finally, let R be the set of cars being used, including those taken by users with instant-access, which will be returned within the end of the planning horizon T. For each rented car $r \in R$, we assume that the estimated time $\sigma_r < T$ for returning the car is known, as well as the parking location $p_r \in P$ where the car will be returned.

The optimization problem can be represented as follows. Let B' be a copy of set B and $U = C \cup R \cup B'$ be the set that contains all the cars

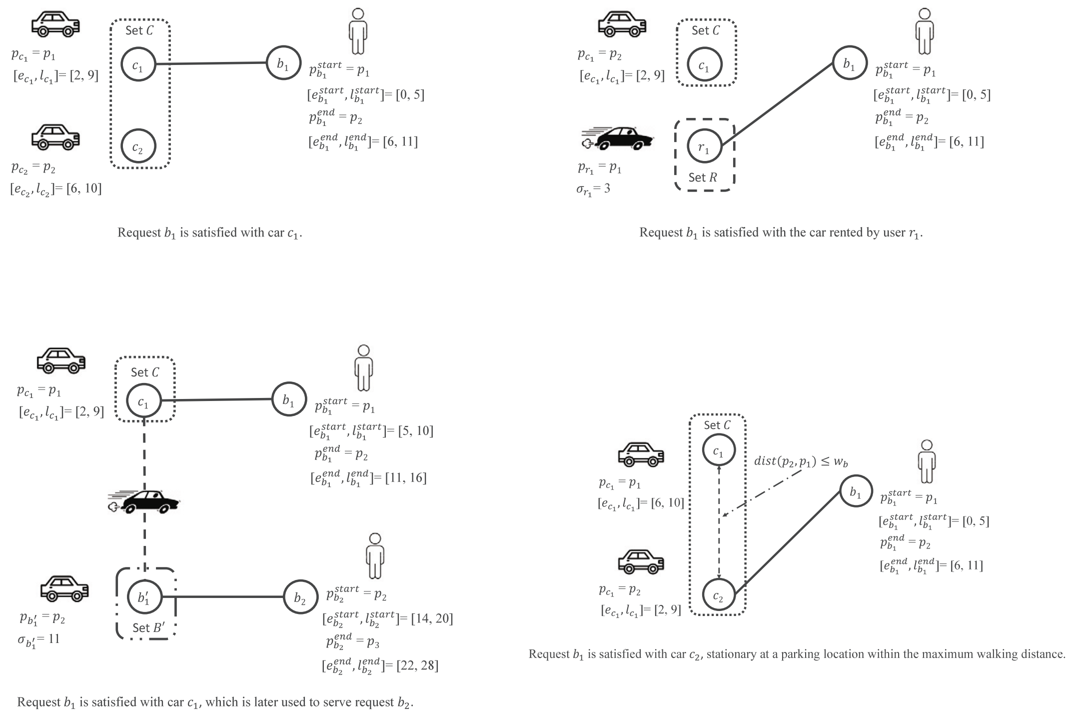

Fig. 1. Four examples of feasible assignments.

that may be used to satisfy the requests, that is the parked cars C, the cars in use R, and the cars that, if assigned to a request, may be later used to satisfy a different request when returned (i.e., set B'). Indeed, the latter set is introduced to allow that the same car is used to satisfy multiple requests. For each copy node $b' \in B'$, let $c(b') \in B$ be the corresponding node in set B. Node b' is associated with an estimated time of availability $\sigma_{b'}$ and with parking location $p_{b'} = p^{end}_{c(b')}$. In the experiments, we assume $\sigma_{b'} = e^{end}_{c(b')}$, i.e., we assume that the car used to satisfy request $c(b')$, if any, will be available at parking location $p_{b'}$ (corresponding to the destination location of request $c(b')$) at time $\sigma_{b'}$ corresponding to the earliest time in the time window within which the car assigned to request $c(b')$ is returned.

Let us consider a bipartite graph $G = (V, E)$, where $V = U \cup B$ is the set of nodes, and E is the set of edges. An edge $(u, b) \in E$ between $u \in U$ and $b \in B$ exists if and only if the following two conditions hold:

1. $[e_u, l_u] \cap [e^{start}_b, l^{start}_b] \neq \emptyset$ if $u \in C$; $\sigma_u \le l^{start}_b$ if $u \in R \cup B'$.
2. Parking location p_u is located within the maximum walking distance from p^{start}_b, i.e. $dist(p_u, p^{start}_b) \le w_b$.

In other words, an edge $(u, b) \in E$ exists in the bipartite graph G if and only if it is feasible to assign booking request $b \in B$ to a car $u \in U$. Some illustrative examples of feasible assignments are shown in Fig. 1.

A summary of the main mathematical notation used in the optimization models can be found in Table 1.

The mathematical formulation is based on binary variable $x_{ub} \in \{0, 1\}$, with $(u, b) \in E$, which takes value 1 if request $b \in B$ is satisfied with car $u \in U$, and 0 otherwise. The optimization problem can be cast as the following BLP:

[Static model]

$$\max \ z = \sum_{(u,b)\in E} x_{ub} - \lambda_1 \sum_{(u,b)\in E} dist(p^{start}_b, p_u) x_{ub} \tag{1}$$

$$\text{s.t.} \quad \sum_{u\in U \mid (u,b)\in E} x_{ub} \le 1 \qquad \forall b \in B \tag{2}$$

$$\sum_{b\in B \mid (u,b)\in E} x_{ub} \le 1 \qquad \forall u \in U \tag{3}$$

$$\sum_{b\in B \mid (b',b)\in E} x_{b'b} \le \sum_{u\in U \mid (u,c(b'))\in E} x_{uc(b')} \qquad \forall b' \in B' \tag{4}$$

$$x_{ub} \in \{0, 1\} \qquad (u, b) \in E. \tag{5}$$

The Static model has a hierarchical objective function. In fact, the objective function (1) comprises two terms, where the first term captures the primary objective, whereas the second term models the secondary goal. The primary objective, to be maximized, is the number of satisfied requests. The secondary objective, to be minimized, is the distance between the desired pickup parking location p^{start}_b and the assigned one p_u. The basic rationale of the secondary objective is the following. If there are two feasible assignments for a given request b, through this second term the optimization model will choose the car available at the parking location closest to the desired one, i.e., p^{start}_b. Hence, parameter λ_1 is a scaling factor set to a small value, namely $\lambda_1 = \frac{0.01}{\max_{(u,b)\in E}\{dist(p^{start}_b, p_u)\}}$. This value of λ_1 is small enough to guarantee that the first term of objective function (1) is prioritized and, when multiple solutions with the same value of the first term are found, the one with the smaller value of the second term is chosen. In fact, note that for a given $(u, b) \in E$ such that $x_{ub} = 1$, the corresponding term in the second sum of objective function (1) becomes $0.01 \frac{dist(p^{start}_b, p_u)}{\max_{(u,b)\in E}\{dist(p^{start}_b, p_u)\}}$ which is considerably smaller than 1. Constraints (2) ensure that each request is served at most once, whereas (3) impose that each car in set U is used to serve at most one request. Constraints (4) guarantee that a car associated with node b' cannot be used if the corresponding request $c(b')$ has not been satisfied. It is worth highlighting that the same car might be used to satisfy multiple requests in sequence. An example of a feasible assignment where the same car is assigned to three subsequent requests is depicted in Fig. 2. Finally, constraints (5) define the domain of the decision variables.

We now prove that the Linear Programming (LP) relaxation of formulation (1)–(5) always provides an optimal integer solution.

Theorem 1. *An optimal integer solution always exists for the LP relaxation of formulation (1)–(5), obtained by replacing (5) with* $x_{ub} \in [0, 1]$, $(u, b) \in E$.

Table 1
A summary of the main mathematical notation.

Sets		Parameters	
Notation	Description	Notation	Description
B	Set of booking requests	*Booking requests*	
B'	A copy of set B	d_b	Estimated usage duration
C	Set of cars available for pickup	$[e_b^{start}, l_b^{start}]$	Time window to pick up the car
E	Set of feasible edges	$[e_b^{end}, l_b^{end}]$	Estimated time window to return the car
P	Set of parking locations	p_b^{start}	Desired pickup parking location
R	Set of rented cars	p_b^{end}	Chosen destination parking location
U	Set of all cars to satisfy the requests	*Cars available for pickup*	
		$[e_c, l_c]$	Time window of availability
		p_c	Associated parking location
		Rented cars	
		σ_r	Estimated time the car will be returned
		p_r	Destination parking location
		Objective functions	
		λ_1	Scaling factor for the term minimizing the distance
		λ_2	Scaling factor for the term prioritizing the urgent requests

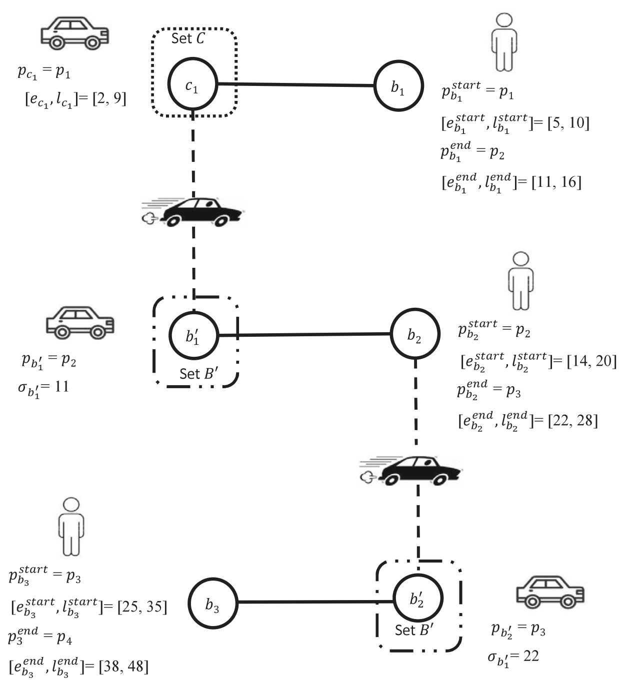

Fig. 2. Car c_1 is used to satisfy requests b_1, b_2, and b_3, sequentially.

Proof. Let us consider the coefficient matrix associated with constraints (2)–(4). Focusing first on constraints (2)–(3), one can notice that the corresponding matrix is Totally UniModular (TUM). In fact, it is the coefficient matrix of a classic assignment problem. Let us now consider constraints (4). Each variable appears in these constraints either once with coefficient −1 or at most twice, once with coefficient +1 and once with coefficient −1. Indeed, each constraint (4) is associated with a different value of b'. Thus, each variable $x_{b'b}$ appears at most once in the left-hand side, with coefficient +1. As for the right-hand side, each variable $x_{b'b}$ appears at most once, specifically, only in the case where $u = b'$ and $c(b') = b$. In this case the coefficient of the variable is −1. Thus, each variable $x_{b'b}$ appears at most twice with coefficient +1 and −1. As for the variables x_{ub} with $u \in U \backslash \{B'\}$, they appear at most once, at the corresponding value of b', with sign −1. Thus, the coefficient matrix for constraints (4) is TUM, as the sufficient condition for a matrix to be TUM is satisfied in this case (see Schrijver, 2003). Given that two adjacent TUM matrices form a TUM matrix (see, again, Schrijver, 2003), the coefficients of the variables in constraints (2)–(4) form a TUM matrix. Thus, the statement follows. □

3.2. Dynamic problem: Description and mathematical formulation

The Static model (1)–(5) is based on the assumption that all the information is available to the decision-maker upfront and is solved only once. We now consider the case where the information is revealed to the service operator dynamically over time. We discretize the planning horizon $[0, T]$ into a number I of equally spaced time intervals $[t_{i-1}, t_i]$, with $i = 1, \dots, I$, $t_0 = 0$, and $t_I = T$. At each time t_i, with $i = 1, \dots, I$ –that is, at the end of each time interval– the BLP detailed below is solved given the information available at that time instant. After each optimization, the acceptance of all the booking requests satisfied is notified to the respective user. The time interval $[t_{i-1}, t_i]$ defines the time between two consecutive optimizations. Note that the choice of this quantity is particularly important: the shorter the time interval, the sooner a user is notified if her/his request has been accepted. If a request b cannot be assigned when optimizing at time t_i, it will be considered also in the optimization models associated with following time instants t_j, $j > i$, up to when $t_j > l_b^{start}$. In the latter case, the request will be rejected and the user will be notified accordingly.

The following optimization model is based on sets B_i, C_i, R_i, B_i', and E_i. These sets have a similar meaning to the corresponding sets defined in Section 3.1, but are constructed according to the information available at time t_i. The mathematical formulation solved at time t_i is the following BLP:

[Dynamic model]

$$\max \quad z = \sum_{(u,b)\in E_i} x_{ub} - \lambda_1 \sum_{(u,b)\in E_i} dist(p_b^{start}, p_u) x_{ub} - \lambda_2 \sum_{(u,b)\in E_i} (l_b^{start} - t_i) x_{ub} \tag{6}$$

$$\text{s.t.} \quad \sum_{u\in U_i|(u,b)\in E_i} x_{ub} \le 1 \qquad \forall b \in B_i \tag{7}$$

$$\sum_{b\in B_i|(u,b)\in E_i} x_{ub} \le 1 \qquad \forall u \in U_i \tag{8}$$

$$\sum_{b\in B_i|(b',b)\in E_i} x_{b'b} \le \sum_{u\in U_i|(u,c(b'))\in E_i} x_{uc(b')} \qquad \forall b' \in B_i' \tag{9}$$

$$x_{ub} \in \{0, 1\} \qquad \forall (u, b) \in E_i. \tag{10}$$

Akin to the Static model, the Dynamic model has a hierarchical objective function. In contrast to the former model, the objective function

(6) comprises an additional third objective. The latter, to be minimized, represents the interval of time between the latest instant of time to pick up a car for request $b \in B$ (i.e., l_b^{start}) and the time of optimization t_i. Its basic rationale is as follows. Assume there are two feasible assignments of two different requests. The first request has to be satisfied within t_i, whereas the second can be satisfied within t_{i+1}. With the third term, the optimization model will prioritize the assignment of the request that is more urgent — i.e., the first request. Parameter λ_2 is set to $\frac{0.01}{\max_{b\in B}\{l_b^{start}-t_i\}}$ in the computational experiments. The meaning of constraints (7)–(10) is akin to the corresponding constraints in the Static model.

4. Experimental study

In this section, we present the experimental study we carried out in order to assess the benefits of the look-ahead carsharing system. In Section 4.1, we describe the data we used concerning the requests of cars made by the users. This data was simulated by exploiting real-world data for the city of Milan (Italy). The simulator was developed in MATLAB. In Section 4.2, we detail the carsharing systems considered in the evaluation, along with the statistics employed to evaluate their performance. The optimization models were implemented in Java, and the LP relaxations of the Static and Dynamic Models were solved by using the ILOG Concert Technology API (CPLEX version 12.10). The results of the experimental study are discussed in Section 4.3. All the experiments were conducted on a Workstation HP Intel(R)-Xeon(R), equipped with a 3.5 GHz 64-bit processor, 64 GB RAM, and Win 10 Pro as Operating System. The processor was equipped with 6 cores, but all tests were performed by using only one thread.

4.1. Experimental environment

We simulated the carsharing demand by exploiting the survey on the mobility of people conducted in the Milan area by the Agency for Mobility, Environment and Territory of the Municipality of Milan (AMAT). For administrative purposes, the city of Milan is divided into several census zones, where each zone has uniform environmental and socio-economic characteristics. The original data refers to private car trips from/to different census zones in Milan and is organized in an Origin–Destination (O–D) matrix. The trips are classified according to their scope (such as, business, study, and occasional) and three different time frames: morning (7:00 to 10:00; peak hours), off-peak (10:00 to 16:00), and evening (16:00 to 20:00; peak hours). We focused on the data describing the occasional trips, as they are the most representative of the carsharing users. Hence, from the original O–D matrix, we extracted the data concerning these occasional trips. To the sake of brevity, in the following the latter is simply referred to as the O–D matrix.

The simulator works as follows. The planning horizon $[0, T]$ is discretized in time slots, lasting 1 min each. For each census zone, we computed its centroid and the node of the road network closest to such centroid. Centroids and the corresponding nodes were determined by means of the ArcGIS software by ESRI (2021). Hereafter, we refer to the road network nodes as *centroid nodes*. We assume that set P of parking locations coincides with this set of centroid nodes. This assumption is motivated by the observation that in our case study the census zones are relatively small. Hence, the centroid nodes can be considered as good approximations of parking locations. In the simulation, we considered the census zones located within the ring road of Milan (see Fig. 3) since most of the carsharing trips occur in this area. Following this procedure, we obtained 105 census zones, whose centroid nodes are shown in Fig. 4.

At the beginning of the simulation, the cars composing the fleet are randomly positioned among the centroid nodes available. At the beginning of every time slot, the simulator generates random car trips

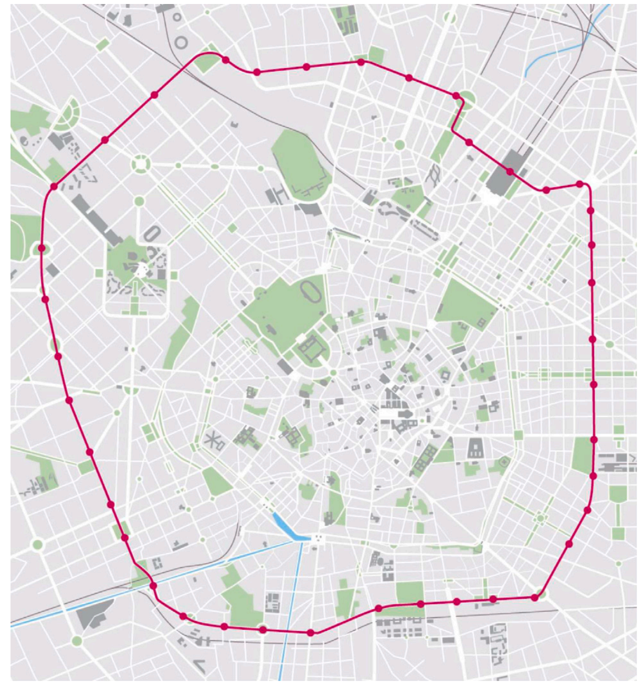

Fig. 3. Ring road (in red) in the city of Milan. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

between centroid nodes with a probability distribution that is consistent with both the O–D matrix and the fleet size. Then, according to the simulated trips, the availability of cars in each centroid node is updated and, consequently, set C. Consider the following example. Assume that car c is positioned in centroid node 50 from the beginning of the planning horizon until the beginning of time slot 35. As a consequence, the simulator sets $p_c = 50$ and its time window $[e_c, l_c] = [0, 35]$.

We now turn our attention to the cars moving among census zones. We assume that a random percentage of the simulated trips correspond to booking requests (i.e., of users that made a reservation) and, therefore, compose set B. The remaining trips are checked before inserting them in set R or B. Recall that trips in set R are associated with users that had instant-access to the cars. To ensure that these trips can be satisfied with the cars composing the fleet, the simulator checks for each trip if there is a car available at a compatible time and at the associated desired parking location (or within the maximum walking distance allowed). If this check is passed, the random trip is inserted in set R, otherwise it is inserted in set B.

For each booking request $b \in B$, p_b^{start} and p_b^{end} are the origin and the destination centroid nodes of the random trip associated with it. Similarly, for each $r \in R$, p_r is the destination centroid node of the associated random trip. The travel time between each pair of centroid nodes is estimated by computing the shortest path with ArcGIS, and then dividing the corresponding length by the average speed, as provided by AMAT, associated with the instant of time when the trip begins. Consequently, based on these travel times we determined the return time σ_r, for each $r \in R$, and the usage duration d_b, for each $b \in B$. For each booking request $b \in B$, the time window $[e_b^{start}, l_b^{start}]$ within which the user wants to pick up the car is determined by setting $e_b^{start} = t_b - \rho$ and $l_b^{start} = t_b + \rho$, where t_b is the instant of time in which the car associated with request b starts its trip in the simulator and ρ is a random parameter generated in the interval $[\frac{d_b}{4}, \frac{d_b}{2}]$. In this way, we can guarantee that the pickup and return time windows do not overlap, and that their width is at least $\frac{d_b}{2}$.

Our experimental study was conducted using 12 instances randomly generated by means of the simulator described above. The input parameters of the simulator, for each of the instances, are the following. The

Fig. 4. Centroid nodes (squares) located within the ring road in the city of Milan.

Table 2
The characteristics of some instance parameters.

	Min (min)	Mean (min)	Max (min)
Width of $[e_b^{start}, l_b^{start}]$	13.00	27.23	38.00
Width of $[e_c, l_c]$	5.00	98.20	179.00
d_b	15.00	27.04	61.00

length of the planning horizon T is 180 min, which is the length of the time frames corresponding to the morning and evening peak hours in the O–D matrix. 3 out of the 12 instances are generated according to the trips in the O–D matrix for the morning peak hours; 3 instances for those in the evening peak hours; and 6 instances for those that occurred during the off-peak hours. The fleet of the carsharing systems simulated comprises 50 cars. In our testbed, the average number of simulated trips is 125. On average, the percentage of the trips associated with booking requests is 74.4% (set B), whereas the remaining 25.6% trips are associated with users that had instant-access (set R).

Table 2 reports some statistics measuring the width of the pickup time window, in minutes, for the booking requests (i.e., $[e_b^{start}, l_b^{start}]$), the width of the time window, in minutes, of availability for the cars in set C (i.e., $[e_c, l_c]$), and the estimated usage duration d_b, in minutes. The columns show the minimum value (Min), the average value (Mean), and the maximum value (Max). The statistics are computed over all the generated instances. We do not report similar statistics for the remaining instance parameters as they can either be obtained from these values (like the width of the return time window $[e_b^{end}, l_b^{end}]$), or the parameter is part of the experiments where we conducted a sensitivity analysis on it (like w_b that, to the sake of simplicity, in the following is denoted as α).

4.2. Carsharing systems and performance indicators

In the experimental study discussed in the following, we compare three business models for carsharing systems operating according to different management policies. The first one, hereafter referred to as Current(α), is adopted by the main carsharing systems currently active in the city of Milan (e.g., Car2Go and Enjoy). The second system, henceforth called Look-Ahead(α/β), is the look-ahead carsharing system defined in Section 3. The third system, from now on referred to as PerfectInfo(α), is used as a benchmark for the other two. It is a reservation-based carsharing system where complete information about all trips over the entire planning horizon is known before time 0. Note that, in the figures presented in the following section, we refer to the three business models for carsharing systems simply as 'carsharing systems'.

The characteristics of the systems, together with the details regarding how they were simulated in the experiments, are summarized in the following.

A. Current(α): This is a free-floating instant-access car sharing system mimicking the most widespread and currently active systems in the city of Milan. In this system, a request is accepted only if, at the moment the request is placed, a car is available at the desired parking location, or within a maximum walking distance. In this system, no information regarding where and when the car will be returned is available to the service operator. We assume that the moment the request $b \in B$ is placed is e_b^{start}. As introduced in Section 3.1, parameter w_b refers to the maximum distance, in Kilometers (km), user b is available to walk to pick up a car, starting from the desired parking location. To the sake of simplicity, and following other authors (e.g., see Soppert et al., 2023), we assume that w_b has the same value for all users and, to simplify the notation, we denote it as α. Note that running this system does not give rise to any optimization problem. In fact, after sorting chronologically all the requests based on the moment each of them is placed, each request is analyzed one at a time to determine if it can be accepted or must be rejected. In case more than one car is available, the car located in the parking location closest to the one desired by the user is assigned. In case of ties, the car in C having the closest latest time l_c is assigned. In case of further ties, the car is assigned randomly. After accepting each request, the availability of the car assigned is updated accordingly;
B. Look-Ahead(α/β): This is the carsharing system considered in this paper: a free-floating carsharing system with both instant-access and real-time reservations, where users provide complete trip information before their journey starts. As above, α refers to the maximum distance (in km) a user is available to walk to pick up a car, whereas β represents the minimum amount of anticipation requested to users for booking requests. More precisely, a user is requested to place the reservation at least β and at most $\beta + \gamma$ minutes before the opening of the desired pickup time window. The moment a user places the request is then randomly generated in the interval $[e_b^{start} - (\beta + \gamma), e_b^{start} - \beta]$, with $b \in B$. To simulate this system, we discretized the planning horizon $[0, T]$ in a number I of time intervals $[t_{i-1}, t_i]$, all having a length of 5 min. Then, at each time t_i, with $i = 1, \ldots, I$, the LP relaxation of the Dynamic model (6)–(10) was solved. In the computational experiments, the value of γ was set to 5 min;
C. PerfectInfo(α): This is a free-floating reservation-based carsharing system where complete information about all trip requests is known to the service operator before the beginning of the planning horizon. Its solutions are obtained by solving the LP relaxation of the Static model (1)–(5) at time 0 over the entire planning horizon. These solutions are used to benchmark the performance of the former two carsharing systems. Again, α refers to the maximum distance (in km) a user is available to walk to pick up a car.

The performance of the above carsharing systems is assessed by computing two of the main KPIs used, in practice and in the scientific literature, to assess the performance of a carsharing system. Let RA be the number of requests accepted by each system, from set B as well as set R.

Table 3
Detailed performance of the three carsharing systems (base case).

Instance	`Current(0.7)`		`Look-Ahead(0.7/0)`		`PerfectInfo(0.7)`	
	Accep. ratio	Util. ratio	Accep. ratio	Util. ratio	Accep. ratio	Util. ratio
CS_01A	18.40%	6.90%	52.00%	18.20%	52.00%	18.20%
CS_01B	27.20%	9.97%	64.00%	24.27%	67.20%	25.17%
CS_01C	29.60%	10.37%	61.60%	22.00%	61.60%	22.20%
CS_01D	37.60%	14.43%	69.60%	26.67%	72.80%	27.57%
CS_02A	17.60%	6.40%	42.40%	14.40%	43.20%	14.53%
CS_02B	29.60%	11.60%	63.20%	23.27%	64.00%	23.67%
CS_02C	34.40%	12.87%	68.80%	25.37%	71.20%	26.90%
CS_02D	36.80%	13.57%	68.00%	24.47%	68.80%	24.93%
CS_03A	13.60%	5.10%	51.20%	18.07%	51.20%	18.07%
CS_03B	39.20%	15.93%	69.60%	26.67%	69.60%	26.67%
CS_03C	34.40%	12.50%	70.40%	27.30%	73.60%	28.03%
CS_03D	38.40%	15.57%	77.60%	30.27%	77.60%	30.27%
Average	**29.73%**	**11.27%**	**63.20%**	**23.41%**	**64.40%**	**23.85%**

Let TR be the total number of requests received by the system. Akin to RA, it considers both the requests in set B and R, that is, $TR = |B| + |R|$, where $|\cdot|$ denotes the cardinality of a set. Let TTU be the total time the cars have been used. Finally, let TTA be the total time potentially available, computed as $T \times f$, where f denotes the fleet size. Note that in the experiments presented below $TTA = 180 \times 50$.

The KPIs used are the following:

1. Acceptance ratio: $100 \times \frac{RA}{TR}$;
2. Utilization ratio: $100 \times \frac{TTU}{TTA}$.

Computing times for solving the LP relaxations of both the Dynamic and Static models are negligible (in the order of fractions of a second) and, therefore, are not reported.

4.3. Results

In this section, we compare the performance of the `Look-Ahead` (α/β) carsharing system with the `Current`(α) system, using the `PerfectInfo`(α) system as a benchmark. In fact, the KPIs of the `PerfectInfo`(α) system give upper bounds on the performance of two other systems. We identify as a *base case* to conduct the comparison the solutions obtained by setting the maximum walking distance $\alpha = 0.7$ and, for the `Look-Ahead`(α/β) system, the minimum amount of anticipation requested to the users $\beta = 0$.

To simplify the discussion, in the following we comment the results obtained by taking the average over the 12 instances composing the testbed. For the base case, we also provide the detailed results obtained for each instance in Table 3. The 12 testbed instances, along with the complete set of results, are publicly available at https://or-brescia.unibs.it/instances.

Fig. 5 shows the average value of each KPI achieved by each of the considered carsharing systems. The main insights that we can gain from this figure are as follows:

- ✓ The average value of the acceptance ratio achieved by the `Look-Ahead`(0.7/0) system is more than twice that of the `Current`(0.7) system (63.20% against 29.73%);
- ✓ The average value of the acceptance ratio achieved by the `Look-Ahead`(0.7/0) system is only slightly smaller than the benchmark value provided by the `PerfectInfo`(0.7) system (63.20% against 64.40%);
- ✓ Similar conclusions can be drawn observing the average value of the utilization ratio achieved by each system.

The box-and-whisker plots depicted in Fig. 6 show the distributions of the values taken by the two KPIs for each charsharing system. The left and right edges of each box represent the first (Q1) and third (Q3) quartiles of these values, while the line inside the box represents the second quartile (i.e., the median). Hence, by construction the middle half of the values falls within the box. The lines extending horizontally from each box (whiskers) indicate the variability outside the first and third quartiles. The front whisker goes from Q1 to the smallest value, while the back whisker goes from Q3 to the largest value. The two box-and-whisker plots clearly confirm that the `Look-Ahead`(0.7/0) significantly outperforms the `Current`(0.7) system, while performing only slightly worse than the `PerfectInfo`(0.7) system.

To gain some insights on the impact of the maximum distance a user is available to walk on the performance of each system, we solved each instance for several values of parameter α, ranging from 0.193 km to 1 km, and observed the values of the two KPIs considered. The results, averaged over all the instances tested, are shown in Fig. 7. Note that the minimum distance between any pair of centroid nodes is 0.193 km. Hence, any value of α between 0 and such threshold (indicated with a red dashed vertical line in Fig. 7) has no impact on the performance of the systems. Note also that only for a very small percentage of pairs of centroid nodes (less than 1%) the distance is smaller than 0.45 km. From Fig. 7 we can gain the following insights:

- ✓ As expected, the longer the maximum walking distance a user is available to walk, the better the performance of each carsharing system is, both in terms of acceptance and utilization ratios;
- ✓ For each system, the improvements start for α greater than 0.4 km. This is due to the fact that in our case study very few centroid nodes are closer than that distance. Therefore, very few, if any, cars available in a different parking location than the desired one can be assigned;
- ✓ The average values of both the acceptance and utilization ratios for the `Look-Ahead`$(\alpha/0)$ system are significantly outperforming those of the `Current`(α) system, whatever the value of α is;
- ✓ The over-performance of the `Look-Ahead`$(\alpha/0)$ system, compared to the `Current`(α) system, slightly reduces when the value of α increases. Nevertheless, even in the worst-case, the over-performance is remarkable. More exactly:
  - For $\alpha = 0$, the average acceptance ratio of the `Look-Ahead`$(\alpha/0)$ system is more than 35% greater than that of the `Current`(α) system, whereas the utilization ratio is about 13% higher;
  - For $\alpha = 0.7$, the average acceptance ratio of the `Look-Ahead`$(\alpha/0)$ system is about 33% greater than that of the `Current`(α) system, whereas the utilization ratio is about 12% higher;
  - For $\alpha = 1$, the average acceptance ratio of the `Look-Ahead`$(\alpha/0)$ system is about 28% greater than that of the `Current`(α) system, whereas the utilization ratio is about 10%.
- ✓ The average values of both KPIs of the `Look-Ahead`$(\alpha/0)$ system are very close to those of the `PerfectInfo`(α) system.

To investigate the impact of parameter α on the quality of service provided to the users, we computed for each instance two additional KPIs: the average distance walked by the users, and the percentage of requests that are assigned to a parking location different from the desired one. Let TDW be the total distance walked, in km, by the users in a given solution. Let RR be the number of relocated requests, i.e., assigned to a parking location different from the desired one. The two KPIs mentioned above are computed as follows:

3. Average distance walked: $\frac{TDW}{RR}$;
4. Relocated requests: $100 \times \frac{RR}{RA}$.

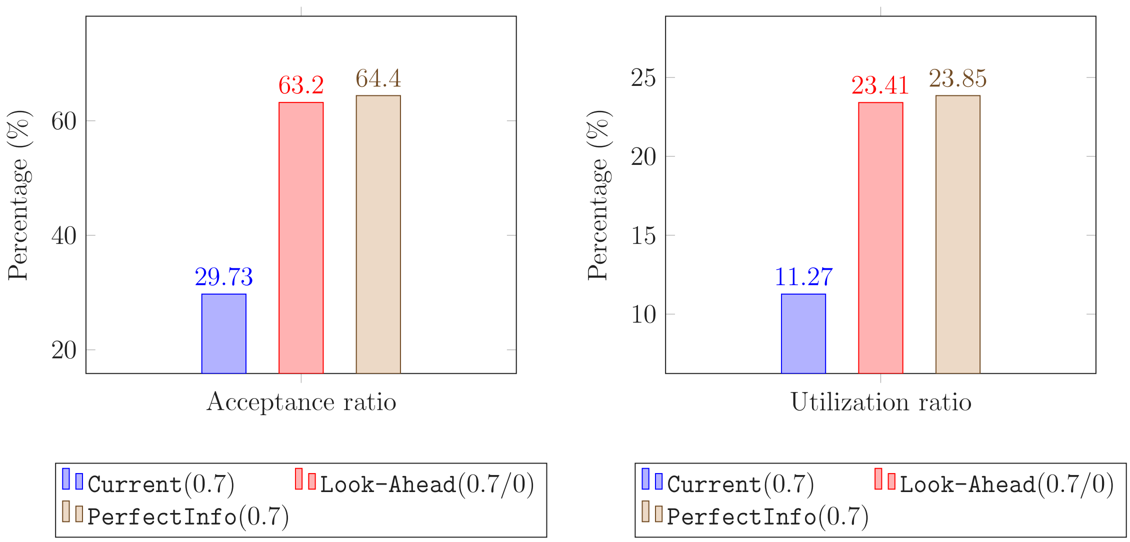

Fig. 5. Average performance of the three carsharing systems (base case).

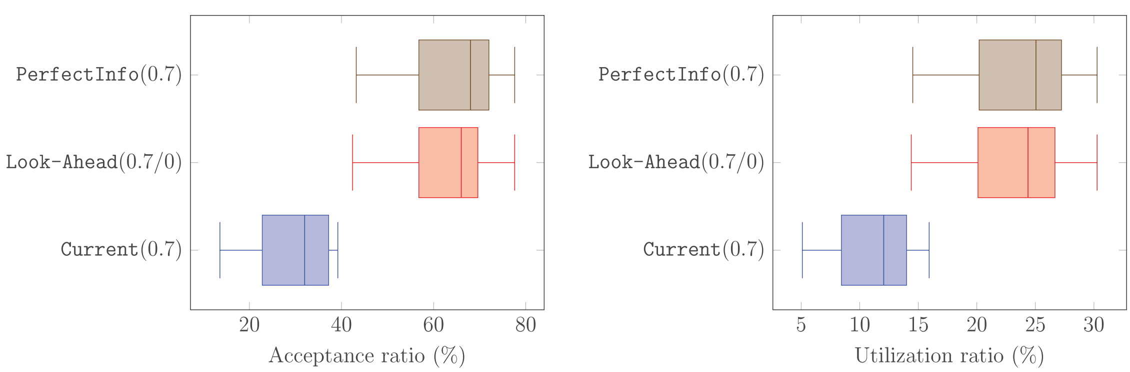

Fig. 6. Box-and-whisker plots showing the distribution of the two KPIs (base case).

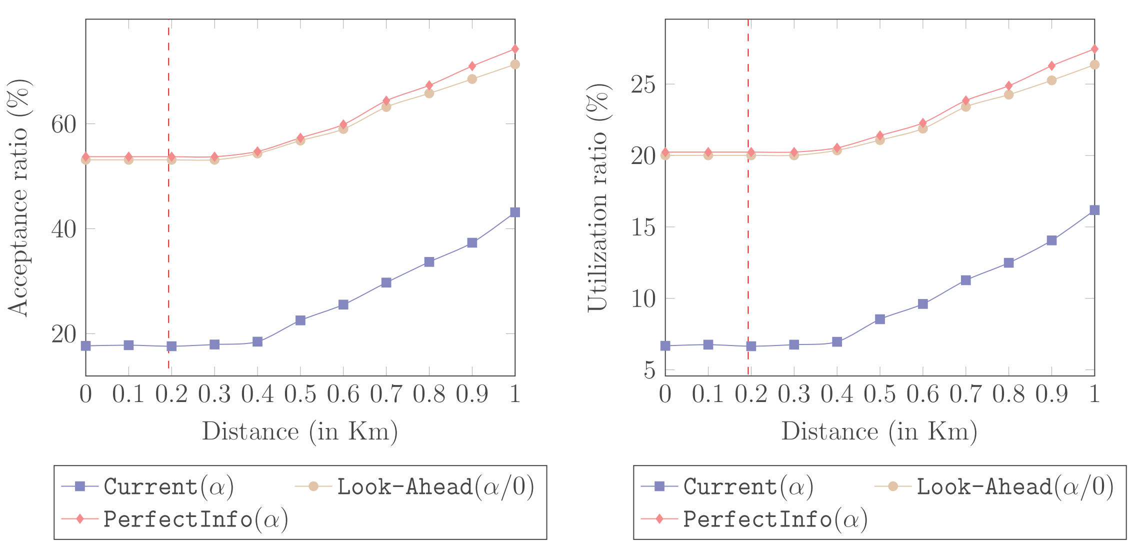

Fig. 7. The impact of the maximum walking distance α on the performance.

Fig. 8 depicts the value of the latter two KPIs, averaged over all the instances, for different values of α. From this figure, we can gain the following main insights:

✓ As expected, the average distance walked increases with α for all carsharing systems;

✓ No remarkable difference can be noticed among the three carsharing systems in terms of the average distance walked;

✓ The average distance walked is smaller than the value of α. For instance, for $\alpha = 1$, the average walking distance in the solutions provided by the `Current`(α) system is 0.75 km;

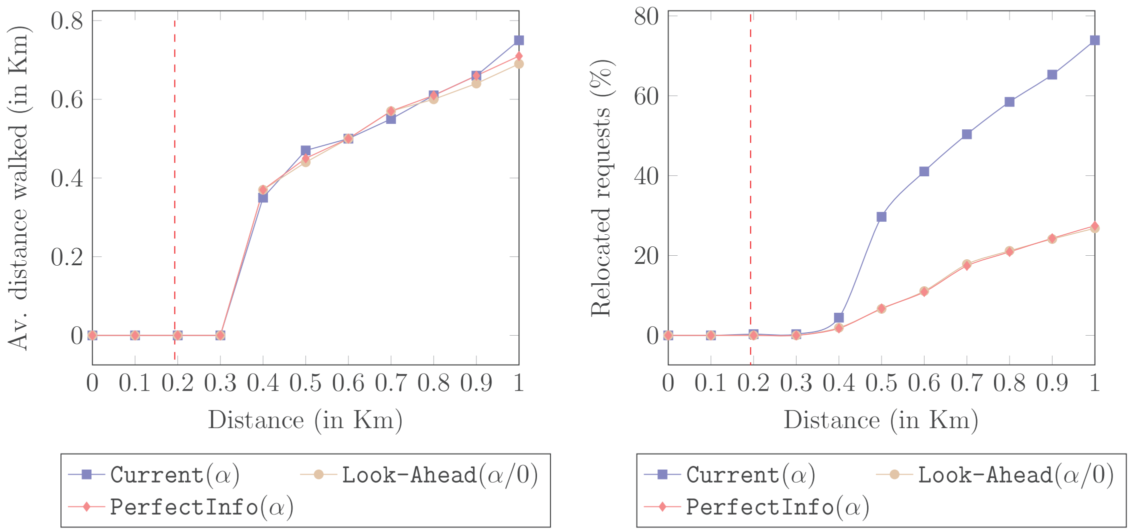

Fig. 8. The impact of the maximum walking distance α: Average distance walked (left panel) and percentage of requests relocated (right panel).

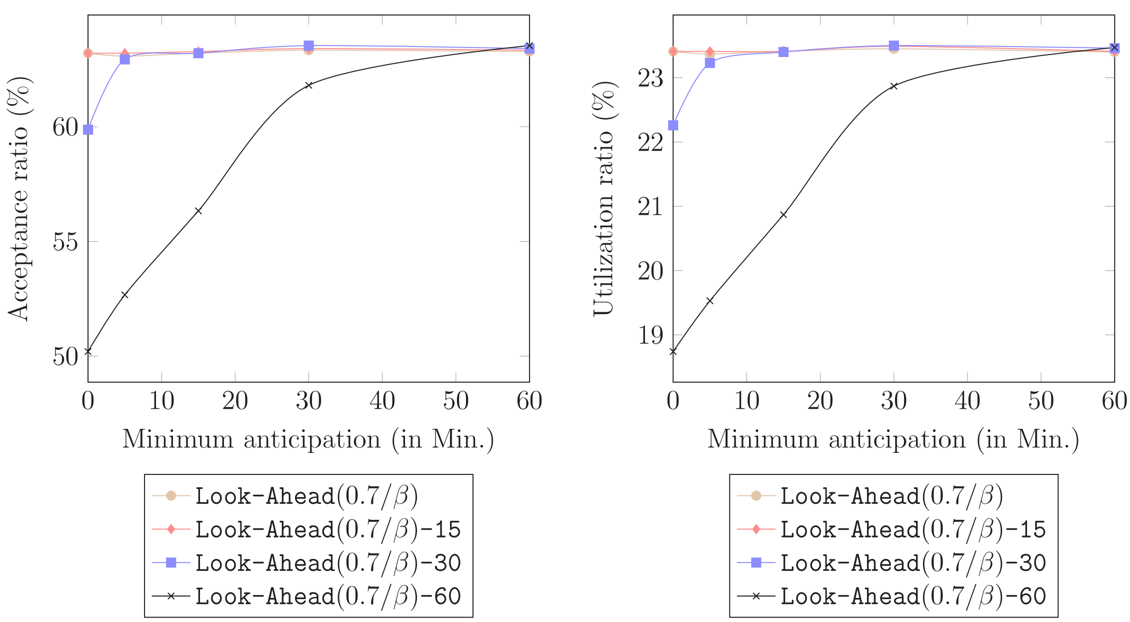

Fig. 9. The impact of the minimum amount of anticipation β on the performance.

- ✓ The percentage of relocated requests is very similar for the Look-Ahead(α/0) and PerfectInfo(α) systems;
- ✓ Compared to the other two systems, the percentage of relocated requests for the Current(α) system deteriorates sharply for values of α greater than 0.4 km. It is worth highlighting that, for a given value of α, the values of RR are rather similar among the three systems. Nevertheless, the value of the denominator RA for the Current(α) system is notably smaller than that of the other two systems, considerably increasing the percentage of relocated requests by the former system.
- ✓ For $\alpha = 1$, the percentage of relocated requests is approximately 74% for the Current(α) system, whereas it is about 27% for the remaining two systems.

We conducted the following analysis to evaluate the impact of the minimum amount of anticipation β on the performance of the Look-Ahead(α/β) system. Firstly, we considered the base case - i.e., Look-Ahead(0.7/0) - and iteratively increased the value of β up to 60 min. No remarkable effects were observed on both the acceptance and the utilization ratios. Recall that in the base case the time between two consecutive optimizations is equal to 5 min. One possible explanation of the negligible impact on the performance is that with a time frame of 5 min the re-optimizations are very frequent, and a booking request is immediately, or within a short time, considered in one of them. Thus, in this case, it is not useful to encourage the users to anticipate their requests. To support this explanation, we increased the time between two consecutive optimizations to 15 min, 30 min, and 60 min. The results, averaged over all the instances, are reported in Fig. 9. From this figure, one can notice that the value of β has negligible impacts on both performance indicators for the base case and its variant with 15 min between two consecutive optimizations (denoted as Look-Ahead(0.7/β)-15). The impacts are noticeable when the time between optimizations is equal to 30 min and, more particularly, 60 min. In the latter two cases, the negative impacts on the two KPIs is remarkable for the smallest values of β, and tends to improve when the value of β increases.

As detailed in Section 4.1, we generated 3 sets of instances, according to the three time frames considered in the original O–D matrix.

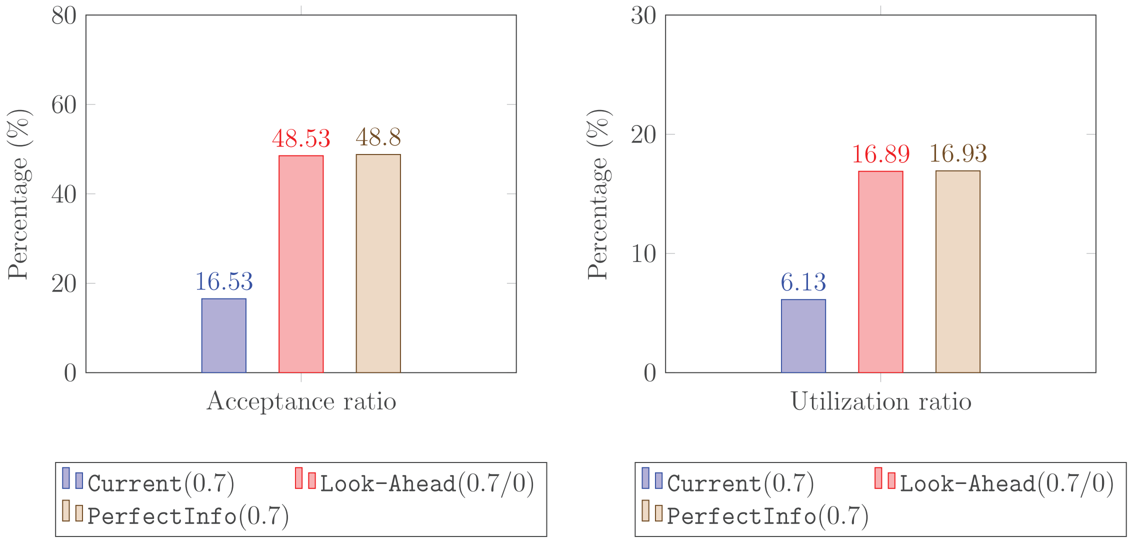

Fig. 10. Average performance of the three carsharing systems: Base case during the morning time frame (7:00 to 10:00; peak hours).

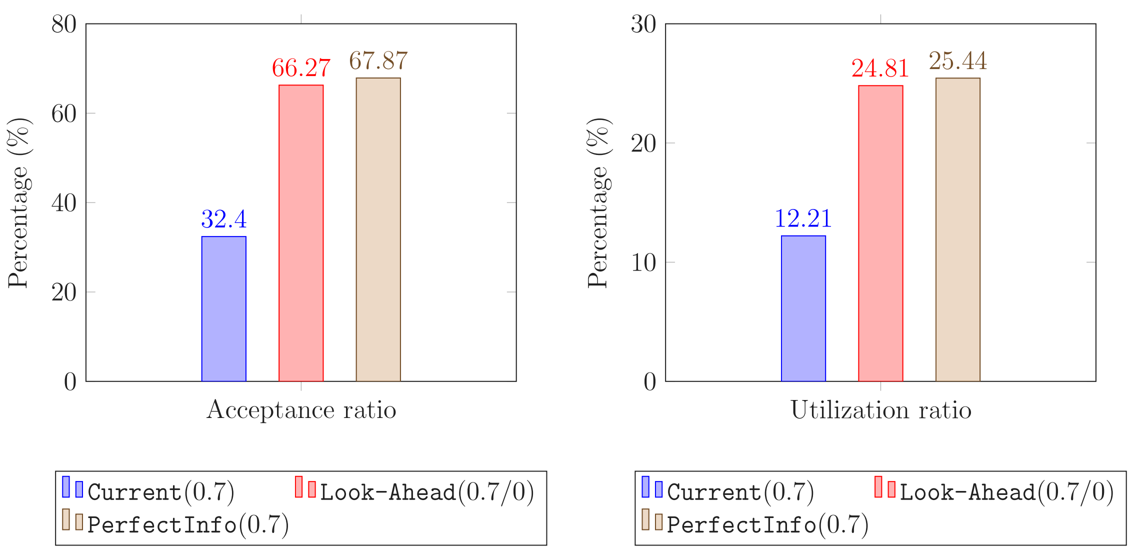

Fig. 11. Average performance of the three carsharing systems: Base case during the off-peak time frame (10:00 to 16:00).

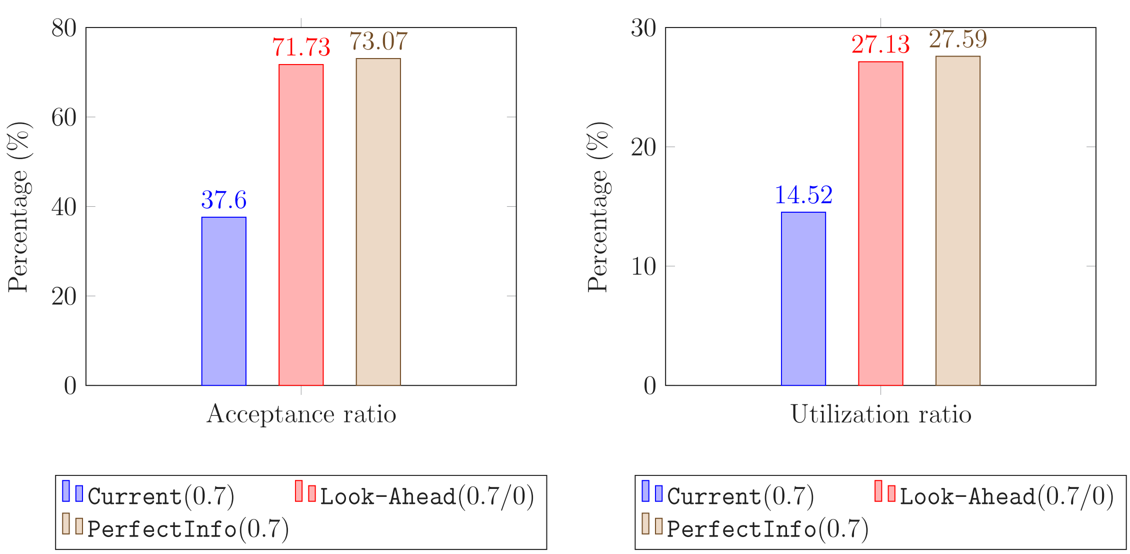

Fig. 12. Average performance of the three carsharing systems: Base case during the evening time frame (16:00 to 20:00; peak hours).

Figs. 10–12 provide some insights on the impacts of the time frame and the related traffic conditions on the performance of the three systems. Fig. 10 shows the performance of the three systems, averaged over the instances generated by the trips that occurred in the morning peak hours. Figs. 11 and 12 show the average performance of the systems for the off-peak and the evening peak-hours, respectively. While there are differences in the values of the KPIs across the three time frames, the results confirm that the `Look-Ahead`(0.7/0) system consistently outperforms the `Current`(0.7) one and is only slightly worse than the `PerfectInfo`(0.7) system. It is worth noting that the values of the KPIs in the morning peak hours are smaller than those achieved in the remaining two time frames. A possible explanation is that, at the beginning of the day, cars are positioned randomly among the centroid nodes (see the description of the instance generation provided in Section 4.1). Instead, as time goes on, cars are positioned according to the solution of the models, which anticipate future requests. As a consequence, the system has better chances to match cars with requests.

To analyze the impact of the most relevant instance parameters, we conducted the additional experiments detailed below. To ensure that the impact on the performance can be attributed to the changes of the parameters values, and not to the randomness of the trip generation, all the experiments detailed in the following were conducted solving the 12 instances considered in the base case with the changed values of one parameter at a time.

As a first experiment, we investigated the impact of the width of the pickup time window for the booking requests (i.e., $[e_b^{start}, l_b^{start}]$). We solved each instance with several values of such width. We modified the instances so that, compared to the average value for the base case displayed in Table 2, the average width is reduced up to 50% and increased up to 50%. Note that, when increasing the width of the pickup time window, the assumption $e_b^{end} \geq l_b^{start}$ can be violated. To avoid this situation, we carried out a procedure that, at each increase of the width, verifies if such condition is violated. If it is found to be violated, the estimated usage duration d_b is increased by the minimum amount necessary to make the above condition respected again. Recall that, as detailed in Section 3.1, $e_b^{end} = e_b^{start} + d_b$. The results of this experiment, averaged over all the instances tested, are shown in Fig. 13. The main insight that we can gain from the latter figure is the following. As expected, the larger the width of the pickup time window, the better the performance of the `Look-Ahead`(0.7/0) and the `PerfectInfo`(0.7) systems, both in terms of acceptance and utilization ratios. In fact, the increase of the average width of the pickup time window implies that the number of possible assignments is bigger, making it possible to satisfy more requests. Recall that in the `Current`(0.7) system users are not associated with a pickup time window. Hence, changing its width has no impact of the performance of the latter system.

As a second experiment, we investigated the impact of the estimated usage duration. We solved each instance for several values of parameter d_b, increasing its average value w.r.t. to the base case (see Table 2) up to 50%. We did not investigate the reduction of parameter d_b, as average values smaller than 27.04 min are not realistic in car sharing systems operating in cities of similar size than Milan. The results, averaged over all the instances tested, are displayed in Fig. 14. From the latter figure, we can gain the following main insights. The average values of the acceptance ratio for the `Look-Ahead`(0.7/0) and the `PerfectInfo`(0.7) systems decrease as the estimated usage duration increases. The basic reason is that, when d_b increases, cars are used, on average, for longer time periods. In other words, cars are idle, and hence available to satisfy requests, for shorter periods of time. Note that the estimated usage duration seems to have no relevant impact on the acceptance ratio for the `Current`(0.7) system. In terms of the utilization ratio, its average value improves as the estimated usage duration increases, for all the systems. This is an expected outcome, as parameter d_b has a direct impact on the numerator of such ratio.

To gain some additional insights on the impact of providing complete trip information, we conducted the following experiments. We considered a variant of the `Look-Ahead`(0.7/0) system, denoted as `Look-Ahead`(0.7/0/θ), where it is not mandatory for users to provide complete trip information to the service provider, and only a given percentage θ of them provides that information. The remaining $(1-\theta)$ users do not provide information on where and when the car will be returned. Thereby, the cars assigned to the latter users cannot be used to satisfy other requests until they have been returned. Note that in the `Look-Ahead`(0.7/0/θ) system, users associated with a booking request in set B are still assumed to make a reservation. We solved each instance for several values of θ, namely 0%, 25%, 50%, and 75%. Recall that in the `Look-Ahead`(0.7/0) system, 100% of the users provide complete trip information. The results, averaged over all the instances tested, are shown in Fig. 15. From this figure, we can notice that the average values of both the acceptance and the utilization ratios deteriorate as long as the value of θ reduces. The growing performance reduction, from `Look-Ahead`(0.7/0) to `Look-Ahead`(0.7/0/0), is due to an increasing lack of knowledge on where and when some of the cars will be returned, preventing their assignment to satisfy booking requests before the latter instant of time. Notice that, even when $\theta = .0$ the performance of the system is considerably better than that of the `Current`(0.7) system (e.g., the acceptance ratio is 55.07% against 29.73%). One explanation is that users associated with a booking request in set B are still assumed to make a reservation through the system, thereby anticipating the trip request which, along with all booking requests arising at similar times, are included in the optimization of the Dynamic model. This provides a remarkable overperformance compared to a system, as the `Current`(0.7) one, that operates without any optimization.

We now summarize the foremost managerial insights that can be drawn from the previous analysis:

- ✓ The availability of complete trip information to the service operator can have a critical impact on the performance of the carsharing system;
- ✓ A dynamic service with frequent re-optimizations, like the `Look-Ahead`(α/β), can provide a comparable performance to a system in which all the information is known to the service operator before the beginning of the planning horizon, like the `PerfectInfo`(α) system;
- ✓ The best performance are obtained re-optimizing frequently - i.e., every 5 min. Less frequent re-optimizations can have negative impacts on the performance of the system;
- ✓ If the system is frequently re-optimized, the minimum amount of anticipation requested to the users has negligible effects on the performance;
- ✓ Larger values of the maximum distance users are willing to walk impact positively on the performance of the system;
- ✓ Larger values of the width of the pickup time windows for the requests booked by the users impact positively on the performance of the system;
- ✓ Larger values of the estimated usage duration decrease the acceptance ratio, while increasing the utilization ratio.

5. Conclusions and future research directions

In this paper, we studied the performance of a free-floating carsharing system, called *look-ahead system*, where users are requested to provide complete trip details beforehand. This complete trip information, revealed dynamically to the service operator over a given planning horizon, enables assigning a user request to a car, even if that car is not yet available at the time the users place the request, contrary to what is commonly done in most of the modern carsharing systems. The functioning of the studied system is simulated by solving iteratively an optimization model to match requests and cars. The computational results carried out on instances generated using real-

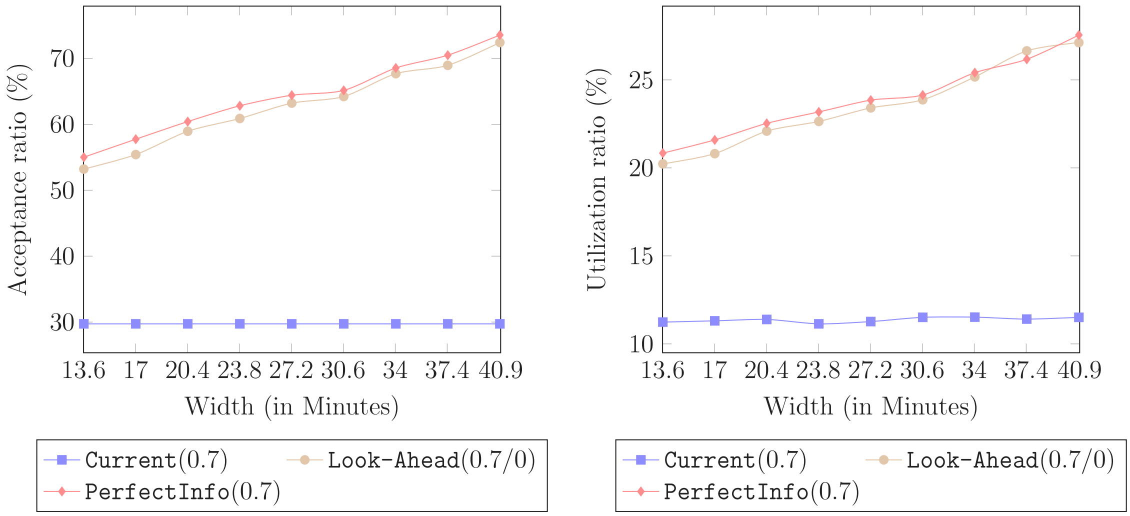

Fig. 13. The impact of the width of the pickup time window $[e_b^{start}, l_b^{start}]$ for the booking requests.

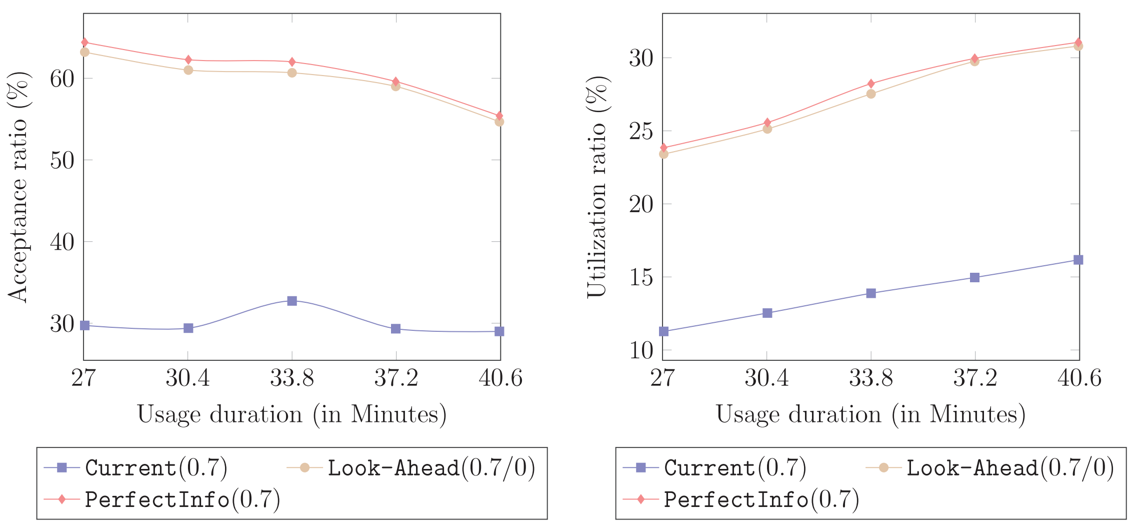

Fig. 14. The impact of the estimated usage duration d_b.

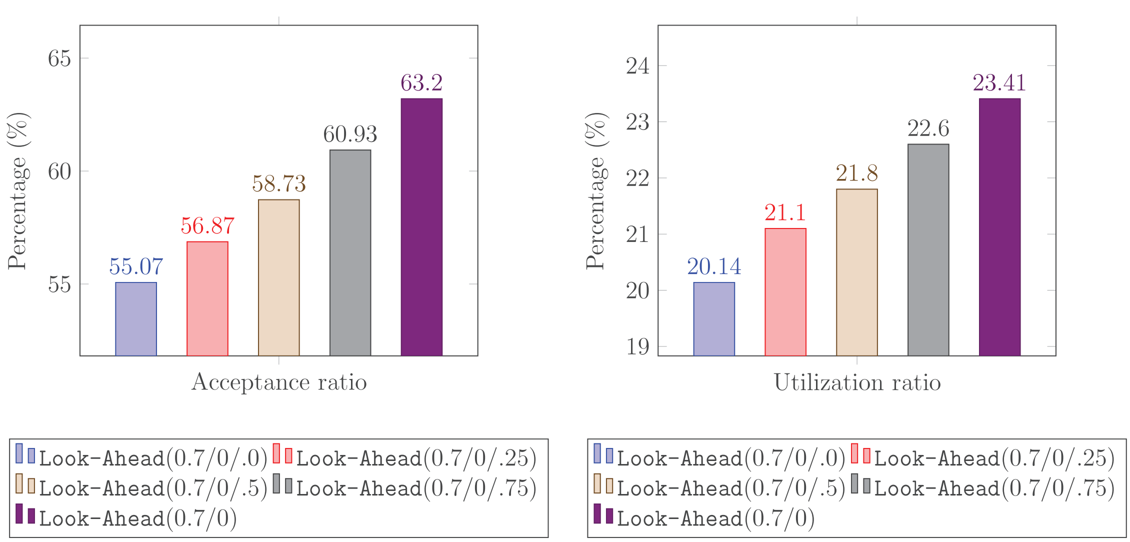

Fig. 15. The impact of complete trip information on the Look-Ahead(0.7/0) system.

world data from the city of Milan (Italy) show that the look-ahead system largely outperforms the most commonly operating systems, in terms of both acceptance and utilization ratios, and that it performs only slightly worse than a benchmark system based on assuming that complete information about the users requests is available before the start of the planning horizon.

Despite it is known that the profit yielded by a service operator is positively correlated with the number of requests accepted, an interesting extension of the present research is to incorporate into the optimization framework revenues and costs associated with the trips. This would enable us to envisage a carsharing system that, rather than imposing users to provide upfront complete trip information, offers an incentive (e.g., in the form of a discount, whose amount has to be determined) to share that information.

As future research directions, it would be interesting to consider the extension of the proposed models to a stochastic environment, where stochasticity might be related to the time (or the location) where cars will be delivered, the trip duration, or the distribution of future requests. For example, the Dynamic model presented in Section 3.2 could be adapted to embed uncertainty through stochastic programming or robust optimization approaches.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful to two anonymous reviewers whose comments and suggestions allowed them to improve the original version of this paper.

References

Alfian, G., Rhee, J., Kang, Y.-S., Yoon, B., 2015. Performance comparison of reservation based and instant access one-way car sharing service through discrete event simulation. Sustainability 7 (9), 12465–12489.
Barth, M., Todd, M., 1999. Simulation model performance analysis of a multiple station shared vehicle system. Transp. Res. C 7 (4), 237–259.
Barth, M., Todd, M., 2000. Intelligent transportation system architecture for a multi-station shared vehicle system. In: ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 00TH8493). pp. 240–245.
Barth, M., Todd, M., Xue, L., 2004. User-based vehicle relocation techniques for multiple-station shared-use vehicle systems. Transp. Res. Rec. 1887, 137–144.
Bruglieri, M., Colorni, A., Lué, A., 2014. The relocation problem for the one-way electric vehicle sharing. Networks 64 (4), 292–305.
Bruglieri, M., Pezzella, F., Pisacane, O., 2017. Heuristic algorithms for the operator-based relocation problem in one-way electric carsharing systems. Discrete Optim. 23, 56–80.
Bruglieri, M., Pezzella, F., Pisacane, O., 2018. A two-phase optimization method for a multiobjective vehicle relocation problem in electric carsharing systems. J. Combin. Optim. 36, 162–193.
Bruglieri, M., Pezzella, F., Pisacane, O., 2019. An adaptive large neighborhood search for relocating vehicles in electric carsharing services. Discrete Appl. Math. 253, 185–200.
Cepolina, E., Farina, A., 2014. A methodology for planning a new urban car sharing system with fully automated personal vehicles. Eur. Transp. Res. Rev. 6, 191–204.
Chapuis, R., Tadjeddine, K., Chinn, D., Holmes, R., Knol, A., Speksnijder, L., Wolfs, K., Lotz, C., Stern, S., 2020. Restoring Public Transit Amid COVID-19: What European Cities Can Learn from One Another. McKinsey & Company.
Chun, Y.-Y., Matsumoto, M., Tahara, K., Chinen, K., Endo, H., 2019. Exploring factors affecting car sharing use intention in the southeast-Asia region: A case study in java, Indonesia. Sustainability 11, 5103.
Cocca, M., Giordano, D., Mellia, M., Vassio, L., 2019. Free floating electric car sharing design: Data driven optimisation. Pervasive Mob. Comput. 55, 59–75.
Correia, G.H.d.A., Antunes, A.P., 2012. Optimization approach to depot location and trip selection in one-way car sharing systems. Transp. Res. E 48, 233–247.
ESRI, 2021. ArcGIS. Available at http://www.esri.com/software/arcgis/arcgis-for-desktop.
Eurostat, 2017. Passenger transport statistics. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Passenger_transport_statistics&oldid=333720 [Last accessed: 27th January, 2023].
Ferrero, F., Perboli, G., Rosano, M., Vesco, A., 2018. Car-sharing services: An annotated review. Sustainable Cities Soc. 37, 501–518.
Folkestad, C.A., Hansen, N., Fagerholt, K., Andersson, H., Pantuso, G., 2020. Optimal charging and repositioning of electric vehicles in a free-floating carsharing system. Comput. Oper. Res. 113, 104771.
Gambella, C., Malaguti, E., Masini, F., Vigo, D., 2018. Optimizing relocation operations in electric car-sharing. Omega 81, 234–245.
George, D.K., Xia, C.H., 2011. Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. European J. Oper. Res. 211 (1), 198–207.
Golalikhani, M., Oliveira, B.B., Carravilla, M.A., Oliveira, J.F., Antunes, A.P., 2021a. Carsharing: A review of academic literature and business practices toward an integrated decision-support framework. Transp. Res. E 149, 102280.
Golalikhani, M., Oliveira, B.B., Carravilla, M.A., Oliveira, J.F., Pisinger, D., 2021b. Understanding carsharing: A review of managerial practices towards relevant research insights. Res. Transp. Bus. Manage. 41, 100653.
Hayashi, T., Hidaka, K., Teshima, S., 2014. Carsharing and IT enabled services. In: 2014 Annual SRII Global Conference. pp. 274–280.
Hildebrandt, B., Hanelt, A., Piccinini, E., Kolbe, L., Nierobisch, T., 2015. The value of IS in business model innovation for sustainable mobility services - the case of carsharing. In: Wirtschaftsinformatik Proceedings 2015, Vol. 68. pp. 1008–1022.
Illgen, S., Höck, M., 2019. Literature review of the vehicle relocation problem in one-way car sharing networks. Transp. Res. B 120, 193–204.
Jorge, D., Correia, G., 2013. Carsharing systems demand estimation and defined operations: A literature review. Eur. J. Transp. Infrastr. Res. 13 (3), 201–220.
Li, X., Ma, J., Cui, J., Ghiasi, A., Zhou, F., 2016. Design framework of large-scale one-way electric vehicle sharing systems: A continuum approximation model. Transp. Res. B 88, 21–45.
Li, L., Petering, M.E.H., 2018. Discrete event simulation analysis of a reservation-based, one-way car-sharing system. J. Simul. 12 (1), 1–22.
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. Transp. Res. C 98, 298–322.
Nair, R., Miller-Hooks, E., 2014. Equilibrium network design of shared-vehicle systems. European J. Oper. Res. 235 (1), 47–61.
Nourinejad, M., Roorda, M.J., 2014. A dynamic carsharing decision support system. Transp. Res. E 66, 36–50.
Pasaoglu Kilanc, G., Fiorello, D., Martino, A., Scarcella, G., Alemanno, A., Zubaryeva, A., Thiel, C., 2012. Driving and parking patterns of European car drivers – A mobility survey. https://publications.jrc.ec.europa.eu/repository/handle/JRC77079 [Last accessed: 27th January, 2023].
Schiffer, M., Hiermann, G., Rüdel, F., Walther, G., 2021. A polynomial-time algorithm for user-based relocation in free-floating car sharing systems. Transp. Res. B 143, 65–85.
Schrijver, A., 2003. Combinatorial Optimization: Polyhedra and Efficiency, Vol. 24. Springer Science & Business Media.
Shaheen, S.A., Chan, N.D., Micheaux, H., 2015. One-way carsharing's evolution and operator perspectives from the americas. Transportation 42, 519–536.
Shaheen, S., Sperling, D., Wagner, C., 1999. A short history of carsharing in the 90's. J. World Transp. Policy Pract. 5 (3), 16–37.
Shoup, D., 2006. Cruising for parking. Transp. Policy 13, 479–486.
Soppert, M., Steinhardt, C., Müller, C., Gönsch, J., Bhogale, P.M., 2023. Matching functions for free-floating shared mobility system optimization to capture maximum walking distances. European J. Oper. Res. 305 (3), 1194–1214.
Xu, M., Meng, Q., 2019. Fleet sizing for one-way electric carsharing services considering dynamic vehicle relocation and nonlinear charging profile. Transp. Res. B 128, 23–49.
YouGov Italy, 2020. Private car use frequency in Italy in 2020. https://www.statista.com/statistics/1122212/private-car-use-frequency-in-italy/ [Last accessed: 27th January, 2023].
Zhao, M., Li, X., Yin, J., Cui, J., Yang, L., An, S., 2018. An integrated framework for electric vehicle rebalancing and staff relocation in one-way carsharing systems: Model formulation and Lagrangian relaxation-based solution approach. Transp. Res. B 117, 542–572.