DESIGN AND OPERATION FOR AN ELECTRIC TAXI FLEET

BERNAT GACIAS AND FRÉDÉRIC MEUNIER

Abstract. The deployment of electric taxi fleets is highly desirable from a sustainable point of view. Nevertheless, the weak autonomy of this kind of taxis requires a careful operation. The management of such a fleet looking to prevent possible run out of power is studied in this paper. The related question of locating charging terminals for the taxis is also addressed. Methods for dealing with these tasks are proposed and their efficiency is proved through simulations.

electric taxis and charging terminal location and fleet management system and mixed integer programming and simulation and taxi dispatching

1. Introduction

1.1. **Context.** *Centrale OO*[1] is a pioneering project aiming to deploy in Paris a fleet of 100% electric taxis. The company in charge of the management of the fleet is the *Société du Taxi Electrique Parisien (STEP)*. The deployment of such fleets finds its main motivation in sustainable issues: electric taxis release almost no air pollutants at the place where they are operated and have less noise pollution than internal combustion engine taxis. However, the main drawback of an electric taxi is its weak autonomy – 80 km in the case of the *Centrale OO* project. In taxi fleet management, two kind of requests can be differentiated: *booking requests* and *opportunistic requests*. The first ones can be immediate or in advance of travel and have to be processed by the taxi dispatching system which assigns the request to a taxi. The opportunistic requests correspond with the traditional taxi services picking up passengers at cab-ranks or from the side of the road. Of course, this kind of requests is not processed by the dispatching system. The constraints of the management, as expressed by the *STEP*, are

- A taxi must never run out of power
- An opportunistic demand inside Paris and its suburbs must always be satisfied (legal environment of Paris)
- The number of booking demands accepted has to be maximized

At a strategic level, the charging problem includes the determination of the best location for the charging terminals. The significant initial investment (the cost of an electric charging terminal is about 20.000 euros) and the restricted taxi autonomy give a high relevancy to the charging terminal location task. Indeed, a wrong placement may in effect lead to a poor fleet management with taxis having difficulties to charge the batteries due to charging terminals saturation or even with taxis constantly running out of charge to keep operating. Our purpose is to propose a practical way for computing the "best" locations for the charging terminals.

At an operational level, the charging problem of the taxis must therefore be carefully addressed. A good assignment of the trips to the taxis is crucial. We propose an efficient way to manage the electric fleet in real-time while taking into account the charging tasks.

1.2. **Model.** We describe now formally the model we deal with in this paper. We derive also some elementary relations, which gives some details on the capacity of a given system (in terms of number of trips that can be realized by unit of time). They are not used elsewhere in the paper.

1.2.1. *Input description.* A complete directed graph $G = (V, A)$ models the network. The vertices are points in the city at which trips start and finish. They can moreover be used to locate taxi charging terminals. The arcs model the possible trips. The duration of a trip is a random variable T_a of expectation τ_a. The mean

[1] See the website http://taxioo.com/index.html for an artistic view.

number of demands for trip a per unit of time is denoted λ_a. The demand for each possible trip $a \in A$ is split between a *booking demand* and an *opportunistic demand*, see Section 5 for a more accurate description.

There are n taxis. A taxi consumes γWh by unit of time when it is moving. It stores ρWh by unit of time when it is charging.

The number of charging terminals is denoted by r. Several terminals can be located at a same vertex.

1.2.2. *Elementary relations.* Let us denote by $\tilde{\lambda}_a$ the average number of demands for a trip a that are accepted by unit of time. We have $\tilde{\lambda}_a \leq \lambda_a$. Let $\tilde{\lambda} = \sum_{a \in A} \tilde{\lambda}_a$ be the average number of trips accepted by unit of time and let $\tau = \frac{1}{\tilde{\lambda}} \sum_{a \in A} \tilde{\lambda}_a \tau_a$ be the average duration of an accepted trip.

The energy consumption of the system by unit of time is $\gamma \tilde{\lambda} \tau$. The maximal rate of supply in energy is ρr. Therefore, we have the following inequality

$$\gamma \tilde{\lambda} \tau \leq \rho r \tag{1}$$

A second inequality can be derived by considerations on the time needed to perform the different tasks. Let us consider a taxi over a time window of sufficiently large duration T. Denote by x the time during which it stores energy at a charging terminal. Over the time window, it spends in average $\frac{T\tilde{\lambda}\tau}{n}$ unit of time with a customer on board. Therefore, we have

$$\frac{T\tilde{\lambda}\tau}{n} + x \leq T$$

During this duration x, it stores a quantity of energy that must cover in average the consumption over the time window. Hence

$$\frac{\gamma T\tilde{\lambda}\tau}{n} \leq \rho x$$

Combining these two inequalities leads to

$$(\gamma + \rho)\tilde{\lambda}\tau \leq n\rho \tag{2}$$

Equations (1) and (2) can be summarized in the following inequality.

$$\tilde{\lambda} \leq \min\left(\frac{n\rho}{(\gamma + \rho)\tau}, \frac{\rho r}{\gamma \tau}\right) \tag{3}$$

Knowing the number of taxis, their efficiency (encoded by γ), the number of charging terminals, and their efficiency (encoded by ρ), then an upper bound of the number of trips that can be accepted by unit of time can be calculated.

Note that the results of this section have not been used in the rest of the article. However, the relations presented are interesting and could be exploited in order to guide future system investments.

1.3. **Plan.** Section 2 is devoted to the literature review for the two problems addressed in this paper, namely charging terminal location and fleet management. The following sections − Section 3 and Section 4 − detail the approaches proposed for each of these problems. Next, we describe a simulator that has been implemented for the evaluation of the proposed approaches (Section 5). The results of the experiments are described in Section 6. The paper ends with concluding remarks (Section 7).

2. Literature review

2.1. **Location issue.** The location problem was originally defined by A. Weber when he considered how to position a single warehouse minimizing the total distance between the warehouse and a set of customers [26]. In 1964, Hakimi [7] defines the *P-median problem*, the problem consists in determining the best location for a set of limited facilities (facilities with finite capacities) in order to minimize the sum of the weighted distances between the clients and the facilities serving these clients.

The problem increases its relevance during the last decades. High costs related to property acquisition and facility construction make facility location projects a critical aspect of strategic planning for a wide range of private and public firms. Indeed, the fact that facility location projects are long-term investments leads the researchers to focus on dynamic and stochastic location problems (see [17] for a review of this extension of the problem). Another important variant of the problem is the *Capacitated Facility Location Problem (CFLP)* where facilities have a constraining upper limit on the amount of demand they can satisfy [15]. An extension of the CFLP closely to our problem is the *Capacitated Plant Location Problem with Multiple facilities in the same site (CPLPM)* [6]. In charging terminal location the positions of the terminals are not the only decision variables, the number of terminals at each position have to be fixed too.

However, in some real-world applications selecting the best location for distance minimization is not the best suitable choice. For example, in electric taxi charging terminal location, like in other critical applications such as ambulance and fire terminal location, the interest is to guarantee that the different geographic zones are covered by a facility (closer than a previously fixed covering distance). This class of problems are known as *Covering Location Problems* (see [27], [20], and more recently [23] for a complete review of covering problems). In that context, the covering issue can be sometimes modelled as a problem constraint. However, if the covering distance is fixed to a small value the problem might become unfeasible. The *Maximal Covering Location Problem (MCLP)* [2] locates the facilities in order to maximize the number of covered customers (customers with a distance to the nearest facility smaller than an initial fixed distance). An extension of the problem very interesting for critical applications is the maximal covering with mandatory closeness problem which imposes a maximal distance (less stringent than the covering distance) between the geographical zones and the nearest facility [2]. These covering models implicitly assume that if a geographical zone is covered by a facility then the facility will always be available to serve the demand. However, in some applications, when facilities have a fixed capacity, being covered is not sufficient to guarantee the demand satisfaction. We find in the literature some models attempting to overcome this issue by maximizing the number of geographical zones covered by multiple facilities [3, 8, 5].

In the electric taxi context, Wang [25] proposes a facility location model for recreation-oriented scooter recharge stations in order to satisfy single origin-destination journeys. More recently, Frade et al. [4] propose a model in order to optimize the demand covered within an acceptable level of service and to define the number and capacity of the stations to be installed in a neighborhood of Lisbon (Portugal).

We present in Section 3 the linear programming models proposed to solve the electric taxi charging location problem.

2.2. **Taxi dispatching.** Traditional taxi dispatching systems are characterized by two principles. First, simple rules such as for example “nearest taxi first” or “least utilized first” are used for dynamic taxi assignment and second, the geographical space is usually divided into zones. In the literature, most of works on the topic basically focus on customer waiting time minimization by proposing improved methods for rule-based systems. In this context, Shrivastava [22] describe a fuzzy model for rule selection and Alshamsi [1] propose a new technique for dynamically divide the dispatch areas.

The recent emergent use of transportation technologies (GPS, EDI, GIS) has widely increased the opportunities for fleet management optimization. It is also the case for taxi dispatching. For example, Seow [21] propose a collaborative model for taxi dispatching where a set of n taxis of the same zone are defined as the agents of the model and a set of n customers as the service-requests. The objective is then to maximize the total service quality solving a collaborative linear assignment problem. However, taxi dispatching is not the only aspect that can be optimized. For example, Lee [12] and Jia [10] use real-time taxi information to propose a model for taxi relocation recommendation based on demand forecasting and a probability model for the design of taxi stops, respectively.

Another approach for fleet management optimization consists in modeling the problem as a variant of the Pick-up and Delivery taxi Routing Problem with Time Windows (PDVRPTW). The idea is to plan a set of routes satisfying known in advance customer requests. In the taxi management context, Wang [24] propose a bi-criteria two-phase method with an initial feasible assignment first and a tabu search improvement later in order to minimize the number of taxis and the sum of travel times for advanced bookings. However, the idea to block some taxis only for advanced booking might in some cases yield to a fleet underutilization.

The *online* PDVRPTW seems to be more adapted for the problem as discussed in [18]. In that context, Horn [9] and Meng [14] try to fill the gap between simple non-optimized rule-based taxi dispatching systems and static routing approaches. The second paper describes a genetic network programming method in order to find the optimal balance between the waiting time and the detour time. The work of Horn [9] is of particular interest in relation to the present work, proposing a taxi dispatching system architecture similar to our fleet management system. He proposes a system for taxi travel time minimization composed by a set of insertion algorithms to decide whether a new customer is accepted or not and a set of optimization mechanisms in order to improve the solution. However, some important differences exist between our work and these last two fleet management systems. The first difference is that in our case, the constraints related to the restricted autonomy of the taxis have also to be taken into account by scheduling charging tasks in the routes of the taxis. The second difference is that, unlike us, both articles deal with the multi-customer problem authorizing customers to share the same taxi at the same time.

3. Electric taxis charging terminal location

The electric vehicle (EV) charging terminal location problem consists in determining the best locations of the charging terminals. The binary linear program has to take into account two important aspects. First, the charging terminals have to be conveniently spread over the geographical area in order to avoid remote geographical zones with difficult taxi operability and fleet management. The second aspect is that the model has to determine the number of charging points at each location in order to facilitate the charging process of the taxis by minimizing the risks of terminals saturation. For these purposes, we propose two models, one called the *P-median model*, the other the *Demand-based model*.

V is the set of geographical points of the problem and $J \subseteq V$ is the set of potential locations where the charging terminals can be located. The number of terminals is limited to r.

3.1. P-median model.

Following Hakimi [7], we define z_j as the decision variables indicating if a facility is located to the point j and l_{ij} as the variables indicating that the geographical point i is assigned to the facility located in j. The binary linear program minimizing the sum of the distances between clients and facilities can be written as follows.

$$\min \sum_{i \in V} \sum_{j \in J} dist_{ij} l_{ij} \tag{4}$$

s.t.

$$\sum_{j \in J} l_{ij} = 1 \text{ for all } i \in V \tag{5}$$

$$l_{ij} \leq z_j \text{ for all } i \in V,\ j \in J \tag{6}$$

$$\sum_{j \in J} z_j \leq r \tag{7}$$

$$z_j \in \{0,1\} \text{ for all } j \in J \tag{8}$$

$$l_{ij} \in \{0,1\} \text{ for all } i \in V,\ j \in J \tag{9}$$

3.2. Demand-based model.

Another approach consists in defining a model with two distances β_{far} and β_{close} as proposed by Church and ReVelle [2]. The idea is then to spread the terminals by fixing a maximal distance (β_{far}) between the different geographical zones and the nearest charging terminal and, at the same time, trying to maximize the demand that will be covered by a nearby charging terminal (β_{close}). Furthermore, we assume given a demand d_i attached to each possible location i. This demand is defined hereafter.

We define then J_i^{far} (resp. J_i^{close}) as the subset of points in J at distance less than β_{far} (resp. β_{close}) from $i \in V$. Conversely, V_j^{close} is the set of points at distance less than β_{close} from the point $j \in J$.

Let x_j be the decision variable indicating the number of terminals located at point $j \in J$ and y_{ij} be the fraction of the demand d_i for $i \in V$ covered by a charging terminal located in j at distance less than β_{close} from i.

The mixed-integer linear programming model proposed to solve the problem called *Demand-based model* is the following.

$$\max \sum_{j \in J} \sum_{i \in V_j^{close}} d_i y_{ij} \tag{10}$$

s.t.

$$\sum_{j \in J_i^{far}} x_j \geq 1 \quad \text{for all } i \in V \tag{11}$$

$$\sum_{j \in J_i^{close}} y_{ij} \leq 1 \quad \text{for all } i \in V \tag{12}$$

$$\sum_{i \in V_j^{close}} d_i y_{ij} \leq x_j \quad \text{for all } j \in J \tag{13}$$

$$\sum_{j \in J} x_j \leq r \tag{14}$$

$$x_j \in \mathbb{Z}_+ \quad \text{for all } j \in J \tag{15}$$

$$y_{ij} \in \mathbb{R}_+ \quad \text{for all } i \in V,\ j \in J_i^{close} \tag{16}$$

The objective (Eq. (10)) consists in maximizing the pointwise demand covered by a charging terminal considering the distance β_{close}. Eq. (11) imposes that a geographical zone $i \in V$ must be covered by at least one charging terminal considering the distance β_{far}. We stress that an adequate β_{far} makes it possible to spread the charging terminals over the geographical area. Eq. (12) specifies that for each geographical zone $i \in V$ the sum of the fractions of demand covered by a charging terminal considering the distance β_{close} has to be less than or equal to one. The idea here is that the demand of each geographical point can be satisfied by different charging terminals and our interest is to maximize the potential energy required being supplied by a terminal closer than β_{close}. Eq. (13) are the constraints linking the variables x_j with the variables y_{ij}. For a given potential charging terminal location $j \in J$, this last set of variables can only be positive if a charging terminal is finally located to $j \in J$, that means $x_j > 0$. Besides, Eq. (13) also imposes that the demand allocated to a charging terminal does not exceed its capacity. The goal of this constraint is to assign a larger number of terminals on the geographical points with a great demand decreasing that way the risk of saturation for the charging terminals. Finally, Eq. (14) limits the number of terminals of the problem.

Estimating a pointwise demand in energy. The proposed charging location model takes as input a pointwise demand. A way to build such a demand d_i attached to a vertex i consists in computing the energy needed by unit of time for the trips starting at i: it is precisely $\gamma \sum_{a \in \delta^+(i)} \lambda_a \tau_a$. Dividing this quantity by ρ provides the number of charging terminals ensuring this supply. It suggests to define

$$d_i^{out} = \frac{\gamma}{\rho} \sum_{a \in \delta^+(i)} \lambda_a \tau_a$$

In this equation, the pointwise demand is calculated considering the trips starting at vertex i. This model presumes that taxis charging tasks take place mostly at the origin of the trips. However, other realistic strategies can be also envisaged. One of these strategies is to consider that taxis charge the batteries at the end of the trips. A pointwise demand considering the energy consumed by unit of time for the trips arriving at i is also proposed. We can then define

$$d_i^{in} = \frac{\gamma}{\rho} \sum_{a \in \delta^-(i)} \lambda_a \tau_a$$

Finally, a third strategy is proposed considering a pointwise demand as a linear combination of d_i^{out} and d_i^{in}:

$$d_i^{mix} = \alpha d_i^{out} + (1 - \alpha) d_i^{in}$$

4. Fleet management

We describe in this section two ways for managing the fleet, a classical and rule-based one (Subsection 4.1), and an improved one trying to address explicitly the charging issue (Subsection 4.2). Let us first introduce some notations. Let CR_i be a booking customer request. Each customer request CR_i is defined by a start time S_i and an origin-destination pair $O_i - D_i$. The S_i is fixed by the customer when the customer request arrives. The completion time of a trip is $C_i = S_i + \tau_{O_i D_i}$, where $\tau_{O_i D_i}$ is the travel time between the origin and destination of the customer request CR_i. Finally, let $R : CT_j$ be a taxi charging task scheduled on the charging terminal CT_j.

4.1. A classical rule-based taxi dispatching system.

A taxi dispatching system based on the principles of the most common real-world systems (see for example [1], [11], or [22]) is described in this section. The architecture of the current taxi dispatching systems are very similar to the system illustrated in Figure 1. The two main components of the system are (1) a customer acceptation mechanism deciding for each new customer if it is accepted (the accepted customers are inserted into a queue of customers) or rejected and (2) a rule-based mechanism assigning accepted customer requests (trips) to the free taxis. For each accepted trip i, the assigning process has to start a few minutes (Θ) before the fixed start time (S_i) in order to maximize the chances to find a taxi to attend the demand. Once a trip is assigned to a taxi, the taxi is automatically blocked and the taximeter begins counting.

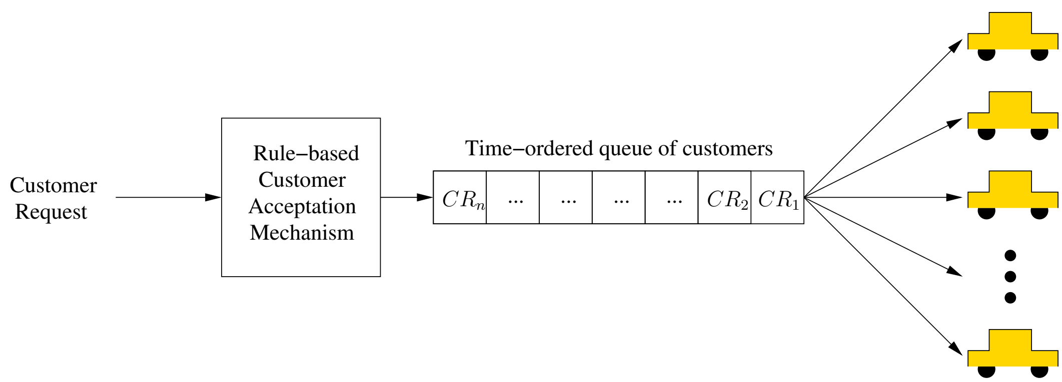

Figure 1. Rule-based taxi dispatching system

A rule for customer acceptation using the time windows for the trips already accepted is proposed. The idea is to limit the trips that must to be performed at the same time in order to minimize the number of not served customers and to establish a margin of k taxis to attend opportunistic customers. For each new customer request CR_{new} the Algorithm 1 determines if it is accepted or not.

Algorithm 1: Rule-based checking for customer acceptation for a margin of k taxis

$L = \{CR_1, CR_2, \ldots, CR_n\}$, list of already accepted customers
CR_{new}, new booking customer request
$nC \leftarrow 0$, number of trips performed at the same time as CR_{new}
foreach CR_i *of* L **do**
  if CR_i *is executed at the same time as* CR_{new} *(*$(S_i \leq S_{new} < C_i)$ *or* $(S_{new} \leq S_i < C_{new})$*)* **then**
    Step 1: Increase the number of customers performed at the same time as CR_{new} $(nC \leftarrow nC + 1)$
if *condition to accept the customer* $(nC < n - k)$ **then**
  Step 2: Insert CR_{new} to the list of accepted customers L

Once the customer request CR_i is accepted, it remains in the queue of customers until $S_i - \Theta$ (this Θ is usually fixed around 20 minutes). At that moment, the system automatically starts looking for a free taxi having sufficient charge to operate the trip. If different taxis are available, the system assigns the trip to the taxi minimizing the customer waiting time (a parameterizable not announced customer waiting time Δ can be authorized). In the case of no vacant taxis are available, the system waits for a taxi to become available. If the waiting time for any request exceeds the authorized maximal customer waiting time Δ, the customer request is then canceled. Note that the number of unsatisfied customers can be reduced by using a more restrictive rule for the customer acceptation mechanism.

The main advantage of such a system where no future work is planned is the high degree of independence for taxi drivers. On the other hand, the drawbacks are the underutilization of the fleet and the loss of efficiency during the peak hours when most of the companies have to close their booking requests systems in order to avoid unsatisfied customers. Furthermore, the charging tasks of the taxis cannot be controlled and it may leads to a poor fleet management with taxis having difficulties to charge the batteries due to charging terminals saturation. The experiments lead in Section 6 confirm these drawbacks.

4.2. **The improved electric taxi management system.** An improved fleet management system aiming to overcome the weakness of the rule-based taxi dispatching system is proposed in this section. The main objective is to maximize the number of accepted customers. One of the major issues is how to deal with opportunistic demand. Indeed, this kind of demand is unpredictable and must always be satisfied, so free taxis must be at any moment able to satisfy the longest trip without running out of charge. This constraint makes the problem considerably more complex forcing the system to provide a mechanism ensuring the feasibility of the already accepted trips each time an opportunistic demand is accepted.

The approach proposed consists in maintaining continuously a feasible planning for the taxis and the charging terminals (see Figure 2). Each time a customer asks for a trip, a simple *insertion algorithm* is run, at the end of which either the trip has been successfully inserted or not. The objective is to assign the customer to the taxi minimizing the customer waiting time (a parameterizable announced customer waiting time Δ can again be authorized). If none of the tried delays on the pick-up time leads to a feasible planning, a *rescheduling algorithm* allowing to reallocate the already accepted customers to the taxis is run.

In all these processes, a key routine is often called, namely the *charging task manager*, which schedules the charging tasks of a taxi, given a planning for the other taxis and the charging terminals.

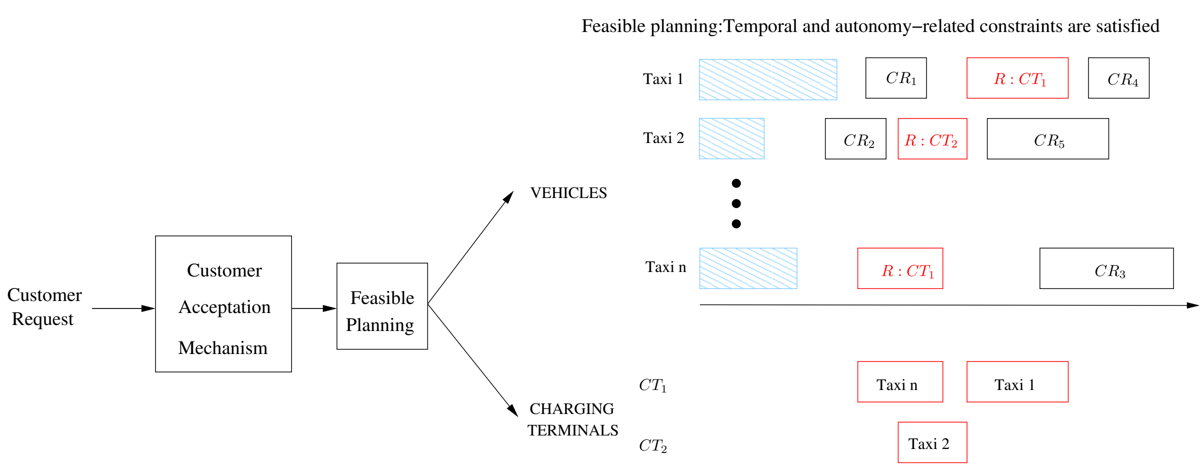

FIGURE 2. Customer acceptation mechanism for the electric taxi management system architecture

In the case of an opportunistic demand, which is necessarily accepted, we follow exactly the same scheme except that there is no degree of freedom in the insertion process: the trip is inserted at the front of the planning of the taxi stopped by the customer, and the rescheduling algorithm is also run if it is necessary.

4.2.1. *Insertion algorithm.* This algorithm is the first step in order to decide if a new trip CR_{new} is accepted or not. The objective is to assign the trip to the taxi minimizing the delay on the pick-up time (see Algorithm 2). The algorithm tests the different authorized pick-up times in increasing order. For each tentative start time, we sequentially try for each taxi (in an arbitrary and prefixed order) to insert the new request. First the scheduled charging tasks of the taxi currently tried are removed. Then the new request is accepted only if it can be inserted with no constraint violation (the pick-up times of the rest of customers are respected and the current autonomy of the taxi, without any charging task, is sufficient). In the case that the taxi autonomy-related constraint is violated, a greedy algorithm trying to schedule a charging task between each pair of trips is proposed (with the help of the "Charging task manager", see Section 4.2.3). After the charging tasks have been inserted, if the taxi is able to perform the trips without running out of charge, then the customer request is accepted.

Algorithm 2: New request insertion algorithm for a maximal authorized delay of Δ minutes

```
TX = {TX_1, TX_2, ..., TX_n}, list of taxis
CR_new, new booking customer request
accepted ← false, variable indicating if the new request is accepted
st ← S_new, start time of the trip
while st ≤ S_new + Δ and accepted = false do
    foreach TX_i of TX do
        Step 1: Delete the charging tasks of the taxi TX_i
        if CR_new starting at st can be inserted in the route of the taxi TX_i then
            if the taxi autonomy-related constraint is satisfied then
                Step 2: CR_new starting at st is inserted in the route of the taxi TX_i (accepted ← true)
            else
                Step 3: Insert charging tasks for TX_i between each pair of trips
                if the taxi autonomy-related constraint is satisfied then
                    Step 2: CR_new starting at st is inserted in the route of the taxi TX_i
                    (accepted ← true)
    if accepted = false then
        Step 4: Increase the pick-up time for the CR_new (st ← st + 1)
```

4.2.2. *Rescheduling algorithm.* The rescheduling algorithm is proposed when the new customer is still not accepted after the insertion algorithm. As for the insertion algorithm, the goal is to find a new feasible planning for the taxis integrating the new request CR_{new}. The main difference is that the trips can be reassigned to different taxis.

This rescheduling problem, without taking into account the autonomy-related constraints, can be solved in polynomial time [16]. The idea is to convert the schedule of trips (without the charging tasks) into a graph and to verify using a max flow computation that all trips can be performed by the taxis. To construct the network two vertices are considered for each customer request CR_i already accepted, the first one v_i represents the pick-up time and the second one v_i' the completion time of the customer request. Such a pair of vertices is also considered for CR_{new}. In addition, four dummy vertices are required: 0, $0'$, a source s, and a sink t. The arcs of the graph are $(s, 0)$, $(0', t)$, all the (s, v_i), all the (v_i', t), all the (v_i, v_i'), and all the (v_i', v_j) such that the customer request CR_j can be performed by the same taxi than the customer request CR_i and after CR_i, that means if $S_j \geq C_i + \tau_{D_iO_j}$. Except the arcs $(s, 0)$ and $(0', t)$, they all have a capacity equal to 1. The arcs $(s, 0)$ and $(0', t)$ have a capacity equal to n (the number of available taxis). A maximum flow computation in this directed graph determines the schedule feasibility and also proposes a new planning for the taxis respecting the customers pick-up times. Indeed, if the max flow $flow_{max}$ is equal to the number of customer requests $|CR|$ plus the number of taxis n ($flow_{\max} = |CR| + n$), then all customer requests can be processed by the n taxis satisfying customer requests time windows (see [16] for a complete proof). The sequences of customers for each taxi can be straightforwardly deduced following the paths of the flow from s to t.

This max flow computation checks the feasibility of the schedule for a given pick-up time $st \in [S_{new}, S_{new} + \Delta]$ and, if it is feasible, finds a reference planning (planning without charging tasks). The greedy algorithm for charging task scheduling is then sequentially applied to the taxis that do not satisfy autonomy-related constraints (that is, taxis whose current charge is not enough to perform all the trips assigned to them without adding charging tasks). The "Charging task manager" is used, see Section 4.2.3. If the schedule appears to be not feasible regarding the electric autonomy, a local search is applied. It explores the neighborhood of the reference planning. This local search is defined by the *swap* and the *reallocation* operators [19] between the schedules of the various taxis. For each explored planning respecting temporal constraints, the greedy algorithm for charging task scheduling is again sequentially applied to the taxis that do not satisfy autonomy-related constraints. If a feasible solution is found, the new customer is then accepted. Otherwise, it is rejected.

4.2.3. *Charging task manager.* As we have already seen, the *insertion* and the *rescheduling algorithms* constantly run a greedy algorithm aiming to insert a charging task between each pair of successive trips of the same route. The algorithms proposed to determine if a new charging task can be integrated in a specific charging terminal planning are described in this subsection. The main feature of our problem is that the processing time of the new charging task is not fixed, instead it is a decision variable defined between the interval limited by the minimal charging time for a taxi $p^{\min}$ (customizable parameter) and the maximal charging time corresponding to the time necessary for a full charge.

The problem to be solved by the charging terminal manager can be then formally stated as follows. A *charging task* R_j is defined by its time window $[r_j, d_j]$, where r_j is the earliest start time (earliest arrival time to the terminal) and d_j the latest end time (latest departure time from the terminal). Let p_j be the decision variable corresponding to the processing time of the task R_j, then $r_j \leq T_j$ and $T_j + p_j \leq d_j$, where T_j is the effective start time of R_j. In this scheduling problem, each task R_j has a maximal processing time $p_j^{\max}$, which is the time necessary for a full charge (taking into account the current charge of the taxi).

We are given a feasible schedule of l charging tasks $\mathcal{S}^l = \{(T_1, p_1), (T_2, p_2), \ldots, (T_l, p_l)\}$ for the charging terminals located at the same geographical position. A new charging task R_{l+1} with a time window $[r_{l+1}, d_{l+1}]$ and a processing time p_{l+1} inside the interval $p^{\min} \leq p_{l+1} \leq p_{l+1}^{\max}$ has to be inserted in the schedule. The problem consists in finding a new feasible schedule

$$\mathcal{S}^{l+1} = \{(T'_1, p_1), (T'_2, p_2), \ldots, (T'_l, p_l), (T_{l+1}, p_{l+1})\}$$

maximizing the processing time of the task R_{l+1}, the processing times for the other tasks being fixed.

The mechanism tests first a task insertion aiming to find quickly a feasible solution. The complexity of the algorithm for task insertion maximizing the processing time of the new task is $O(n)$ where the start times and completion times of the scheduled jobs are sorted in non-decreasing order. If no solution is found after the task insertion algorithm, a *dichotomous algorithm* allowing to reschedule the tasks is proposed in order to find a solution maximizing the processing time of the new task. For each iteration of the algorithm, a *satisfiability test* based on constraint propagation involving *energetic reasoning* is first triggered. The goal of the feasibility test is to detect an inconsistency indicating that it is not possible to find a feasible schedule integrating the new task. Finally, if the energetic reasoning is not conclusive a local search algorithm is proposed in order to find a solution.
Satisfiability test: Energetic reasoning. A satisfiability test based on constraint propagation involving energetic reasoning is proposed [13]. A fictitious energy (which has nothing to do with the electricity) is produced by the charging terminals and it is consumed by the charging tasks. We determine the fictitious energy consumed by the tasks ($E_{consumed}$) over a time interval $\delta = [t_1, t_2]$ and we compare this fictitious energy with the available fictitious energy produced by the m charging terminals located at the same geographical position ($E_{produced} = m \times (t_2 - t_1)$). The minimal fictitious energy consumed by the tasks in an interval $\delta = [t_1, t_2]$ is:

$$E_{consumed} = \sum_{i=1}^{n+1} \max\{0, \min\{p_j, t_2 - t_1, r_j + p_j - t_1, t_2 - d_j + p_j\}\} \tag{17}$$

If $E_{consumed} > E_{produced}$, then it is impossible to find a feasible schedule $\mathcal{S}^{n+1}$ integrating the new task. The relevant intervals δ for a complete satisfiability analysis can be enumerated in $O(n^2)$. The test is restricted to the intervals $[t_1, t_2]$ specified by $\{r_j\} \cup \{d_j\} \cup \{r_j + p_j\} \cup \{d_j - p_j\}$ where the new task R_{l+1} may consume ($t_1 \leq d_{l+1}$ and $t_2 \geq r_{l+1}$).

Dichotomous algorithm. A dichotomous algorithm maximizing the processing time of the new task is described in this section (see Algorithm 3). A dichotomy is run on the processing time p_{l+1} as follows. For processing times $p_{l+1} \in [p^{\min}, p_{l+1}^{\max}]$, the satisfiability test based on the energetic reasoning indicates whether the necessary conditions are satisfied or not. If it is the case, a local search mechanism tries to find a feasible schedule. The parallel machine scheduling problem with time windows can be solved by a list scheduling algorithm. It means there exists a total ordering of the jobs (i.e., a list) that, when a given machine assignment rule is applied, reaches the optimal solution. For our problem, this rule consists in allocating each task to the machine that allows it to start at the earliest (Earliest Start Time or EST rule). The local search mechanism proposed to solve the problem is based on this result. First, the tasks are ordered in non-decreasing order of their due dates (Earliest Due Date or EDD rule), then the local search consists in exploring different permutations of the list defined by the insertion neighborhood. The insertion neighborhood consists in taking a task from the list, removing it from its current position, and inserting it at any other position of the list. These new lists are computed for each possible task, providing a neighborhood of size $O(n^2)$. For each list of tasks, the machines (i.e. the terminals at the considered location) are assigned according to the EST rule in order to reach a feasible solution. If no feasible schedule is eventually found, the request is rejected.

Algorithm 3: Dichotomous algorithm for processing time maximization

```
min ← p^min
max ← p_{l+1}^max
S_best^{l+1} ← ∅
while min ≤ max do
    Step 1: Fix the processing time p_{l+1} of the new task R_{l+1} (p_{l+1} ← ⌊(min + max)/2⌋)
    if SatisfiabilityTest() then
        Step 2: Sort the tasks according to the EDD rule
        Step 3: Local search using the insertion operator
        if a feasible schedule S^{l+1} = {(T_1, p_1), (T_2, p_2), ..., (T_l, p_l), (T_{l+1}, p_{l+1})} is found then
            Step 4: Update the lower limit (min ← p_{l+1} + 1)
            Step 5: Update the best solution (S_best^{l+1} ← S^{l+1})
        else
            Step 6: No solution exists, update the upper limit (max ← p_{l+1} − 1)
    else
        Step 7: No solution exists, update the upper limit (max ← p_{l+1} − 1)
if S_best^{l+1} = ∅ then
    Step 8: No solution is found (return ∅)
else
    Step 9: A feasible solution is found (return S_best^{l+1})
```

5. Taxi simulator

The taxi behaviour simulator consists in a discrete-events simulator programmed in C++. It simulates the model described in Section 1.2. The demand for each possible trip $a \in A$ is assumed to follow a Poisson process of rate λ_a. Each possible trip between the points of the city is characterized by two parameters, λ_{book} and λ_{opp}, representing the rates of the Poisson process for booked and opportunistic demand, respectively. For sake of simplicity, the duration and the distance of each trip are considered constant over the time. The inputs of the simulator are a description of the system state and the simulation parameters. The system is originally defined by:

- A set of points defined by their coordinates representing the geographical points of the city.
- The location of the charging terminals.
- The fleet of taxis. Each taxi is defined by the following parameters:
  - AUT: it is the current autonomy of the taxi.
  - MAX_AUT: it is the autonomy of the taxi when it is fully charged.

– POS: it is the initial position of the taxi.

Opportunistic and booking demands are treated differently. The booking demands are managed by the taxi dispatching system deciding whether a demand is accepted or not. The opportunistic demands always have to be satisfied, the simulator assigns the trip to a free taxi located at the same geographical point. As we can note, an opportunistic demand is only considered when it exists spatial and temporal coincidence between the demand and a free taxi, otherwise the demand is simply ignored. It is worth recalling that the opportunistic demands may lead to unsatisfied booking demands initially accepted by the taxi dispatching system.

6. Computational experiments

The fleet management systems and the charging terminal location models are compared and evaluated by simulation. The network $G = (V, A)$ used for the simulations has 100 vertices and mimics the city of Paris and its suburbs. The vertices are randomly spread over a zone of 40 km^2. The geographic area has been divided into two zones; a central area of 20 km^2 with 60 vertices and the remainig with only 40 vertices. The demand for a trip a, i.e. λ_a, is more important for the trips belonging to the central area. Besides, the two airports of Paris and the four main train stations have been also modelled as vertices in V, with higher demands for the trips starting at or finishing to one of these vertices. We consider a first set of simulations with a *weak demand* ($\lambda_{book}^{weak} \approx 0.4$ and $\lambda_{opp}^{weak} \approx 1.0$) and a second set of simulations with a *strong demand* ($\lambda_{book}^{strong} \approx 0.8$ and $\lambda_{opp}^{strong} \approx 2.0$). A simulation is replicated 10 times with various seeds and consists in a choice of a number of terminals $r \in \{5, 20, 40\}$ and a number of taxis $n \in \{50, 100, 200\}$. Each replication simulates 900 minutes of working.

The tests were run on a Pentium(R) Dual-Core CPU E5500 2.80 GHz with 3.8 Gio of Ram under a linux Debian operating system. The maximal authorized delay on pick-up time is fixed to $\Delta = 15$ minutes for both systems and the minimal charging time for a taxi is fixed to $p^{\min} = 10$ minutes.

6.1. **Charging terminal location.** The demand-based linear model for charging terminals location presented in Section 3.2 has been compared with the P-median model minimizing the distance between the geographical points and the nearest charging terminal. The improved electric taxi management system introduced in Section 4.2 has been used to compare the models. Both models have been solved using IBM-ILOG CPLEX Optimizer with Concert Technology in C++.

Table 1 shows the results for the comparison between both models for the 40 instances with 5 charging terminals. The first column (*NbTrips*) displays the average of accepted customers. *NbBooking* is the average of served booking requests. The average percentage of operating time (free taxi looking for an opportunistic customer, a taxi going to pick-up a customer and an occupied taxi) and the average percentage of time when a taxi is waiting for an available charging terminal are displayed on the last two columns. The last two indicators are appropriate to assess the efficiency of the models by evaluating the profitable time and the waste of time due to charging terminals saturation.

When the number of charging terminals is equal to five, the demand-based model outperforms the P-median model, for any criterion. More customers are satisfied, the operating time of taxis is higher and the time waiting for an available charging terminal is drastically reduced. In some cases, this last value is even divided by two. The different ways proposed to estimate the pointwise demand have been also compared (for d_{mix}, we set $\alpha = 0.5$). We observe that no strategy clearly outperforms the others.

When the number of charging terminals is larger as in Table 2 and Table 3 (20 and 40 charging terminals), there is no real difference between the two models. This is not surprising since in those cases charging terminals are no longer a critical resource. This fact explains also the small waiting time for charging. In fact, for large number of charging terminals, we expect similar behaviours for any locations regularly spread over the area of experimentation.

6.2. **Fleet management.** The improved fleet management system has been compared with the rule-based taxi dispatching system presented in Section 4.1. The charging terminals have been located using the demand-based model with the d_{out} strategy to compute the pointwise demand. Table 4, Table 5, and Table 6 display the results for the comparison between both systems for the instances with 5, 20, and 40 charging terminals, respectively. The first column (*NbTrips*) displays the average of served customers. *NbBooking* is the average

$r = 5$ *weak demand*		*NbTrips*	*NbBooking*	*%Operating*	*%Waiting for charging*
$n = 100$	*P-median model*	481.6	358.9	85.41 %	9.52 %
	Demand-based model (d_{out})	497.6	361.6	90.98 %	4.67 %
	Demand-based model (d_{in})	488.7	359.1	91.84 %	3.63 %
	Demand-based model (d_{mix})	488.5	358.4	91.17 %	4.41 %
$n = 200$	*P-median model*	551.6	366.3	90.16 %	7.06 %
	Demand-based model (d_{out})	567.9	367.8	94.97 %	2.80 %
	Demand-based model (d_{in})	563.3	364.9	94.99 %	2.70 %
	Demand-based model (d_{mix})	560.3	366.1	94.95 %	2.80 %
strong demand					
$n = 100$	*P-median model*	744.3	652.8	55.19 %	38.04 %
	Demand-based model (d_{out})	777.1	671.8	64.08 %	29.44 %
	Demand-based model (d_{in})	803.4	677.2	68.83 %	24.28 %
	Demand-based model (d_{mix})	789.9	672.7	65.63 %	27.67 %
$n = 200$	*P-median model*	886.0	702.1	66.01 %	30.33 %
	Demand-based model (d_{out})	943.5	722.4	75.04 %	21.48 %
	Demand-based model (d_{in})	943.0	716.7	77.30 %	19.09 %
	Demand-based model (d_{mix})	953.0	721.8	77.43 %	19.03 %

TABLE 1. Comparison between different programming models for instances with 5 charging terminals

$r = 20$ *weak demand*		*NbTrips*	*NbBooking*	*%Operating*	*%Waiting for charging*
$n = 100$	*P-median model*	497.0	360.2	94.46 %	0.19 %
	Demand-based model (d_{out})	495.9	360.6	93.99 %	0.28 %
	Demand-based model (d_{in})	499.0	361.8	94.19 %	0.22 %
	Demand-based model (d_{mix})	495.1	362.5	93.98 %	0.31 %
$n = 200$	*P-median model*	566.7	366.0	96.79 %	0.11 %
	Demand-based model (d_{out})	564.5	366.5	96.49 %	0.17 %
	Demand-based model (d_{in})	561.2	367.3	96.57 %	0.16 %
	Demand-based model (d_{mix})	561.1	367.3	96.49 %	0.20 %
strong demand					
$n = 100$	*P-median model*	894.8	731.9	90.02 %	0.89 %
	Demand-based model (d_{out})	895.4	731.8	88.97 %	1.26 %
	Demand-based model (d_{in})	884.7	731.1	89.21 %	1.15 %
	Demand-based model (d_{mix})	895.5	734.7	88.59 %	1.36 %
$n = 200$	*P-median model*	994.9	738.3	93.85 %	0.73 %
	Demand-based model (d_{out})	1005.4	742.0	93.40 %	0.88 %
	Demand-based model (d_{in})	997.0	737.8	93.28 %	0.87 %
	Demand-based model (d_{mix})	1008.9	741.2	93.19 %	0.95 %

TABLE 2. Comparison between different programming models for instances with 20 charging terminals

of served booking requests. The number of unsatisfied customers (customers initially accepted by the system but they are never served) is indicated by *Unsatisfied Customers* at the third column. Finally, the last column indicates the average customer overcharge (number of minutes in the taximeter when the taxi arrives at the pick up point).

The results show that the proposed fleet management improves the rule-based taxi dispatching system. The number of served customers is in general more important with the improved fleet management system.

$r = 40$		*NbTrips*	*NbBooking*	*%Operating*	*%Waiting for charging*
weak demand					
$n = 100$	*P-median model*	492.6	361.6	94.88 %	0.06 %
	Demand-based model (d_{out})	493.8	360.7	94.71 %	0.06 %
	Demand-based model (d_{in})	493.3	361.0	94.85 %	0.04 %
	Demand-based model (d_{mix})	492.8	360.7	94.69 %	0.05 %
$n = 200$	*P-median model*	549.7	366.1	97.08 %	0.03 %
	Demand-based model (d_{out})	556.1	364.7	97.04 %	0.03 %
	Demand-based model (d_{in})	560.5	365.2	97.06 %	0.03 %
	Demand-based model (d_{mix})	559.1	366.8	96.99 %	0.02 %
strong demand					
$n = 100$	*P-median model*	902.1	729.0	91.30 %	0.24 %
	Demand-based model (d_{out})	897.6	729.3	91.66 %	0.13 %
	Demand-based model (d_{in})	896.4	730.8	91.44 %	0.14 %
	Demand-based model (d_{mix})	897.0	730.2	91.31 %	0.13 %
$n = 200$	*P-median model*	999.0	738.1	95.00 %	0.14 %
	Demand-based model (d_{out})	1009.3	738.1	94.96 %	0.09 %
	Demand-based model (d_{in})	1008.8	737.5	94.90 %	0.11 %
	Demand-based model (d_{mix})	1006.5	737.5	94.91 %	0.12 %

TABLE 3. Comparison between different programming models for instances with 40 charging terminals

Furthermore, and this may be its main advantage, the percentage of unsatisfied customers remains acceptable and almost constant with this system, contrarily to what happens for the rule-based taxi dispatching. For this latter, the percentage of unsatisfied customers grows with the demand size when charging terminals are saturated, as we can see for the instances with 50 and 100 taxis and 5 charging terminals. This fact can be explained because the taxi charging aspects are completely ignored in the customer acceptation rule of the rule-based taxi dispatching system. This behaviour is not acceptable for a real-life taxi management system based on customer loyalty and good quality of service. Finally, customer overcharge is very similar for both systems, for the instances with 20 and 40 resources here again the improved fleet management is a better system than the rule-based taxi dispatching. However, we did not find a satisfactory explanation for the higher curstomer overcharge observed for the improved fleet management when $r = 5$.

Concerning the time needed to compute the answer to a booking request, it is in general under 1 ms, and in less than 2% of the cases above 5 s. In any case, the computation time is always reasonable, and, with faster computers, the time needed to compute the answer could be reduced if required.

6.3. **Interaction between location decisions and fleet management.** The goal of this section is to assess the interaction between strategic and operational decision levels. The idea is to evaluate the effect of the higher level-decisions at a strategic level on the lower-level (or operational) outcomes. Concretely, we want to see the impact of using a demand-based location model instead of a P-median one when the fleet is managed according to the rule-based taxi dispatching policy, and similarly, to see the impact of managing the fleet with the improved management system instead of the rule-based one when the location model is the P-median one.

The simulations have been performed on the strong demand set of instances generated from different values for the number of terminals ($r \in \{5, 20, 40\}$) and for the number of taxis ($n \in \{50, 100, 200\}$). Table 7, Table 8, and Table 9 display the results when we compare the effects of modifying strategies for the different decision levels. Note that the demand-based model has been used with the d_{mix} strategy, which explains the differences with Table 4, Table 5, and Table 6.

The results show that a less performant fleet management system like the rule-based TD can be indeed compensated by using the demand-based model for most of the cases, especially when the number of charging terminals is small. Even for $r = 20$, see Table 8, we see that the demand-based location model has a positive impact on the results. In contrast, when the charging terminals form a critical resource, $r = 5$ (Table 7),

$r = 5$		*NbTrips*	*NbBooking*	*Unsatisfied Customers (%)*	*Overcharge (min)*
weak demand					
$n = 50$	*Rule-based TD*	428.4	353.3	19.5 (5.23 %)	6.56
	Improved FM	437.2	354.2	0.6 (0.17 %)	6.68
$n = 100$	*Rule-based TD*	498.8	361.8	11.0 (2.95 %)	6.07
	Improved FM	499.7	362.2	1.0 (0.28 %)	6.34
$n = 200$	*Rule-based TD*	568.9	368.3	4.5 (1.21 %)	5.78
	Improved FM	571.5	368.4	0.6 (0.16 %)	5.94
strong demand					
$n = 50$	*Rule-based TD*	659.1	614.3	144.8 (19.08 %)	5.89
	Improved FM	672.4	624.1	0.9 (0.14 %)	6.79
$n = 100$	*Rule-based TD*	778.9	671.6	87.5 (11.53 %)	5.96
	Improved FM	788.2	675.5	1.0 (0.15 %)	6.48
$n = 200$	*Rule-based TD*	943.6	723.4	35.7 (4.70 %)	5.87
	Improved FM	941.7	717.0	2.1 (0.29 %)	6.13

TABLE 4. Comparison between different fleet management systems for instances with 5 charging terminals

$r = 20$		*NbTrips*	*NbBooking*	*Unsatisfied Customers (%)*	*Overcharge (min)*
weak demand					
$n = 50$	*Rule-based TD*	433.9	356.3	16.5 (4.43 %)	6.47
	Improved FM	433.4	354.9	0.9 (0.25 %)	6.37
$n = 100$	*Rule-based TD*	488.9	360.6	12.2 (3.27 %)	6.05
	Improved FM	492.9	361.4	1.0 (0.28 %)	6.01
$n = 200$	*Rule-based TD*	554.8	368.5	4.3 (1.15 %)	5.77
	Improved FM	549.9	365.7	1.5 (0.41 %)	5.56
strong demand					
$n = 50$	*Rule-based TD*	776.3	710.6	47.3 (6.24 %)	6.60
	Improved FM	781.9	710.1	2.3 (0.32 %)	6.42
$n = 100$	*Rule-based TD*	881.9	732.3	26.8 (3.53 %)	6.00
	Improved FM	887.3	730.9	2.8 (0.38 %)	5.72
$n = 200$	*Rule-based TD*	991.4	744.3	14.8 (1.95 %)	5.74
	Improved FM	994.0	744.8	2.8 (0.37 %)	5.40

TABLE 5. Comparison between different fleet management systems for instances with 20 charging terminals

the improved fleet management does not seem to be able to correct their poor position computed according to the P-median model. However, the rule-based taxi dispatching approach always leads to unsatisfied users (we have not diplayed these numbers again).

Finally, we observe here the same behaviour as in the last sections, the strategy used for charging terminal location and taxi dispatching becomes less relevant when the system has enough capacity to satisfy the demand.

$r = 40$		NbTrips	NbBooking	Unsatisfied Customers (%)	Overcharge (min)
weak demand					
$n = 50$	*Rule-based TD*	430.2	351.9	20.9 (5.61 %)	6.39
	Improved FM	432.4	355.7	1.2 (0.34 %)	6.45
$n = 100$	*Rule-based TD*	486.7	359.8	13.0 (3.49 %)	6.13
	Improved FM	498.8	361.3	0.6 (0.17 %)	5.89
$n = 200$	*Rule-based TD*	557.7	364.8	8.0 (2.15 %)	5.80
	Improved FM	560.3	366.0	1.1 (0.30 %)	5.55
strong demand					
$n = 50$	*Rule-based TD*	797.8	716.7	41.1 (5.42 %)	6.48
	Improved FM	792.2	711.9	2.4 (0.34 %)	6.29
$n = 100$	*Rule-based TD*	906.0	731.2	27.9 (3.68 %)	5.99
	Improved FM	905.4	729.5	2.3 (0.31 %)	5.72
$n = 200$	*Rule-based TD*	1003.8	742.0	17.1 (2.25 %)	5.64
	Improved FM	1014.2	738.4	3.5 (0.47 %)	5.39

Table 6. Comparison between different fleet management systems for instances with 40 charging terminals

$r = 5$		NbTrips	NbBooking	%Operating
strong demand				
$n = 50$	*P-median model - Rule-based TD*	600.4	565.7	49.77 %
	Demand-based model (d_{mix}) - Rule-based TD	663.7	612.8	59.67 %
	P-median model - Improved FM	619.6	583.5	52.64 %
$n = 100$	*P-median model - Rule-based TD*	741.6	649.6	54.69 %
	Demand-based model (d_{mix}) - Rule-based TD	793.2	676.1	63.87 %
	P-median model - Improved FM	744.3	652.8	55.19 %
$n = 200$	*P-median model - Rule-based TD*	903.8	706.7	66.27 %
	Demand-based model (d_{mix}) - Rule-based TD	945.1	720.7	76.03 %
	P-median model - Improved FM	886.0	702.1	66.01 %

Table 7. Evaluation of decision levels interaction for instances with 5 charging terminals

$r = 20$		NbTrips	NbBooking	%Operating
strong demand				
$n = 50$	*P-median model - Rule-based TD*	782.9	708.5	85.11 %
	Demand-based model (d_{mix}) - Rule-based TD	784.0	714.0	82.94 %
	P-median model - Improved FM	779.1	707.2	84.97 %
$n = 100$	*P-median model - Rule-based TD*	888.9	728.8	90.23 %
	Demand-based model (d_{mix}) - Rule-based TD	890.9	734.4	88.88 %
	P-median model - Improved FM	894.8	731.9	90.02 %
$n = 200$	*P-median model - Rule-based TD*	999.3	740.9	93.33 %
	Demand-based model (d_{mix}) - Rule-based TD	1014.1	747.4	93.01 %
	P-median model - Improved FM	994.9	738.3	93.85 %

Table 8. Evaluation of decision levels interaction for instances with 20 charging terminals

$r = 40$ *strong demand*		*NbTrips*	*NbBooking*	*%Operating*
	P-median model - Rule-based TD	786.5	711.0	87.26 %
$n = 50$	*Demand-based model* (d_{mix}) *- Rule-based TD*	786.5	711.6	87.30 %
	P-median model - Improved FM	793.1	712.2	86.67 %
	P-median model - Rule-based TD	902.6	733.9	91.70 %
$n = 100$	*Demand-based model* (d_{mix}) *- Rule-based TD*	889.3	727.6	92.00 %
	P-median model - Improved FM	902.1	729.0	91.30 %
	P-median model - Rule-based TD	999.3	740.4	95.23 %
$n = 200$	*Demand-based model* (d_{mix}) *- Rule-based TD*	1006.7	741.8	95.04 %
	P-median model - Improved FM	999.0	738.1	95.00 %

TABLE 9. Evaluation of decision levels interaction for instances with 40 charging terminals

7. Conclusion

In this paper, we deal with the management of a fleet of electric taxis. Two different problems have been solved: the electric charging terminal location problem and the taxi dispatching problem. Different approaches have been proposed to solve each problem.

For the electric charging terminal location problem, the first approach is based on classical models of the literature to solve the facility location problem. The second approach is a demand-based mixed-integer linear programming model adapted to the particularities of our problem. The idea is to consider a pointwise demand in order to maximize the demand covered by a close charging terminal. The originality of the approach lies in the way how initial dynamic demands are estimated as static pointwise demands. Both models have been tested and compared on a set of realistic instances randomly generated. The results show that the proposed demand-based model outperforms the model minimizing the sum of distances when the number of charging terminals is small. When this number is large, charging terminals are no longer a critical resource and both models show similar behaviour. In short, for the location of the charging terminals, we clearly recommend to use the demand-based model. The demand-based model could furthermore be incorporated in a framework in which it is possible to move the terminals (this option is considered by the *STEP* company). We think that it would then be able to manage the breakdown of some terminals or strong changes in the user demand for the system, and that it would also be able to deal with a sharing of the terminals with vehicles not belonging to the fleet (see below).

For the taxi dispatching problem, two different approaches are proposed: a rule-based system inspired on the most real-life taxi dispatching systems and an improved fleet management system adapted to the problem constraints. The idea of the fleet management system is to continuously maintain a feasible solution for the problem (a feasible planning for the taxis and for the charging terminals). Scheduling algorithms and a local search scheme have been proposed to solve the problem. Both systems have been compared by simulation on randomly generated instances. The results show that rule-based systems cannot be easily adapted to the particularities of electric taxis. Indeed, a considerable loss in efficiency results when the number of charging terminals is small. A centralized management system seems to be more appropriate, to facilitate a better use of the resources, and to avoid too many unsatisfied users.

A possible direction of future research would precisely consist in studying the impact of sharing the charging terminals with vehicles not belonging to the fleet. The *STEP* company considers the possibility of allowing other vehicles (private, of from other companies) to use the charging terminals. In such a situation, even if there are many charging terminals (large r), they may remain a critical resource. The demand-based model would probably provide suitable locations for the terminals, but playing with the prices or the time windows on which there are open to other vehicles could improve the management of the system and increase the profit. How to compute these prices and to choose the time windows would certainly require a careful study and would probably be interesting for urban electric systems in general.

References

[1] Alshamsi, A., Abdallah, S., Rahwan, I.: Multiagent self-organization for a taxi dispatch system. In: 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2009)

[2] Church, R.L., ReVelle, C.: The maximal covering location problem. Papers of the Regional Science Association **32**, 101–118 (1974)

[3] Daskin, M.S., Stern, E.H.: A hierarchical objective set covering model for emergency medical service vehicle deployment. Transportation Science **15**(2), 137–152 (1981)

[4] Frade, I., Ribeiro, A., Gonçalves, G., Antunes, A.P.: Optimal location of charging stations for electric vehicles in a neighborhood in Lisbon. Transportation Research Record: Journal of the Transportation Research Board **2252**, 91–98 (2011)

[5] Gendreau, M., Laporte, G., Semet, F.: Solving an ambulance location model by tabu search. Location Science **5**(2), 75–88 (1997)

[6] Ghiani, G., Guerriero, F., Musmanno, R.: The capacitated plant location problem with multiple facilities in the same site. Computers & Operations Research **29**(13), 1903–1912 (2002)

[7] Hakimi, S.L.: Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research **12**(3), 450–459 (1964)

[8] Hogan, K., ReVelle, C.S.: Concept and applications of backup coverage. Management Science **32**(11), 1434–1444 (1986)

[9] Horn, M.E.T.: Fleet scheduling and dispatching for demand-responsive passenger services. Transportation Research Part C: Emerging Technologies **10**(1), 35–63 (2002)
[10] Jia, Y.: Improvement program of urban taxi stops based on simulated annealing algorithm in the context of China. In: Proceedings of the 2008 Second International Conference on Genetic and Evolutionary Computing, pp. 356–359. IEEE Computer Society, Washington, DC, USA (2008)
[11] Lee, D.H., Wang, H., Cheu, R.L., Teo, S.H.: Taxi dispatch system based on current demands and real-time traffic conditions. Transportation Research Record (1882), 193–200 (2004)
[12] Lee, J.H., Shin, I.H., Park, G.L.: Pick-up pattern for taxi location recommendation. In: 4th International Conference on Networked Computing and Advanced Information Management (2008)
[13] Lopez, P., Esquirol, P.: Consistency enforcing in scheduling: A general formulation based on energetic reasoning. In: 5th International Workshop on Projet Management and Scheduling (PMS'96), pp. 155–158. Poznan (Poland) (1996)
[14] Meng, Q., Mabu, S., Yu, L., Hirasawa, K.: A novel taxi dispatch system integrating a multi-customer strategy and genetic network programming. Journal of Advanced Computational Intelligence and Intelligent Informatics **14**(5) (2010)
[15] Mirchandani, P., R. Francis, e. (eds.): Discrete Location Theory. John Wiley and Sons, Inc., New York (1990)
[16] Neumann, K., Schwindt, C., Zimmermann, J.: Project Scheduling with Time Windows and Scarce Resources. Springer (2002)
[17] Owen, S.H., Daskin, M.S.: Strategic facility location: A review. European Journal of Operational Research **111**(3), 423–447 (1998)
[18] Parragh, S., Doerner, K., Hartl, R.: Part 2: Transportation between pickup and delivery locations. In: A survey on pickup and delivery problems, vol. 58, pp. 81–117. Journal für Betriebswirtschaft (2008)
[19] Savelsbergh, M.W.P.: The vehicle routing problem with time windows: Minimizing route duration. INFORMS Journal on Computing **4**(2), 146–154 (1992)
[20] Schilling, D.A., Vaidyanathan, J., Barkhi, R.: A review of covering problems in facility location. Location Science **1**(1), 25–55 (1993)
[21] Seow, K.T., Dang, N.H., Lee, D.H.: A collaborative multiagent taxi-dispatch system. IEEE Transactions on Automation Science and Engineering **7**(3), 607–616 (2010)
[22] Shrivastava, M., Chande, P.K., Monga, A.S., Kashiwagi, H.: Taxi dispatch: A fuzzy approach. In: IEEE Conference on Intelligent Transportation System (1997)
[23] Vijeyamurthy, C.N., Panneerselvam, R.: Literature review of covering problem in operations management. International Journal of Services, Economics and Management **2**(3-4), 267–285 (2010)
[24] Wang, H., Lee, D.H., Cheu, R.: PDPTW based taxi dispatch modeling for booking service. In: Proceedings of the 5th international conference on Natural computation, ICNC'09, pp. 242–247. IEEE Press, Piscataway, NJ, USA (2009)
[25] Wang, Y.W.: An optimal location choice model for recreation-oriented scooter recharge stations. Transportation Research Part D: Transport and Environment **12**(3), 231–237 (2007)
[26] Weber, A.: Über den Standort der Industrien - 1. Teil: Reine Theory des Standorts. Mohr, Tübingen (1909)
[27] White, J.A., Case, K.E.: On covering problems and the central facilities location problems. Geographical Analysis **6**(3), 281–294 (1974)

B. Gacias, Artelys, 12 rue du 4 Septembre, 75002 Paris, France bernat.gacias@gmail.com and F. Meunier, Université Paris Est, CERMICS, 6-8 avenue Blaise Pascal, Cité Descartes, 77455 Marne-la-Vallée, Cedex 2, France frederic.meunier@cermics.enpc.fr