The Relocation Problem for the One-Way Electric Vehicle Sharing

Maurizio Bruglieri and Alberto Colorni
Dipartimento di Design – Politecnico di Milano, Via Durando 38/A, 20158 Milano, Italy

Alessandro Luè
Poliedra – Politecnico di Milano, via Giuseppe Colombo 40, 20133 Milano, Italy

Dipartimento di Design – Politecnico di Milano, Via Durando 38/A, 20158 Milano, Italy

1. INTRODUCTION

As a result of the current financial crisis and within the context of environmental and social changes, a collective reflection is taking place on how to design and build more sustainable a new kind of prosperity. Consequently, the idea of sharing objects and services has been gaining popularity [9, 30]. As regards urban mobility, the number and dimension of car sharing services have increased, especially in the last decades [39]. The earliest origins of car sharing, understood as an organized form of shared use of the car, can be traced back to 1948 when an association started a service in Zurich [38]. In the following years, particularly in the 1970s and 1980s, several car sharing systems were set up, but still on a small scale, with poor results [37]. The original idea has been gradually replaced by an offer structurally organized according to strict business criteria, to achieve economies of scale, which resulted in increased benefits to users in terms of low rates and diversification of the available fleet.

The design and management of a car sharing service raise several optimization problems, which have been tackled in the literature (e.g., [6, 21, 26]), in particular to determine the optimal size of the fleet and identify the location of the parking stations. For instance, in [16], the authors propose a methodology based on fuzzy logic algorithm, where the users' needs are modeled by way of suitable performance indicators, with the objective of ensuring the balance between costs, number of stations and level of service. The problems of the optimal location of the charging stations and of the optimal electric vehicle (EV) routing are considered in [40], where their mathematical programming formulations consider variants of well-known combinatorial optimization problems. Reference [24] introduced a tool for evaluating the performance of a network of car sharing stations in function of variations of the service demand. The tool is based on discrete event simulation and seeks to maximize the satisfaction level of the users and to minimize the number of vehicles.

Traditional car sharing services are based on the two-way (or round trip) scheme, where the user picks up and returns the vehicle at the same parking station. Some new services, such as Car2Go (www.car2go.com) or Autolib (www.autolib.eu/en/) permit also one-way trips, which allow

Received January 2013; accepted February 2014
Correspondence to: M. Bruglieri; e-mail: maurizio.bruglieri@polimi.it
DOI 10.1002/net.21585
Published online 27 November 2014

the user to return the vehicle in another station. The one-way scheme is more attractive for the users, but may lead to an unbalance between the demand and the availability of vehicles (e.g., near the railway stations at the beginning of a working day) or vice versa between the request for returning a vehicle and the availability of vacant parking lots. In such cases, the service provider may develop strategies to reallocate the fleet and restore a better distribution of the vehicles among the parking stations to maximize their availability to the users. Such a distribution could be based on the immediate needs at a particular station, or on a historical prediction (i.e., estimating the vehicle demand in the future to determine when and from where a relocation event must occur) [6]. The relocation activity can be carried out by the user himself or by the service provider [7]. In the first case, the user is incentivized to choose another station and/or reservation time, or to car pool with other users; in the second case, which is the most common in real services, the vehicles are physically moved by the service provider. The relocation problem exists also for bike sharing services (see, for instance, [2, 11, 36]); in this case, the practical solution is easier since the bikes can be simply moved loading them in a truck. In the car sharing relocation literature, also the platooning of the vehicles has been considered [17], where the platoon is composed by a chain of technologically innovative vehicles, led by vehicle head. This procedure, however, is still of little use for safety reasons and due to pending future technological developments [12].

In the following, a brief literature review regarding the methods proposed to model and solve the vehicle relocation problem is presented.

In [14], the use of a fleet of auto transport trucks is considered, minimizing the number of cars to move between the stations and minimizing the travel time of the auto-transport trucks. The first problem is an uncapacitated transportation problem, solved through the Hitchcock algorithm [29]; while the second problem is addressed through a heuristic algorithm. One development is proposed in [15], which makes the assumption of using a single auto transport truck that runs continuously on a circuit, arbitrarily established. In [22], a heuristic based on the immediate needs at the stations is presented, that is, the next station to be visited by the auto-transport truck is chosen according to the current state of the system. In such a way, the algorithm gives priority to visit the stations that have the greatest likelihood of running out of vehicles. [31] proposed two techniques, namely "shortest time" and "inventory balancing," for the relocation carried out by the service provider. Relocating by "shortest time" means moving vehicles to or from a neighboring station in the shortest possible time to answer to an immediate need. Relocating by "inventory balancing" means keeping the balance between the stations, following a predetermined desired distribution of the vehicles. A simulation model was used to test such techniques using historical data from a car sharing operator in Singapore. Results have shown that the system can afford a 10% reduction in car parking lots and a 25% reduction in staff numerousness. In [19], the complex dynamics of the system, using a discrete event system simulation, is represented. The article considers the relocation made by both users and staff, simulating different scenarios, with the objectives of reducing the number of required staff and minimizing the number of car sharing vehicles to satisfy the service demand. In [32], a decision support system to determine a set of near-optimal operating parameters (e.g., number of vehicles) is presented, formulating the relocation problem as a mixed integer programming problem. In [34], the authors consider a strategy that involves anticipative fleet redistribution that operators initiate to correct short-term demand asymmetry, developing a stochastic mixed-integer program to take into account demand uncertainty.

As regards the engine of the vehicles, the general expectation, confirmed by the investments of the principal car makers, is a shift from internal combustion to electricity [23]. Such a change will have a particular impact on urban areas, which will experience less local emissions and a better air quality. In addition, greenhouse gas emissions will decrease if the electricity needed for the vehicles will be generated by cleaner and renewable technologies, such as solar power plants. The problem of the relocation of EVs has been faced in [20], as a generalization of the pickup and delivery problem, formulated as a mixed integer program and solved using different solution methods including constraint programming, Lagrangian relaxation and a modified A* heuristic. The algorithm is applied to a car sharing service with 50 EVs and five charging stations, offered in the French town of Saint Quentin en Yvelines. The same car sharing service was studied also in [28], where the authors determined the needed number of auto transport trucks with an exact algorithm, and then minimized the total travel time of reallocation, studying three different heuristics. In [13], the authors consider the issue of recharging the vehicles, and an optimal level of battery charge that makes the vehicle available for the users is defined. Because of the battery-limited range, two major aspects have to be taken into account in the relocation system: the physical and energy availability of vehicles at stations. In [10], the authors proposed a model for supporting strategic decisions (i.e., number and location of charging stations and number of vehicles) and tactical planning decisions (i.e., allocation of personnel for the vehicle relocation). The model was applied to plan an electric one-way vehicle sharing service in Nice, France.

In our opinion, the relocation approach based on auto-transport trucks may be not well suitable for an urban settings, from a practical point of view, because stations may not be easily reachable by the trucks, and the operations of loading/unloading EVs is time consuming. For the EV relocation problem, we propose, therefore, the use of a staff of car sharing operators (hereafter called workers). They may move easily and in ecosustainable way from a delivery point to a pickup point using a folding bicycle that can be loaded in the trunk of the EV which needs to be moved. Such a relocation approach generates a challenging Paired Pickup and Delivery Problem with Time Windows (PPDPTW) [35] with features that, to the best of our knowledge, have been never considered in the literature. We call such a problem the

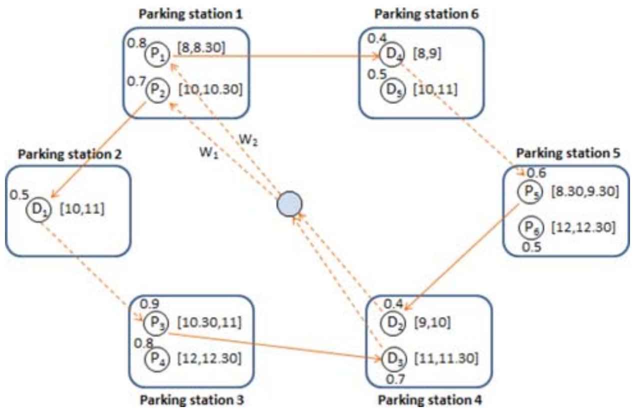

FIG. 1. Instance of the EVRP with six pickup requests (P_i for i = 1,...,6) and five delivery requests (D_i for i = 1,...,5). Both the battery charge level and time window [$\tau^{\max}$, $\tau^{\min}$] are indicated beside each request node. Two workers (W_1, W_2) leave a single depot, indicated by the shaded node, to relocate the EVs. The dashed arcs denote that a worker is biking while the solid arcs denote that is driving an EV. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

EV relocation problem (EVRP). In Fig. 1, an instance of the EVRP with six pickup relocation requests and six delivery relocation requests spread in six parking stations is given. Beside each request node, the battery charge level of the EV is indicated as well as a time window [$\tau^{\max}$, $\tau^{\min}$] being $\tau^{\max}$ and $\tau^{\min}$, respectively, the earliest time and the latest time when the request can be satisfied. An example of feasible solution with two workers is also shown: it consists of two cycles (one for each worker) starting from the depot and alternating a pickup request with a delivery request, which is compatible for battery charge level and time window (taking into account both the battery consumption and the time spent to arrive to the parking station of the delivery request with the EV picked up); moreover, the duration of each cycle must not to exceed the working time of the workers.

EVRP shares some features with the 1-skip vehicle routing problem [3] and the rollon–rolloff problem ([5, 8, 18]), that is, the fact that just one item at the time can be picked up and delivered, and routes starting and ending at a single depot cannot exceed a given maximum duration. However, EVRP is more challenging than the above mentioned problems since it is complicated by the fact that the distance covered by a vehicle depends also by the item picked up, that is, the residual electrical charge of the EV picked up. This further complication does not allow for instance to map the problem into a static bipartite graph like for the rollon–rolloff problem presented in [5] because the feasibility of an arc connecting a pickup request node with a delivery request node depends on the time when the pickup request node is reached since the battery level of a parked EV increases over the time. Another important difference between the EVRP and the rollon–rolloff problem is that in the EVRP both the routes and the schedules of the vehicles are decided, whereas in the latter only the routes.

We faced such a problem within the Green Move project [4, 33], which has been studying and testing a new system based on small, electric, and shared vehicles. Financed by Regione Lombardia, the project aims to face both the technology aspect and the service design part to identify a successful business model of vehicle-sharing. The main idea is to create a flexible sharing service, based on EVs and open to a wide range of different categories of users. The system will be made easily accessible thanks to a device, the Green-e-Box [1], which allows the inclusion into the vehicle sharing service also of private vehicles.

In this work, we yield the first mixed integer linear programming (MILP) formulation of the EVRP, and we illustrate some techniques to strengthen it and to speedup its solution through a state-of-the art solver (CPLEX). Finally, we test our approach on verisimilar instances built on the Milan road network.

The article is organized as follows. In Section 2, we describe the problem, in Section 3, we present its MILP formulation, in Section 4, we introduce some techniques to speedup its solution, in Section 5, we show the results of some numerical experiments, and in Section 6, we draw some conclusions.

2. PROBLEM DESCRIPTION

We consider a one-way car sharing service with a homogeneous fleet of EVs. Let L be the maximum distance that an EV can cover when its battery is fully charged. Such distance depends on the vehicle model; for instance, L can vary from 50 km for a Liberty Piaggio to 400 km for a Tesla (in the experimental campaign, we assume that L =150 km). Note that when the battery of an EV is not fully charged, the maximum distance that can be covered is linearly proportional to the residual charge of the battery (i.e., an EV with residual charge at 50% can travel for $L/2$ km). Concerning the recharge time Γ of a battery, typically the recharge process comprises two phases: the first one is intensity-constant, the second one is tension-constant. In the first phase, the battery charge level grows linearly with the time and at the end the battery is almost fully recharged. In the second phase, conversely, the charging pace is not linear with the time; such a phase can require some hours to achieve the full charge of the battery and to ensure an uniform recharge of all the cells that compose the battery. For sake of simplicity, we do not consider the second phase to model the EVRP. The maximum time needed to complete the first phase depends on the recharge technology used and can vary for instance from Γ = 1 hour for a 380 V Superfast Recharger to Γ = 5 h for a 220 V Multifast Recharger (in the experimental campaign we consider Γ = 4 h).

We suppose that all the lots of the parking stations are equipped with a charging infrastructure. In a one-way car sharing service, the relocation problem consists in establishing how to move the vehicles to prevent a station from running out of EVs or having all the parking lots occupied. Let D be the set of delivery requests (i.e., requests of EVs that need to be delivered to prevent a station from running out of them) and let P be the set of pickup requests (i.e., requests of EVs that need to be moved to vacant parking lots). Each request $r \in P \cup D$ is characterized by a parking location v_r, that is, a node of the road network containing a parking station, by the residual charge of the battery ρ_r and by a time window $[\tau_r^{\max}, \tau_r^{\min}]$ where $\tau_r^{\max}$ and $\tau_r^{\min}$ represent, respectively, the earliest time and the latest time of the request r. For instance, if r is a pickup request then $\tau_r^{\min}$ is the time before which the EV is not available, while $\tau_r^{\max}$ is the time after which it is not convenient to pickup the EV (because from $\tau_r^{\max}$, the vehicle may be used by some user in the parking station where it is). Note that for a delivery request r, ρ_r indicates the minimum charge level that the EV battery must have at time $\tau_r^{\max}$. Therefore, if an EV is delivered before $\tau_r^{\max}$, the charge level of its battery may be less than ρ_r on condition that at least the charge level ρ_r is achieved at the time $\tau_r^{\max}$. Whereas for a pickup request r, ρ_r indicates the battery charge level at $\tau_r^{\min}$. Since the fleet of EV is homogeneous, each delivery request can be satisfied picking up every EV of a pickup request provided that the constrains on time windows and battery charge level are satisfied.

We propose to relocate the EVs using a staff of car sharing operators (workers). They may move easily and in an ecosustainable way from a delivery point to a pickup point using a folding bicycle that can be loaded in the trunk of the EV which needs to be moved. Note that some relocation requests may not be satisfied through this relocation approach (for instance, when $|P| \neq |D|$): in this case, the car sharing provider should consider other actions to satisfy the users requests. For instance, whether some delivery request remains unsatisfied, and thus, a parking station runs out of EVs, then the car sharing provider may put at disposal of the users a taxi (e.g., this policy is used in Milan by the *e-vai* company: www.e-vai.com).

Given a team of K workers which leave a single depot—even at different times—using folding bicycles, we want to determine their routes and their schedules in such a way that: (1) each route consists of an alternating sequence of pickup requests and delivery requests, (2) the duration of each route does not exceed a given threshold T (i.e., the working time of the workers), (3) each route ends at the depot, and (4) the number of satisfied requests is maximized respecting the time windows and battery charge level constraints. We call such a problem the EVRP.

3. MATHEMATICAL PROGRAMMING FORMULATION

The formulation of the EVRP is based on a directed graph G=(N, A) that models all the possible actions rather than considering straightly the road network. The set of nodes of G is given by $N = P \cup D \cup \{0\}$ where 0 indicates the depot node. The set of arcs can be partitioned into two sets: the EV arcs and the bike arcs. The EV arcs model the action of a worker when he is traveling using an EV from a pickup point to a delivery point; the bike arcs model the action of a worker when he is traveling using a bike from a delivery point or from the depot to a pickup point or to the depot. Therefore, for each $i \in P$ and for each $j \in D$, the EV arcs link i and j through the ordered pairs (i,j), while the bike arcs are defined by the ordered pairs (j,i). Moreover, the bike arcs also include the arcs $(0, i)$ $\forall i \in P$ and the arcs $(j, 0)$ $\forall j \in D$.

TABLE 1. Operational times c_{ij} of every arc of graph G.

Arcs	Operational times	Involved nodes
(i, j)	$\frac{d_{ij}}{s'} + q' + q''$	$\forall i \in P, \forall j \in D$
(j, i)	$\frac{d_{ji}}{s''}$	$\forall i \in P, \forall j \in D$
$(0,i)$	$\frac{d_{0i}}{s''}$	$\forall i \in P$
$(j,0)$	$\frac{d_{j0}}{s''}$	$\forall j \in D$

For each $i \in P$ and for each $j \in D$, let d_{ij} denote the length of the shortest path from v_i to v_j with an EV, let d_{ji} denote the length of the shortest path from v_j to v_i with a bike, let s' indicate the average speed of an EV, let s'' indicate the average speed by bicycle of a worker, let q' be the time to park the EV and take the bike from the EV trunk, let q'' be the time to load the bike in the EV trunk and leave the parking lot with the EV. We associate an operational time c_{ij} with every kind of arc as reported in Table 1.

There are two main advantages to deal with the graph G rather than directly with the road network. The first one is that an elementary cycle on graph G corresponds always to every feasible route of a worker, whereas this is not true in the original road network when there are multiple requests in the same parking and modeling nonelementary cycles is by far harder (see [20]). The second advantage, even in the case of a single request for each parking, is that a formulation based on graph G requires by far less variables than a formulation based on the road network, because variables are defined on the arcs and nodes of the used graph. The dimension of graph G depends only on the number of requests (since $|N|$=$|P|$+$|D|$+1 and $|A|$=2$|P||D|$+$|P|$+$|D|$) and not by the number of the physical nodes (road intersections) and road links. For instance, the Milan road network considered in Section 5 contains more than 23,000 road links, which are by far greater than $|A|$ even for a high number of EVs to be redistributed.

Let us introduce the binary routing variables x_{ijk} equal to 1 if the kth worker visits node $j \in N$ immediately after node $i \in N$, 0 otherwise. Let us also introduce the continuous variables t_{ik} to model the arrival time to the parking v_i of the kth worker. We state that the EVRP can be modeled by way of the following MILP (note that for all $i \in N$, we indicate with $\delta^+(i)$ and $\delta^-(i)$ the forward star and the backward star of node i, respectively, and in constraints (5)–(6) $M = \max_{i \in P \cup D} \{\tau_i^{\max}\}$).

$$\max \sum_{k=1}^{K} \sum_{(i,j) \in A : i \neq 0} x_{ijk} \tag{1}$$

subject to:

$$\sum_{j \in \delta^+(0)} x_{0jk} \leq 1 \quad \forall k = 1, \ldots, K \tag{2}$$

$$\sum_{k=1}^{K} \sum_{j \in \delta^+(i)} x_{ijk} \leq 1 \quad \forall i \in P \cup D \tag{3}$$

$$\sum_{j \in \delta^+(i)} x_{ijk} - \sum_{j \in \delta^-(i)} x_{jik} = 0 \quad \forall i \in P \cup D \cup \{0\}, \quad \forall k = 1, \ldots, K \tag{4}$$

$$t_{ik} + c_{ij} x_{ijk} \leq t_{jk} + M(1 - x_{ijk}) \quad \forall (i,j) \in A : j \neq 0, \quad \forall k = 1, \ldots, K \tag{5}$$

$$t_{ik} + c_{i0} x_{i0k} \leq t_{0k} + T + M(1 - x_{i0k}) \quad \forall i \in \delta^-(0), \quad \forall k = 1, \ldots, K \tag{6}$$

$$\tau_i^{\min} \leq t_{ik} \leq \tau_i^{\max} \quad \forall i \in P \cup D, \quad \forall k = 1, \ldots, K \tag{7}$$

$$d_{ij} x_{ijk} \leq L \left(\rho_i + \frac{t_{ik} - \tau_i^{\min}}{\Gamma} \right) \quad \forall (i,j) \in A : i \in P, j \in D, \quad \forall k = 1, \ldots, K \tag{8}$$

$$\rho_i + \frac{t_{ik} - \tau_i^{\min}}{\Gamma} - \frac{d_{ij}}{L} x_{ijk} \geq \rho_j - \frac{\tau_j^{\max} - t_{jk}}{\Gamma} - (\rho_j + 1)(1 - x_{ijk}) \quad \forall (i,j) \in A : i \in P, j \in D, \quad \forall k = 1, \ldots, K \tag{9}$$

$$1 - \frac{d_{ij}}{L} x_{ijk} \geq \rho_j - \frac{\tau_j^{\max} - t_{jk}}{\Gamma} - (\rho_j + 1)(1 - x_{ijk}) \quad \forall (i,j) \in A : i \in P, j \in D, \quad \forall k = 1, \ldots, K \tag{10}$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i,j) \in A, \quad \forall k = 1, \ldots, K \tag{11}$$

$$t_{ik} \geq 0 \quad \forall i \in P \cup D \cup \{0\}, \quad \forall k = 1, \ldots, K \tag{12}$$

Since each arc connects a pair of requests, the objective function (1) represents the total number of satisfied requests. Constraints (2) take into account that at most K workers are available, and therefore, at most K routes can be generated imposing that at most one arc for each worker can leave the depot node 0. Constraints (3) impose that each request is satisfied at most once. Flow conservation constraints (4) ensure that the solution is a collection of cycles. Constraints (5) rule the time variables ensuring that the visit time of a node is given by the sum of the visit time of its predecessor and the operational time to go from the predecessor to the current node. Note that such constraints are not imposed for the depot node to ensure that the route can pass through the depot and at the same time they prevent the solution from containing isolated cycles that do not pass through the depot. In this way, the formulation does not require additional subtour elimination constraints. Constraints (6) ensure that the duration of each route does not exceed the threshold T. Constraints (7) enforce the time windows for the pickup and delivery requests. Constraints (8) model the fact that the maximum distance traveled by an EV is linearly proportional to the residual charge. Note that if $\rho_i + \frac{t_{ik} - \tau_i^{\min}}{\Gamma} > 1$, such constraints become redundant since the graph topology prevent already the existence of arcs (i, j) with $d_{ij} > L$. Finally, constraints (9) and (10) ensure that an EV is delivered with a battery level such that at the time $\tau_j^{\max}$ a charge level not lower than ρ_j will be achieved.

4. SOLVING THE EVRP

4.1. *Speedup Techniques for the MILP Formulation*

In this subsection, we present some techniques to speedup the solution of MILP formulation (1)–(12) of the EVRP when a commercial MILP solver (e.g., CPLEX) is used.

The first technique consists in reducing the number of arcs considered in the graph representation of the EVRP introduced in Section 3, by excluding the arcs that cannot model feasible actions. In particular, the EV arcs (i, j) are defined not for every pair of node $i \in P$ and $j \in D$ but only for the ones that satisfy the following four conditions:

$$\frac{d_{0i}}{s''} + \frac{d_{ij}}{s'} + \frac{d_{j0}}{s''} + q' + q'' \leq T \tag{13}$$

$$\tau_j^{\min} - \tau_i^{\max} + \frac{d_{0i}}{s''} + \frac{d_{j0}}{s''} \leq T \tag{14}$$

$$d_{ij} \leq L \cdot \min \left\{ 1, \rho_i + \frac{\tau_i^{\max} - \tau_i^{\min}}{\Gamma} \right\} \tag{15}$$

$$\max \left\{ \Gamma \left(\frac{d_{ij}}{L} + \rho_j - \rho_i \right), 0 \right\} + \frac{d_{ij}}{s'} + q' + q'' \leq \tau_j^{\max} - \tau_i^{\min} \tag{16}$$

Condition (13) takes into account the time to travel—within the working time T—from the depot to v_i, from v_i to v_j and then to the depot. Condition (14) is necessary to respect both the request time windows and the working time when a worker needs to wait to serve request j. Condition (15) ensures that even if the EV is picked up as later as possible (i.e., at $\tau_i^{\max}$), the battery level achieved is sufficient to cover the trip from v_i to v_j along the shortest path. Condition (16) checks if the time necessary to reach the battery level of the delivery request—starting from the battery charge level of the pickup request ($\Gamma(\rho_j - \rho_i)$)—is compatible with the time windows of both the requests, taking into account also the battery consumption ($\Gamma d_{ij}/L$) and the time (d_{ij}/s') used to travel from v_i to v_j, and the related parking operations ($q' + q''$).

In a similar way, the bike arcs (j,i) are defined for each $j \in D$ and for each $i \in P$ such that the following condition holds:

$$\tau_i^{\max} \geq \tau_j^{\min} + \frac{d_{ji}}{s''} \tag{17}$$

We call $\tilde{A}$, the set of arcs that satisfy conditions (13)–(17).

A second technique is based on the observation that the feasible region of MILP delimited by (2)–(12) may contain several equivalent optimal solutions. In an optimal solution, if any, the route of one generic worker can be swapped with the one of any other worker, yielding a different optimal solution in terms of variables x_{ijk} and t_{ik}. The presence of such multiple optimal solutions is harmful for a MILP solver, since it may require more CPU time. To prevent such situation, we add to Formulation (1)–(12) the following group of constraints that "breaks" the symmetry of the feasible region:

$$\sum_{(i,j)\in A: i\neq 0} c_{ij}x_{ijk'} \geq \sum_{(i,j)\in A: i\neq 0} c_{ij}x_{ijk''} \qquad \forall k', k'' = 1, \ldots, K : k' < k'' \tag{18}$$

Constraints (18) prevent the presence of multiple optimal solutions mentioned above, since they impose that the routes are assigned to the workers according to the non-increasing operative cost ordering.

4.2. A MILP-Based Heuristic

We observed that the MILP formulation of EVRP can often be solved in reasonable time for K=1. Therefore, we exploited such outcome to build a simple but effective heuristic procedure based on the iterative solution with the MILP (1)–(12) of at most K instances of the EVRP with one worker. At the kth iteration, the tour of the kth worker is determined considering only the requests that are not already satisfied in the previous iterations, until k is equal to K or all the requests are satisfied.

The quality of such heuristic solution can be evaluated considering the gap with an upper bound to the optimal value (when the latter is not available), as will be explored in the next subsection.

4.3. MILP-Based Upper Bounds

A first method to compute an upper bound to the optimal value of the satisfied requests in the EVRP when $K > 1$ consists straightforward in multiplying by K the maximum number of satisfied requests by one worker [i.e., the optimal value of the EVRP obtained solving MILP (1)–(12) with K=1]. Indeed let s_1 be the maximum number of satisfied requests when K=1 and let s_K be the maximum number of requests served by a worker in a team of $K > 1$ workers, then we have by definition $s_K \leq s_1$. Therefore, the total number of requests served by $K > 1$ workers is upper bounded by the product Ks_1. In the following, we call U_1 the upper bound given by $U_1 = \min\{\mathrm{K}s_1, |P \cup D|\}$.

A second method to compute an upper bound, U_2, consists in solving with one worker the Formulation (1)–(12) where Constraint (7) is relaxed by elimination and the worker's working time is extended to KT. The optimal value obtained in this way is an upper bound for the original problem because any feasible solution of the latter consists in at most K routes, each one assigned to a worker, that satisfy all Constraints (2)–(12) and therefore they can be also traveled, one after the other, by one worker within the working time KT satisfying certainly all constraints except (7). Hence, any feasible solution of the original problem is also feasible for the Formulation (1)-(12) where Constraints (7) are removed and the working time is extended to KT.

Finally, a third kind of upper bound, U_3, can be obtained considering the linear programming relaxation of MILP (1)–(12), that is, the binary variables x_{ijk} become continuous variables in the interval [0,1].

4.4. Upper Bound with Unlimited Workers

In this subsection, we describe a method to compute an upper bound, U_4, which considers an unlimited number of workers. Such an upper bound will be useful in the experimental campaign to detect those cases where, although not all the relocation requests are satisfied, no further improvement is possible increasing the number of workers.

Let us consider the bipartite graph $\hat{G} = (P \cup D, \tilde{A}_{\mathrm{EV}})$ where $\tilde{A}_{\mathrm{EV}}$ is the subset of the arc set $\tilde{A}$ (introduced in Subsection 4.1) made up only by the EV arcs: if a pickup request $i \in P$ and a delivery request $j \in D$ are compatible (i.e., the request j can be satisfied using the EV of request i) then the EV arc (i, j) is present in $\tilde{A}_{\mathrm{EV}}$. Therefore, twice the maximum matching value on graph $\hat{G}$ gives an upper bound on the maximum number of relocation requests that can be satisfied by an unlimited number of workers (U_4 may be strictly greater than the maximum number of the feasible relocation requests since Conditions (13)–(17) are necessary but may be not sufficient for the compatibility). Note that such an upper bound can be tighter than $|P \cup D|$ since it exploits the information on the compatibility between pickup and delivery requests given by $\tilde{A}_{\mathrm{EV}}$.

5. EXPERIMENTAL RESULTS

This section presents some numerical experiments made on the road network of Milan. The network is based on the database set up by the Milan transport agency [27], which contains information about road topology, nodes, permitted maneuvers, and link attributes. The road network consists of about 23,000 links. The experiments consider nine charging stations located nearby some main attractors, that is, Loreto, Cadorna, Porta Genova, Porta Garibaldi, Piola, Duomo, Stazione Centrale, Turati, and San Babila (Fig. 2).

We built 100 instances of the EVRP considering randomly generated pickup and delivery requests on the nine parking stations with random battery charge levels and random time windows between 8.00 a.m. and 8.00 p.m. In all the instances, $|P|$ = $|D|$ and the total number of relocation requests can assume five values: 10, 20, 30, 40, and 50 (we refer to them as "instance size"). Note that the maximum instance size considered is consistent with most of real world car sharing case studies (e.g., [10]). The 100 instances have been obtained considering, for each instance size, five different samples on the random choice of parking, battery charge level and time window of each pickup and delivery request, and four values of K = 1,...,4. Some statistics on the battery levels and on the time windows of the relocation requests considered in each instance are shown, respectively, in Tables 8 and 9 of Appendix.

FIG. 2. The road network and the locations of the nine charging stations considered in the numerical experiments. Point D denotes the depot, located at the charging station of Piola, where the workers start their routes. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Through a preprocessing algorithm, we eliminated from each instance the relocation requests that cannot be satisfied because infeasible. More precisely, the algorithm removes the requests $i \in P$ if there exists $j \in D$ such that Conditions (13)–(17) are not satisfied. An analogous check is made for each $j \in D$. The number of relocation requests removed by the preprocessing is very small (more details can be found in Table 10 of Appendix).

The main input data values used for the MILP formulation of the EVRP presented in Section 3 are summarized in Table 2.

The MILP Formulation (1)–(12) has been implemented in AMPL [25] and solved with the state of the art solver CPLEX11.0 on a PC Intel Xeon 2.80 GHz with 2GB RAM.

The average percentage cardinality saving passing from arc set A to the reduced arc set $\tilde{A}$ introduced in Section 4 is around the 50% (the details can be found in Table 10 of Appendix). Therefore, the application of the first technique proposed in Section 4 halves on average the number of binary variables used by the MILP formulation of the EVRP, since the only binary variables of the model are the routing variables x_{ijk} defined for each arc (i,j). In the experimental campaign, we always use the speedup techniques presented in Section 4. Moreover, we always consider a CPU time limit of 1800 s.

Table 3 shows the numerical results obtained with the MILP formulation of the EVRP on the whole test bed of instances. The first column indicates the instance name: note that each name is in the format *AmatX_Y* where *X* indicates the instance size and *Y* the sample considered. The successive four columns represent the percentage of satisfied requests for a number of workers K, respectively, equal to 1, 2, 3, and 4. Such percentage is computed on the total number of original requests, that is, considering also infeasible requests identified by the preprocessing algorithm. We emphasize in boldface the the values for which the 100% of requests have been served or the percentage of satisfied requests corresponds to the maximum achievable since the upper bound U_4 (described in Subsection 4.4) has been reached (this is done only for the minimum value of K). The sixth column (headed with $K = \infty$) indicates the maximum percentage of satisfied requests achievable with an unlimited number of workers on the basis of upper bound U_4. The last four columns show the CPU times (measured in seconds) spent to obtain the solutions for $K = 1, 2, 3, 4$, respectively. In such columns, we emphasize in boldface the cases where the maximum CPU time limit has been reached, and therefore, there is no guarantee that the solution yielded by CPLEX is optimal. The average results on each group of instances with the same size are reported in Table 4; the meaning of the columns is similar to the one of Table 3.

Concerning the quality of the obtained solutions, we note that with $K = 4$ it is possible to satisfy a high percentage of relocation requests (on average around 83%), with an average gap of only 7% with the maximum number of requests that may be satisfied with an unlimited number of workers. Three workers are already sufficient to satisfy all instances with 10 requests and $K = 3$ seems to be also the suitable number of workers to be used for the instances up to 30 requests, because the average marginal improvement switching from $K = 3$ to $K = 4$ is only 3.11%.

Concerning the CPU time, the MILP Formulation (1)–(12) with the speedup techniques is able to solve on average instances up to 20 requests in few seconds, the instances with 30 requests in few minutes, while it can require several minutes to solve the instances of greater sizes (for 20 instances, the maximum CPU time allowed is reached). We can also notice that the CPU time of the instances with size greater than 20 strongly depends on K (for $K > 1$, the CPU time can also be the double of the one for $K = 1$). However, we note that for some instances (e.g., Amat30_1, Amat30_2, Amat40_1, Amat50_1) the CPU time can decrease when K increases (for instance switching from K=3 to K=4). In such cases, the marginal improvement in the number of satisfied requests is small (even null for Amat30_1 and Amat30_2) and therefore the MILP solver obtains easier the optimal solution with an additional worker, because the feasible region for K=4 contains the feasible region for K=3.

Table 5 presents the characteristics of the solutions obtained considering only the instances with the minimum

TABLE 2. Main input data values used in the experiments.

Input data	T (min)	s' (km/h)	s'' (km/h)	q' (min)	q'' (min)	L (km)	Γ (min)
Values	300	25	15	1	1	150	240

TABLE 3. Numerical results of MILP (1)–(12) with the speedup techniques.

	Served %					CPU			
Instance	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = \infty$	$K = 1$	$K = 2$	$K = 3$	$K = 4$
Amat10_1	60.00	**100.00**	100.00	100.00	100.00	0.01	0.04	0.07	0.05
Amat10_2	60.00	**100.00**	100.00	100.00	100.00	0.01	0.03	0.03	0.04
Amat10_3	60.00	**80.00**	80.00	80.00	80.00	0.01	0.01	0.02	0.02
Amat10_4	40.00	80.00	**100.00**	100.00	100.00	0.01	0.02	0.14	0.22
Amat10_5	40.00	60.00	**80.00**	80.00	80.00	0.01	0.01	0.01	0.01
Amat20_1	50.00	**100.00**	100.00	100.00	100.00	0.38	0.21	1.09	0.38
Amat20_2	50.00	80.00	**90.00**	90.00	90.00	0.89	2.22	0.94	0.42
Amat20_3	50.00	70.00	**80.00**	80.00	80.00	4.78	21.42	0.19	0.28
Amat20_4	30.00	50.00	70.00	80.00	90.00	0.05	0.38	0.26	3.67
Amat20_5	30.00	50.00	70.00	80.00	90.00	0.04	0.16	0.36	1.33
Amat30_1	46.67	86.67	**100.00**	100.00	100.00	27.74	224.24	615.52	8.85
Amat30_2	46.67	80.00	**93.33**	93.33	93.33	1.62	35.46	153.96	7.35
Amat30_3	33.33	60.00	73.33	80.00	93.33	26.36	168.81	1327.92	**1800.00**
Amat30_4	33.33	53.33	66.67	80.00	93.33	0.67	2.48	17.59	17.94
Amat30_5	26.67	46.67	60.00	66.67	86.67	0.03	0.53	2.51	18.00
Amat40_1	40.00	75.00	85.00	**95.00**	95.00	**1800.00**	**1800.00**	**1800.00**	105.35
Amat40_2	35.00	65.00	80.00	80.00	90.00	410.41	**1800.00**	**1800.00**	**1800.00**
Amat40_3	30.00	55.00	75.00	80.00	90.00	47.78	**1800.00**	**1800.00**	**1800.00**
Amat40_4	25.00	45.00	55.00	65.00	85.00	1.46	8.15	122.16	554.99
Amat40_5	25.00	50.00	60.00	70.00	90.00	0.53	4.27	25.55	59.84
Amat50_1	44.00	72.00	84.00	**88.00**	88.00	**1800.00**	**1800.00**	**1800.00**	1633.19
Amat50_2	36.00	60.00	72.00	80.00	84.00	566.56	**1800.00**	**1800.00**	**1800.00**
Amat50_3	32.00	52.00	64.00	72.00	84.00	67.57	**1800.00**	**1800.00**	**1800.00**
Amat50_4	24.00	44.00	56.00	64.00	80.00	1.22	13.47	162.97	**1800.00**
Amat50_5	32.00	52.00	64.00	72.00	88.00	0.29	13.54	78.18	116.27
Average	**39.19**	**66.67**	**78.33**	**83.04**	**90.03**	**190.34**	**451.82**	**532.38**	**533.13**

value of K for which the maximum value of satisfied requests has been reached (e.g., K=2 for Amat10_1, K =3 for Amat10_5). After the instance name indicated in the first column, the next four columns show the duration of the maximum relocation trip that each worker handles in a shift (the field is empty if a worker is not used in a solution). The sixth column represents the maximum travel time of a worker, the seventh column the total travel time of all workers, and the last three columns the total time spent by all the workers for, respectively, driving an EV, riding a bicycle, and waiting. All the times are expressed in minutes. We notice that on average 66% of the working time is spent by the workers for waiting, 23% for riding a bicycle, and only 11% for driving an EV. The fact that the waiting time is so high can be due to two reasons: (1) sometimes the workers really need to wait for respecting the time windows of the relocation requests; (2) the only objective of the EVRP is the optimization of the total number of relocation satisfied requests and not the optimization of the time spent by the workers (the latter is indirectly optimized only if the time saved allows to serve additional relocation requests).

In Table 6, we compare the results obtained with the heuristic described in Subsection 4.2 with the results of the MILP formulation. The first column indicates the instance name, the successive three columns indicate the percentage of satisfied requests for a number of workers K, respectively, equal to 2, 3, and 4. The successive three columns indicate the relative gap (expressed in percentage) between the requests satisfied by the heuristic and by the MILP formulation (the ratio is with the latter). Finally, the last three columns indicate the CPU times in seconds to obtain the results for $K = 2, 3, 4$, respectively. We notice that the heuristic obtains high-quality

TABLE 4. Average numerical results of MILP (1)–(12) with the speedup techniques.

	Served %					CPU			
$\lvert P \cup D\rvert$	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = \infty$	$K = 1$	$K = 2$	$K = 3$	$K = 4$
10	52.00	84.00	92.00	92.00	92.00	0.01	0.02	0.05	0.07
20	42.00	70.00	82.00	86.00	90.00	1.23	4.88	0.57	1.22
30	37.33	65.33	78.67	84.00	93.33	11.28	86.30	423.50	370.43
40	31.00	58.00	71.00	78.00	90.00	452.04	1082.48	1109.54	864.04
50	33.60	56.00	68.00	75.20	84.80	487.13	1085.40	1128.23	1429.89

TABLE 5. Statistics on the longest trips and on the time spent by the workers biking, driving and waiting (in min).

	Max Trip								
Instance	Worker 1	Worker 2	Worker 3	Worker 4	Max Travel	Tot Travel	Drive	Bike	Wait
Amat10_1	16.78	8.80			98.82	154.55	49.74	104.81	445.45
Amat10_2	9.15	10.77			100.45	160.87	38.20	122.67	378.73
Amat10_3	9.78	7.62			80.77	117.80	29.27	88.53	277.97
Amat10_4	9.59	9.78	4.79		61.29	137.14	32.00	105.14	762.86
Amat10_5	15.66	16.78	11.84		47.07	119.57	55.05	64.52	471.09
Amat20_1	16.78	9.59			133.31	249.84	91.24	158.60	350.16
Amat20_2	9.78	11.07	6.62		98.59	232.76	65.89	166.87	667.24
Amat20_3	13.47	16.86	11.41		81.70	219.47	90.40	129.07	458.36
Amat20_4	11.36	7.09	13.47	4.72	63.37	196.43	63.98	132.45	1003.57
Amat20_5	16.86	11.41	15.66	9.21	88.09	237.89	92.42	145.47	962.11
Amat30_1	16.42	14.79	11.07		165.50	359.08	119.09	239.99	467.64
Amat30_2	11.53	9.78	9.78		143.99	314.41	102.29	212.11	525.11
Amat30_3	8.21	9.78	14.79	11.53	94.49	350.16	103.08	247.09	560.91
Amat30_4	11.41	14.79	16.78	9.59	97.00	337.17	114.99	222.19	707.43
Amat30_5	9.39	11.53	11.07	16.86	95.18	308.35	86.80	221.55	865.32
Amat40_1	16.78	12.41	16.86	9.33	148.43	484.27	174.30	309.97	694.62
Amat40_2	11.53	11.07	14.79		144.20	390.89	131.82	259.07	451.99
Amat40_3	9.78	9.39	11.53	14.79	117.78	414.70	123.55	291.15	621.99
Amat40_4	9.78	16.78	9.39	11.53	114.26	375.72	117.85	257.87	824.28
Amat40_5	12.41	9.06	16.78	9.21	129.39	365.76	111.96	253.80	550.21
Amat50_1	9.78	9.59	16.86	11.84	154.59	521.53	177.97	343.57	621.35
Amat50_2	11.53	14.79	9.59	9.15	156.50	497.69	154.20	343.49	488.50
Amat50_3	14.79	11.53	11.53	9.78	156.38	448.65	159.28	289.37	694.23
Amat50_4	9.39	9.78	11.53	16.78	138.33	414.19	127.43	286.76	785.81
Amat50_5	11.84	9.78	12.41	16.78	155.13	425.90	151.55	274.35	765.36

solutions with often by far less CPU time of the MILP formulation. Indeed the average relative gap with the MILP is less than 3.28%, and on average the heuristic CPU time is less than half of that one necessary for the MILP.

Table 7 compares the performances of the four methods to obtain upper bounds described in Subsections 4.3 and 4.4. The first column indicates the instance name, the Columns 2, 3, and 4 indicate the values of upper bound U_1, expressed as percentage on the total number of requests, for a number of workers K, respectively, equal to 2, 3, and 4. The fifth column has the same meaning for the upper bound U_2 since it yields the same results for K = 2, 3, and 4: this seems to happen because, removing Constraints (7), already two workers are able to satisfy a high number of relocation requests, and therefore, there is no improvement switching to $K = 3$ or $K = 4$. The successive three columns indicate the values of upper bound U_3. The ninth column contains the results for the upper bound U_4. Finally, the last three columns indicate the relative gap (expressed in percentage) between the best of the three bounds and the optimum value $z*$ obtained with MILP (1)–(12) (or the value of the best feasible solution found if the CPU time limit has been reached), represented by the following:

$$100\frac{\min\{U_1, U_2, U_3, U_4\} - z*}{z*}$$

We do not report the CPU time to generate the upper bounds, because for U_1 the CPU time is given by the sixth column of Table 3, while for U_2, U_3, and U_4, the CPU time is negligible being always less than 1 s (on average 0.12 s). Concerning the quality of the upper bounds, we notice that they are by far better for $K = 4$ than for $K = 2, 3$ since in the former case, the relative gap with the CPLEX solution value is on average 3.33% against a value around 11% for $K = 2{,}3$. Moreover, we observe that in general there is no domination between the two upper bound methods U_1 and U_2. Indeed for $K = 2$, $U_1 < U_2$ in 16 cases on 25, but for $K = 3, 4$, $U_2 \leq U_1$ is always true. The deterioration of the quality of upper bound U_1 when K increases probably depends on the fact that the value U_1 linearly increases with K and for the highest values of K, it easily reaches the total number of requests. Also between the two upper bound methods U_1 and U_4 in general there is no domination since for $K = 2$, $U_1 < U_4$ in 16 cases on 25, but for $K = 3$, the number of cases decreases to 4 against 13 cases with $U_4 < U_1$ and for $K = 4$, it always results $U_4 \leq U_1$. While U_2 almost dominates U_4 since it always results $U_2 \leq U_4$ except for an instance (Amat20_3). Upper bound U_3 yields performances very similar to those concerning upper bound U_4, but slightly better being $U_3 < U_4$ in 5 cases for $K = 2$ and in 3 cases for $K = 3$ while $U_3 = U_4$ in any other case. Therefore, considering also that the CPU time required by U_2 is by far less than the one for U_1, we can conclude that in our test bed the most convenient upper bound method is U_2 for $K \geq 3$.

Finally, we note that for two instances (Amat50_2 and Amat50_4 with K=4), for which the CPU time limit was

TABLE 6. Comparison of the heuristic results with the MILP results.

Instance	Served by heuristic %			Relative gap %			CPU		
	$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$	$K = 4$	$K = 2$	$K = 3$	$K = 4$
Amat10_1	80.00	100.00	100.00	20.00	0.00	0.00	0.21	0.03	0.01
Amat10_2	80.00	100.00	100.00	20.00	0.00	0.00	0.02	0.01	0.03
Amat10_3	80.00	80.00	80.00	0.00	0.00	0.00	0.02	0.02	0.01
Amat10_4	80.00	100.00	100.00	0.00	0.00	0.00	0.06	0.03	0.04
Amat10_5	60.00	80.00	80.00	0.00	0.00	0.00	0.02	0.02	0.03
Amat20_1	100.00	100.00	100.00	0.00	0.00	0.00	0.08	0.15	0.21
Amat20_2	80.00	90.00	90.00	0.00	0.00	0.00	0.63	0.77	0.79
Amat20_3	70.00	80.00	80.00	0.00	0.00	0.00	4.74	4.83	4.69
Amat20_4	50.00	70.00	80.00	0.00	0.00	0.00	0.02	0.06	0.07
Amat20_5	50.00	60.00	70.00	0.00	14.29	12.50	0.04	0.11	0.06
Amat30_1	86.67	93.33	100.00	0.00	6.67	0.00	31.03	30.52	30.22
Amat30_2	80.00	86.67	93.33	0.00	7.14	0.00	2.13	2.60	2.36
Amat30_3	53.33	66.67	73.33	11.11	9.09	8.33	27.86	26.64	26.87
Amat30_4	53.33	53.33	53.33	0.00	20.00	33.33	0.74	1.09	0.81
Amat30_5	46.67	60.00	66.67	0.00	0.00	0.00	0.06	0.19	0.06
Amat40_1	75.00	85.00	95.00	0.00	0.00	0.00	1800.00	1800.00	1800.00
Amat40_2	65.00	75.00	80.00	0.00	6.25	0.00	671.97	650.69	650.25
Amat40_3	55.00	70.00	80.00	0.00	6.67	0.00	65.19	65.15	64.71
Amat40_4	45.00	55.00	65.00	0.00	0.00	0.00	1.23	1.25	1.43
Amat40_5	50.00	60.00	70.00	0.00	0.00	0.00	0.49	0.53	0.49
Amat50_1	72.00	84.00	88.00	0.00	0.00	0.00	1800.00	1800.00	1800.00
Amat50_2	60.00	68.00	76.00	0.00	5.56	5.00	745.96	787.55	787.54
Amat50_3	52.00	60.00	68.00	0.00	6.25	5.56	93.77	95.39	99.17
Amat50_4	44.00	56.00	64.00	0.00	0.00	0.00	1.71	1.33	1.34
Amat50_5	52.00	64.00	72.00	0.00	0.00	0.00	0.76	0.47	0.51
Average	**64.80**	**75.88**	**80.99**	**2.04**	**3.28**	**2.59**	**209.95**	**210.78**	**210.87**

TABLE 7. Performances of upper bounds U_1, U_2, U_3, and U_4.

Instance	U_1			U_2	U_3			U_4	Relative gap %		
	$K = 2$	$K = 3$	$K = 4$	$K = 2,3,4$	$K = 2$	$K = 3$	$K = 4$	$K = \infty$	$K = 2$	$K = 3$	$K = 4$
Amat10_1	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00
Amat10_2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00
Amat10_3	100.00	100.00	100.00	80.00	80.00	80.00	80.00	80.00	0.00	0.00	0.00
Amat10_4	80.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00
Amat10_5	80.00	100.00	100.00	80.00	80.00	80.00	80.00	80.00	33.33	0.00	0.00
Amat20_1	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00
Amat20_2	100.00	100.00	100.00	90.00	90.00	90.00	90.00	90.00	12.50	0.00	0.00
Amat20_3	100.00	100.00	100.00	90.00	80.00	80.00	80.00	80.00	14.29	0.00	0.00
Amat20_4	60.00	90.00	100.00	80.00	90.00	90.00	90.00	90.00	20.00	14.29	0.00
Amat20_5	60.00	90.00	100.00	80.00	90.00	90.00	90.00	90.00	20.00	14.29	0.00
Amat30_1	93.33	100.00	100.00	100.00	100.00	100.00	100.00	100.00	7.69	0.00	0.00
Amat30_2	93.33	100.00	100.00	93.33	93.33	93.33	93.33	93.33	16.67	0.00	0.00
Amat30_3	66.67	100.00	100.00	86.67	93.33	93.33	93.33	93.33	11.11	18.19	8.33
Amat30_4	66.67	100.00	100.00	86.67	93.33	93.33	93.33	93.33	25.01	29.99	8.33
Amat30_5	53.33	80.00	100.00	73.33	86.67	86.67	86.67	86.67	14.28	22.22	9.99
Amat40_1	80.00	100.00	100.00	95.00	95.00	95.00	95.00	95.00	6.67	11.76	0.00
Amat40_2	70.00	100.00	100.00	90.00	87.50	87.50	90.00	90.00	7.69	9.38	12.50
Amat40_3	60.00	90.00	100.00	90.00	90.00	90.00	90.00	90.00	9.09	20.00	12.50
Amat40_4	50.00	75.00	100.00	70.00	80.00	82.50	85.00	85.00	11.11	27.27	7.69
Amat40_5	50.00	75.00	100.00	75.00	90.00	90.00	90.00	90.00	0.00	25.00	7.14
Amat50_1	88.00	100.00	100.00	88.00	88.00	88.00	88.00	88.00	22.22	4.76	0.00
Amat50_2	72.00	100.00	100.00	80.00	84.00	84.00	84.00	84.00	20.00	11.11	0.00
Amat50_3	64.00	96.00	100.00	84.00	82.00	84.00	84.00	84.00	23.08	31.25	16.67
Amat50_4	48.00	72.00	96.00	64.00	74.00	78.00	80.00	80.00	9.09	14.29	0.00
Amat50_5	64.00	96.00	100.00	72.00	86.00	88.00	88.00	88.00	23.08	12.50	0.00
Average									**12.28**	**10.65**	**3.33**

TABLE 8. Statistics on the battery charge level of all relocation requests considered in each instance.

Instance	min_R_P	max_R_P	av_R_P	dev_R_P	min_R_D	max_R_D	av_R_D	dev_R_D
Amat10_1	0.40	1.00	0.78	0.21	0.30	0.60	0.48	0.12
Amat10_2	0.40	1.00	0.74	0.20	0.30	0.70	0.54	0.14
Amat10_3	0.30	1.00	0.80	0.26	0.40	0.90	0.64	0.16
Amat10_4	0.40	1.00	0.72	0.21	0.40	0.90	0.58	0.17
Amat10_5	0.20	0.90	0.50	0.24	0.40	0.80	0.56	0.14
Amat20_1	0.20	1.00	0.63	0.26	0.30	0.70	0.48	0.12
Amat20_2	0.20	1.00	0.71	0.27	0.30	0.90	0.56	0.16
Amat20_3	0.30	1.00	0.82	0.23	0.40	0.90	0.64	0.14
Amat20_4	0.30	1.00	0.71	0.24	0.40	0.90	0.61	0.13
Amat20_5	0.20	1.00	0.66	0.27	0.30	0.80	0.55	0.14
Amat30_1	0.20	1.00	0.65	0.27	0.30	0.70	0.47	0.12
Amat30_2	0.20	1.00	0.64	0.25	0.30	0.90	0.53	0.14
Amat30_3	0.30	1.00	0.74	0.24	0.30	0.90	0.62	0.15
Amat30_4	0.30	1.00	0.75	0.24	0.30	0.90	0.57	0.15
Amat30_5	0.20	1.00	0.58	0.26	0.30	0.80	0.57	0.16
Amat40_1	0.20	1.00	0.65	0.27	0.30	0.70	0.47	0.11
Amat40_2	0.20	1.00	0.67	0.27	0.30	0.90	0.56	0.15
Amat40_3	0.20	1.00	0.67	0.27	0.30	0.90	0.59	0.15
Amat40_4	0.20	1.00	0.65	0.27	0.30	0.90	0.59	0.15
Amat40_5	0.20	1.00	0.64	0.25	0.30	0.80	0.54	0.16
Amat50_1	0.20	1.00	0.58	0.27	0.30	0.70	0.48	0.11
Amat50_2	0.20	1.00	0.60	0.28	0.30	0.90	0.55	0.14
Amat50_3	0.20	1.00	0.60	0.28	0.30	0.90	0.57	0.14
Amat50_4	0.20	1.00	0.58	0.28	0.30	0.90	0.57	0.14
Amat50_5	0.20	1.00	0.58	0.25	0.30	0.80	0.52	0.16

reached solving the MILP (1)–(12), the gap is equal to 0, allowing to conclude that the solutions found by CPLEX are optimal for these two cases too.

6. CONCLUSIONS

In this work, a new approach to redistribute the vehicles of a one-way electric car sharing service has been proposed. The EVs are transferred by a team of workers, who can travel from a delivery point to a pickup point using folding bicycles that can be loaded in the trunk of the EVs. Such approach generates a new challenging Paired Pickup and Delivery Problem with Time Windows for which we propose the first MILP formulation. The formulation is based on a graph modeling of the problem rather than directly on the road network for two reasons: avoiding nonelementary cycles for the worker route representation and reducing the number of binary variables used. To further reduce the number of binary variables, we established necessary conditions for which a pickup relocation request and a delivery relocation request are compatible. Such conditions are also used in a preprocessing algorithm to eliminate from the instances possible relocation requests that certainly cannot be satisfied.

To evaluate the impact of the number of employed workers on the number of satisfied relocation requests, we developed a method to compute an upper bound on the maximum number of requests that can be satisfied with an unlimited number of workers.

We tested the MILP formulation of the EVRP on a test bed of 100 instances based on the road network of Milan (about 23,000 road links) considering up to 50 relocation requests and a number of workers $K = 1, 2, 3, 4$. The results show that with $K = 4$ a high percentage of relocation requests can be satisfied: on average 83%, with an average gap of only 7% with the maximum number of requests that may be satisfied with an unlimited number of workers. The employment of $K = 3$ workers is already sufficient to satisfy all instances with 10 requests, and this seems to be also the suitable number of workers to be used for the instances up to 30 requests since in this case, the average marginal improvement switching from $K = 3$ to $K = 4$ is only 3.11%.

Concerning the CPU time, while for instances up to 20 requests the MILP formulation solved by CPLEX 11.0 on a PC Intel Xeon 2.80 GHz with 2 GB RAM requires on average less than 5 s, it can require a few minutes for instances up to 30 requests, and more than 15 min for instances with 40 and 50 requests (for 20 instances, the maximum CPU time allowed of 30 min has been reached). To overcome such computational difficulties due to the non-deterministic polynomial-time (NP)-hardness of the EVRP, we have also developed a MILP-based heuristic and three upper bound methods (in addition to the one for an unlimited number of workers) to estimate the quality of the heuristic when the optimal solution cannot be found by CPLEX.

The experiments on the same test bed used for the MILP formulation show that the heuristic obtains high quality solutions with by far less CPU time than solving the MILP by CPELX. Indeed the average relative gap with the CPLEX solution values is less than 3.28%, and on average the CPU time is less than half the one necessary to CPLEX.

TABLE 9. Statistics on the time windows of all relocation requests considered in each instance.

Instance	min_TW_P	max_TW_P	av_TW_P	dev_TW_P	min_TW_D	max_TW_D	av_TW_D	dev_TW_D
Amat10_1	30	100	48.00	27.13	30	80	46.00	17.72
Amat10_2	30	100	46.00	27.28	30	80	51.00	18.00
Amat10_3	20	88	44.00	23.66	10	45	33.00	12.08
Amat10_4	18	35	25.60	6.28	10	25	21.00	5.83
Amat10_5	20	25	22.00	2.45	20	25	21.00	2.00
Amat20_1	27	180	58.90	45.79	30	80	53.50	16.59
Amat20_2	20	100	47.80	27.63	10	80	44.50	19.93
Amat20_3	20	88	36.70	19.33	10	45	34.00	10.91
Amat20_4	15	40	25.00	7.60	10	35	24.00	7.68
Amat20_5	15	27	21.70	3.47	10	35	23.50	6.73
Amat30_1	27	180	63.93	42.27	25	80	49.67	16.07
Amat30_2	15	100	47.33	25.38	10	80	41.67	19.38
Amat30_3	20	88	35.33	16.31	10	60	37.00	11.94
Amat30_4	15	40	25.87	7.24	10	50	25.67	9.98
Amat30_5	15	47	22.93	7.74	10	35	22.93	6.20
Amat40_1	27	180	64.35	37.20	25	80	51.75	16.98
Amat40_2	15	100	43.35	23.36	10	80	42.25	17.64
Amat40_3	15	88	36.25	16.75	10	70	36.00	14.37
Amat40_4	15	47	26.75	8.28	10	60	27.45	12.19
Amat40_5	10	47	22.25	8.31	10	35	21.95	6.91
Amat50_1	23	180	57.20	36.32	20	80	46.92	18.09
Amat50_2	15	100	39.60	22.25	10	80	36.52	17.20
Amat50_3	15	88	34.32	15.52	10	70	33.12	14.14
Amat50_4	15	47	26.32	8.35	10	60	27.08	11.13
Amat50_5	10	47	22.32	8.50	10	35	22.28	6.48

Future work on the EVRP concerns the investigation of a multiobjective version of the problem where beside the maximization of the total number of the satisfied relocation requests, also the optimization of the working time is considered. Concerning the instances, a possible improvement consists in the generation of the pickup and delivery requests in more verisimilar ways exploiting the origin-destination traffic matrix yielded by the Milan transport agency. Moreover, we have also an interest in investigating the combination of the EVRP operator-based relocation approach with pricing policies, that is, the promotion of the vehicle relocation made by the users by varying the rental fare in function of the vehicles availabilities at the stations.

APPENDIX

Table 8 shows some statistics on the battery charge level of the considered relocation requests in each instance of the experimental campaign presented in Section 5. The first column indicates the instance name, the successive four columns indicate, respectively, the minimum, the maximum, the average, and the deviation value of the battery charge levels on all pickup requests; the successive four columns have the same meaning for the delivery requests.

Table 9 shows some statistics on the time window size of the considered relocation requests in each instance (the meaning of the columns is analogous to the one of Table 8).

Finally the columns of Table 10 show for each instance, the number of pickup requests, the number of delivery requests, the cardinality of the arc set A defined in Section 3 to model the EVRP, the cardinality of the reduced arc set $\tilde{A}$ introduced in Section 4 and the percentage cardinality saving passing from A to $\tilde{A}$.

TABLE 10. Information on the number of relocation requests and on the arcs considered in the model for each instance.

Instance	$\lvert P \rvert$	$\lvert D \rvert$	$\lvert A \rvert$	$\lvert \tilde{A} \rvert$	Arc saving %
Amat10_1	5	5	60	38	36.67
Amat10_2	5	5	60	37	38.33
Amat10_3	4	5	49	33	32.65
Amat10_4	5	5	60	32	46.67
Amat10_5	4	5	49	27	44.90
Amat20_1	10	10	220	114	48.18
Amat20_2	10	10	220	122	44.55
Amat20_3	8	8	144	100	30.56
Amat20_4	10	10	220	100	54.55
Amat20_5	9	10	199	99	50.25
Amat30_1	15	15	480	236	50.83
Amat30_2	15	15	480	228	52.50
Amat30_3	14	15	449	244	45.66
Amat30_4	15	15	480	221	53.96
Amat30_5	14	15	449	192	57.24
Amat40_1	20	20	840	418	50.24
Amat40_2	20	20	840	390	53.57
Amat40_3	20	20	840	374	55.48
Amat40_4	20	20	840	340	59.52
Amat40_5	19	20	799	332	58.45
Amat50_1	25	25	1300	592	54.46
Amat50_2	25	25	1300	554	57.38
Amat50_3	25	25	1300	537	58.69
Amat50_4	25	25	1300	502	61.38
Amat50_5	24	25	1249	500	59.97

ACKNOWLEDGMENTS

The authors thank Giovanni Alli and Andrea Giovanni Bianchessi of the Electronic and Information Technology Department of Politecnico di Milano for the technical information on EVs. The authors also thank the anonymous referees for their insightful comments.

REFERENCES

[1] G. Alli, L. Baresi, A. Bianchessi, G. Cugola, A. Margara, A. Morzenti, C. Ongini, E. Panigati, M. Rossi, and S. Rotondi, "Green move: Towards next generation sustainable smartphone-based vehicle sharing," Sustainable internet and ICT for sustainability (SustainIT), Pisa, Italy, IEEE, 2012, pp. 1–5.

[2] P. Angeloudis, J. Hu, and M. G. Bell, A strategic repositioning algorithm for bicycle-sharing schemes, Transportation Research Board 91st Annual Meeting, Washington DC, 2012.

[3] C. Archetti and M. Speranza, Vehicle routing in the 1-skip collection problem, J Oper Res Soc 55 (2004), 717–727.

[4] M. Arena, G. Azzone, A. Colorni, A. Conte, A. Luè, and V. Paruscio, Innovative configurations for electric vehicle sharing services, 19th Intelligent Transport Systems World Congress, Vienna, Austria, 2012.

[5] R. Aringhieri, M. Bruglieri, F. Malucelli, and M. Nonato, An asymmetric vehicle routing problem arising in the collection and disposal of special waste, Electron Notes Discrete Math 17 (2004), 41–47.

[6] M. Barth and M. Todd, Simulation model performance analysis of a multiple station shared vehicle system, Transp Res Part C Emerg Technol 7 (1999), 237–259.

[7] M. Barth, M. Todd, and L. Xue, User-based vehicle relocation techniques for multiple-station shared-use vehicle systems, Transp Res Record 1887 (2004), 137–144.

[8] L. Bodin, A. Mingozzi, R. Baldacci, and M. Ball, The rollon–rolloff vehicle routing problem, Transp Sci 34 (2000), 271–288.

[9] R. Botsman and R. Rogers, What's mine is yours: The rise of collaborative consumption, HarperCollins, 2010.

[10] B. Boyacı, N. Geroliminis, and K. Zografos, An optimization framework for the development of efficient one-way car-sharing systems, 13th Swiss Transp Res Conf, Ascona, Switzerland, 2013.

[11] L. Caggiani and M. Ottomanelli, A dynamic simulation based model for optimal fleet repositioning in bike-sharing systems, Proc Soc Behav Sci 87 (2013), 203–210.

[12] E. Chan, P. Gilhead, P. Jelínek, and P. Krejei, Sartre cooperative control of fully automated platoon vehicles, 18th ITS World Congress, Orlando, FL, 2011.

[13] F. Chauvet, N. Hafez, and J. M. Proth, Electric vehicles: Effect of the availability threshold on the transportation cost, Appl Stoch Model Bus Ind 15 (1999), 169–181.

[14] F. Chauvet, N. V. Hafez, J.-M. Proth, and N. Sauer, Management of a pool of self-service cars, J Intell Manuf 8 (1997), 459–465.

[15] F. Chauvet, A. O. Haouba, and J.-M. Proth, Pool of self-service cars: A balancing method, Technical report 3650, Institut national de recherche en informatique et en automatique (INRIA), 1999.

[16] T. Cucu, L. Ion-Boussier, Y. Ducq, and J.-M. Boussier, Management of a public transportation service: Carsharing service, Proc 6th Int Conf Theory Pract Perform Meas Manage, Dunedin, New Zeland, 2009.

[17] P. Daviet and M. Parent, Platooning techniques for empty vehicle distribution in the praxit ele project, Pro 4th Mediterr Symp New Dir Control Automat, Maleme, Greece, 1996.

[18] L. De Meulemeester, G. Laporte, F. Louveaux, and F. Semet, Optimal sequencing of skip collections and deliveries, J Oper Res Soc (1997), 57–64.

[19] A. Di Febbraro, N. Sacco, and M. Saeednia, One-way carsharing: Solving the relocation problem, Transp Res Board 91st Ann Meet, Washington DC, 2012.

[20] M. Dror, D. Fortin, and C. Roucairol, Redistribution of self-service electric cars (a case of pickup and delivery), Technical Report 3543, Institut national de recherche en informatique et en automatique (INRIA), 1998.

[21] Y. Du and R. Hall, Fleet sizing and empty equipment redistribution for center-terminal transportation networks, Manage Sci 43 (1997), 145–157.

[22] C. Duron, M. Parent, and J.-M. Proth, Analysis of the balancing process in a pool of self-service cars, Technical report 3949, Institut national de recherche en informatique et en automatique (INRIA), 2000.

[23] European Commission, White paper: Roadmap to a single European transport area—Towards a competitive and resource efficient transport system, COM, vol. 144, 2011.

[24] A. E. Fassi, A. Awasthi, and M. Viviani, Evaluation of carsharing network's growth strategies through discrete event simulation, Expert Syst Appl 39 (2012), 6692–6705.

[25] R. Fourer, D. Gay, and B. W. Kernighan, The ampl book, Duxbury Press, Pacific Grove, 2002.

[26] 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 211 (2011), 198–207.

[27] Grafo di offerta stradale versione 0.4 (road network database), Agenzia Mobilità Ambiente e Territorio, 2008 (Publicly downloadable at http://www.amat-mi.it/it/documenti/dati/).

[28] N. Hafez, M. Parent, and J.-M. Proth, Managing a pool of self service cars, Proc Intell Transp Syst, IEEE, Oakland, CA, 2001, 943–948.

[29] F. L. Hitchcock, The distribution of a product from several sources to numerous localities, J Math Phys 20 (1941), 224–230.

[30] F. Jégou and E. Manzini, "Collaborative services," P. Bala, C. Cagnin, C. Cipolla, J. Green, T. van der Horst, B. de Leeuw, H. Luiten, I. Marras, A. Meroni, S. Rocchi, P. Strandbakken, E. Stø, J. Thakara, S. Un, E. Vadovics, P. Warnke, and A. Zacarias (Editors), Social Innovation and Design for Sustainability with essay, Edizioni Poli. design, Milan, 2008.

[31] A. G. Kek, R. L. Cheu, and M. L. Chor, Relocation simulation model for multiple-station shared-use vehicle systems, Transp Res Rec 1986 (2006), 81–88.

[32] A. G. Kek, R. L. Cheu, Q. Meng, and C. H. Fung, A decision support system for vehicle relocation operations in carsharing systems, Transp Res Part E: Logist Transp Rev 45 (2009), 149–158.

[33] A. Luè, A. Colorni, R. Nocerino, and V. Paruscio, Green move: An innovative electric vehicle-sharing system, Proc Soc Behav Sci 48 (2012), 2978–2987.

[34] R. Nair and E. Miller-Hooks, Fleet management for vehicle sharing operations, Transp Sci 45 (2011), 524–540.

[35] S. Parragh, K. Doerner, and R. Hartl, A survey on pickup and delivery problems, J für Betriebswirtschaft 58 (2008), 21–51.

[36] M. Rainer-Harbach, P. Papazek, B. Hu, and G. Raidl, "Balancing bicycle sharing systems: A variable neighborhood search approach," M. Middendorf and C. Blum (Editors), Evolutionary computation in combinatorial optimization, vol. 7832, Springer, Berlin, 2013, pp. 121–132.

[37] S. Shaheen, D. Sperling, and C. Wagner, Carsharing in Europe and North American: Past, present, and future, Transp Q 52 (1998), 35–52.

[38] S. Shaheen, D. Sperling, and C. Wagner, A short history of carsharing in the 90's, J World Transp Policy Pract 5 (1999), 18–40.

[39] S. A. Shaheen and A. P. Cohen, Carsharing and personal vehicle services: Worldwide market developments and emerging trends, Int J Sustain Transp 7 (2012), 5–34.

[40] N. Touati-Moungla and V. Jost, Combinatorial optimization for electric vehicles management, J Energy Power Eng 6 (2012), 738–743.