A Multi-Stage Optimisation Approach to Design Relocation Strategies in One-Way Car-Sharing Systems With Stackable Cars

Riccardo Iacobucci, Raffaele Bruno, *Member, IEEE*, and Chiara Boldrini

***Abstract*— One of the main operational challenges faced by the operators of one-way car-sharing systems is to ensure vehicle availability across the regions of the service areas with uneven patterns of rental requests. Fleet balancing strategies are required to maximise the demand served while minimising the relocation costs. However, the design of optimal relocation policies is a complex problem, and global optimisation solutions are often limited to very small network sizes for computational reasons. In this work, we propose a multi-stage decision support system for vehicle relocation that decomposes the general relocation problem into three independent decision stages to allow scalable solutions. Furthermore, we adopt a rolling horizon control strategy to cope with demand uncertainty. Our approach is highly modular and flexible, and we leverage it to design user-based, operator-based and robotic relocation schemes. Besides, we formulate the relocation problem considering both conventional cars and a new class of compact stackable vehicles that can be driven in a road train. We compare the proposed relocation schemes with two recognised benchmarks using a large data set of taxi trips in New York. Our results show that our approach is scalable and outperforms the benchmark schemes in terms of quality of service, vehicle utilisation and relocation efficiency. Furthermore, we find that stackable vehicles can achieve a relocation performance close to that of autonomous cars, even with a small workforce of relocators.**

***Index Terms*— Car sharing, vehicle relocation, optimisation.**

I. Introduction

MANY experts agree that we are at the dawn of a revolution in the automotive industry, which is driven by technological advances, digitalisation of mobility services, changes in people's mobility behaviours, as well as their perspective towards car ownership and the environmental impact of transportation systems [1], [2]. In particular, two global trends in the urban mobility landscape are particularly relevant to this study: i) the increasing popularity and diffusion of shared mobility solutions, and ii) the emergence of specialised vehicle concepts that attempt to reduce the *road footprint* (namely, public space that is occupied by cars), providing more convenient personal urban mobility [3].

A wide range of different shared and on-demand mobility services, especially in dense urban environments, have emerged to enable users to gain short-term access to transport on an "as-needed" basis [4]. One of the most prominent examples of these new on-demand mobility solutions is the *one-way car-sharing* scheme. Members of such systems can rent a shared-used vehicle from a fleet operated by a private company or a public entity for one-way short trips, typically using a web-based or mobile app [5]. One-way car-sharing systems can be further categorised as station-based or free-floating schemes if the vehicle can be picked up and dropped off only from designated stations or at any location in the service area, respectively.

A critical operational challenge for one-way car-sharing systems is how to ensure that there are sufficient available vehicles in each station (or across the regions of the service area) to satisfy the current and future rental requests. Indeed, it is well known that the distribution of a car-sharing fleet gets temporally and spatially imbalanced due to uneven demand patterns [6], [7]. The most straightforward approach to ensure the system balance is to over-dimension the fleet and station capacity to absorb demand fluctuations [8]. A more effective approach for the car-sharing operator is to relocate empty vehicles where they are most needed, based on forecasted vehicle demand or short-term bookings. In particular, vehicle relocation can be performed: i) by the users themselves, who are incentivised to carpool or to choose another trip destination; or ii) by dedicated drivers, which is currently more common [9]. There is a trade-off between the revenue loss due to lost rental requests and the costs associated with relocation operations, namely, fuel costs for the vehicle travelling empty, personnel cost for operator-based schemes [10], or fare discounts for user-based relocation schemes [11].

There is abundant literature on the design of optimal relocation policies under different operational and business constraints (see Section II for a detailed review). Typically, existing optimisation models are based on queuing theory [12], stochastic optimisation frameworks [13], mixed-integer linear programming (MILP) [14], or model predictive control (MPC) tools [15]. However, such models are computationally expensive and, for this reason, exact solutions are limited to small

Manuscript received 5 September 2019; revised 22 April 2020, 18 November 2020, and 26 May 2021; accepted 30 March 2022. Date of publication 13 April 2022; date of current version 11 October 2022. This work was supported in part by the European Institute of Innovation & Technology (EIT) Climate-Knowledge and Innovation Communities (KIC) through the DRIVE2 Project under Grant 200286 and in part by the Consiglio Nazionale delle Ricerche (CNR) through the TIRS Project under Grant FOE 2020. The Associate Editor for this article was M. J. Barth. *(Corresponding author: Raffaele Bruno.)*

Riccardo Iacobucci is with Tensor Energy, Tokyo 102-0093, Japan (e-mail: iacobucci.riccardo@gmail.com).

Raffaele Bruno and Chiara Boldrini are with the Ubiquitous Internet Lab, IIT-CNR, 56124 Pisa, Italy (e-mail: raffaele.bruno@iit.cnr.it; chiara.boldrini@iit.cnr.it).

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2022.3164989, provided by the authors.

Digital Object Identifier 10.1109/TITS.2022.3164989

problem sizes. Moreover, most of the existing approaches develop the relocation model under the assumption that the time-ordered sequence of pickups and drop-offs in the car-sharing system (namely, the traffic demand) is known. However, deterministic approaches may be ill-suited to consider demand uncertainties that occur in real systems. On the other hand, stochastic optimisation problems based on probability distributions of demand are difficult to solve, and they require sophisticated heuristic algorithms, such as in [13].

To cope with the intractability of finding an optimal global solution to the relocation problem for reasonable problem sizes and over a large time horizon, in this study we propose a novel *modular* and *multi-stage* decision-making tool for fleet rebalancing in one-way car-sharing systems. We show how to split the general relocation problem into three independent decision stages that can be solved sequentially. We will show that this approach scales well to very large instances of the relocation problem. The first decision stage focuses on the assessment of fleet imbalance; the second one on the selection of redistribution flows and routes between stations with over-accumulated vehicles to stations that experience vehicle shortages; and the third one on the scheduling of relocation tasks. Our optimisation framework relies only on short-term predictions of localised vehicle surpluses and deficiencies (also known as inventory imbalance). This approach mitigates the impact of demand uncertainties and dynamic traffic conditions on operational efficiency. Furthermore, it adopts a *rolling* horizon control strategy to adjust relocation decisions to the evolving system state.

The flexibility and generality of the proposed optimisation framework are demonstrated in two ways. First, we leverage our modular design to specify operator-based, user-based and robotic relocation policies that maximise the satisfied demand while minimising the time a vehicle spends in relocation operations. Second, we formulate our optimisation models not only for conventional cars, but also for a class of *stackable electric cars* that can be coupled together when parked (to save space), and driven together as a "road train". The design of compact electric cars that can be folded and stacked in line, which are intended to be used for short-distance urban trips in car-sharing systems, is not a novel concept, being the folding MIT CityCar[1] the most famous example. More recently, this concept has been further developed and also extended to include coupling capabilities that allow the creation of a train of vehicles than can be driven (and even recharged) together. An example of such concept is the ESPRIT car (illustrated in Fig. 1), which has been prototyped in 2018 by a European consortium.[2] One important feature of the ESPRIT prototype is semi-autonomous towing capabilities, whereby each vehicle, when being parked at a station, is stacked in an automated way to the vehicles already parked. Preliminary studies have shown that ESPRIT cars could be particularly useful for car-sharing systems, as they would enable more efficient redistribution of the fleet (one driver can relocate two or more vehicles [16]), as well as more efficient utilisation of the charging infrastructure (one charging station could serve multiple vehicles simultaneously [17]). There are similarities between the relocation problem for stackable cars and the relocation problem in bike-sharing systems, which typically use trucks to relocate simultaneously a large amount of bicycles [18], [19]. However, stackable cars offer more flexibility as they can be relocated without using a dedicated fleet of service vehicles. An operator-based relocation strategy using dedicated towing vehicles was also experimented in the past in the car-sharing system deployed at University of California Riverside, called Intellishare [20]. However, manual towing to relocate vehicles is very inefficient and slow. To the best of our knowledge, this paper is the first to formalise the relocation problem in a car-sharing system by considering stackable cars that have *semi-autonomous towing capabilities*, and to apply this concept to both operator-based and user-based policies.

Fig. 1. The ESPRIT train of vehicles.

To assess the efficiency and capabilities of the proposed framework, we tested different relocation techniques, using taxi trip data from New York City in 2018. The data set contains pickup and drop-off times/locations of more than 200,000 trips per day, offering a large-scale instance of the relocation problem. We also compare our relocation schemes with two state-of-the-art solutions, one designed for station-based one-way car-sharing systems [14], and one designed for free-floating bike-sharing systems [21]. Trade-offs between the level of service offered, fleet size, relocation efficiency, model complexity are discussed. Our results show that our approach ensures lower computation times than the considered benchmark schemes, as well as improved quality of service, vehicle utilisation and relocation efficiency. Furthermore, we demonstrate that a small team of drivers relocating stackable cars is sufficient to approach the efficiency of a shared-used fleet of self-driving vehicles.

The remainder of this paper is organised as follows. Section II reviews related work and further elaborates on the novelty of the proposed approach. Section III introduces the system model and the proposed methodology. Section III presents the relocation models. Section V compares the proposed relocation techniques with two benchmark schemes, while Section VI discusses the research conclusions and provides recommendations for future research.

II. Related Work

The fleet balancing problem in one-way car-sharing systems and, more in general, in shared mobility systems, has been extensively studied in recent years. The reader is referred to [9], [14], [22] for a comprehensive review. Given the focus

[1] https://www.media.mit.edu/projects/citycar/overview/.

[2] http://www.esprit-transport-system.eu/.

of our study, this overview concentrates on the *proactive* relocation problem, where vehicles are dispatched to certain stations or service zones to satisfy future predicted demand.

Two main control approaches are adopted for proactive relocation. A first approach leverages a stochastic representation of traffic demands. One class of solutions leverages stochastic fluid models, which describe the car-sharing system as a queuing network [12], [23], [24]. These models assume that the demand pattern can be simply represented as a set of aggregated flows of vehicles between network nodes, and modelled as a Poisson arrival process. Moreover, queuing theory is used to model available vehicles and waiting customers in the system. Then, average relocation rates are determined by finding the optimal steady-state solution of the system dynamics. However, under time-varying demand conditions, a steady state might not exist. Moreover, this modelling approach is not readily adaptable to varying levels of demand over time. A second class of solutions developed stochastic programming models for fleet rebalancing with uncertain demands that are described through probability functions [13], [25]. However, these models are typically solved using heuristic approaches or Monte Carlo methods. Dynamic operator-based relocation strategies are presented in [26] for a station-based car-sharing system with short-term reservations and limited station size. The first strategy is based on the requirement to have at least one vehicle at each station at any time, while a second strategy is based on a Markov model to estimate future shortage of vehicles or parking spots at stations.

A second approach relies on the construction of space-time networks to describe system dynamics with individual rental requests. Then, vehicle relocation is typically formulated as a mixed-integer linear programming (MILP) problem [10], [14], [15], [27]–[29]. Specifically, the authors in [27] formulate a MILP problem to minimise the total generalised cost of relocation, taking into consideration movement and relocation costs, staff cost and penalty costs of rejected rental requests. The branch-and-bound technique is used to find exact solutions to the problem for a small dataset containing less than 2000 trips. Relocation strategies for free-floating systems are proposed in [28] by combining a macroscopic relocation optimisation policy of moving vehicles between zones, with a rule-based heuristic for the relocation of individual vehicles. In [30] a rolling horizon optimisation approach is proposed to maximise the number of relocations performed over a short period of time ahead considering both in advance reservations and last-minute trip requests. In [14] an integrated optimisation framework is developed to model operator-based relocation, station planning and fleet sizing jointly to maximise the net revenue of the car-sharing operator. The work does not take into account operator rebalancing, but it only guarantees that the total time spent to relocate vehicles does not exceed the total available working hours for a shift of the relocation personnel. Vehicle and personnel relocation in a car-sharing system with reservations is modelled in [10]. A model predictive control approach with dynamic continuous relocation is presented in [15]. The main limitation of these works is that the MILP formulations are computationally very demanding as they explicitly model the routing of vehicles in the system, i.e. which route a vehicle should take when moved by a relocator or rented by customers. Thus, either very small systems are tested, or approximate algorithms (e.g., aggregated models with relaxed constraints [14], station clustering [10], [28], or zoning schemes [29]) are proposed to determine sub-optimal solutions in large-scale scenarios. Furthermore, the creation of a time-space model of the state of the car-sharing system requires a fixed and *deterministic* demand pattern (e.g. the time-ordered sequence of pickups and drop-offs in the car-sharing system). This approach is ill-suited to consider demand uncertainties that occur in real systems.

A few studies deal with user-based relocation, where users are encouraged and compensated for changing their behaviour to relocate vehicles [11], [31]. While these works show that user-based relocation can increase the system profitability and the served demand, the system performance significantly depends on the level of spatial and temporal flexibility of the demand [32]. A potential advantage of stackable vehicles is that user-based relocation may be possible without changing trip patterns just by appending extra vehicles to specific passenger trips [16], [33].

Operator-based relocation schemes suffer from the intrinsic complexity of optimising the routing and rebalancing of operators, as these too become unbalanced [34]. In practice, the relocation of operators between tasks is generally accomplished through a second operator in a service car, or by having operators move among regions by other means, such as public transport or by a folding bike which can be stowed in the relocating vehicle [9]. Both approaches have limitations: the first one forces to double the number of relocators, thus increasing relocation costs. The second approach requires longer times for the relocation of operators between tasks. A hybrid approach is to have relocators drive vehicles with passengers when relocating themselves, akin to a mixed car-sharing/taxi service [34].

The car-sharing relocation problem is also related to relocation in bike-sharing systems, where a dedicated fleet of trucks is used to relocate multiple bikes at once. Most of the literature on bike-sharing relocation has addressed the rebalancing problem as a variant of the one-commodity pickup-and-delivery capacitated vehicle routing problem. Most studies have considered a static problem in which the changes in the bike usage rate are negligible during the repositioning period (i.e., night relocation) [35], [36]. Other works consider the dynamic problem in which the usage rate varies over time (e.g. [31], [37]) or they allow the repositioning trucks to visit a station multiple times (e.g. [38]). Models for the relocation problem with multiple trucks and multiple visits to stations have also been developed [19], but they are computationally costly. A simplified and tractable method assumes single visits to stations and divides the problem into two separate vehicle routing problems: one for bike pick-up, and the following one for dropping off bikes and returning to the depot [21]. To conclude, it is important to point out that bike-sharing systems generally assume few relocation operations during the day, since the cost of the operator/truck is much higher than the bikes themselves. Conversely, in a car-sharing system, the cost of the vehicles is one of the highest costs for the system, so a strategy more reliant on frequent relocations and a smaller fleet is favourable.

The optimisation model proposed in this paper differs in many respects from the papers mentioned above. In particular, we deal with the computational complexity of the problem by using a multi-stage modelling approach, rather than a unified global optimisation. Furthermore, our solution is also robust to demand uncertainties as it leverages a rolling horizon control approach. Finally, the developed relocation models are not restricted to conventional cars but are generalised to be also used with stackable cars. While bike sharing and car sharing with stackable cars have many similarities, especially at the system planning level, stackable cars offer additional features (e.g., vehicle coupling) and increased flexibility, which demand new model formulations.

III. A Multi-Stage Decision Support System for Vehicle Relocation

Before introducing the model formulation, we present the design principles and the modular architecture of our decision support system for fleet rebalancing in one-way car-sharing systems. For the sake of simplicity, let us assume that time is discretised into time slots. A vehicle relocation strategy is a function that takes as input the car-sharing system status in a given time slot and outputs the set of relocation actions for a number of future time slots during which the relocation process is to be optimised. A relocation action consists in the transfer of one or more vehicles from a zone/station where there is an accumulation of vehicles (called *feeder*) to a zone/station where there is a shortage of vehicles (called *receiver*).[3] Depending on the relocation strategy, vehicle relocation can be carried out by a professional driver, a user, or even autonomously if self-driving cars are available.

As discussed in Section II, for practical size problems solving an optimisation model for the selection of individual relocation actions over an infinite control horizon may be a computationally intractable task. To address this issue, in this study we propose a *problem decomposition* approach that splits the original relocation problem into three simpler sub-problems that can be solved sequentially to determine a near-optimal (or at least well-performing) relocation policy. The structure of this approach is presented in Fig. 2.

A. Prediction Module

The first stage of the decision chain consists in predicting the maximum inventory imbalance (i.e. surplus or lack of vehicles) of each zone in the car-sharing operational area during a future time interval. Then, a feeder is any region (or station) with a positive imbalance, i.e., a foreseen surplus of vehicles. At the same time, a receiver is any region (or station) with a negative imbalance, i.e., a foreseen deficiency of vehicles. This component takes as *input* the current inventory levels of the zones (i.e., how many vehicles are currently available) and the future demand patterns, and it provides as *output* the estimate of the vehicle inventory imbalance of each station. A precise characterisation of the imbalance level is difficult due to the uncertainty of future demands [7], [39], and the complex interactions between the demand processes of different zones [18]. The design of our predictors is detailed in Section IV-B.

[3]Note that our model formulation can be applied to both station-based and free-floating systems.

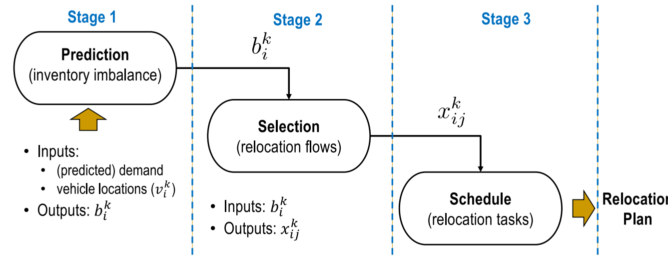


Fig. 2. Functional architecture of the multi-stage decision support system for vehicle relocation we propose in this study. The blocks in the diagram represent the main components of the systems with the input and output variables. Variable v_i^k and b_i^k denotes the available vehicles and the vehicle inventory imbalance, respectively, of station i at the decision point k, while x_{ij}^k is the relocation flow from station i to station j during the k-th decision interval. The formal definition of these variables is provided in Section IV.

B. Selection Module

This component of our decision-support system takes as *input* the inventory imbalance of each zone and provides as *output* the total number of vehicles to move between each feeder-receiver pair. The amount of time the car-sharing operator looks in the future to determine the relocation flows defines the planning horizon of the redistribution process. As discussed in Section II, long-term planning horizons (e.g., full operating days) typically assume known and static demand (e.g., a reservation-based system, where all the customer requests must be performed in advance), or a nearly idle system (e.g. when relocation is carried out during the night) [40]. On the contrary, short-term planning horizons are more suitable to cope with uncertain demands and to continuously exploit feedback about redistribution efficiency. Finally, different objective functions can be defined for the relocation process. There is a cost to move a vehicle between zones, and this cost typically depends on the distance between zone centroids. The model formulation of the second decision stage is detailed in Section IV-C.

C. Schedule Module

The third and final stage of the decision chain takes as *input* the set of relocation flows and schedules the sequence of relocation tasks that implement those relocation flows. A relocation task defines the relocation operations for individual (or group of) vehicles, namely how to split the aggregated relocation flow into a sequence of relocations of individual vehicles or trains of vehicles. Intuitively, feasible relocation tasks depend on the specific relocation technique. For instance, in case of an operator-based scheme, if the driver is not already available at the selected feeder, the car-sharing operator has to send one from another zone, incurring additional costs and delays. On the other hand, in a user-based scheme, an incentive (monetary or otherwise) should be offered to customers to contribute to vehicle rebalancing. Moreover, the time before a customer willing to relocate a vehicle arrives at a given feeder zone is uncertain.

A key advantage of the multi-stage decision process in Fig. 2 is that different relocation techniques can be easily plugged into the system by only adapting the last stage of the decision chain. Least-cost scheduling algorithms for different relocation techniques, both for conventional, stackable and robotic cars are developed in Section IV-D. A pseudo-code of the implementation of the full relocation algorithm that is executed by the proposed multi-stage decision support system is also provided in Appendix A of the supplemental material.

IV. Relocation Models

In this section, we present the mathematical formalisation of the proposed relocation models. In Section IV-A, we first define the sets and indices used to describe the model, as well as the functions, variables and parameters. Then, we describe in details the models for the three optimisation stages of the relocation policy presented in Section III.

A. Problem Definition

Let us assume that the operational area of the car-sharing system is partitioned into N zones. Let $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ be the set of zones. We also assume that there is an unlimited number of parking stalls reserved for shared vehicles within each service zone. Then, a demand pattern is associated with a zone, which represents an aggregation of pickup and drop-off events of vehicles within that zone. For the sake of simplicity, and without loss of generality, we cluster together the origin and destination locations of a car-sharing trip into the centroid of the related zone.[4] Then, the travel time $T(i, j)$ between centroids of zones s_i and s_j is constant and derived from historical traffic data. Without ambiguity, in the following, we use the index i to refer to zone s_i. Finally, in our model, we assume that customers do not wait for a vehicle, nor they change departure or arrival zones. In other words, a request for a car-sharing trip departing from a zone i with destination j (with the possibility that $j = i$) is admitted only if an empty vehicle is available in zone i. Otherwise, the user leaves the system.

For computational efficiency, time is discretised into times slots of duration τ. Then, all the time variables of the model are expressed as multiples of this time unit. Similarly to [40], we adopt a *rolling horizon* approach to decide the relocation plan. Specifically, each operation day is split into time periods of duration equal to n_C time slots (i.e. $T_C = n_C\tau$), called *planning periods*. Then, the relocation plans are computed at the beginning of each planning period using information about vehicle locations, rental requests, and surpluses/deficiencies of vehicles at each zone. Our relocation model assesses the effect of relocation decisions on the imbalance of vehicle supply considering a look-ahead time window of n_O time slots. In other words, $T_O = n_O\tau$ is the model *prediction time horizon*, with $n_O > n_C$ (see Fig. 3). Following the rolling horizon approach, the relocation decision process is iterated over the subsequent planning periods. When the time horizon rolls from the decision point k to the decision point $(k+1)$, the predicted state of the car-sharing system for the next n_O time slots is updated, (see Section IV-B for the details about the prediction methods). Note that when the system state is updated at time $(k + 1)$, the decision taken at the decision point k and not yet completed may not be optimal anymore, according to the state information and updated trip information that is available at decision point (k+1). Typically, in a rolling-horizon control approach, control decisions taken in a previous stage of the decision process, and not yet completed when the following decision stage starts, should not be modified [41]. Thus, any pending (i.e., not yet finished or started) relocation task from decision period k is not modified when a new relocation plan is computed at time $(k+1)$ (see Fig. 5 for an illustration of a pending relocation task). It is important to point out that the use of short-term demand predictions and decision periods shorter than the typical duration of a relocation trip, help to cope with demand variability.

[4]The same formalism can be readily applied to a station-based car-sharing system by substituting the zone centroids with the station locations.

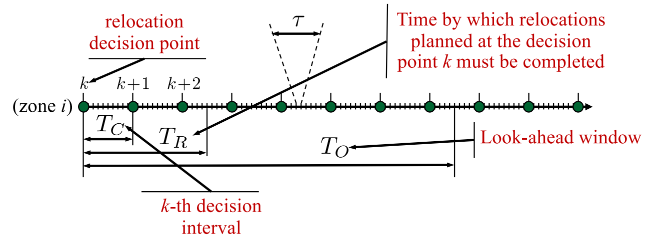


Fig. 3. Illustrative timeline of the relocation decision process with $n_C = 10$, $n_R = 25$ and $n_O = 75$ time slots. Circles represent the time points when relocation decisions are taken.

Without loss of generality, we assume that a relocation task starts at the beginning of a time slot. However, relocation operations (i.e., the physical redistribution of vehicles) can generally last more than one time slot. Furthermore, a relocation decision can entail multiple relocation operations from the same pair of feeders and receivers. However, we require that the sequence of consecutive relocation operations between a feeder-receiver pair that follow the decisions calculated at time kT_C are completed within a time $T_R = n_R\tau$, with $n_C \le n_R \le n_O$. A bound on the maximum time to complete a relocation task is beneficial to limit the relocation scope and to avoid relocating vehicles between very distant stations. Furthermore, relocation operations must finish within the model prediction time horizon to ensure that only the latest predicted state is used when calculating the relocation decision. Finally, setting up $T_C \le T_R$ reduces the probability that the system is idle because all relocation tasks are completed before a new decision interval starts. For the sake of clarity, Fig. 3 illustrates an example of the relation between τ, T_C, T_O and T_R.

B. Prediction of Inventory Imbalance

As defined in Section IV-A, at the k-th decision interval, the relocation policy needs to update the information about the

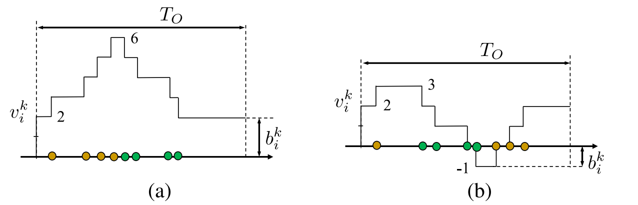


Fig. 4. Examples of the virtual inventory level of a (a) feeder station and a (b) receiver station. Orange circles denote vehicle arrivals, while green circles denote customers' trip requests. Initial inventory level $v_i^k = 2$. Both cases include four vehicle arrivals and four customers' trip requests in total, but different time orders.

inventory imbalance of each zone in the future n_O time slots to decide, first, which zones can be feeders and receivers and, second, how many vehicles should be relocated. Let b_i^k denote the expected inventory imbalance for zone i during interval $[kT_C, kT_C + T_O]$ (see Fig. 3) in case no new relocations are scheduled at time kT_C. Ideally, to exactly compute the b_i^k value we would need the in-advance knowledge of the functions $A_i^k(t)$ and $C_i^k(t)$, with $t = 1, \dots, n_O$, to model the vehicle arrival and departure processes of zone i in a deterministic manner.[5] More formally, let us assume that $A_i^k(t)$ provides the exact number of vehicles (either driven by a relocator or a customer) that arrive in zone i during time slot t in $[kT_C, kT_C + T_O]$. Similarly, let us assume that $C_i^k(t)$ provides the exact number of customers requesting a shared vehicle from zone i during time slot t in $[kT_C, kT_C + T_O]$. Let v_i^k be the number of already available vehicles within zone i at the beginning of the k-th decision interval (a.k.a. the initial inventory level of zone i). Now, let us introduce the variable $I_i^k(t)$, which measure the *virtual* inventory level of zone i at time slot t, namely the number of vehicles in zone i at time slot t if all customer trip requests would be accepted. It holds that $I_i^k(t)$ is given by:

$$I_i^k(t) = v_i^k + \sum_{n=1}^{t} \left[A_i^k(n) - C_i^k(n) \right] . \tag{1}$$

For the sake of clarity, Fig. 4 shows two illustrative examples of the temporal evolution of the $I_i^k(t)$ function for a zone i with an initial inventory level equal to two vehicles. Owing to the definition of $I_i^k(t)$, the inventory imbalance of zone i during interval $[kT_C, kT_C + T_O]$ can be computed as:

$$b_i^k = \min_{t=1,\dots,n_O} I_i^k(t) \tag{2}$$

A positive inventory imbalance quantifies the maximum number of vehicles that can be removed from zone i ensuring that $I_i^k(n) \geq 0$ throughout time interval $[kT_C, kT_C + T_O]$ (see Fig. 4a). Similarly, a negative inventory imbalance quantifies the minimum number of vehicles that should be moved to zone i to ensure that $I_i^k(n) \geq 0$ throughout time interval $[kT_C, kT_C + T_O]$ (see Fig. 4b). Then, the set $\mathcal{F}^k$ of feeder zones at the k-th decision interval is simply given by $\mathcal{F}^k = \{s_i \in \mathcal{S} | b_i^k > 0\}$. Similarly, the set $\mathcal{R}^k$ of receiver zones at the k-th decision interval is given by $\mathcal{R}^k = \{s_i \in \mathcal{S} | b_i^k < 0\}$.

[5]Clearly, in a real-world system $A_i^k(t)$ and $C_i^k(t)$ can only be predicted.

As shown in Equation 2, a precise estimate of the b_i^k variable would require to know the *time-ordered* sequence of vehicles and customers arrivals at each zone. Unless a strict reservation-based system is employed, car-sharing operators typically have an uncertain knowledge of $A_i^k(t)$ and $C_i^k(t)$, mainly leveraging historical data for rental requests. Depending on the quality and granularity of available data, different approaches can be devised to estimate b_i^k, and two of them are discussed in the following.

1) Worst-Case Estimate of b_i^k: To obtain a worst-case estimate of the b_i^k parameter, the car-sharing operator can leverage *average* estimates of the demand over the observation interval T_O. Specifically, let us assume that the car-sharing operator knows: i) the total number C_i^k of expected customers' requests for trips departing from zone s_i in $[kT_C, kT_C + T_O]$; ii) the average number D_i^k of passenger trips that start after time kT_C and terminate at zone i in $[kT_C, kT_C + T_O]$; and iii) the number R_i^k of vehicles that are currently en route to zone i, including empty vehicles and vehicles with passengers, and that are expected to arrive before $kT_C + T_R$. Intuitively, a rough estimate of b_i^k at each zone could be given by the net balance of vehicles arriving, vehicles leaving, and vehicles that are already parked there. Among these components, the estimate of D_i^k is typically the most unreliable, as it depends on the complex interactions between the demand processes of different zones [18]. Thus, a conservative estimate of b_i^k could ignore the contribution of D_i^k. Note that this does not affect the quality of service perceived by the customers: if the D_i^k vehicles do indeed arrive in the end, customers will experience a greater availability in zone i.

Based on the above considerations, it is easy to observe that the worst-case estimate of b_i^k is given by

$$b_i^k = v_i^k + R_i^k - C_i^k . \tag{3}$$

Basically, Eq. (3) assumes that all user requests are generated before new vehicles are dropped off at a given zone.

2) Probabilistic Estimate of b_i^k: We can obtain a probabilistic estimate of the b_i^k parameter if we assume that the car-sharing operator at least knows the *demand probability distributions* of each zone. More formally, let $f_A^{k,t}(n; i)$ and $f_C^{k,t}(m; i)$ denote the probabilities that n vehicles arrive at zone i during time slot t in $[kT_C, kT_C + T_O]$, and m user requests for trips departing from zone i arrive during time slot t in $[kT_C, kT_C + T_O]$, respectively. Moreover, we assume that $f_A^{k,t}(n; i)$ is defined over the finite set $[0, \beta_V]$, with $\beta_V \in \mathbb{Z}_{>0}$, while $f_C^{k,t}(n; i)$ is defined over the finite set $[0, \beta_C]$, with $\beta_C \in \mathbb{Z}_{>0}$. In general, these demand probability distributions can be estimated by historical trip data using simple averaging (as explained in Section V), or more sophisticated methods, such as kernel density estimation (e.g. in [25]) or ML techniques (e.g. in [7]). Now, we can model the virtual inventory variable $I_i^k(t)$ as a time-varying Markov chain. To compute the transition probabilities we introduce the auxiliary random variable $\Delta_i^k(t) \in \mathbb{Z}$, denoting the increment of the virtual inventory level in time slot t, i.e., $\Delta_i^k(t) = [A_i^k(t) - C_i^k(t)]$. We remind that the inventory level is increased by one for each vehicle arrival and decreased by one for each rental request. For the sake of simplicity and computational efficiency, we require that users issuing a rental request during time slot t can pick-up the vehicle only at the end of the time slot. In this case, it holds that:

$$\Pr\{\Delta_i^k(t)=l\} = \sum_{n=0}^{\beta_V}\sum_{m=0}^{\beta_C}\left[f_A^{k,t}(n;i) - f_C^{k,t}(m;i)\right]\delta_{(n-m),l} \quad \text{with } l \in [-\beta_V, \beta_C]\,, \tag{4}$$

where $\delta_{i,j}$ is the Kronecker delta function: 0 if $i \neq j$; 1 if $i = j$. Owing to the law of total probability, we can write:

$$\Pr\{I_i^k(t+1)=z\} = \sum_{l=-\beta_V}^{\beta_C} \Pr\{I_i^k(t) = z-l\} \cdot \Pr\{\Delta_i^k(t)=l\}\,, \tag{5}$$

From a generic initial state $I_i^k(0)$, the inventory level evolves following paths that are constrained by the β_V and β_C parameters. More formally, let ${}^{+}z_i^t$ and ${}^{-}z_i^t$ be the maximum and minimum values such that $\Pr\{I_i^k(t) = {}^{+}z_i^t\}$ and $\Pr\{I_i^k(t) = {}^{-}z_i^t\}$ are not null. It is straightforward to note that it holds:

$${}^{+}z_i^t = I_i^k(0) + \beta_V t \tag{6}$$

$${}^{-}z_i^t = I_i^k(0) - \beta_C t\,. \tag{7}$$

The rationale of Eqs. (6) and (7) is that after each time slot the maximum increment of the $I_i^k(t)$ random variable is β_V, while β_C is the maximum decrement.

Now we can define a methodology to provide a probabilistic estimate of the inventory imbalance during interval $[kT_C, kT_C + T_O]$. Let $F_i^k(v_i^k)$ denote the total probability of observing a negative $I_i^k(t)$ value over the period T_O if $I_i^k(0) = v_i^k$. It holds that:

$$F_i^k(v_i^k) = \sum_{t=1}^{n_O}\sum_{z=-z_i^t}^{-1} \Pr\{I_i^k(t) = z | I_i^k(0) = v_i^k\}\,. \tag{8}$$

Intuitively, $F_i^k(v_i^k)$ represents the probability that there is shortage of vehicle at zone i during $[kT_C, kT_C + T_O]$, given a demand probability distribution. Thus, $F_i^k(v_i^k) > 0$ implies that zone i is a receiver and the inventory imbalance b_i^k is the smallest negative value that ensures that $F_i^k(v_i^k + b_i^k) \leq \epsilon$, with ϵ small. On the contrary, $F_i^k(v_i^k) = 0$ implies that zone i is a feeder and the inventory imbalance b_i^k is the smallest positive value that ensures that $F_i^k(v_i^k - b_i^k) \geq \epsilon$, with ϵ small.

In Appendix B of the supplemental material we present a comparison of the two proposed methods and the evaluation of their impact on the relocation efficiency.

C. Selection of Relocation Flows

In this section, we formulate an optimisation model to calculate the set of relocation flows between pairs of feeders and receivers that satisfy a two-fold objective: 1) to balance the inventory level of the maximum number of receivers; and 2) to minimise the time a vehicle drives empty performing load balancing operations. Our conjecture is that the faster the fleet gets rebalanced, the higher is the demand served.

We start by defining the *utility value* J_{ij}^k assigned to the pair of zones i and j as follows:

$$J_{ij}^k = \begin{cases} T_R - T(i,j) & \text{if } i \in \mathcal{F}^k,\ j \in \mathcal{R}^k \\ -T_R & \text{otherwise} \end{cases} \tag{9}$$

where $T(i,j)$ is the travel time between centroids of zones s_i and s_j (see Section IV-A). Owing to Eq. (9), the utility of a feeder-receiver pair decreases as the time to complete a single relocation increases. The optimisation problem we formulate based on (9) is described in Eqs. (10)-(13). The decision variable x_{ij}^k expresses the number of vehicles that should be relocated from feeder i to receiver j.

Problem 1 : Optimal selection of relocation flows

$$\max \quad \sum_{i\in\mathcal{S}}\sum_{j\in\mathcal{S}} J_{ij}^k x_{ij}^k \tag{10}$$

$$s.t. \quad \sum_j x_{ij}^k \leq b_i^k \quad \forall i \in \mathcal{F}^k \tag{11}$$

$$\sum_i x_{ij}^k \leq -b_j^k \quad \forall j \in \mathcal{R}^k \tag{12}$$

$$x_{ij}^k \in \mathbb{N}_0 \tag{13}$$

The objective function (10) maximises the overall utility of the relocation process. Since each relocated vehicle satisfies one user request, we assign the same utility to each relocated vehicle. Implicitly, this also means that every successful trip returns the same profit to the car-sharing operator. Constraint (11) ensures that the number of vehicles relocated from feeder i cannot exceed the positive imbalance at the station. Similarly, constraint (12) ensures that the number of vehicles relocated to receiver j does not exceed the negative imbalance at the zone j. Constraint (13) ensures that x_{ij}^k belongs to the set of natural numbers (including 0). Note that Problem 1 is a variation of a 0-1 Multiple Knapsack Problem (MKP), which is an NP-hard problem. However, several relaxation techniques and heuristic approaches exist to obtain tight upper bounds of the problem solution in polynomial time [42].

D. Schedule of Relocation Tasks

The output of the second stage of the decision chain of Fig. 2 is a list of matched feeder-receiver pairs with associated the total number of vehicles to relocate (i.e. a relocation flow). Now, we want to answer the following research question: which is the most *efficient* way to relocate x_{ij}^k vehicles from station i to station j. Clearly, the answer depends on the constraints imposed by the specific redistribution technique that is employed. In the following sections, we develop a range of optimisation models for operator-based, user-based and robotic relocation schemes. The key novelty of our solutions is to consider stackable cars that can be driven in a train. Our optimisation framework is general enough to be applied also to car-sharing systems using conventional cars.

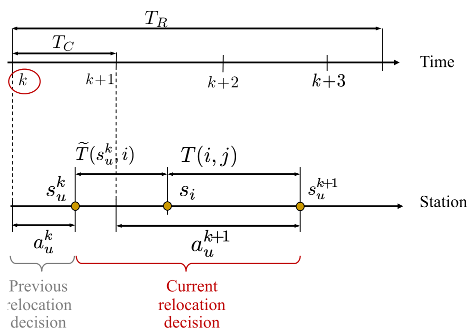


Fig. 5. Example of relocation decisions for operator o_u at decision interval k in the case $y_{uijl}^k = 1$.

1) Operator-Based Scheme: Let $\mathcal{O} = \{o_1, o_2, \ldots, o_M\}$ denote the set of M professional drivers (*relocators*) that are employed by the car-sharing operator to relocate vehicles. Without ambiguity, in the following, we use the index u to refer to relocator o_u.

Let us consider the k-th decision interval. To account for varying working shifts, let $\widehat{\mathcal{O}}^k \subseteq \mathcal{O}$ be the subset of relocators that are available during the k-th decision interval. Let us assume that relocator $u \in \widehat{\mathcal{O}}^k$ is already relocating a vehicle at time kT_C. Then, two variables can be associated with relocator u: 1) s_u^k, defined as the destination zone of u, and 2) a_u^k, defined as the residual time (in time slots) to reach s_u^k (see Fig. 5). A relocation flow can be split into multiple relocation tasks depending on the maximum allowable train size. Specifically, we assume that each relocator uses a service car to move between zones, and that up to $(\eta + 1)$ vehicles can be stacked together. Thus, each operator can relocate at most η customer vehicles during a single relocation task, since one vehicle in the relocated train is the service used by the operator to travel to the next feeder. Hence, a relocation flow x_{ij}^k can be split into $L_{ij}^k = 1 + \lfloor x_{ij}^k/(\eta) \rfloor$ relocation tasks, each one used to relocate p_{ijl}^k vehicles. In principle, a conventional car-sharing system can be modelled by setting $\eta = 1$ and assuming that the service car is driven by a second operator.

By definition, it holds that:

$$x_{ij}^k = \sum_{l=1}^{L_{ij}^k} p_{ijl}^k \tag{14}$$

$$p_{ijl}^k \leq \eta\ . \tag{15}$$

For the sake of notation brevity, we introduce the set $\mathcal{P}_{ij}^k = \{p_{ij1}^k, p_{ij2}^k, \ldots, p_{ijL_{ij}^k}^k\}$ that contains the relocation tasks that compose the relocation flow x_{ij}^k. The optimisation problem we formulate to assign relocators to relocation tasks is described in Eqs. (17)-(20). The decision variable y_{uijl}^k expresses the assignment of relocator u to relocation task l in the set $\mathcal{P}_{ij}^k$.

The objective function in Eq. (17) expresses the maximisation of the relocation efficiency, accounting not only for the vehicle relocation but also for relocator rebalancing. Specifically, the first part of Eq. (17) ensures that the maximum number of vehicles are relocated. The second part of Eq. (17) refers to the total time that relocator u spends in completing the ongoing relocation tasks (i.e. a_u^k) and reaching the feeder zone i from the destination zone s_u^k of the previous relocation task. The former time is denoted with $\widetilde{T}(s_u^k, i)$, and, in theory, it can be different from $T(s_u^k, i)$ (for instance, because the relocator travels between stations using a bike). Hence, this second term minimises the delay incurred by an operator before starting a new relocation task. Given that the second term takes values in the range [0, 1], our formulation prioritises over the number of relocation trips. Constraint (17) restricts to one the number of new tasks that a relocator can perform in T_R. Note that the model can be expanded to remove this restriction but in real-world cases, the length of the T_R period is comparable to the $T(i, j)$ values and do not allow the same relocator to perform multiple relocation tasks within the same relocation period. Constraint (18) ensures that a relocation task is assigned to a single relocator. Constraint (19) restricts the assignment of a relocation task to relocators that can complete it within the relocation interval.

Problem 2 : Optimal assignment of relocation tasks to operators

$$\max \sum_{u \in \widehat{\mathcal{O}}^k} \sum_{i \in \mathcal{F}^k} \sum_{j \in \mathcal{R}^k} \sum_{l \in \mathcal{P}_{ij}^k} \left(p_{ijl}^k - \frac{a_u^k + \widetilde{T}(s_u^k, i)}{T_R} \right) y_{uijl}^k \tag{16}$$

$$s.t. \quad \sum_{i \in \mathcal{F}^k} \sum_{j \in \mathcal{R}^k} \sum_{l \in \mathcal{P}_{ij}^k} y_{uijl}^k \leq 1 \quad \forall u \in \widehat{\mathcal{O}}^k \tag{17}$$

$$\sum_{u \in \widehat{\mathcal{O}}^k} y_{uijl}^k \leq 1 \quad \forall i \in \mathcal{F}^k, \quad \forall j \in \mathcal{R}^k, \quad \forall l \in \mathcal{P}_{ij}^k \tag{18}$$

$$y_{uijl}^k \left(a_u^k + \widetilde{T}(s_u^k, i) + T(i, j) \right) \leq T_R \tag{19}$$

$$\forall u \in \widehat{\mathcal{O}}^k, \quad \forall i \in \mathcal{F}^k, \quad \forall j \in \mathcal{R}^k, \quad \forall l \in \mathcal{P}_{ij}^k$$

$$y_{uijl}^k \in \{0, 1\} \tag{20}$$

After assigning relocators to relocation tasks we can update the s_u^{k+1} and a_u^{k+1} variables as follows:

$$a_u^{k+1} = \begin{cases} \max(0, a_u^k + \left[\widetilde{T}(s_u^k, i) + T(i, j)\right] - T_C) & \text{if } \ y_{uijl}^k = 1 \\ \max(0, a_u^k - T_C) & \text{otherwise} \end{cases} \tag{21}$$

$$s_u^{k+1} = \begin{cases} j & \text{if } \ y_{uijl}^k = 1 \\ s_u^k & \text{otherwise} \end{cases} \tag{22}$$

It is easy to observe that when relocator u is not performing a relocation task in the k-th decision interval, then a_u^{k+1} is the previous residual time minus the length of the decision interval (the residual time cannot be less than zero). If a relocation task is assigned, this is added to the next residual. The new destination station s_u^{k+1} for relocator u is j if $y_{uijl}^k = 1$, otherwise it remains unchanged.

2) User-Based Scheme: In traditional user-based relocation schemes, the car sharing operator needs to convince customers, leveraging fare discounts, to change the destination of their trips [11]. Stackable vehicles add a new dimension to the problem: customers can now take an additional vehicle with them, and this can be beneficial for the rebalancing, even without a change in the destination (e.g., if the customer is already travelling towards a zone that needs additional cars). Note that we assume that only professional drivers with a bus driving license can relocate trains of more than 2 vehicles, while users with conventional driving license can take at most an additional car. Furthermore, customers are not required to manually tow vehicles by themselves as vehicle towing occurs at dedicated stations in a semi-automated way.[6] Obviously, intervening only on the number of towed cars and not on the destination may be ineffective in some situations (e.g., there might not be enough users travelling in the "right" direction for relocation), but it still provides an improvement over traditional approaches with very limited inconvenience for the customers. Thus, in the following we focus on this scenario, and we leave for future work the investigation of the willingness of customers to change their destinations as a function of the offered fare discount. In the following, we simply assume that a user accepts the request from the car-sharing of driving a train of two vehicles with a constant probability equal to γ, independently of the price discount offered by the car-sharing operator to reward customers for the relocation of cars. Then, a relocation task is performed from zone i to zone j during the k-th decision interval if the following conditions are satisfied: 1) a customer heading toward zone j arrives at zone i during the k-th decision interval; 2) zone i and zone j have a surplus and deficiency of vehicles, respectively[7]; and 3) the customer is willing to relocate a vehicle. The design of optimised pricing schemes for incentivising user-based relocation is left as future work.

3) Robotic Scheme: It is widely recognised that vehicles having self-driving capabilities enable a more efficient relocation process because empty vehicles can autonomously relocate to a zone when needed without waiting for a relocator or a user. Thus, relocation tasks are only constrained by the availability of cars.

As explained in Section IV-C, the output of the second decision stage is the number x_{ij}^k of vehicles that should be relocated from station i to station j. The simplest approach for a robotic relocation scheme would be to transfer vehicles from station i to station j as soon as they are available, and until the number of relocated vehicles is equal to x_{ij}^k. However, in our relocation model, we follow the same approach as in [12], which assumes that vehicles autonomously relocate themselves from zone i to zone j with a constant rate equal to α_{ij} vehicles. The optimal relocation rate is updated at the beginning of the k-th decision interval as follows:

$$\alpha_{ij}^k = \frac{x_{ij}^k}{T_R - T(i, j)}\,, \tag{23}$$

[6]An example of automatic towing operations of real ESPRIT vehicles is available at `https://youtu.be/ayZ4-7O6rSs`.

[7]This condition requires that zone i is a feeder and zone j is a receiver. After each relocation the value of the inventory imbalance at the feeder (receiver) is decreased (increased) by one.

where x_{ij}^k is the solution of Problem 1. It is important to point out that we can not use the rebalancing strategy defined in [12] because users were allowed to wait for an available vehicle, and the deficiency of vehicles was measured in terms of waiting customers. Nevertheless, our model formulation in Problem 1 and the one in [12] are equivalent if the surplus of vehicles in the feeders is sufficient to rebalance all receivers.

V. Evaluation

To evaluate the effectiveness of the proposed relocation framework, we have implemented the optimisation models described in Section III in Matlab. We have also developed a discrete-time simulation model of a *station-based* one-way car-sharing system to validate the feasibility of the solutions generated by the optimisation models in realistic settings. Before presenting the results, we describe the simulation setup, the data we use to characterise the demand for pickup and drop-off of vehicles, and the benchmarks we adopt for performance comparison.

A. Data and Simulation Setup

There are a few publicly available data sets about real-world car-sharing services [7]. However, these data sets primarily contain data about pickup and drop-off times/locations of rented vehicles, but they do not disclose trip trajectories or rejected rental requests, as this is private and valuable commercial information. Furthermore, commercial car-sharing systems typically implement only overnight relocation. Hence, daytime demand patterns extracted from these data sets are necessarily balanced (since customers cannot pick up vehicles that are not there), and they are ill-suited for studying the efficiency of relocation policies. To circumvent this limitation, we use the data set of trips by New York's yellow taxis, available from the New York Taxi and Limousine Commission (TLC) [43]. These trips are likely to be unbalanced and to reflect the effective passenger demand, as taxi drivers can relocate their vehicles without passengers. An extensive analysis of the traffic patterns and demand imbalance properties of this data set are reported in Appendix B of the supplemental material. We have chosen the first 10 Wednesdays of 2018 (3 January to 7 March) as representative weekdays for our simulations. The number of trips in the data set for each day considered are reported in Table I. Each day includes over 200,000 trips. We found out that the beyond the classical morning peak period between 8:00AM to 10:00AM (with 300 trip requests per minute), there is also a similar demand peak in the evening between 6:00PM to 8:00PM. A substantial demand is also observed during off-peak periods, with an average of 200 trip requests per minute. Regarding the simulation model, it is a time-stepping simulation with $\tau = 1$ minute. At each time step, the simulation model begins by first checking if there is a trip request and if there is a vehicle available. Vehicle are assigned on a first-come-first-served basis. Customers are not allowed to wait at stations for available vehicles. If there are no rental requests, relocation tasks are executed depending on the current status of the system. For simplicity, we assume that a station is deployed at each centroid of a zone. Moreover,

TABLE I
NUMBER OF TRIPS FOR THE TEN DAYS USED IN THE EVALUATION (NEW YORK CITY, TAXI DATA, YEAR: 2018)

day	3 Jan	10 Jan	17 Jan	24 Jan	31 Jan
trips	224062	245844	261854	270451	273514
day	7 Feb	14 Feb	21 Feb	28 Feb	7 Mar
trips	273723	275086	251767	264750	204263

trips start from and arrive at the station since we are ignoring access walking times. Travelling times between pairs of zone centroids are assumed constant during the day, and they are estimated as the average duration of the taxi trips in the data set with pickups and drop-offs in those zones. In the following tests, we consider varying fleet size. We run the model in [14] without relocation to determine the initial position of vehicles at stations than minimise the probability of rejecting trip requests.

Five different relocation policies are tested in the following experiments: i) the operator-based scheme described in section IV-D.1 with three different train sizes, $\eta = 7$ (labelled OPR-E7), $\eta = 2$ (labelled OPR-E2), and $\eta = 1$ (labelled OPR-E1); ii) the user-based relocation described in section IV-D.2 (labelled USR) under the assumption that $\gamma = 1$ (i.e., customers always accept the relocation offer); and iii) the autonomous relocation scheme described in section IV-D.3 (labelled AR). We recall that with stackable cars the relocator can use a service vehicle that is connected to the train of vehicles to be relocated. The advantage is that the journey from a receiver to a feeder is faster than using alternative transportation like a bicycle or public transport (namely $\widetilde{T}(i, j) = T(i, j)$). The downside is that in a train of k vehicles, $(k - 1)$ vehicles satisfy the relocation needs while one is used by the operator to relocate himself. Note that in OPR-E1 the driver is allowed to relocate only one vehicle in addition to his service car. Thus, it is the policy that the most closely resembles an operator-based scheme used in conventional car-sharing systems. If not otherwise stated, the following results are obtained by using the worst-case estimate of the inventory imbalance described in Section IV-B.1. The parameter $C_i^k(t)$ in equation (3) is estimated using a simple averaging of trip requests for the NYC taxi dataset.

The key performance metric we use to assess system performance is the percentage of rejected trip requests. The trade-offs of fleet size and relocation resources are analysed. From the operator's perspective, we also consider the running time of relocation models and the relocation efficiency, measured in terms of the number of relocation tasks and travelled distances of relocated vehicles. To compute performance statistics and 95% confidence intervals, we have replicated each simulation ten times using data from the ten selected days.

B. Benchmarks

For performance comparison, we consider three representative state-of-the-art approaches.

The first benchmark is inherited from the body of work about vehicle relocation in bike-sharing systems. Specifically, we have implemented the truck-based relocation algorithm presented in [21] (labelled as TRR). We assume that the relocation is carried out by multiple trucks, each one with a capacity of 20 vehicles. The depot from which the trucks start and complete their routes is located in the centre of the operational area. We recall that TRR allows a relocator to visit and drop-off vehicles at multiple stations during the route. This is different from our operator-based relocation method, which does not allow fractional relocation. We recognised that this is an idealised system, as truck-based relocation is not possible with cars due to the difficulty of loading and unloading cars from stations. Nevertheless, a car-sharing system using stackable cars and a bike-sharing system have a somehow similar mobility concept. Thus, the model in [21] can provide a useful baseline of the potential benefits of vehicle towing during relocation operations.

The second benchmark is inspired by the operator-based relocation policy developed in [14]. As discussed in Section II, the original model in [14] is quite sophisticated as it jointly considers strategic and operational decisions to maximise the net revenue of the car sharing operator. In that study, the number of relocated trips between each pair of origin-destination stations are selected so that the total time spent to relocate vehicles does not exceed the total available working hours for a shift of the relocation personnel. Then, this benchmark relocation scheme, labelled as AGGR, minimises the time needed to perform planned relocations under a similar constraint on the maximum total relocation time. As in [14], we assume that demand is known for the whole day, and that an operating day is divided into time intervals (not necessarily equally long) and each operation (i.e. rental, relocation, charging) starts at the beginning and finishes at the end of a time interval.[8] Moreover, we do not consider the time that the relocation personnel spend moving from a receiver station to a feeder station: we assume that relocators are "teleported" to the station where they are needed when their previous relocation trip is finished. Note that it may happen that a relocation trip is planned to start when the vehicle is not yet available. This implies that not all the selected relocation decisions may be feasible, as this depends on the real availability of vehicles and relocators at feeders. Unfeasible relocation plans are simply ignored during the simulations of the real system.

The third benchmark, labelled as UB, consists in the reference model developed in [26] (Appendix A in that work), which simultaneously decides which requests to accept and which relocations to perform with in-advance full knowledge of the demand. This model provides an approximate upper bound for conventional car-sharing systems not using stackable cars, which maximises the number of accepted requests, subject to flow conservation constraints of vehicles and relocators, and vehicle reservation constraints. Note that this model can reject some trip requests even if vehicles are available at the booking times, if this increases the total number of accepted trips. As assumed in [26], the day is divided in periods of 5 minutes, and the starting and ending times of each trip are discretised using the same time granularity. Since the

[8]In the following tests we use 50 time intervals, chosen so that each interval has about the same number of trip requests.

TABLE II

BEST PERFORMING MODEL PARAMETERS

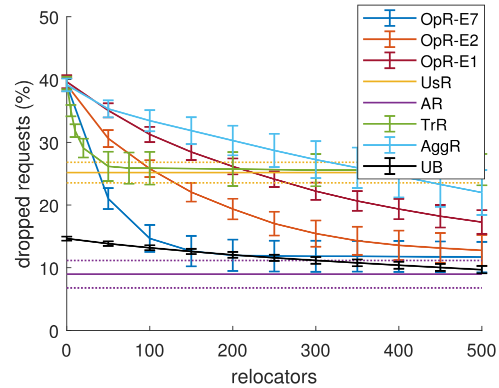
Relocation policy	T_C	T_R	T_O
OpR-E7 (worst case)	15	30	45
OpR-E7 (probabilistic)	20	40	70
TrR	60	60	120

model in [26] cannot be solved efficiently for our problem size, we adopt the same approach as in [14], and we assume that relocated vehicles are firstly accumulated at an imaginary hub and then distributed from that hub to the stations. This transformation significantly reduces the number of relocation variables in the model [26] and has a limited impact on the results. To reach feasible solutions at our scales, we also relax the integer constraints and we solve the equivalent linear optimisation. We think this is a reasonable approximation given the size of our problem. We note that in general this leads to slightly better results for this model thanks to the reduced constraints.

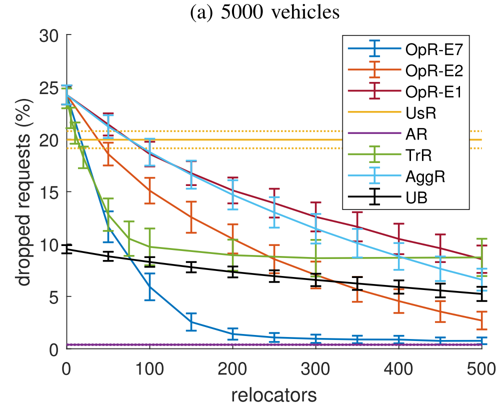
C. Effect of Fleet Size

If not otherwise stated, the following results have been obtained using the model parameters listed in Table II. An extensive parameter-sensitivity analysis has been conducted to select the best model parameters, which is reported in Appendix C of the supplemental material. Fig. 6a and Fig. 6b shows the percentage of rejected trip requests versus the number O of operators for the considered relocation policies with a fleet of 5,000 and 10,000 vehicles, respectively. Clearly, the relative performance of the user-based relocation scheme and the robotic one is invariant with the number of relocators. For the TRR scheme, the number on the x axis represents the number of trucks.

Important observations can be derived from the results shown in Fig. 6a. First, the robotic relocation scheme achieves the best performance among the considered strategies, but about 9% of the trip requests are still rejected. This is not surprising, as the relocation process in the robotic scheme is constrained only by the vehicle availability at the feeders. The user-based relocation scheme produces a 150% increase in the percentage of rejected trip requests when compared to the robotic scheme. Clearly, the efficiency of the USR scheme depends on several factors, including the willingness of users to perform relocation tasks, the demand patterns, and the probability that there is a passenger trip from a feeder to a receiver when needed. On the contrary, our OPR-E7 scheme approximates the efficiency of the robotic scheme with only 200 relocators (i.e. one relocator every 25 vehicles). Moreover, OPR-E7 scheme significantly outperforms the AGGR benchmark in all the considered scenarios. Even with a large relocation workforce (i.e. one relocator every 10 vehicles), the number of rejected rental request with AGGR is twice as much as that of OPR-E7. Clearly, the efficiency of our operator-based relocation scheme degrades when using smaller train sizes, namely $\eta = 2$. Nevertheless, OPR-E7 and OPR-E2 achieve similar performance when the number of relocators is



(a) 5000 vehicles



(b) 10000 vehicles

Fig. 6. Percentage of rejected trip requests as a function of the number of relocators for (a) 5,000 vehicles and (b) 10,000 vehicles.

sufficiently large. Interestingly, OPR-E1 behaves slightly better than AGGR even if both polices relocate a single vehicle. We observe that the car-sharing system is severely underserved when a fleet of 5,000 vehicles is used. In these conditions, the AGGR model is not able to find a relocation plan that can satisfy the full demand, while OPR-E1 only searches for a relocation plan that minimises the time spent relocating vehicles. It is also important to point out that using large trains is not necessarily beneficial for relocation efficiency. Indeed, the performance of the TRR scheme rapidly flattens out after a few trucks are employed. Besides, returning the trucks to a central depot rather than rebalancing them from receivers to feeders introduces an excessive delay in the relocation process. Thus, the truck-based relocation scheme achieves performance that is comparable to the user-based scheme with stackable cars. The results show that the percentage of dropped trip requests when no relocation is performed is significantly lower with the UB benchmark than the other schemes. This is due to the fact that the UB has a perfect advance knowledge of the trip requests and the model selects the trips to accept so that it minimises the fleet imbalance over the whole day. On the contrary, our system and the other benchmarks assign vehicles to passengers on a first-come first-served basis, which ensures a fair treatment of customers but a less efficient system. In addition, vehicle relocation has a small effect on the performance of the UB benchmark, as trips than cause inventory imbalance are removed in advance by UB. Thus,

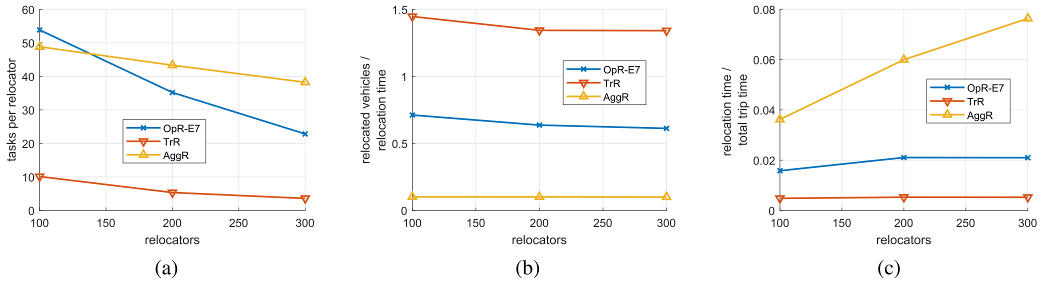


Fig. 7. Relocation efficiency in terms of: a) number of relocation tasks per relocator; b) number of relocated vehicles over daily relocation time; c) daily relocation time over the total duration of trips with passengers. Fleet size equal to 10,000 vehicles.

TABLE III
SPATIO-TEMPORAL ANALYSIS OF DROPPED REQUESTS (%). THE RESULTS REFER TO A RELOCATION STAFF OF 200 DRIVERS

Spatio-temporal domain	No relocation	OPR-E7	AGGR
5000 vehicles			
Morning peak (8am-10am)	42.48	17.55	31.71
Off-peak (noon-2pm)	27.71	4.82	18.33
5 most imbalanced	48.56	10.82	39.86
10000 vehicles			
Morning peak	19.57	1.18	7.62
Off-peak	18.57	0.03	10.37
5 most imbalanced	34.66	2.09	23.53

we can observe that OPR-E7 approaches UB performance with 150 relocators and when the number of relocators is sufficiently high (about 500 in the considered scenario), the performance gain of UB model over OPR-E2 is limited.

The results of Fig. 6b for 10,000 vehicles generally confirm the trends observed Fig. 6a. Clearly, doubling the fleet size increases the capacity of the car-sharing system. Indeed, the performance of all relocation schemes improves, with the robotic scheme that is able to satisfy the demand fully. Interestingly, we can observe that the use of longer road trains ($\eta = 7$) provides a faster performance gain because there are more opportunities for a relocator to find a full train of vehicles parked at the feeder when starting a new relocation task (see also Table IV). Therefore the performance gap between OPR-E7 and OPR-E2 increases with a fleet of 10,000 vehicles. AGGR performs slightly better than OPR-E1 with a fleet of 10,000 vehicles. In this case, the unfilled demand is low and the relocation policy of AGGR is able to determine more efficient relocation plans. As already shown in Fig. 6a, UB performs significantly better than the other relocation schemes when the number of relocators is low since it can select the most beneficial trips to serve. However, with 10,000 vehicles the relative performance gain of UB model is smaller than what observed with 5,000 vehicles. In addition the efficiency of the UB model increases very slowly when increasing the size of the relocation personnel. Thus, the relocation with stackable cars can outperform the UB benchmark when the number of relocators is higher than 75 for OPR-E7 and 300 for OPR-E2.

The results in Fig. 6a and Fig. 6b show the performance gains over the whole day. Table III reports the percentage of rejected trips in particularly relevant time intervals or stations for a scenario with 200 relocators. Specifically, we investigate the relocation efficiency during the morning peak, from 8:00AM to 10:00AM, and the off-peak period, from noon to 2:00PM. Moreover, we analyse the relocation efficiency in the five stations that show the deepest traffic imbalance. The results indicate that the off-peak period benefits more than the morning peak, with a five-fold decrease of dropped requests for 5,000 vehicles, but a two-fold decrease of dropped requests is achieved also in the off-peak period. As expected the receiver stations with the largest deficiency of vehicles are the ones with the highest percentage of dropped requests, but they also benefit the most from relocated vehicles.

D. Relocation Efficiency

In this section we investigate the efficiency of the operator-based relocation schemes from the car-sharing operator's perspective. Intuitively, it is important for the car-sharing operator to efficiently utilise the relocation workforce by ensuring that each relocator completes the largest possible number of tasks and that the distance vehicles drive empty is minimised. To this end, Fig. 7a shows the number of tasks each relocator performs on average with a fleet of 10,000 vehicles. Our results show that AGGR uses relocators more frequently than other schemes. This can be explained by noting that both OPR-E7 and TRR rapidly reach their optimal performance (see Fig. 6b), and the relocation efficiency is mainly constrained by vehicle availability rather than relocator availability. In this condition, the more relocators are employed, the longer they are idle. Clearly, the simultaneous relocation of multiple vehicles with a single relocation task also allows OPR-E7 and TRR to relocate vehicles more rapidly than AGGR (see Fig. 7b). Finally, Fig. 7c shows the ratio between the total time vehicles drive empty for relocation (namely, the relocation time) and the time vehicles drive with passengers. We point out that a train of relocated vehicles contributes to the total relocation time only once. The results show that OPR-E7 significantly outperforms AGGR as it ensures a higher utilisation of the shared vehicles for profitable trips.

To conclude this study, Table IV shows statistics about train length for OPR-E7 for different fleet sizes and numbers of relocators. The results show that between 60% and 80% of relocation actions are performed with the maximum train

TABLE IV
LENGTH OF TRAINS FOR OPR-E7 (%). O DENOTES THE NUMBER OF RELOCATORS, AND K THE FLEET SIZE

	K=5000		K=10000	
Length (x)	O=100	O=200	O=100	O=200
$x < 3$	14.0	16.6	6.0	9.7
$3 \leq x < 5$	12.0	12.4	6.9	8.9
$5 \leq x < 8$	9.2	10.7	6.0	7.5
$x = 8$	64.8	60.4	81.1	73.9

TABLE V
FRACTION OF TIME THAT RELOCATORS SPEND TRAVELLING FROM RECEIVER TO FEEDER STATIONS

	K=5000		K=10000	
Model	O=100	O=200	O=100	O=200
OPR-E2	52.8	53.1	52.0	52.6
OPR-E7	52.0	50.4	52.4	51.0

length of 8 vehicles. However, a non-negligible fraction of relocation trips involves smaller trains, which allows taking advantage of feeder stations with a small surplus of vehicles. The length of vehicle trains increases with the number of vehicles and decreases with the number of available relocators. This is because the larger the fleet, the larger the number of vehicles that are available at each station, and relocators give higher priority to tasks that relocate the most vehicles (see problem formulation in 2). Finally, Table IV shows the fraction of time relocators spend rebalancing themselves when completing a relocation task. The results show that about 50% of the travel time of operators is used to travel from receiver to feeder stations. This further justifies the AGGR, which considers only vehicle relocation and not operator relocation in the formulation of the relocation problem.

VI. CONCLUSION

Most of the research effort on car sharing systems is focused on vehicle relocation, which is considered the most difficult operational aspect of these systems. Previously proposed solutions are generally very complex, slow and with poor scalability. This limits their practical implementation. In this work, we presented a novel fast and scalable relocation optimisation framework considering stackable cars. We use problem decomposition to split the relocation problem into three simpler sub-problems. We showed that our algorithm outperforms previously proposed relocation algorithms while keeping computational complexity low. The capabilities of the proposed approach are demonstrated on a large scale data set comprising over 200,000 taxi trips per day in New York. Furthermore, our approach relies on limited aggregate information about the demand and does not require detailed advance knowledge, which is generally not available in real-world cases.

REFERENCES

[1] *Automotive Revolution—Perspective Towards 2030*, McKinsey, 2016.
[2] G. S. Bauer, J. B. Greenblatt, and B. F. Gerke, "Cost, energy, and environmental impact of automated electric taxi fleets in Manhattan," *Environ. Sci. Technol.*, vol. 52, no. 8, pp. 4920–4928, Apr. 2018.
[3] W. J. Mitchell, C. Borroni-Bird, and L. D. Burns, *Reinventing the Automobile Personal Urban Mobility for the 21st Century*. Cambridge, MA, USA: Massachusetts Institute of Technology, 2010.
[4] S. Shaheen, N. Chan, A. Bansal, and A. Cohen, "Shared mobility: Definitions, industry developments, and early understanding innovative mobility research," UC Berkeley, CA, USA, White Paper, Nov. 2015.
[5] B. Schwieger, P. Victorero-Solares, and D. Brook, "Global carsharing operators report 2015," Team Red, Germany, White Paper, Dec. 2015.
[6] M. Barth, M. Todd, and L. Xue, "User-based vehicle relocation techniques for multiple-station shared-use vehicle systems," in *Proc. IEEE Transp. Res. Board 80th Annu. Meeting*, Washington, DC, USA, Jan. 2004.
[7] C. Boldrini, R. Bruno, and M. H. Laarabi, "Weak signals in the mobility landscape: Car sharing in ten European cities," *EPJ Data Sci.*, vol. 8, no. 1, p. 7, Mar. 2019.
[8] D. K. George and C. H. Xia, "Fleet-sizing and service availability for a vehicle rental system via closed queueing networks," *Eur. J. Oper. Res.*, vol. 211, no. 1, pp. 198–207, May 2011.
[9] S. Illgen and M. Höck, "Literature review of the vehicle relocation problem in one-way car sharing networks," *Transp. Res. B, Methodol.*, vol. 120, pp. 193–204, Feb. 2019.
[10] B. Boyacı, K. G. Zografos, and N. Geroliminis, "An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations," *Transp. Res. B, Methodol.*, vol. 95, pp. 214–237, Jan. 2017.
[11] A. Di Febbraro, N. Sacco, and M. Saeednia, "One-way car-sharing profit maximization by means of user-based vehicle relocation," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 628–641, Feb. 2019.
[12] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for mobility-on-demand systems," *Int. J. Robot. Res.*, vol. 31, no. 7, pp. 839–854, Jun. 2012.
[13] R. Nair and E. Miller-Hooks, "Fleet management for vehicle sharing operations," *Transp. Sci.*, vol. 45, no. 4, pp. 451–566, 2011.
[14] B. Boyaci, K. G. Zografos, and N. Geroliminis, "An optimization framework for the development of efficient one-way car-sharing systems," *Eur. J. Oper. Res.*, vol. 240, no. 3, pp. 718–733, Feb. 2015.
[15] R. Zhang, F. Rossi, and M. Pavone, "Model predictive control of autonomous mobility-on-demand systems," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 1382–1389.
[16] C. Boldrini, R. Incaini, and R. Bruno, "Relocation in car sharing systems with shared stackable vehicles: Modelling challenges and outlook," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.
[17] E. Biondi, C. Boldrini, and R. Bruno, "Optimal charging of electric vehicle fleets for a car sharing system with power sharing," in *Proc. IEEE Int. Energy Conf. (ENERGYCON)*, Apr. 2016, pp. 1–6.
[18] T. Raviv and O. Kolka, "Optimal inventory management of a bike-sharing station," *IIE Trans.*, vol. 45, no. 10, pp. 1077–1093, Oct. 2013.
[19] T. Bulhões, A. Subramanian, G. Erdoğan, and G. Laporte, "The static bike relocation problem with multiple vehicles and visits," *Eur. J. Oper. Res.*, vol. 264, no. 2, pp. 508–523, Jan. 2018.
[20] M. J. Barth and M. Todd, "UCR INTELLISHARE: An intelligent shared electric vehicle testbed at the University of California, Riverside," *IATSS Res.*, vol. 27, no. 1, pp. 48–57, 2003.
[21] L. Caggiani, R. Camporeale, M. Ottomanelli, and W. Y. Szeto, "A modeling framework for the dynamic management of free-floating bike-sharing systems," *Transp. Res. C, Emerg. Technol.*, vol. 87, pp. 159–182, Feb. 2018.
[22] S. C. Ho, W. Szeto, Y.-H. Kuo, J. M. Y. Leung, M. Petering, and T. W. Tou, "A survey of dial-a-ride problems: Literature review and recent developments," *Transp. Res. B, Methodol.*, vol. 111, pp. 395–421, May 2018.
[23] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in Singapore," in *Road Vehicle Automation*, G. Meyer and S. Beiker, Eds. Cham: Springer, 2014, pp. 229–245.
[24] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: A queueing-theoretical perspective," *Int. J. Robot. Res.*, vol. 35, nos. 1–3, pp. 186–203, Jan. 2016.
[25] X. Li, C. Wang, and X. Huang, "A two-stage stochastic programming model for car-sharing problem using kernel density estimation," 2019, *arXiv:1909.09293*.
[26] M. Repoux, M. Kaspi, B. Boyacı, and N. Geroliminis, "Dynamic prediction-based relocation policies in one-way station-based carsharing systems with complete journey reservations," *Transp. Res. B, Methodol.*, vol. 130, pp. 82–104, Dec. 2019.

[27] A. Kek, R. Cheu, and C. F. Q. Meng, "A decision support system for vehicle relocation operations in carsharing systems," *Transp. Res. E, Logistics Transp. Rev.*, vol. 45, no. 1, pp. 149–158, 2009.

[28] S. Weikl and K. Bogenberger, "Relocation strategies and algorithms for free-floating car sharing systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 4, pp. 100–111, Dec. 2013.

[29] L. Wang, Q. Liu, and W. Ma, "Optimization of dynamic relocation operations for one-way electric carsharing systems," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 55–69, Apr. 2019.

[30] M. Repoux, B. Boyaci, and N. Geroliminis, "Simulation and optimization of one-way car-sharing systems with variant relocation policies," in *Proc. Transp. Res. Board 94th Annu. Meeting*, Jan. 2015, pp. 1–19.

[31] J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari, "Dynamic vehicle redistribution and online price incentives in shared mobility systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1567–1578, Aug. 2014.

[32] B. Boyacı and K. G. Zografos, "Investigating the effect of temporal and spatial flexibility on the performance of one-way electric carsharing systems," *Transp. Res. B, Methodol.*, vol. 129, pp. 244–272, Nov. 2019.

[33] C. Boldrini and R. Bruno, "Stackable vs autonomous cars for shared mobility systems: A preliminary performance evaluation," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 232–237.

[34] S. L. Smith, M. Pavone, M. Schwager, E. Frazzoli, and D. Rus, "Rebalancing the rebalancers: Optimally routing vehicles and drivers in mobility-on-demand systems," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 2362–2367.

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

[36] Y. Li, W. Szeto, J. Long, and C. Shui, "A multiple type bike repositioning problem," *Transp. Res. B, Methodologica*, vol. 90, pp. 263–278, Aug. 2016.

[37] L. Caggiani and M. Ottomanelli, "A modular soft computing based method for vehicles repositioning in bike-sharing systems," *Proc. Social Behav. Sci.*, vol. 54, pp. 675–684, Oct. 2012.

[38] J. J. Salazar-González and B. Santos-Hernández, "The split-demand one-commodity pickup-and-delivery travelling salesman problem," *Transp. Res. B, Methodol.*, vol. 75, pp. 58–73, May 2015.

[39] F. Miao *et al.*, "Data-driven robust taxi dispatch under demand uncertainties," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 1, pp. 175–191, Jan. 2019.

[40] C. Gambella, E. Malaguti, F. Masini, and D. Vigo, "Optimizing relocation operations in electric car-sharing," *Omega*, vol. 81, pp. 234–245, Dec. 2018.

[41] R. Scattolini, "Architectures for distributed and hierarchical model predictive control—A review," *J. Process Control*, vol. 19, no. 5, pp. 723–731, May 2009.

[42] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009.

[43] (2016). NYC Taxi and Limousine Commission (TLC). *TLC Trip Record Data*. [Online]. Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Riccardo Iacobucci received the Laurea degree in mechanical engineering and the master's degree in energy engineering from the Politecnico di Milano, Italy, in 2011 and 2014, respectively, and the Ph.D. degree in energy science from Kyoto University, Japan, in 2018. He is currently responsible for simulation and data science at Tensor Energy. His research interests are mainly in developing simulation and optimization models of sustainable energy systems, especially related to transport and power grid integration of renewable energy.

Raffaele Bruno (Member, IEEE) is currently a Senior Researcher with IIT-CNR, Pisa, Italy. His H-index is 32. His research interests are in designing network architecture and protocols for the IoT and 5G/B5G systems, fog/edge computing, and shared urban mobility systems. He has been involved in several national and European projects on those topics, among which, H2020 SLICE-DS, H202 SLICES-SC, H2020 ESPRIT, and H2020 REPLICATE. He was the TPC Co-Chair of ACM Q2SWinet 2012 and IFIP/IEEE SustainIT 2013, IEEE WoWMoM 2016, and IFIP WWIC 2019; and the General Co-Chair of IEEE WoWMoM 2021. He is on the Editorial Board of *Computer Communications* (Elsevier) and *Sustainable Computing* (Elsevier). He is the Editor-in-Chief of the IoT section of *Sensors* journal (MDPI).

Chiara Boldrini is currently a Researcher with IIT-CNR, Italy. Her research interests are in social computing and computational social science, human-centered AI, smart transportation and urban data science, mobile, and ubiquitous systems. She has published more than 60 papers on these topics, which have received more than 2800 citations. She is the IIT Co-PI of the projects CHIST-ERA SAI, H2020 SoBigData++, and H2020 HumanE-AI-Net. She has been involved in several national and European projects, since FP7, among which HAGGLE, SOCIALNETS, and ESPRIT. She is on the Editorial Board of *Pervasive and Mobile Computing* (Elsevier) and *Computer Communications* (Elsevier). She has been in the organizing committee of several IEEE and ACM conferences and workshops, among which IEEE PerCom, IEEE SECON, and ACM MobiHoc.