



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna Archivio istituzionale della ricerca

Optimizing relocation operations in electric car-sharing

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Claudio Gambella, Enrico Malaguti, Filippo Masini, Daniele Vigo (2018). Optimizing relocation operations in electric car-sharing. OMEGA, 81, 234-245 [10.1016/j.omega.2017.11.007].

Availability:

This version is available at: <https://hdl.handle.net/11585/645472> since: 2018-10-04

Published:

DOI: <http://doi.org/10.1016/j.omega.2017.11.007>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

*Claudio Gambella, Enrico Malaguti, Filippo Masini, Daniele Vigo, **Optimizing relocation operations in electric car-sharing**, Omega, Volume 81, 2018, Pages 234-245, ISSN 0305-0483*

The final published version is available online at:

<https://doi.org/10.1016/j.omega.2017.11.007>

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)

When citing, please refer to the published version.

Optimizing Relocation Operations in Electric Car-Sharing

Claudio Gambella^{#,°}, Enrico Malaguti[#], Filippo Masini[#], Daniele Vigo[#]

[#] DEI “Guglielmo Marconi”, Università di Bologna,
Bologna, Italy

{enrico.malaguti; filippo.masini; daniele.vigo} @unibo.it

[°] IBM Research Ireland, IBM Technology Campus,
Mulhuddart, Dublin 15, Ireland
claudio.gambella1@ie.ibm.com

In this paper we consider a station-based electric car-sharing system which allows one-way trips, and uses relocation to re-balance the vehicle distribution. We adopt the point-of-view of a service provider, whose objective is to maximize the profit associated with the trips performed by users. We introduce an exact relocation model for operating hours, and we explicitly consider the consumption and recharge process of electric vehicles batteries. In addition, the model is extended to the relocation operations to be performed at night, namely when the system is not operating. We also describe two model-based heuristics developed to solve the relocation model for operating hours on large-scale systems. The paper is concluded by a set of computational experiments on realistic data derived from an existing car-sharing system. The experiments investigate the scalability of the proposed model and highlight the circumstances under which the relocation operations can improve the system performance.

Keywords: Electric Car-Sharing, Relocation, Operations, Mathematical Models, Heuristic Algorithms, Computational Experiments.

1 Introduction

The first car-sharing system was introduced in Zurich, back in 1948. Nowadays, because of the increasing concerns regarding oil supplies and sustainable transportation, car-sharing systems are increasingly popular: sharing a car can help to lower the idle time of the vehicle, thus achieving a reduction of both the emissions and the total number of vehicles in cities (see Millard-Ball et al. [17]). On top of that, new technologies are making the

management of such systems easier: GPS allows to track vehicle movements, while on-line tools enable to book cars in advance and monitor the real-time status of the system.

Car-sharing providers may offer two kinds of services, namely two-way and one-way services. Serving only two-way trips means accepting only travels that end at the same spot where they started. On the other hand, one-way trips may end everywhere else, within the designated zone. Clearly, one-way trips may determine an unbalance in the distribution of the vehicles, which needs to be addressed by the car-sharing provider through the possible relocation of idle vehicles.

In electric car-sharing systems, only Electrical Vehicles (denoted as *EVs* in the following) are used. EVs are characterized by relatively large costs and limitations in their autonomy. These issues may have hindered the diffusion of EVs among the general public ; however, car-sharing initiatives contributed to their penetration in the market. EVs are now widespread and their popularity is also related to the availability of alternatives (and environmentally friendly) technologies to produce electricity. Therefore, switching to EVs represents the natural evolution of car-sharing as a way to move towards sustainable means of transport.

In the first implementations of electric car-sharing (denoted as Ecar-sharing) services, station-based systems were adopted, which require the customer to pick-up (and drop-off) the vehicles at specific sites in the city, each equipped with recharge infrastructures. Recently, however, some free-floating systems have been introduced. The free-floating option allows the user to return the car at any idle parking spot under the area controlled by the car-sharing provider.

The design of an Ecar-sharing system requires to take long-term decisions (strategical planning level), while daily operations (operational planning level) are faced during the management of the service. In this framework, the main optimization challenges can be summarized as follows:

Strategic	Infrastructure design
	Sizing of the EVs fleet
	Sizing of the relocation team
Operational	Assign travels to EVs
	Re-balance the one-way system
	Avoid battery depletion

This paper focuses on the operational level. Therefore, the strategic decisions are considered as fixed (i.e., the infrastructure and the number of available EVs are given) and their relative costs are considered as sunk costs. At the operational level, the EVs must be assigned to certain paths in order to serve as many customers' trips as possible. Simultaneously, the relocation operations can be scheduled. Each relocation requires an operator (in the following referred to as *relocator*) to move the vehicle. The assignment of relocators to EVs must also ensure the feasibility of the vehicle routes with respect to the battery charge at the departure station. In this paper, we assume that all requests are done in advance, say the day before. Within a prefixed time interval, say one hour, from the closing of the request system the users are notified about the acceptance or not of their transportation requests based on the available system resources and the system optimum found by the optimization system. Such process is similar to that implemented in some existing systems and compatible with the use of the exact and heuristic methods proposed in this paper.

The literature considers both Ecar-sharing systems and traditional ones. A brief review is provided here with focus on relocation; for an extensive survey on optimization problems arising in electric car-sharing systems, the reader is referred to the recent survey by Brandstätter et al. [6]. As far as relocation strategies are concerned, two approaches are identified in the literature: *user-based* (UB) and *operator-based* (OB) strategy.

Reference	Strategy	Objective	methodology
[3]	UB	min. relocation costs	Simulation
[9]	UB	max. revenue and max. user's benefit	Simulation
[12, 13]	OB	min. relocation cost and min. rejected demand	Exact/Heuristic/Simulation
[18]	OB	min. relocation costs	Exact
[15]	OB	min. relocation distance	Simulation
[11]	OB	max. profit	Exact/Simulation
[7]	OB	max. number of relocations served	Exact
[5]	OB	max. revenue and max. user's benefit	Exact

Table 1: Classification of the literature related to vehicles relocation (from Brandstätter et al. [6])

The UB relocation is performed by the customers and consists in modifications of their trips in order to help the system to restore a balanced distribution of vehicles in the network. Barth et al. [3] introduce *trip joining*, where two customer trips are converted into a single one who satisfies both customers, and *trip splitting*, where a single trip is divided into multiple trips, as a mean to reduce imbalances in the distribution. Furthermore, Clemente et al. [9] simulate a real-time monitoring that suggests trip alternatives to better balance the system. However, no realistic data on the customers willingness to accept the suggested trips after economic incentives are available. A necessary condition for adopting UB strategies is to gather information on the trips booked by users and this may cause privacy issues.

Other papers rely solely on OB strategies. Lee and Park [15] present an operation planner: given the actual EVs distribution, a relocation strategy is chosen, then the demand for relocation is computed, and finally the staff operations are scheduled. These tasks are carried out at the end of the service by a single team. A genetic algorithm is developed in order to minimize the distance traveled. The battery charge of the EV is not considered during the scheduling.

Nair and Miller-Hooks [18] consider various scenarios defined by a stochastic demand, and solve a Mixed-Integer Programming (MIP) model with joint chance constraints. To achieve a specified reliability of the system and minimize the relocation costs, no restriction on the number of relocators is considered.

Bruglieri et al. [7] explicitly consider linear battery consumption. Given a customer demand from under-supplied stations to over-supplied stations, an Electric Vehicle Relocation Problem (EVRP) is developed to maximize the total number of requests served.

Jorge et al. [11] present a MIP model based on a time-space network which focuses on maximizing the profit obtained by the car-sharing operator, given a set of stations and a demand. Besides, a simulation model is used to compare two different OB relocation strategies. The authors consider traditional combustion cars.

Boyacı et al. [5] introduce a bi-objective MIP model to support strategic, tactical and operational decisions (i.e., station positioning and capacity, assigning trips to EVs, performing relocations) in an integrated framework. A time-space network models a one-way station-based service. The EVs battery is assumed to be recharged after every travel. This introduces a fixed waiting time for the vehicle, but the charging process is not explicitly

modeled. The economical viability of relocations is evaluated under different scenarios. In order to reduce the number of relocation arcs, nodes are aggregated in virtual hubs.

A relevant issue in EVs systems is related with charging operations. Indeed, charging time for these vehicles is not negligible, charging infrastructure is a scarce resource, and charging operations require large electric power, affecting the status of the power grid. Kim et al. [14] propose a stochastic model and charge scheduling methods for EV battery charging infrastructure. The model considers relevant random factors and constraints, which include parking times, requested amounts of electricity, number of parking lots and demand level. Umetani et al. [20] consider the use of EVs as battery storages for stabilizing large fluctuations in the power grid through the vehicle-to-grid power system. They develop a linear programming based heuristic algorithm on a time-space network model for charge and discharge scheduling of EVs, and also present an improved two-stage heuristic algorithm to cope with uncertain demands and departure times of EVs.

Finally, in a one-way system each vehicle has a pick-up location and a return location, and hence associated optimization problems share some elements with pick-up and delivery problems, see, e.g., Hernández-Pérez and Salazar-González [10], Malaguti et al. [16]. However, in classical pick-up and delivery problems the visit order of the “locations” is not predetermined, while in one-way system each vehicle must visit the return location right after the pick-up location.

In this paper we consider a station-based electric car-sharing system which allows one-way trips, and uses OB relocation to re-balance the vehicles distribution. The system is managed by a service provider whose objective is to maximize the profit associated with the trips performed by the users. We introduce an exact relocation model for operating hours, and we explicitly consider the EVs battery consumption and recharge process. Our operational model assumes the number and features of vehicles, and the number of relocators as given. However, it can be used to assess the system performance under different configurations, and hence answers to tactical questions on the system management. In particular, we run computational experiments on realistic data derived from an existing car-sharing system, and discuss under which circumstances relocation operations can provide an improvement to the system performance.

This work was motivated and developed within the European project *e4-share* (Models for Ecological, Economical, Efficient, Electric Car-Sharing). The project lays the foundations for efficient and economically viable electric car-sharing systems by studying and solving the optimization problems arising in their design and operations.

The paper is organized as follows. Section 2 presents the exact model for the relocation problem, and Section 3 describes an auxiliary model for overnight relocation. Section 4 describes the heuristic approaches. The computational experiments and results are described in Section 5. Section 6 presents a graphical tool for displaying the solution computed on an electronic map. Finally, Section 6 draws the conclusions and presents possible future extensions of the work.

2 Relocation Model for Operating Hours

This section describes a Mixed-Integer Programming (MIP) model representing the operations of a one-way Ecar-sharing system. The system consists of a set of stations, where customers can pick-up or return a vehicle, and where idle vehicles are parked and recharged. We consider a reservation-based system, where all the customer requests must be performed in advance (say, the day before) and the users are notified within a prescribed deadline

(say, one hour after the requests period closing) about the acceptance or not of their request based on the available resources. To this end, each request is associated with a *profit* including the revenue for its service, but possibly also some other system-related or customer-related factors, such as bonuses to increase customers retention. The objective of the car-sharing provider is to maximize the profit associated with served requests. This is accomplished by choosing the requests to be served, by assuring the vehicles have enough battery charge before starting a trip, and by relocating idle vehicles. Relocations is performed by the so-called relocators, who drive the vehicles where needed.

More in detail, let H be the set of EVs and Q the set of relocators. The car-sharing system is modeled as a time-space network where both $h \in H$ and $q \in Q$ move, and $\{0, \dots, t, \dots, T_{\max}\}$ is the discretized set of time instants. Vehicles can either move only in time (i.e., wait at a station), or move in time and space when used by a customer or moved by a relocator. Similarly, relocators move in time only, or move in time and space when relocating a shared vehicle (in this case they travel using one of the vehicles, and each vehicle can transport at most B relocators) or alone (i.e., by walking, bike, public transportation, etc.). The model decides the optimal initial station for each vehicle at the beginning of the day; this information can be used to organize overnight relocation. The models also decides the optimal initial station for each relocator. Therefore, the model can manage every decision that may have an impact on the final daily performance.

Let S be the indexed station set. Each station $i \in S$ can accommodate and charge a limited number C_i of EVs. The node set V of the time-space network is defined by $S \times \{0, \dots, t, \dots, T_{\max}\}$. In the arc set A , each arc $a = (i_t, j_{t'})$ connects a node i_t to a node $j_{t'}$, with $t' > t$. When traversed by a vehicle, each arc a is associated with an energy variation c_a , which can be either positive or negative, as well as a profit (or cost) p_a , and a positive demand d_a for arcs used by customers. In detail, the arc set A is partitioned into several subsets of specific arcs as follows:

- A_c : arcs $a = (i_t, j_{t'})$, encoding the customer requests. Each of these arcs corresponds to a request starting in a pick-up station i at time t and ending in a return station j at time t' . Each time the arc is traversed by a vehicle (i.e., the associated request is satisfied, up to the demand $d_a > 0$), the vehicle energy variation is $c_a < 0$, and the profit earned by the service provider is $p_a > 0$;
- A_w : arcs $a = (i_t, i_{t+1})$ where either vehicles or the relocators wait at a station i between periods t and $t + 1$. These arcs are associated with a positive variation of the energy $c_a > 0$ when traversed by a vehicle, and do not provide profit, hence, $p_a = 0$. There are $O(|S|T_{\max})$ of such arcs in the time-space network (see 2.1 for more details).
- A_r : arcs $a = (i_t, j_{t'})$ where vehicles are relocated from station i at time t to station j at time t' (at least one relocator is onboard). These arcs are defined for each pair of stations and each pair of time instants t, t' such that t' is the first instant (in the time discretization) which allows to reach j when leaving i at time t . These arcs are associated with a variation of the energy $c_a < 0$ when traversed by a vehicle, and are associated with a cost for the service provider, hence, $p_a < 0$. There are $O(|S|^2T_{\max})$ of such arcs in the time-space network.
- A_t : transfer arcs $a = (i_t, j_{t'})$ travelled by relocators when they do not move on a shared vehicle or wait at a station. Similarly to the relocation arcs, they are defined for pair of stations and time intervals depending on the speed of transportation mode

of the relocators. These arcs are associated with a null variation of energy $c_a = 0$ and null profit $p_a = 0$. There are $O(|S|^2 T_{\max})$ of such arcs in the time-space network.

Three sets of decision variables are used:

- Vehicle arc variables: x_a^h , $a \in A_r \cup A_w \cup A_c$, taking value 1 when vehicle h travels on arc $a = (i_t, j_{t'})$ and 0 otherwise;
- Relocator arc variables: y_a^q , $a \in A_r \cup A_w \cup A_t$, taking value 1 when relocator q travels on arc $a = (i_t, j_{t'})$ and 0 otherwise;
- Vehicle battery charge variables: z_t^h , denoting the charge of vehicle h battery at time t , where z_0^h is the initial battery charge.

The MIP model is as follows

$$\max \sum_{h \in H} \sum_{a \in A_c \cup A_r} p_a x_a^h \quad (1)$$

$$\text{s.t.} \quad \sum_{h \in H} x_a^h \leq d_a \quad a \in A_c \quad (2)$$

$$\sum_{h \in H} \sum_{a=(i_t, i_{t+1}) \in A_w} x_a^h \leq C_i \quad i \in S, 0 \leq t < T_{\max} \quad (3)$$

$$\sum_{h \in H} \sum_{a \in \delta^+(i_0) \cap (A_c \cup A_w \cup A_r)} x_a^h \leq C_i \quad i \in S \quad (4)$$

$$\sum_{i \in S} \sum_{a \in \delta^+(i_0) \cap (A_c \cup A_w \cup A_r)} x_a^h = 1 \quad h \in H \quad (5)$$

$$\sum_{a \in (\delta^+(i_t)) \cap (A_c \cup A_w \cup A_r)} x_a^h = \sum_{a \in (\delta^-(i_t)) \cap (A_c \cup A_w \cup A_r)} x_a^h \quad h \in H, i \in S, \quad (6)$$

$$\sum_{i \in S} \sum_{a \in \delta^+(i_0) \cap (A_r \cup A_t \cup A_w)} y_a^q = 1 \quad q \in Q, 1 < t < T_{\max} \quad (7)$$

$$\sum_{a \in (\delta^+(i_t)) \cap (A_r \cup A_t \cup A_w)} y_a^q = \sum_{a \in (\delta^-(i_t)) \cap (A_r \cup A_t \cup A_w)} y_a^q \quad q \in Q, i \in S, 1 < t < T_{\max} \quad (8)$$

$$(\mathbf{x}, \mathbf{z}) \in \mathcal{B} \quad (9)$$

$$\sum_{h \in H} x_a^h \leq \sum_{q \in Q} y_a^q \quad a \in A_r \quad (10)$$

$$\sum_{q \in Q} y_a^q \leq B \sum_{h \in H} x_a^h \quad a \in A_r \quad (11)$$

$$x_a^h \in \{0, 1\} \quad h \in H, a \in A_c \cup A_w \cup A_r \quad (12)$$

$$y_a^q \in \{0, 1\} \quad q \in Q, a \in A_r \cup A_t \cup A_w \quad (13)$$

The objective function (1) maximizes the profit associated with the customers requests that are satisfied, minus the cost for relocation of vehicles. Constraints (2) guarantee that the customers demand associated with each arc in A_c is not exceeded. Constraints (3) ensure that the number of vehicles parked (and recharging) at each station at any time

$t > 0$ does not exceed the capacity, while (4) impose the same condition for instant $t = 0$ (i_0 denotes the node associated with the initial time instant for station i). Constraints (5) impose that each vehicle departs from only one station in the time-space network at time 0, and Constraints (6) impose vehicle flow conservation at the network nodes. Constraints (7) and (8) impose the same conditions for the relocators. Constraints (9) require that the set of trips assigned to each car is feasible with respect to that car battery charge. Actual implementations of such constraints are discussed next. Constraints (10) and (11) are used to match vehicles and relocators on relocation arcs A_r : (10) impose that the number of vehicles traveling on a relocation arc $a \in A_r$ is not larger than the number of relocators (i.e., relocators may either drive the vehicle or travel as passengers), and (11) impose that no more than B relocators can travel on a vehicle.

2.1 Battery Charge Constraints

In order to model constraints (9), we introduce a continuous variable z_t^h , $t = 1, \dots, T_{max}$, $h \in H$, denoting the battery charge of vehicle h at time t . At each time instant, the battery charge of each vehicle must be non-negative, and cannot exceed a maximum level Z_{max} , as expressed in the following constraints

$$0 \leq z_t^h \leq Z_{max} \quad h \in H, t \in \{1, \dots, T_{max}\} \quad (14)$$

Next, for a given vehicle h , we have to impose an upper bound on the battery level at each time t , which is defined as the battery level at any previous instant τ , plus the charge and discharge of the battery during the path on the time-space network from τ to t , as expressed by the following constraints

$$z_t^h \leq z_\tau^h + \sum_{a=(i_\theta, j_\rho) \in A_c \cup A_w \cup A_r: \tau \leq \theta < \rho \leq t} c_a x_a^h \quad h \in H, t \in \{1, \dots, T_{max}\}, \tau < t \quad (15)$$

Note that, since an arc $(i_t, j_{t'})$ can connect two non-consecutive time instants in the time discretization (i.e., $t' > t+1$), it is not possible to express (15) for consecutive time instants only but, for a given t , all previous instants τ have to be considered. The effect of (14) and (15) is to allow a vehicle to reach the maximum charge level when it is waiting at a charging station, and then to remain at the station (i.e., traversing arcs in A_w in the time-space network), without further charging, so that the maximum charge level Z_{max} is not exceeded. Parameter z_0^h defines the initial charge for each vehicle $h \in H$ at the beginning of the time horizon. i.e., for $t = 0$.

Preliminary numerical testing showed that the large number of constraints (15) affects the capability to solve the model. The constraint generation is demanding both in terms of CPU time and memory required. Therefore, an alternative formulation was developed, replacing constraints (15) with the following

$$z_t^h = z_0^h + \sum_{a=(i_\theta, j_\tau) \in A_c \cup A_w \cup A_r: \theta < \tau \leq t} c_a x_a^h \quad h \in H, t \in \{1, \dots, T_{max}\} \quad (16)$$

However, since the battery of an EV cannot exceed Z_{max} , (16) would prevent fully charged EVs from staying at stations (overcharging) and force them to move uselessly. In order to avoid this undesired effect, each wait arc $a = (i_t, i_{t+1})$, denoting wait and recharge

at node i , is paired with a (new) parallel wait arc $\bar{a} = (i_t, i_{t+1})$, with $c_{\bar{a}} = 0$, $p_{\bar{a}} = 0$. These new waiting arcs allow the EV to stay at station i without increasing the vehicle battery charge.

Comparing the two alternatives, constraints (15) are $|H|T_{max}(T_{max} - 1)/2$ and ask to define $A_w = |S|T_{max}$ wait variables; (16), are $|H|T_{max}$ and ask to define $A_w = 2|S|T_{max}$ wait variables. Note that the two ways of modeling the battery charging are not equivalent: the former allows the battery level to reach any value between 0 and Z_{max} , while the latter only allows to charge by exactly the energy c_a associated with the arc a , so the maximum charge level may not be reached. Testing proved that the introduction of new variables is worthy both in terms of computing time and memory usage, and that the two alternatives for modeling the battery charge are equivalent for practical purposes. An experimental comparison of the two formulation is discussed in Section 5.2.1.

3 Relocation Model For Non-Operating Hours

The model described in Section 2 computes the optimal initial position for each vehicle in the network. This optimal initial distribution can be obtained through an overnight relocation performed when the system is not operating. An additional model was developed, in order to schedule the staff operations during the night.

Let R_i be the number of EVs that are requested at a station i at the beginning of the next day. We operate on the same network described beforehand, however, no travel arc A_c is available during the night. The initial position o_h and the initial battery z_h^0 of vehicle h are given by the system configuration at the end of the day.

The MIP model for relocation in non-operating hours is as follows

$$\max \quad Z \quad (17)$$

$$\text{s.t.} \quad Z \leq z_h^{T_{max}} \quad h \in H \quad (18)$$

$$\sum_{a \in (\delta^+(o_h)) \cap (A_w \cup A_r)} x_a^h = 1 \quad h \in H \quad (19)$$

$$\sum_{a \in (\delta^-(i_{T_{max}})) \cap (A_w \cup A_r)} x_a^h = R_i \quad h \in H, i \in S \quad (20)$$

$$(3) - (13)$$

The objective function (17) maximizes Z , which is defined in Constraints (18) as the charge level of the EV with the most depleted battery at the end of the working day. An alternative is to maximize the sum of the battery levels for all vehicles. Constraints (19) impose that each EV departs from its initial position. Constraints (20) impose to achieve the required EV distribution.

This formulation allows to stop relocations after a specified time, therefore taking into account the length of a shift for the operators. When relocation operations are over, the model still allows the EVs parked at charging stations to be recharged for the next period (in the model, relocation arcs are disabled after a certain time). The EVs, from that moment on, can only stay at their station and recharge. Clearly, allowing a larger relocation time leads to better solutions, since more time is available for both relocation and recharging. If relocation time drops below a minimum threshold, there might not be enough time to move the EVs to their final position, thus making the problem infeasible.

4 Heuristic Algorithms

Finding the optimal solution of the relocation model presented in Section 2 may be a computationally impractical task. This section describes model-based heuristics, also known as matheuristics (see, e.g., Boschetti et al. [4], Puchinger and Raidl [19], Archetti and Speranza [2]), developed for solving the relocation model for operating hours, i.e., (1)–(13), on a larger scale. Computing a fast initial feasible solution can be also useful in helping a general-purpose solver in finding the optimal solution of the model. Computational tests on several instances showed that such initial solution can be computed really fast and provides a remarkable decrease of the CPU total time when used to initialize the search of the solver. For the testing we used the CPLEX MIP solver by IBM.

4.1 Removing Relocation - Reducing Relocation Density

Removing or reducing the relocation arcs in A_r defines a model that remains feasible, since relocators can move in the time-space network on wait arcs, while EVs can either serve a customer or move along wait arcs. The model is clearly sub-optimal because the opportunity to relocate vehicles to increase the profit from served requests is lost or reduced. We considered removing or reducing relocation separately:

- By completely removing relocations, the model is significantly easier to be solved. We use the solution computed in this case to initialize the MIP solver with a feasible solution when tackling model (1)–(13). We observed that the built-in heuristics that CPLEX uses are usually slower in finding a feasible solution.
- The effect of reducing relocation arcs on the solution quality depends on how many relocation arcs are removed. Carlier et al. [8] studied a traditional vehicle management problem and proposed to perform relocations only at certain time steps (e.g., every 30 minutes). Reducing in this way the relocation density simplifies the problem, however the obtained solution can be far from the optimal one. In order to find a good trade-off between time and solution quality, random instances were tested with different densities of the relocation arcs. The results obtained using such approaches are reported in Section 5.3.

4.2 Rolling Horizon: Gradual Inclusion of Relocation Arcs

The rolling horizon approach is widely used in planning problems when the same group of decisions must be repeated over time. Instead of dealing with the problem over the whole planning horizon, the time horizon is split into smaller sub-periods. The first sub-periods is solved to optimality and the associated decision variables are fixed, while variables associated with the subsequent periods are relaxed or reduced. Afterwards, the problem is solved in sequence for the next sub-periods in a rolling fashion.

In detail, the rolling horizon algorithm that we apply to solve the relocation model for operating hours works as follows. The original formulation is split into ρ sub-periods and relocation arcs associated with sub-periods after the first one are disabled. The resulting MIP model is solved to optimality, and the value of all variables associated with the first sub-period is fixed. Then the process is iterated for the next sub-period. As the rolling horizon proceeds, the algorithm may update the decisions for future periods, while an increasing set of decision variables remains fixed, making the resulting MIP problems smaller at each iteration.

Figures 1 and 2 represent the procedure in the time-space network. The horizontal axis denotes time steps, and the vertical axis denotes the stations. The vertical line divides the time into 2 sub-periods. The relocater path is represented by a straight line and 3 EVs-paths are depicted by dashed lines. Note that some of the trips requested by customers are two-way, so there can be travel arcs that remain at the same station. When a relocation is performed, both the relocater and the EV travel the same arc. At the first iteration (Figure 1), no relocation is permitted in the second sub-period (sp_2), and the resulting problem is optimally solved for the first sub-period sp_1 . At the next iteration, the solution is fixed for sub-period sp_1 , and relocation is activated in the second sub-period.

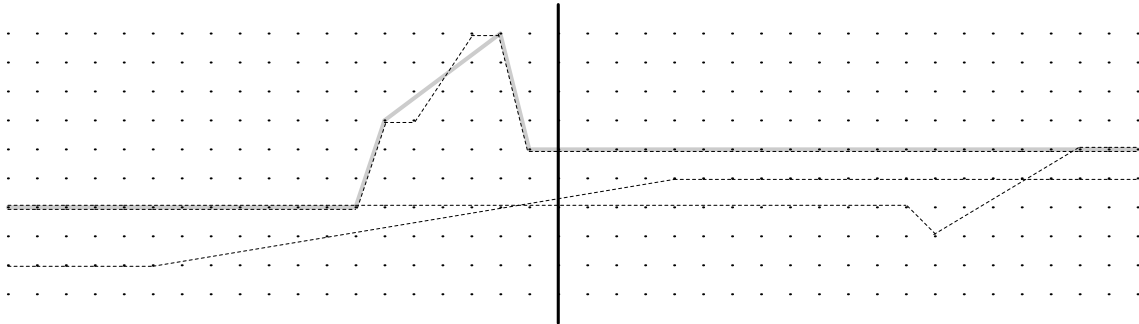


Figure 1: Visualization of solution of subproblem sp_1

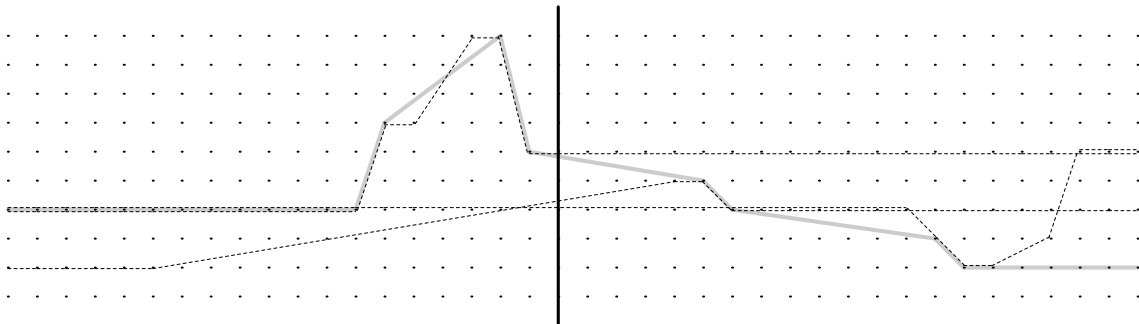


Figure 2: Visualization of solution of subproblem sp_2

The ρ parameter determines the trade-off between size of the models to be solved and quality of the solution. As the number of sub-periods increases, the computational time decreases. However, smaller sub-periods lead to potentially worse solutions.

A rolling horizon approach is very useful when the relocation problem is solved on-line, and not all the customer requests are known in advance. In this case, it is preferable to find the solution of a limited upcoming sub-period, without considering relocations that are late in time, and may be not necessary due to yet unknown customer requests.

5 Computational Experiments

5.1 Test Instances

This section describes the process of data gathering and filtering, which led to the definition of the realistic instances to test the mathematical models and heuristic algorithms. The

presented experiments have two purposes: first, we want to assess the difficulty in solving the presented models with a general purpose MIP solver, and to evaluate the effectiveness of the heuristic algorithms for difficult instances; second, we want to evaluate under which conditions the relocation is significant for improving the performance of a car-sharing system. The algorithms are incorporated in a web tool for planning relocation of cars in a practical context which is briefly illustrated in the Appendix.

ICVS (Intelligent Community Vehicle System) ICVS was a Honda Motor car-sharing initiative in Singapore, which is not operating anymore. It used hybrid Honda Civic vehicles, parked in 14 stations. The ICVS allowed customers to use the vehicles for one-way trips. All customer travels performed from March 2003 until January 2006 were inserted into a dataset. These data are currently the only one publicly available for car-sharing problems and have already been used as test instances in previous papers, see, e.g., Nair and Miller-Hooks [18], Kek et al. [13] and Kek et al. [12].

ICVS cars could rely on a tank of fuel, therefore they could be used for long journeys. However, the battery itself would not have been enough for such trips. Given that the proposed models deal with electric cars solely, the dataset was modified accordingly, and trips longer than 6 hours were deleted. The maximum battery capacity was assumed to be sufficient to cover 150 km. The recharge speed may vary depending on the infrastructures; since we consider station-based systems, we assume that high power chargers are installed. The recharge speed is modelled as a linear function of time.

Model (1)-(13) aims to support operational vehicle management throughout the day, while other activities (such as maintenance) are carried out by night. The vehicle trips are considered in a 10-hours interval, namely from 8.00 to 18.00. In Table 2 we report the seven days with largest number of trips performed in the original dataset, before and after filtering of trips that were too long or exceeded operating hours.

<i>Day</i>	<i>Original</i>		<i>Duration < 6h</i>		<i>Duration < 6h; 8.00 – 18.00</i>	
	<i>#Trips</i>	<i>Rank</i>	<i>#Trips</i>	<i>Rank</i>	<i>#Trips</i>	<i>Rank</i>
1	98	1	60	1	50	1
2	97	2	56	2	50	2
3	96	3	54	3	47	3
4	96	4	53	4	45	4
5	95	5	52	5	45	5
6	95	6	52	6	44	6
7	95	7	52	7	44	7
Dataset Total	45570	Dataset Total	23212	Dataset Total	19633	Dataset Total

Table 2: Number of customer trips; 7 days with highest demand

The 10 hours horizon was split in 40 intervals of 15 minutes. This discretization represents a convenient trade-off between size of the model and accuracy of the instance: after removing the trips longer than 6 hours, average trip duration was 128 minutes.

Network specification - Base Case We constructed a *base-case* time-space network based on ICVS data, with the following characteristics:

- 14 stations with finite capacity;
- Profit of customer arcs $p_a, a \in A_c$: 0.2 per km and 0.1 per minute, between 50 and 125 arcs per instance;

- Profit (negative) of relocation arcs $p_a, a \in A_r$: -1.5 per km;
- Profit of transfer arcs $p_a, a \in A_t$: 0;
- 20 EVs
- 2 relocators
- Battery capacity: 150 km; the battery is recharged of the 10% in each time step

The profit of customer arcs was computed as a function of distance and time duration of the trips. The values have been obtained from the price of existing car-sharing services. The duration and distance associated with relocation arcs were computed from the ICVS data. There is a cost associated with each relocation, hence the corresponding arcs have a negative profit p_a , proportional to the distance. The cost per km considers the EV usage and the labor cost.

Working days corresponding to the filtered data proved to be too easy for the model, because only two days had more than 50 trips. Therefore, test instances were created by randomly picking customer trips from the dataset. The instances have a different number of customer arcs A_c (denoted as *Size* of the instance in the following). For each size, 6 instances were created.

5.2 Exact solution

This section summarizes the results obtained by solving to optimality the previously described models. Each instance was tackled with a time limit of 1 hour by means of CPLEX 12.6 on a Xeon E3-1220 processor clocked at 3.10 GHz, with 8 GB RAM.

In the following tables, the computational times are reported in seconds and the value of the solution obtained by each method is denoted by *Sol*.

5.2.1 Battery charge constraints

The two alternative formulations (14), (15) and (14), (16) of battery charge constraints (described in Section 2.1) were tested on the base instances, so as to define the best performing model to be used in the subsequent experiments. Table 3 displays the best solution values obtained and the associated computing times. The average values on 6 instances with the same size are reported. The smallest instances (size 50 and 75) show a remarkable decrease in computing time if constraints (16) are used in place of (15). The average solution value is the same, with the exception of one instance of size 75, for which the optimal solution was not found with the formulation (14), (15).

<i>Size</i>	<i>Battery constr</i>	<i>Sol</i>	<i>Time</i>
50	(14)-(16)	724.33	75.83
	(14)-(15)	724.33	840.50
75	(14)-(16)	1057.50	95.67
	(14)-(15)	1055.83	2985.33

Table 3: Comparison of computational time [s] between formulations.

On top of the additional computational time, formulation (14), (15) also requires a larger amount of memory. Since Constraints (14), (16) turn out to be much more efficient than (14), (15), they were chosen to be used in the following experiments.

5.2.2 Base instances

Table 4 reports the results obtained by solving model (1)–(13) on the base instances. Each row of the table reports average results over 6 instances. Column *Size* reports the number of available customer arcs; columns *Cons* and *Vars* report the number of constraints and variables of the optimization model, respectively; column *Gap* the percentage optimality gap of CPLEX at the time limit; *Nodes* is number of explored nodes; *Time* is the computing time required; *RootGap* reports the gap at the root node of CPLEX. Finally, *Solved* is the number of instances solved to optimality within time limit.

<i>Size</i>	<i>Cons</i>	<i>Vars</i>	<i>Sol</i>	<i>Gap</i>	<i>Nodes</i>	<i>Time</i>	<i>RootGap</i>	<i>Solved</i>
50	27268	336717	724.33	0.00%	0.00	75.83	0.00%	6/6
75	27293	337267	1057.50	0.00%	0.00	95.67	0.00%	6/6
100	27318	337817	1197.33	0.01%	3568.00	739.00	0.71%	5/6
125	27343	338367	1249.83	0.16%	2055.67	1068.50	0.95%	5/6

Table 4: Average performances of the algorithm, base instances

All instances with up to 75 customer arcs are easily solved at the root node by the CPLEX branch-and-bound algorithm. For larger number of customer arcs, the solver has to start branching, and can solve 10 out of 12 instances within time limit.

Concerning the system performance and the relevance of relocation, in Table 5 we report the number of customer arcs served when relocation is not active, column A_c (*no rel*), and the computational time required by CPLEX. The same information is reported when relocation activities are possible. Clearly, the presence of relocation arcs makes the mathematical model more challenging, however the obtained solution enables to serve more customer requests. Finally, the number A_r of relocation arcs used and the percentage increase in profit are reported.

<i>Size</i>	A_c (<i>no rel</i>)	<i>Time</i> (<i>no rel</i>)	A_c (<i>rel</i>)	<i>Time</i> (<i>rel</i>)	A_r	% <i>Profit</i>
50	42.83	4.79	44.83	75.83	2.67	1.93%
75	56.17	9.97	61.33	95.67	4.83	2.50%
100	61.17	12.33	63.00	739.00	6.00	2.35%
125	60.17	42.95	65.67	1068.50	7.83	3.56%

Table 5: System performance with and without relocation, base instances

5.2.3 Sensitivity to relocation cost

In this section we discuss the effect of relocation cost and other features of the instances related to the relocations.

In the base-case we considered a quite large cost for relocation, and according to the results in Table 5 relocations are performed with low occurrence.

The actual cost of relocations in a real-world application depends on the cost structure of the car-sharing provider, and in particular on the allocation of labour cost, which can be a large component in relocation cost. When labour cost is not allocated as a direct cost to relocations (because, e.g., relocations are performed by personnel which is already hired for other duties and can be used also for relocations), we can assume smaller relocation costs. Tables (6) and (7) report information for the case in which the relocation costs of each arc has been reduced to 1.

<i>Size</i>	<i>Sol</i>	<i>Gap</i>	<i>Nodes</i>	<i>Time</i>	<i>Root node gap</i>	<i>Solved</i>
50	736.17	0.00%	0.00	167.83	0.00%	6/6
75	1084.50	0.00%	0.00	247.17	0.00%	6/6
100	1228.67	0.51%	78.17	820.00	1.54%	5/6
125	1282.83	0.85%	376.67	1403.67	1.17%	4/6

Table 6: Average performances of the algorithm, low relocation cost

<i>Size</i>	<i>A_c (no rel)</i>	<i>A_c (with rel)</i>	<i>A_r</i>	<i>% Profit</i>
50	42.83	48.33	5.33	3.64%
75	56.17	65.17	11.33	5.11%
100	61.17	66.00	12.33	5.14%
125	60.17	68.33	14.50	6.49%

Table 7: System performance with and without relocation, low relocation cost

According to Table 6, the instances become more challenging, with only 9 out of 12 instances solved to optimality when 100 or 125 customer arcs are considered. A possible explanation for the fact that instances are easier when relocation cost are large is that, in this case, several variables related with relocation arcs can be removed by the MIP solver preprocessing, because they could not appear in an optimal solution. Instead, by reducing the relocation cost, many more relocation arcs become attractive and their possible inclusion in an optimal configuration complicates the solution of the model. Concerning the system performance and the relevance of relocation, we observe that the number of performed trips increases in a more sensitive way when relocation is allowed, and relocation can lead to a profit increase of more than 6% on average, for the instances with 125 customer arcs.

A second feature that determines the benefit of performing relocations is the ratio of customer arcs and available EV. On the one hand, when there are many EVs with respect to the customer arcs, a careful initial allocation of the EV may reduce the need for relocation, because we may expect to have available vehicles where needed. Trivially, if the number of available EVs equals the number of customer arcs (and no capacity constraint is active at the stations), no relocation is needed. On the other hand, when there are very few EVs with respect to the customer arcs, again relocation is not attractive. Indeed, when a trip is ended, it may be preferable to keep the EV waiting at a station for the next request, instead of moving it to a different station. Relocation looks appealing in intermediate situations, as shown in Figure 3, where we test the base instances for different numbers of EVs. The figure reports the average number of performed relocations as a function of the ratio between customer arcs and EV. Initially, as the ratio A_c to EV grows, the average number of relocation increases, but only up to a certain threshold of about 4 customer arcs per EV. When the ratio grows over such threshold, the number of relocations decreases.

5.2.4 Sensitivity to battery capacity

In this section we evaluate the effect of having smaller or larger battery capacities on the difficulty of the instances, and on the profit that can be obtained by the car-sharing provider. In the base instances, the maximum battery capacity was set to 150 Km. The model was tested by considering 4 different values of battery capacity: *scarce* (autonomy

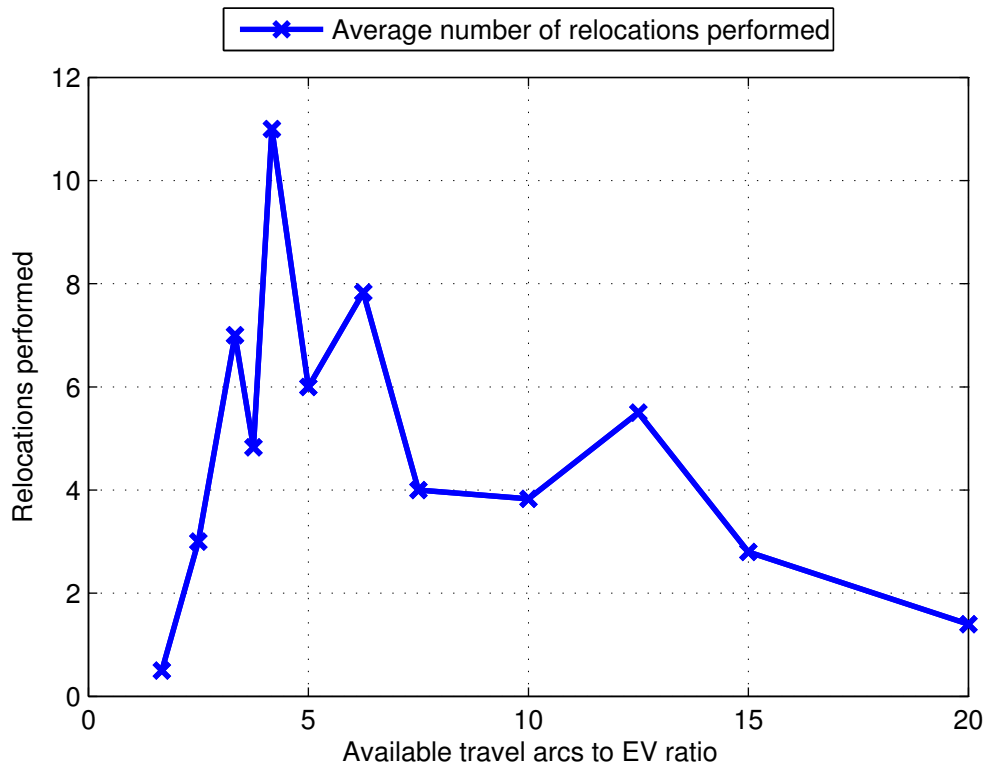


Figure 3: Base instances: average number of relocations and ratio A_c to EV.

of 80 Km), *less* (100 km), *base* (150 Km) and *more* (200 km).

In Table 8 we report the average optimality gaps and computing times for different battery capacities. The table shows that reducing the battery has a big impact on the difficulty of the instances. For low values of the battery capacity, the algorithm rarely solves the instance to optimality, especially when the size is large. The optimality gaps are larger than 10% on average for instances with 125 customer arcs. Instead, when considering the *more* battery capacity, all instances are solved to optimality.

Concerning the profit, in Table 9 we observe that it is not increased when going from *base* to *more* capacity, showing that at the base level the car-sharing operator is already capable to make the maximum profit from the system and larger battery capacity seems not required. For smaller levels of battery capacity, the obtained profit seems to be reduced. However, since most of the small-capacity instances are not solved to optimality, we cannot conclude with high certainty that such profit reduction is only due to the reduced battery capacity.

5.2.5 Overnight relocation

The overnight relocation model of Section 3 is intended to relocate EVs at the end of the day, in order to be available at the requested stations at the beginning of the next day. Hence, the initial network of the overnight relocation model is given by the position and battery level of each vehicle as specified by a solution of model (1)-(13) at the end of a working day. The target is the distribution of the vehicles at the beginning of the next

Size	Gap				Time			
	Scarce	Less	Base	More	Scarce	Less	Base	More
50	0.21%	0.00%	0.00%	0.00%	1619.67	136.67	75.83	108.83
75	4.44%	1.12%	0.00%	0.00%	3348.83	2284.50	95.67	100.83
100	9.41%	3.60%	0.01%	0.00%	3600.00	3188.33	736.33	114.67
125	10.51%	5.31%	0.16%	0.00%	3600.00	3600.00	1036.83	133.67

Table 8: Average optimality gap and computing time for different battery capacities.

Size	Sol			
	Scarce	Less	Base	More
50	706.50	708.33	724.33	724.33
75	995.50	1032.50	1057.50	1057.50
100	1082.83	1148.67	1197.33	1197.67
125	1125.33	1185.67	1249.83	1252.67

Table 9: Average profit for different battery capacities.

working day.

This is an operational model for which several input decisions, depending on the service design, can be varied: schedule of the relocation (to be performed immediately after the previous day or just before new day operations are started), number of relocators, etc. In our experiments we considered 20 EVs, as in the base case. The time available for the relocation operations is set between 2 and 5 hours, i.e., 8 to 20 time steps, and each value was tested on 6 different instances. When a shorter time for relocation is allowed, we consider a larger number of relocators. We assume relocation is performed just after the working day, i.e., when the EVs have the lowest battery level. For the rest of the night, each EV recharges at the destination station.

In Table 10, we consider groups of 6 homogeneous instances, where each group is defined by the number of relocators and by the relocation duration reported in column *Time Steps*. Note that the target distribution may be unreachable with the available resources (i.e., time and staff); the number of instances where this happens is displayed in the *Infeasible* column. On top of that, not all the feasible instances may not be solved to the optimality within 1 hour of computing time. The *Solved* column counts the instances out of the feasible ones for which the optimal solution was found. The table also reports the number A_r of used relocation arcs.

Instance		Performance			
Relocators	Time Steps	Average time	Infeasible	Solved	A_r
1	20	145.50	4	2	10.00
1	24	590.00	1	4	11.25
2	12	95.67	1	5	11.40
2	16	726.33	0	6	13.33
2	20	2232.60	0	5	14.40
3	8	16.66	1	5	11.20
3	12	324.33	0	6	14.50

Table 10: Performances of the overnight relocation with different resources (i.e time and number of operators)

The tables clearly shows that larger problems, having more relocators or more time steps for the relocation, are more difficult to solve.

5.3 Heuristic Algorithms

We compare the heuristic approaches described in Section 4 with the exact solution of models (1)–(13). When considering a reduction of relocation arcs, we tried to halve them, i.e., relocation is performed every half hour, and to reduce them to a quarter, i.e., relocation performed only once per hour. In the following, we denote the corresponding heuristic solutions as *dens2* and *dens4*, respectively. In the rolling horizon algorithm, the planning horizon was divided into 2 intervals (denoted as *rolling2* in the following) and 4 intervals (denotes as *rolling4*). The quality of the heuristic solutions of the base case instances with size 100 and 125 are compared to the exact solutions. All algorithms are run with a time limit of 1 hour. We also considered larger instances with size 200, where the heuristics are run with a time limit of 1 hour, while the exact algorithm was given a longer time limit of 4 hours (so as to compute a near optimal solution value and a good upper bound to allow a meaningful evaluation of the heuristic methods). Average results for groups of 6 homogeneous instances are reported in Table 11 where, for each solution method, we give the average best solution value (*Sol*), the associated computing time, the optimality gap and the subproblem gap. The first gap is equal to the gap between the upper bound associated with the solution of the exact model, and the solution value obtained by the heuristic method itself. The second gap corresponds to the average optimality gap of the subproblems associated with the heuristic approaches. Namely, it is the optimality gap for the models with a reduced number of relocation arcs, and it is the average optimality gap for each subproblem associated with the *rolling* algorithms. If it is equal to zero, it means that the subproblems were solved to the optimum.

For instances of size 100, the *dens* solutions can be obtained with a negligible reduction of computing time, and a loss of profit not exceeding 0.6%. The *rolling* approach is more interesting: with a loss of profit of about 1.5% for *rolling4*, the computing time is reduced by an order of magnitude. For instances of size 125, the *rolling2* approach approximately halves the computing times, at cost of a 1.1% reduction of profit. For large instances of size 200, the use of the heuristic algorithms has a significant impact on the computational tractability of the model. At basically no decrease in terms of average solution quality, computing times are reduced by an order of magnitude and more. On instances of size 200, the *rolling* approach finds solutions of quality comparable to the ones obtained with the *density* approach in approximately half of the time.

We also tested the heuristic methods on the low battery instances of size 125 at the *Less* level, which are quite difficult to solve to optimality. As reported in Table 12, the battery limitation largely worsen the exact resolution, and both heuristics provide a sub-optimal solution better than the best solution found by CPLEX, in comparable computing time.

6 Conclusions and Future Research

In this paper we presented a mathematical model to manage the daily operations of an electric car-sharing system. The system under consideration is station-based, and unbalances in the electric vehicles distribution are managed by relocation of the vehicles, which is performed by a dedicated staff (i.e., an operator-based relocation strategy). Both exact and heuristic approaches were developed, and tested on a set of realistic instances obtained

<i>Size</i>	<i>Method</i>	<i>Sol</i>	<i>Time</i>	<i>Opt. Gap</i>	<i>Suprob. Gap</i>
100	exact	1197.33	739.00	0.01%	-
	dens2	1194.17	650.33	0.27%	0.05%
	dens4	1190.00	642.33	0.62%	0.01%
	rolling2	1189.17	225.17	0.69%	0.00%
	rolling4	1179.67	51.00	1.48%	0.00%
125	exact	1249.83	1068.50	0.16%	-
	dens2	1244.00	769.50	0.63%	0.08%
	dens4	1232.5	704.00	1.54%	0.08%
	rolling2	1236.00	593.33	1.26%	0.00%
	rolling4	1207.00	615.20	3.58%	0.00%
200	exact	1380.00	11544.40	0.44%	-
	dens2	1378.20	2311.20	0.57%	0.26%
	dens4	1375.40	2286.40	0.77%	0.58%
	rolling2	1378.60	894.60	0.54%	0.09%
	rolling4	1377.40	782.40	0.63%	0.04%

Table 11: Heuristic performances compared to exact algorithm

<i>Size</i>	<i>Method</i>	<i>Sol</i>	<i>Time</i>	<i>Opt. Gap</i>	<i>Suprob. Gap</i>
125	exact	1185.67	3600.00	5.31%	-
	dens2	1199.83	3600.00	4.18%	3.51%
	dens4	1196.50	3151.83	4.45%	2.50%
	rolling2	1194.17	2812.17	4.63%	1.85%
	rolling4	1192.00	3600.00	4.80%	1.05%

Table 12: Low-battery stress instances: heuristic solutions compared to exact solutions

from an existing car-sharing service. Furthermore, a model for the overnight relocations was described and tested in integration with the previous daily operations model. Together, the two models provide a comprehensive optimization tool to support the management of an electric car-sharing system.

The performed computational experiments showed that the introduction of relocation in a electric car-sharing system allows to achieve larger profits, in particular when there is a balance between the number of vehicles and the customers requests.

The proposed model is mainly operational. As a future research direction, we plan to develop an integrated model where strategic and tactical design decisions are simultaneously taken into account. In particular, the strategic decision considered will be related to the sizing and location of the charging stations. Moreover, further research may be devoted to different exact approaches that may allow for an increase in the size of solved problems.

Acknowledgements

This research is performed within the European project *e4-share* (Models for Ecological, Economical, Efficient, Electric Car-Sharing) funded by FFG (Austria) under grant 847350, INNOVIRIS (Belgium) and MIUR (Italy) via the Joint Programme Initiative Urban Europe (see <http://www.univie.ac.at/e4-share/> for more details). Research activities undertaken by Claudio Gambella in relation to this paper were entirely performed at DEI, University of Bologna.

The authors would like to thank Kelvin Cheu for kindly sharing the ICVS data, and two anonymous referees for insightful comments, which helped improving the presentation of the paper.

Appendix: Displaying the Solution

A tool for representing the solutions was developed by using Google Maps API [1]. Given the position of the stations, it is possible to show the path of EVs and relocators during operational activities with an *html* page.

Figure 4 depicts the path of one relocator on an example instance. Markers on the map show the waypoints; each waypoint is labelled with an increasing index that marks the order of the performed travels. If one station is visited more than once, markers “scatter” when clicked, allowing to check all the stops at that station. On the right side of the figure, the path is described textually, allowing to find the correct time to the ordinal indexes on the waypoints. A white waypoint marks the starting station of a relocation, whereas an orange waypoint represents the head of a transfer arc.

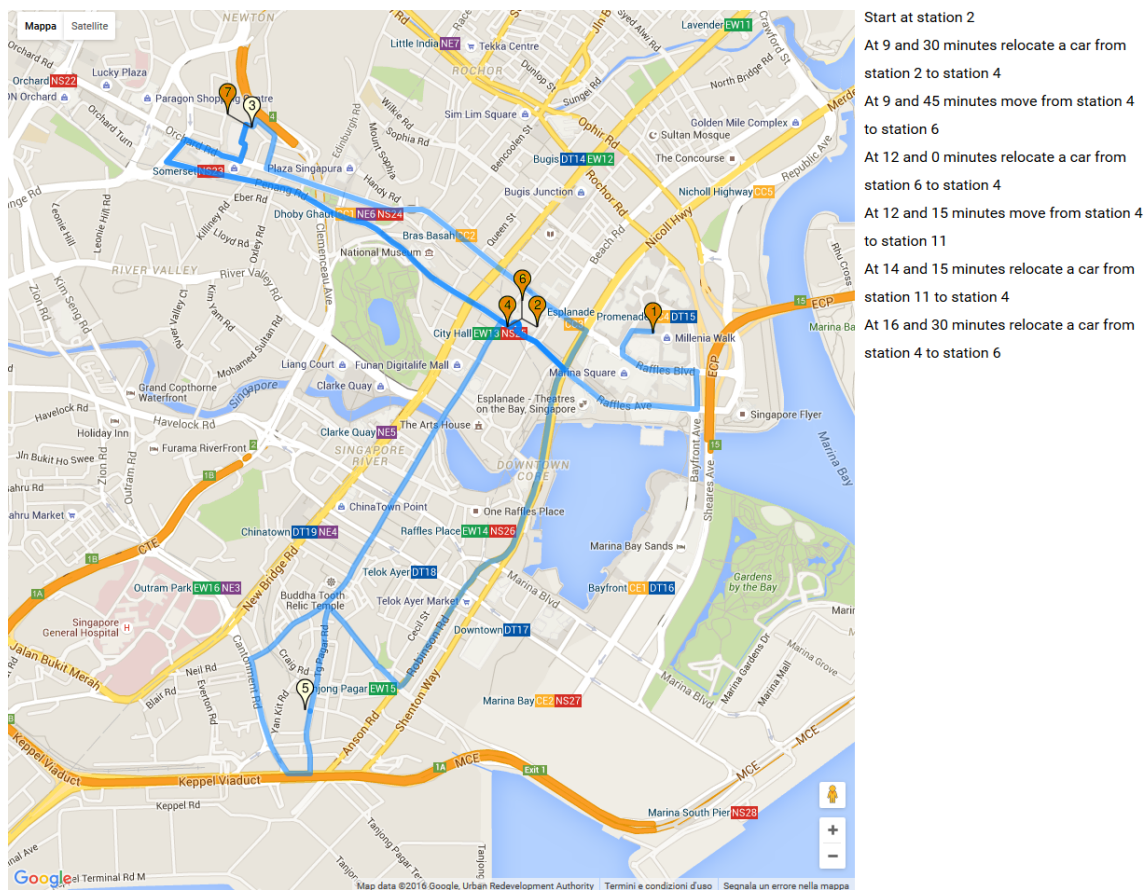


Figure 4: Path of one relocator

The path of an EV can be quite different, as shown in Figure 5. Yellow waypoints correspond to relocations and red waypoints indicate the trips performed: they are labelled with an increasing index showing the visit order. Travel arcs have an unknown route: the customer may follow whichever path he prefers to get from the starting point to the

destination. Such paths are displayed as arrows. On the other hand, relocations follow the shortest path, so the actual route is displayed. Note that some of the travels are two-way (i.e., starting and ending station are coincident).

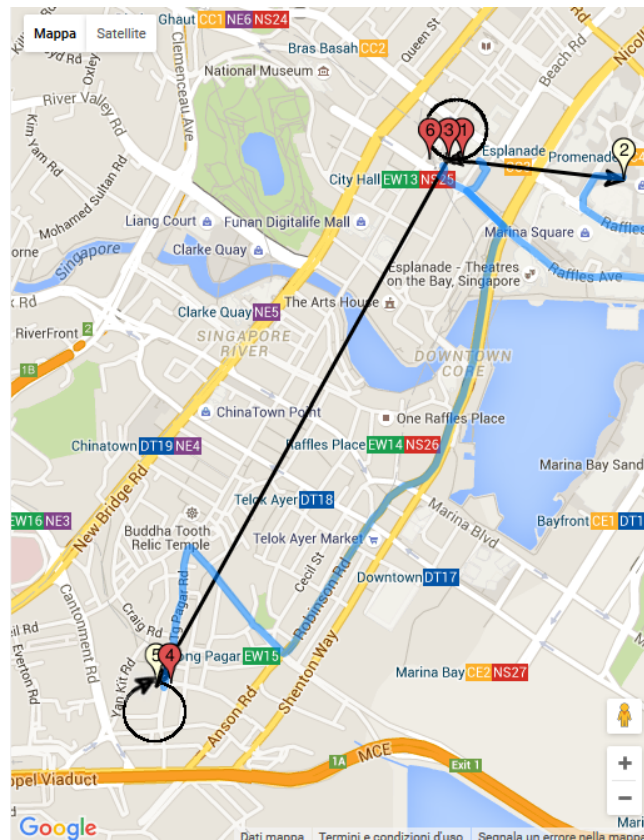


Figure 5: An EV Path

References

- [1] Google Maps API. URL <https://developers.google.com/maps/>.
- [2] Archetti, C. and Speranza, M. G. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246, 2014.
- [3] Barth, M., Todd, M., and Xue, L. User-based vehicle relocation techniques for multiple-station shared-use vehicle systems. In *Transportation Research Board, 80th Annual Meeting*, 2004.
- [4] Boschetti, M. A., Maniezzo, V., Roffilli, M., and Röhrler, A. B. Matheuristics: Optimization, simulation and control. In *International Workshop on Hybrid Metaheuristics*, pages 171–177. Springer, 2009.
- [5] Boyacı, B., Zografos, K. G., and Geroliminis, N. An optimization framework for the development of efficient one-way car-sharing systems. *European Journal of Operational Research*, 240(3):718–733, 2015.
- [6] Brandstätter, G., Gambella, C., Leitner, M., Malaguti, E., Masini, F., Puchinger, J., Ruthmair, M., and Vigo, D. Overview of optimization problems in electric car-sharing system design and management. In Dawid, H., Doerner, K., Feichtinger, G., and Kort, P. M., editors, *Dynamic Perspectives on Managerial Decision Making*. Springer, Berlin, 2016.
- [7] Bruglieri, M., Colorni, A., and Luè, A. The vehicle relocation problem for the one-way electric vehicle sharing: An application to the Milan case. *Procedia - Social and Behavioral Sciences*, 111:18–27, 2014.
- [8] Carlier, A., Munier-Kordon, A., and Klaudel, W. Mathematical model for the study of relocation strategies in one-way carsharing systems. *Transportation Research Procedia*, 10:374–383, 2015.
- [9] Clemente, M., Fanti, M. P., Mangini, A. M., and Ukovich, W. The vehicle relocation problem in car sharing systems: Modeling and simulation in a Petri net framework. *Lecture Notes in Computer Science*, 7927:250–269, 2013.
- [10] Hernández-Pérez, H. and Salazar-González, J. J. The one-commodity pickup-and-delivery traveling salesman problem: Inequalities and algorithms. *Networks*, 50(4): 258–272, 2007.
- [11] Jorge, D., Correia, G. H., and Barnhart, C. Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1667–1675, 2014.
- [12] Kek, A. G., Cheu, R. L., and Chor, M. Relocation simulation model for multiple-station shared-use vehicle systems. *Transportation Research Record*, 1986(1):81–88, 2006.
- [13] Kek, A. G., Cheu, R. L., Meng, Q., and Fung, C. H. A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 45(1):149–158, 2009.

- [14] Kim, J., Son, S.-Y., Lee, J.-M., and Ha, H.-T. Scheduling and performance analysis under a stochastic model for electric vehicle charging stations. *Omega*, 66:278 – 289, 2017.
- [15] Lee, J. and Park, G.-L. Planning of relocation staff operations in electric vehicle sharing systems. In *Lecture Notes in Computer Science*, volume 7803, page 256–265. Springer Science + Business Media, 2013.
- [16] Malaguti, E., Martello, S. and Santini, A. The traveling salesman problem with pickups, deliveries, and draft limits. *Omega*, 2017. <http://dx.doi.org/10.1016/j.omega.2017.01.005i>.
- [17] Millard-Ball, A., Murray, G., Ter Schure, J., Fox, C., and Burkhardt, J. Carsharing: Where and how it succeeds. Technical Report TCRP Report 108, TRB, Washington D.C., 2005.
- [18] Nair, R. and Miller-Hooks, E. Fleet management for vehicle sharing operations. *Transportation Science*, 45(4):524–540, 2011.
- [19] Puchinger, J. and Raidl, G. R. *Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification*, pages 41–53. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [20] Umetani, S., Fukushima, Y., and Morita, H. A linear programming based heuristic algorithm for charge and discharge scheduling of electric vehicles in a building energy management system. *Omega*, 67:115 – 122, 2017.