Data Driven Optimization of Charging Station Placement for EV Free Floating Car Sharing

Data Driven Optimization of Charging Station Placement for EV Free Floating Car Sharing

Michele Cocca[a], Danilo Giordano[a], Marco Mellia[a] and Luca Vassio[a]
[a]Department of Electronics and Telecommunications, Politecnico di Torino, `first.last@polito.it`

***Abstract*— Free Floating Car Sharing (FFCS) is a transport paradigm where customers are free to rent and drop cars of a fleet within city limits. In this work we consider the design of a FFCS system based on Electric Vehicles (EVs). We face the problem of finding the minimum number of charging stations and their placement, given the battery constraints of electric cars, the cost of installing the charging network, and the time-varying car usage patterns of customers.**

Differently from other studies, we base our solution on actual rentals collected from traditional combustion FFCS systems currently in use in two cities. We use about 450 000 actual rentals to characterize the system utilization. We propose a user-behavior model and system policies for the charging events. Then we evaluate via accurate trace driven simulations the performance with different charging station placement policies. We first present greedy solutions, and then perform a local optimization with a meta-heuristic that 1) guarantee system operativeness, i.e., car batteries never get depleted, and 2) minimize users' discomfort, i.e., users are only seldom forced to drop cars in a far-away charging station.

Results show that it is possible to guarantee service continuity by installing charging stations in just 6% of city areas, while 15% of equipped zones guarantee limited impact on users' discomfort.

I. Introduction

The majority of the world population live in an urban environment.[1] Cities increasingly face problems caused by transport and traffic. The question of how to enhance mobility while at the same time reducing congestion and pollution is a common challenge to all major cities worldwide.

Smart and shared mobility are seen as a key component to reduce emissions and reduce traffic congestions in urban centers [1]. Given a fleet of vehicles, Free Floating Car Sharing (FFCS) systems allow users to pick and drop any car everywhere inside an operative area using a smart-phone application, and billing the customer only for the actual rental time. FFCS systems reduce the number of private cars increasing the number of available parkings. A further step is to convert internal combustion cars into Electric Vehicles (EVs), strongly reducing emissions and pollution in cities [2].

This conversion has, however, several challenges. Indeed EVs require to frequently charge the battery at a dedicated Charging Station (CS), with a full battery top-up that could require several hours. This creates discomfort for users whenever they are asked to return the car to a charging pole [3]. The design of the charging station network becomes thus vital to guarantee both service continuity and users' comfort maximization. The optimization of the CSs placement is a key problem given also the cost of charging poles installation.

Past studies already faced the CSs placement problem considering the mobility in a big city. For instance, authors of [4] state how the parking time is an important metric to optimize the CSs distribution, and validate their algorithms with artificial traffic patterns. Other works proposed a mathematical formulation and validated it using simulation, e.g., to minimize the total installation cost [7], or the impact on the electric grid [5], or additional distance the users are called to walk when forced to drop the EV to a CS [6].

The availability of actual data is key to drive these optimizations, and most studies use simulations to evaluate proposed solutions. A common approach is to validate models with trace driven simulations where traffic demand is extrapolated from Interview Travel Surveys (e.g., in Singapore [8], Lisbona [9], or Beijing [10]).

More recently, authors have started using rental traces collected from actual car sharing systems. For instance authors of [12] use real rental traces collected from a station based car charging system in use in France to feed their simulation. Traces in a campus were used also in [13] to predict the usage of CSs given their placement. Authors of [11] study the design of a self-driven EV taxi fleet to replace the current Munich's taxi service, using actual trips recorded by taxis.

In our past works we were among the first to collect longitudinal data of actual rentals of FFCS systems using traditional combustion vehicles [14], [15]. We used this data for dimensioning a FFCS system with EVs and identifying its possible system policies. We also perform a preliminary study of the CSs placement problem.

Here, starting from these two works, we rely on a larger dataset containing about half a million of real FFCS rentals recorded in Turin and Milan (Italy) for more than two months. First we characterize the usage pattern and then we propose a simple customer behavior model and system policy for the charging events. Differently from the other works based on real traces, in this work we focus on finding the required number of CSs and choose their placement. This is a challenging problem since we want to have as less CSs as possible to reduce costs, but at the same time we want the system to be usable and practical for users, e.g., customers should be rarely forced to return the car to a far-away CS.

Given recorded rentals we solve the CSs placement problem using first greedy algorithms, whose performance is thoroughly compared using accurate simulations. Then we

[1]`https://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS`

further optimize CS placement with a meta-heuristic algorithm.

We observe that wisely equipping just 6% of city areas (square region with 500 m side) guarantees to sustain the system and all trips, i.e., car battery never depletes. Moreover, results show that a good charging station placement in ≈15% of city areas would guarantee that more than 98% of the trips successfully end in the desired arrival area, with 2% of users that are rerouted to a charging station, which is less than 1.8 km far.

II. Data collection and characterization

We collect data from a FFCS system having a combustion engine fleet. We first describe the data collection methodology and then we provide a characterization of the data which is instrumental for the CS placement problem.

A. Data collection and filtering

Modern FFCS like car2go, DriveNow or Share 'n Go allow users to book a car through web service accessible with a web interface or a smartphone app. Some offers open APIs to access data about system status.[2] By leveraging these API, we developed UMAP [14], a web-crawler that collects car2go's information and is able to rebuild car rental history by reconstruction the *parkings* and *rentals* periods for each car. In a nutshell, we take a snapshot of the available cars every minute and derive parking and rental periods by observing which cars are available or booked at each snapshot.

We characterize each rental by its starting and final position, duration, and enrich it via a possible traveled distance between starting and final position as provided by Google Map service [3]. Parkings instead record the position and duration. Trip distances reflect the energy consumed during a rental, while parking positions and durations offer information about the energy an electric vehicle could obtain if connected to a CS. From September 5^{th}, 2017 to November 2^{nd}, 2017, we recorded about 125 000 rentals in Turin and 320 000 in Milan. We observed an average daily fleet of 377 vehicles in Turin and 749 in Milan.

B. Rental and parking characterization

Fig. 1 shows the Empirical Cumulative Distribution Function (ECDF) of the rentals duration observed in Turin and Milan.[4] Notice the log scale on x-axis. Turin is smaller and less populated than Milan, condition which is reflected traveled distance. Indeed the maximum distance reaches 28 km in Milan, 19 km in Turin. For both cities approximately 97% of rentals cover less than 10 km. In Turin, the trips are in median 300 meter shorter (3.4 km with respect to 3.7 km) than in Milan. In more details, in Turin it is possible to observe a steep slope at the end of the ECDF which is due to the trips from and to Turin Airport, located 15 km far from the city center. They represent 2.1% of rentals.

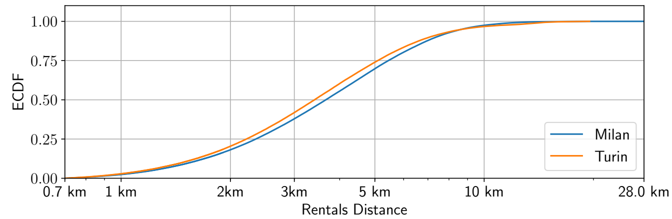
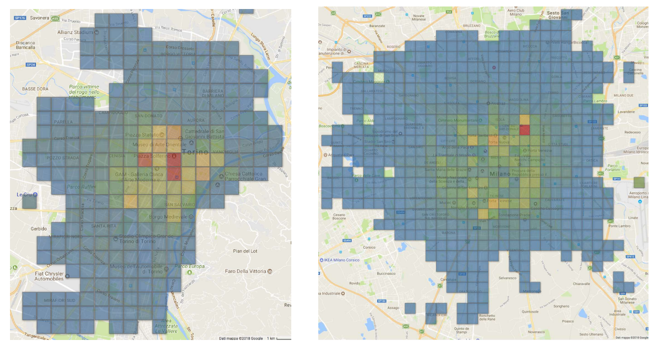


Fig. 1. Empirical Cumulative Distribution Function of rental distances as observed in current FFCS.



(a) (b)

Fig. 2. Heatmap showing frequency of parkings per zone in Turin (a) and Milan (b). The cells cover the operative area.

Next we analyze the parkings habits in the two cities. We divide the operative area with a grid where each cell side is 500 m and count how many parkings are observed in each cell. We consider as active only the zones having at least 1 parking in the data collection period. We observe 261 zones in Turin, and 549 in Milan. Fig. 2 shows them, with colors that reflect the number of parkings: the more the zone is red, the more frequently people parked inside it. In Turin, Fig.2(a), the zone having more parkings (47 per day on average) is in correspondence of the main train station. In general most of the parkings happen to be in the downtown area: here we observe lot of parkings but lasting short time. Fig. 2(b) shows the same trend in Milan. The parkings are concentrated in city center, which is bigger than in Turin. Here the most frequented area recorded about 100 parkings per day on average. It is interesting to notice the big peripheral area which extend for more than 10 km from the city center and where the number of parkings are few units. This is due to the fact that car2go imposes an extra car-drop fee in this area to discourage the users to park too far from the city center.

In a nutshell, people mostly use the FFCS to move to and in the downtown area, where lot of parkings of short duration (less than 1 hour) are recorded. Instead in the periphery there are fewer parkings, which last longer (up to days) – see [14], [15] for more details. Given this habit, we next

[2]car2goAPI, `https://www.car2go.com/api/tou.htm` service subject to approval by car2Go. Approval granted in Sept. 2016 and discontinued in Jan. 2018.

[3]`https://developers.google.com/maps/documentation/distance-matrix/`

[4]We filtered the dataset keeping only those rentals having a traveled distance greater than 700 m, since users may cancel the booking without moving the car and the car GPS sometimes gives inaccurate measure.

investigate different charging station placement policies and observe how they would perform assuming the same FFCS uses a EV fleet. For this, we use a trace driven approach that replays exactly the same trips as done by actual FFCS customers.

III. System Model and Simulator

Our goal is to study a FFCS system that uses EVs. We assume the customers' demand for the service is the same as the one observed in today FFCS that we recorded in our traces. For this, we use an event driven simulator that accurately replays the events recorded in each trace, using rental and parking events. We made the simulator available as open source to the community.[5]

The fleet is the mean of the daily vehicles observed in the trace. Cars are assumed to be Electric Smart ForTwo model, equipped with a battery of capacity $C = 17.6\,\text{kWh}$, and nominal consumption of $\lambda = 0.13\,\text{kWh/km}$.[6]

When a new rental starts, we assume that the customer selects the closest available car with the highest State of Charge (SOC). The simulator then changes the status of the car from available to rented. For each trip of $D\,\text{km}$, the simulator computes fraction of the battery energy consumption $\Delta C = \frac{\lambda \cdot D}{C}$. We keep λ constant because both cities do no present streets difference in altitude in the analyzed areas. When the rental ends, it moves the vehicles to the drop-off location, labels it back as available, and decreases accordingly the battery SOC by ΔC. If, at the end of the trip, the SOC is equal to or smaller than zero, an Infeasible Trip is recorded. The *percentage of Infeasible Trips* accounts for those rentals that would be infeasible because the battery SOC does not allow to cover the desired distance.

For charging, we assume each CS is equipped with 4 poles able to provide 2 kW/h for the time the car is plugged. Let Z be the number of zones in the city (see Fig. 2), and N the number of zones with a CS. We consider the next two mechanisms that regulates when and how to plug the car in a CS at the end of the rental:

- *Safety policy*: Whenever the SOC at the end of the rental is smaller than a threshold α, then the customer is obliged to return the car to the closest CS with at least an available pole, and plug it. In case the customer is forced to drive to a different zone from the one desired, we call this a *reroute* event, and we compute the new SOC of the car for the new route. If there are no available charging poles in the whole city, the customer returns the car in the desired arrival zone.
- *Altruistic behavior*: Whenever there is a free charging pole in the desired arrival zone, the customer voluntarily plugs the car with a probability p. This reflects users' willingness to sustain the system and to easily find a parking spot.

With these two mechanisms, the users could drop the car in a location which is different from the desired one, which has to be reached by walking. Given we assume squared zones of 500 m x 500 m, and place the CS at the center of a zone, the altruistic behavior would cause an average walking distance of about 150 m. Instead, the safety policy would reroute the customer for at least 500 m.

In the following section, we vary N from 1% to 30% of total zones Z in both cities, corresponding to a maximum of 72 and 135 zones in Turin and Milan, respectively. Given the maximum traveled distance observed in the trace is 28 km (Fig.1), we choose $\alpha = 0.25$ which corresponds to guarantee enough charge to support a single trip of at maximum 33 km. The number of poles per CS (equal to 4), the fleet size, the car model, and the supplied power for each plug (2 kW) are considered constant. Finally, we assume $p = 0.5$, i.e., we assume that in the half of the rentals, the customers are willing to plug the vehicle in case there is a free pole in their arrival zone.

Our goal is to find a smart placing of a small number of charging stations for zeroing the percentage of infeasible trips, minimizing at the same time the probability of being rerouted and the average walking distance after a reroute.

IV. Greedy Placement Approach

There are $\binom{N}{Z}$ possible placement solutions, which makes it prohibitive to find exhaustively the optimal solution. For this, we first propose greedy placement policies. Later, we refine the solution by running a local search meta-heuristic.

For greedy placement, we assign to each zone $z \in Z$ a likelihood l_z. Then we assign a CS to the top N zone, sorting them by decreasing likelihood. We consider three policies:

- *random placement - RND*: l_z is an independent and identical distributed random uniform variable, so that CSs result placed at random;
- *average parking time - Avg-Time*: l_z is the average parking duration in z as recorded in the trace, so that CSs are placed in zones where cars remain parked for the longest time.
- *Average number of parkings - Avg-Parking*: l_z is the average number of parkings per day recorded in z in the trace, so that CSs are placed in zones where cars are parked with the highest probability.

We compare now the performance in terms of infeasible trips for the three different CS greedy placement policies with different number of CSs. We consider both Turin and Milan. In both cases, we use the actual data of rentals recorded in the whole period to define the average parking duration and the number of parking per zone that we use then to guide the CS placement. For each resulting placement, we then simulate all rentals as recorded in the trace.

Fig. 3 shows the percentage of infeasible trips for increasing number of CS. Top x-axis reports the actual number N, while bottom x-axis reports the percentages of zone equipped with a CS to ease comparison. Consider Turin first - detailed in Fig. 3(a). As expected, when N has high values, the system is able to guarantee that all rentals can be successfully completed. That is, the CS system guarantees enough charging opportunities so that it is always possible

[5] `https://github.com/michelelt/sim3.0`
[6] https://www.smart.com/uk/en/index/smart-electric-drive.html

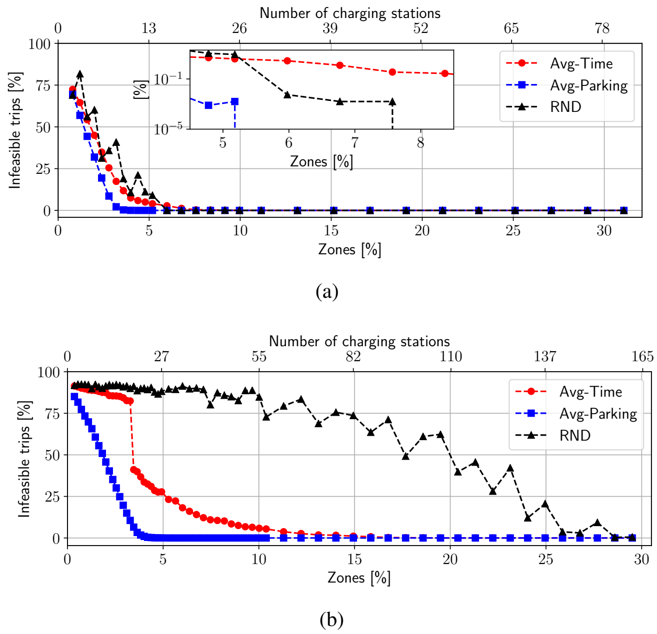


Fig. 3. Percentage of infeasible trips in Turin (a) and Milan (b).

to find a free charging pole when the battery runs below the minimum charing level α. Coupled with the Safety Policy, this guarantees the car battery never depletes.

Considering CS placement, the random placement (RND) has the worst performance. Interestingly, the average parking time (Avg-Time) performs worse than the average number of parking (Avg-Parking) policy. Indeed the zones having the highest average parking time are those in the peripheries. Cars that reach those areas, stay parked for long time and will be rented rarely because the demand in those zones is low. As such, the pole stays occupied for long time, with little benefits (the car battery being at full charge), and eventually precluding the charging chance to vehicles with low battery. The Avg-Parking policy perform the best: 5.3% of zones (14 actual CSs in total) guarantee to sustain all rentals in Turin – as detailed by the inset in Fig. 3(a).

Moving to Milan, Fig. 3(b), we observe the previous differences to be much more striking. Recall that Milan has a much larger operative area than Turin, with a bigger fleet. However, most of the rentals still occur in the downtown area, which is relatively small. Indeed, cars that reach the periphery stay parked for days before someone rents them. Placing a charging station in such areas would be almost useless since cars would stay plugged to a charging pole that results occupied for long time. This diminishes the actual charging capacity of the system, ultimately causing cars to run out of battery. A random placement performs then poorly, even when more than 25% of zones are equipped with CSs. The Avg-Time policy suffers from the same problem, with most CSs uselessly placed in the periphery. Again, placing charging station in areas with high parking probability guarantees the best results. Just $N = 33$ CSs (6.0% of zones) would guarantee all rentals in Milan.

In a nutshell, placing the CSs in an area where cars are frequently returned (and rented) allows to maximize the probability of finding a free charging pole. This guarantees enough top-up battery opportunities to recover the (small) amount of energy consumed by the (short) trips.

Algorithm 1: Local search algorithm

Input : oldSolution = InitialSolution
Output: newSolution
while *#Iterations* $\leq$ *MaxIter* **do**
  followDirection = False;
  CS=ChoseRandomCS(oldSolution);
  for *dir in* $\{$*north, east, south, west*$\}$ **do**
    newSolution = MoveCS(CS, dir);
    Simulate(newSolution);
    if *isBetter(newSolution, oldSolution)* **then**
      oldSolution = newSolution;
      followDirection = True;
      dirDescent=dir;
    end
  end
  if $followDirection == True$ **then**
    for *zone in* $\{$*from CS to border in dirDescent*$\}$ **do**
      newSolution = MoveCS(CS, zone);
      Simulate(newSolution);
      if *isBetter(newSolution, oldSolution)* **then**
        oldSolution = newSolution;
      else
        break;
      end
    end
  end
  #Iteration+=1;
end

V. Charging Station Placement Optimization

In this section we investigate how to minimize the customers' discomfort due to forcing them to plug the car to a nearby zone equipped by a CS. When the system forces a user to drive and park the car in a charging station that is not in her desired zone, it means that the customer has walked by at least 500 m to reach her destination. We compute the Average Walked Distance (AWD) among all these rerouted trips. Intuitively, the AWD decreases as the charging zone percentage increases, since the likelihood of finding a nearby charging zone increases.

Given all trips are guaranteed, the probability of being rerouted, and the Average Walked Distance are secondary performance indexes that we want to minimize. To this end, we design a local search approach. Given our optimization problem, previous placement policies fall in the category of greedy algorithms: they make a local choice that maximize a fitness function at each step (placing the next CS in the zone with the currently highest l_z), without ever tracing back. Local search algorithms are meta-heuristics that instead

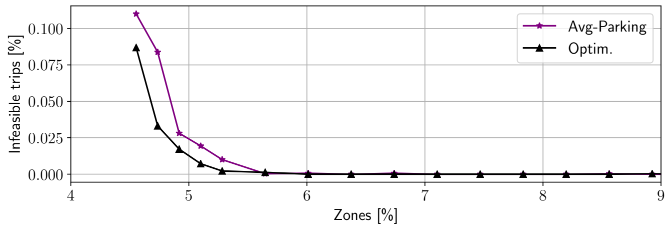


Fig. 4. Infeasible trips in Milan before and after the local search optimization.

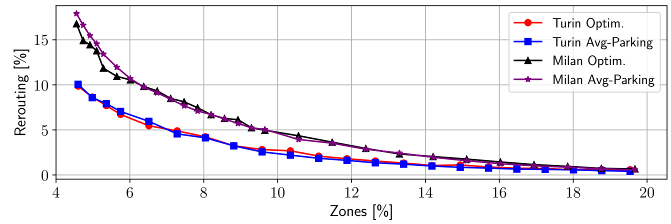


Fig. 5. Reroute probability before and after the local search optimization in Turin and Milan.

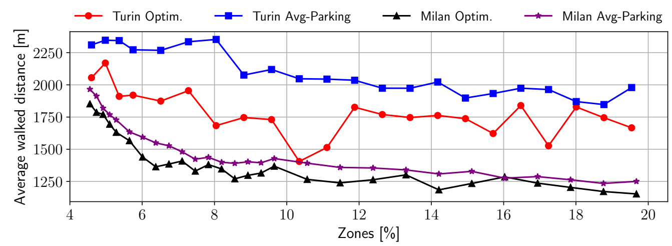


Fig. 6. Average Walked Distance before and after the local search optimization in Turin and Milan.

consider several solutions and select the best one found. The generation of solutions typically is performed by imposing small modifications to the current solution, generating a set of neighbor solutions among which to pick the best.

We consider as single objective to minimize the weighted sum of 1) the number of infeasible trips, 2) the AWD, and 3) the reroutes probability. The weights are given in order to give priority to them in the order here given (i.e., infeasible trips is the most important to minimize).

Our local search algorithm is similar to a hill climb with coordinate descent. Alg. 1 sketches the algorithms pseudo code. It receives in input the set of zones that the Avg-Parking greedy algorithm has equipped with CSs. Then, it randomly extracts one of CS zone, and builds 4 new solutions where the CS is moved to the closest zones (north, east south, and west), if not yet equipped with a CS. These neighbor solutions differ from the current one only by the placement of one CS. We run a trace driven simulation to evaluate if the new solution performs better than the old one. If there are improvements (according to the single objective), the algorithm accepts the new solution. Once identified the direction of improvement, this is followed by moving the CS along the same direction through a line search, until new best solutions are found. After the line search, the two steps are repeated by extracting another CS at random, and trying to move it. The algorithm stops after a maximum number of iterations.

To speed up the local search algorithm, both the two steps of the algorithms are actually parallelized in our code. Moreover, we let it evaluate the performance considering a subset of all events recorded in the trace (the first week of data). At the end the local search, we run a complete simulation considering the best found placement.

VI. Placement Optimization Results

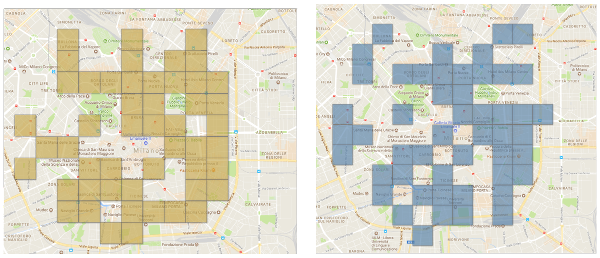
In the following, we run the local search optimization algorithm selecting the Avg-Parking as initial solution. We set the maximum number of iteration equal to 1 000.

We focus first to the percentage of infeasible trips. Fig. 4 shows results in Milan. Here the local search algorithm is able to reduce infeasible trips compared to the greedy solution. In particular it reduces to 0 infeasible trips at 5.6% of zones equipped with CSs. Recall that there are no infeasible trips for more than 6.0% of the zones, already for Avg-Parking policy. Instead, in Turin the optimized placement generate much smaller advantage in terms of infeasible trips and are therefore not shown for brevity.

The number of CSs affects the probability of a forced rerouting. Fig. 5 details this. First notice that Milan has higher rerouting probability than Turin, as a direct consequence of the bigger operative area, fleet and number of customers. Second, with more CSs, the probability to find a CS in the desired arrival zone increases, and the altruistic behavior guarantees a sufficient average SOC, which stays over the security threshold α. With 15% or more of CS zones the rerouting probability goes below 0.02 for both cases. Having less than 2% trips that are rerouted, the FFCS system could adopt some simple relocation policies with employees that move cars that need a recharge, or consider incentives to users to move these cars.

Being the less important part of the weighted objective, the optimization does not influence the rerouting probability much, and it is possible to observe a small improvement between 4.5% and 6.0% of zones in Milan only.

We now analyze the Average Walked Distance after a reroute event. Fig. 6 details AWD metric for both Turin and Milan, before and after the local search optimization. At first sight it is possible to see how the optimization algorithm reduces the AWD in both cities. In Milan, the average walked distance is decreasing monotonically with increasing percentages of zones with CSs. This is explained by the higher probability of finding a nearby free charging station when the number of charging stations increases. The difference between the local search and Avg-Parking placement is in average equal to about 100 m, with the best improvement of 190 m. Recall that in Milan customers have to pay an extra fee to drop the car in the periphery. This discourages them in using the FFCS service, so that most of parkings actually occur within the downtown area limit –



(a) (b)

Fig. 7. Zones equipped with CSs with the Avg-Parking greedy placement algorithm (a) and after the local search optimization (b) in Milan.

easily reachable within 1 250 m. To show this, we graphically illustrate the algorithm placement tendency. Fig. 7(a) depicts the CS placement by the Avg-Parking greedy policy. i.e., before the optimization. Fig. 7(b) depicts the final solution found by the local search algorithm. It moved some CS areas to cover also some residential zones in which we have observed still significant number of parkings but not as high as in the downtown areas. By moving some CS in those areas we reduce the AWD for those rentals that are rerouted back to downtown for a forced charge.

Moving to Turin, we observe a higher average walked distance in Fig. 6. This is due to the fact that people tend to use the FFCS service also to go to the periphery. Since most CSs are in the downtown, customers may be rerouted there, thus having to walk for more than 1 500 m. Having fewer reroutes and more homogeneous usage than in Milan, in Turin the local search algorithm offer a sizable gain. For instance, the average gain is about 300 m, corresponding about 21% improvement. This suggests that is could be possible to find even better solutions, e.g., by using more advanced meta-heuristic approaches. We leave this for future work.

VII. Conclusion and Future Work

In this work we studied gasoline FFCS system conversion into EVs, by leveraging real FFCS rental data. We based our study on a large dataset of actual rentals recorded in operative FFCS systems. We focused on two different cities as use cases.

We showed that a data driven smart placement of the CSs guarantees the system sustainability: equipping the top 6% of zones having the largest parking probability is enough to guarantee no car run out of battery. Next we developed a local search algorithm to further optimize the CS placement, minimizing the number of infeasible trips, and the discomfort for customers, that we evaluated as the probability of being rerouted to a far-away charging station, and the average walked distance they have to cover. Results showed that 15% of equipped zones guarantee limited impact on users' discomfort, with more than 98% of the trips that successfully end in the desired arrival area, and with less than 2% of trips rerouted to a charging station, which is found at a distance less than 1.8 km.

Given the complexity of the model which entails user habits and complicated relationships between charging need and availability, we believe data driven optimization is very promising to optimize the design of future FFCS systems with EVs. For this, we plan in the future to propose more advances optimization algorithms that move one step beyond the simple local search presented in this paper. We also plan to extend our analysis to more cities and to propose guidelines that can drive the policy makers to optimize the design of EV charging station placement in case no data is available.

References

[1] J. Firnkorn and M. Müller, "What will be the environmental effects of new free-floating car-sharing systems? the case of car2go in Ulm" *Ecological Economics*, vol. 70, no. 8, pp. 1519–1528, 2011.

[2] J. Firnkorn and M. Müller, "Free-floating electric carsharing-fleets in smart cities: The dawning of a post-private car era in urban environments?", *Environmental Science& Policy*, vol. 45, pp. 30–40, 2015.

[3] M. C. Falvo, D. Sbordone, S. Bayram, and M. Devetsikiotis, "EV charging stations and modes: International standards," in *2014 Symposium on Power Electronics, Electrical Drives, Automation and Motion*, 2014.

[4] A. Hess, F. Malandrino, M. B. Reinhardt, C. Casetti, K. A. Hummel, and J. M. Barceló-Ordinas, "Optimal deployment of charging stations for electric vehicular networks," in *Proceedings of the First Workshop on Urban Networking*, 2012, pp. 1–6.

[5] P. Phonrattanasak and N. Leeprechanon, "Optimal placement of ev fast charging stations considering the impact on electrical distribution and traffic condition," in *2014 Conference and Utility Exhibition on Green Energy for Sustainable Development (ICUE)*, 2014.

[6] T. Chen, K. Kockelman, and M. Khan, "The electric vehicle charging station location problem: A parking-based assignment method for seattle," in *92nd Transportation Research Board Meeting*, 2013.

[7] A. Y. S. Lam and Y. W. Leung and X. Chu, "Electric vehicle charging station placement," in *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2013.

[8] R. Bi, J. Xiao, D. Pelzer, D. Ciechanowicz, D. Eckhoff and A. Knoll, "A simulation-based heuristic for city-scale electric vehicle charging station placement," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, 2017, pp. 1-7, 2017.

[9] D. Jorge, G. Correia, C. Barnhart, "Testing the Validity of the MIP Approach for Locating Carsharing Stations in One-way Systems," *Procedia - Social and Behavioral Sciences*, vol. 54, pp 138–148, 2012.

[10] Z. Jiao, L. Ran, J. Chen, H. Meng, and C. Li, "Data-driven approach to operation and location considering range anxiety of one-way electric vehicles sharing system," in *8th International Conference on Applied Energy*, 2016.

[11] B. Jger, F. M. M. Agua and M. Lienkamp, "Agent-based simulation of a shared, autonomous and electric on-demand mobility solution," *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama*, pp. 250-255, 2017.

[12] E. Biondi, C. Boldrini and R. Bruno, "Optimal charging of electric vehicle fleets for a car sharing system with power sharing," *2016 IEEE International Energy Conference*, Leuven, pp. 1-6 , 2016.

[13] A. Ramachandran, A. Balakrishna, P. Kundzicz, A. Neti. "Predicting Electric Vehicle Charging Station Usage: Using Machine Learning to Estimate Individual Station Statistics from Physical Configurations of Charging Station Networks,", 2018.

[14] A. Ciociola, M. Cocca, D. Giordano, M. Mellia, A. Morichetta, A. Putina, and F. Salutari, "UMAP: Urban mobility analysis platform to harvest car sharing data," in *Proocedings of the IEEE SCI 2017: IEEE Smart City Innovations*, 2017.

[15] M. Cocca, D. Giordano, M. Mellia, L. Vassio, "Free Floating Car Sharing in Smart Cities: Data Driven System Dimensioning," in 4^{th} *IEEE International Conference On Smart Computing*, 2018.