POLITECNICO DI TORINO

Master’s Degree in Computer Engineering

Study of relocation strategies for electric car sharing system

---

Candidate:

Liu Xian

S261423

Supervisors:

Luca Vassio

Giordano Danilo

Alessandro Ciociola

April 2021

Declaration of Authorship

Liu Xian, declare that this thesis titled, ‘Study of relocation strategies for electric car sharing system’ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed my- self.

Signed: ____

Date: ____

Summary

Sharing Economy is developing rapidly in recent years due to its advantages in resource utilization and environmental protection. As for transportation, electric car sharing system has the potential ability in reducing air pollution and improving transportation efficiency. Free-floating car sharing system is extremely convenient for the users. It allows users drop off the vehicle at anywhere in the operation area instead of returning vehicle in specific place. However, it is easy to cause the asymmetry problem of vehicle supply and demand. Therefore, relocation operation which takes vehicles from oversupplied area to undersupplied area is necessary for the system to run in a sustainable state.

The goal of this Master Thesis is first, to understand if relocation is useful and profitable for car sharing system, then to investigate and compare different relocation strategies for Electric Free Floating Car Sharing (EFFCS) systems. Two kinds of relocation strategies are discussed here: reactive relocation and proactive layer relocation. Reactive relocation means that relocation operations are only triggered at the end of each trip when the battery level of vehicle is below a specific threshold. Proactive layer relocation refers to at the end of each hour, some vehicles will be relocated in order to meet the user demand for the next hour. The number of relocated vehicles depends on the number of system employee which performed the relocation operations. For each relocation strategy, system performance and economic performance are tested and evaluated. Here, system performance refers to metrics like the faction satisfaction of booking request and number of unsatisfied booking request which no available vehicles nearby. Besides, economic performance refers to the relocation related cost including hiring operational workers and extra energy used for relocation, revenues and profit rate for the whole EFFCS system so on.

For this purpose, I adopt and extend an existing date-driven, discrete-event simulator written in Python. I use the dataset which comes from actual rentals in the city of Turin performed by a famous car sharing program car2go. I conduct the case study about the city of Turin under different configurations. Results show that relocation operation has a

positive impact in improving system performance of satisfying more booking request for using vehicles. The faction of satisfied booking requests increases about 5% to 10% compared with no proactive relocation scenario in our experiments. As for choosing charging relocation area, choose the closest area with available charging poles is regarded to be the best solution. Besides, doing relocation operation in a given time frequency such as hourly execution seems meaningful. However, a tradeoff between better performance of system and extra cost needed to be considered. For instance, in our cases, hiring workers to do relocation operations definitely improves system performance. The extra revenues brought from improving performance is around 10,000 €. However, it also leads to more system cost both for paying for workers and relocation the vehicles for about 45,000 €.

My work about discussing different relocation strategy is useful in solving the problem about the unbalance between user demand and actual distribution of vehicles and improve the system performance. What’s more, analysis about the economic performance in relocation is meaningful when considering real world situation. As for further work, research can be expanded to other big cities such as Milan. Besides, more complex relocation strategy such as using machine learning model to make whole operation process adaptive to the real traffic situation could be considered and tested.

Acknowledgements

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Sharing Economy

The sharing economy refers to the sum of economic activities that use modern information technologies such as the Internet, with the sharing of use rights as the main feature, integrate massive and decentralized resources, and meet diverse needs. It is developing rapidly over the world because of saving energy and resource. Sharing economy will become the most important force in the social service industry. In the fields of accommodation, transportation, education services, life services and tourism, excellent sharing economy companies are constantly emerging.

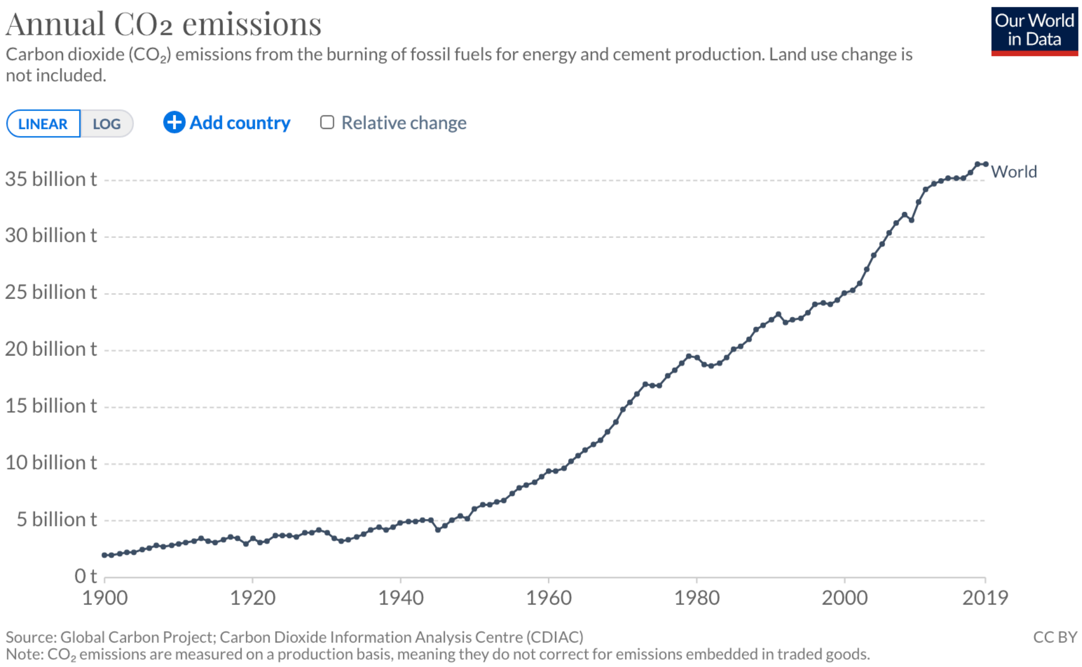


Figure 1.2.1 Annual Carbon Dioxide Emissions during Last Century[1]

1.2 Car Sharing

In terms of transportation, car sharing is becoming more and more popular in the last decades due to its excellent performance in reducing air pollution and fuel consumption, releasing traffic jam. Figure 1.2.1[1] shows the annual carbon dioxide emissions during the last century. We can see that each action of reducing carbon emission is very necessary. Car sharing is a model of car rental where people rent cars for short periods of time, often by the hour. It differs from traditional car rental in that the owners of the cars are often private individuals themselves, and the car sharing facilitator is generally distinct from the car owner. Car sharing is part of a larger trend of shared mobility. Figure 1.2.1[2] shows that the number of users in car sharing is increasing stably. Most of this form of transportation has been taking place in the cities in Europe, North America, Japan and Singapore[3]. There are two main branches in car sharing system: the station based car sharing system, in which the user picks and drops the car in the given parking spots and the Free Floating Car Sharing System(FFCS)[4], in which the user picks up and drops the car anywhere when he starts or ends the trip in the operational area.

The latter solution has given more flexibility to users however it leads to spatiotemporal demand asymmetries. Leaving the system on its own without any intervention saves money cost and human resource, but it has serious results. Vehicles are easily get stuck in areas with low demand causing a loss of money and low customer satisfaction. Things get worse for electric vehicles. When the vehicles are in the state of low battery, they are no longer usable for future period thus make the system corrupted. What relocation do is balancing supply and demand and charging low-battery cars, thus make the whole system run in a virtuous circle. To offer an appropriate level of service in areas with high demand, the operator has to move the vehicles from oversupplied area to undersupplied area. Those transfers are often executed during the day to optimally supply the high demand during peak hours. Moreover, as for electric vehicles(EV), which is more environmental friendly and widely used in public car sharing system, the worker has to relocate cars to the charging hub when they are going to run out of battery in order to make it use in a sustainable way.

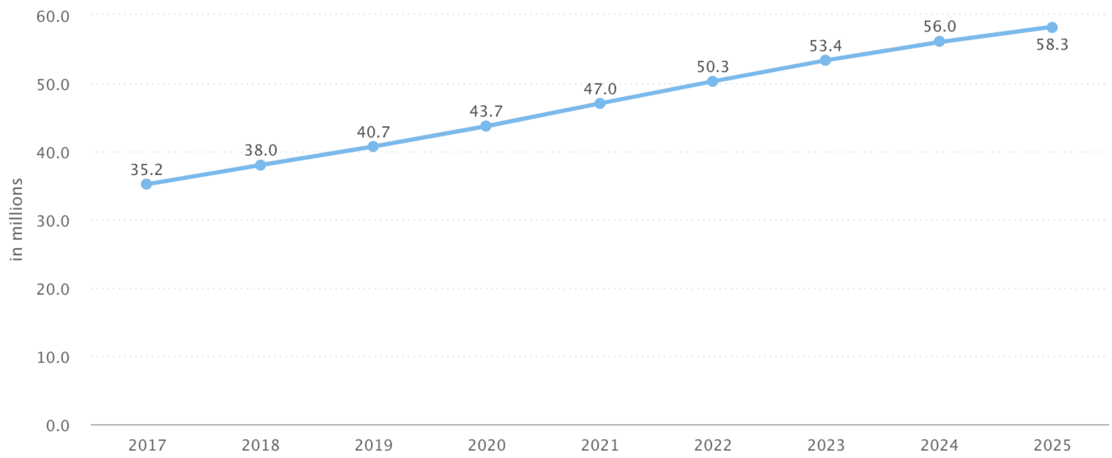


Figure 1.2.1 Number of Car Sharing Users

1.3 Thesis Presentation and Motivation

In this thesis, I will study different relocation strategies for electric vehicles. Relocation operation is very useful to fill the gap between demand and supply in vehicles. Moreover, charging infrastructure is also very necessary for making the electric vehicles run in a long time. In order to improve the usage of charging infrastructure, charging relocation strategy also have to be considered carefully. Futhermore, choosing charging poles in the most efficient way will also improve the whole system performance. I use a simulator to simulate real traffic situation in the city of Turin, Italy. I generate thousands of real FFCS trips in given time period. Firstly, by implementing three different kinds of charging relocation strategies that choose charging poles at the end of each trip, I observe their performance and study the additional cost which relocation operation bring. Post charging relocation are also called reactive relocation in the thesis.

Next, I consider **proactive** relocation. In that way, relocation is not only happened at the end of each trip, but can be scheduled and operated in a given time frequency. I use two different approaches to catch the demand spatial variability. We propose an hourly triggered relocation strategy. We relocate the cars to some zones which are confronted with the demand model at the end of each hour. By changing the fleet size and the number of relocation workers, we analyze many Key Performance Indexes such as satisfaction fraction of booking trip, relocation cost and system revenue so as to consider the availability of this strategy.

The main questions that we try to answer are:

- Can I add more configurations to the existing simulator, able to implement different kinds of both reactive and proactive relocation strategies?

- Can I fairly compare system performance of different relocation strategies?

- Can I fairly compare financial performance of different relocation strategies?

More specifically, the research questions I pose to analyze are the following:

- How do the different post charging relocation strategies influences service quality and operational cost in the given city Turin?

- How do proactive relocation strategy influences service quality and operational cost comparing with no relocation strategy scenario in the given city Turin?

- How do system parameters such as fleet size, the number of relocation workers and charging poles density impact service quality and operational cost in the given city Turin?

Our results show that in the reactive model, taking the low battery car to the nearest available charging poles has the most efficient performance. Besides, proactive relocation strategy make the whole system maximize the satisfied demand by increasing relatively acceptable additional relocation cost.

The thesis is organized as follows: In Chapter II I propose more detailed introduction about previous work about relocation in free floating car sharing system and the simulator I use for the whole thesis. What's more, I also review existing scientific literature that talks about the simulator modelling topics. Then comes to Chapter III that describes the simulator and dataset that used for the experiment in detail. Besides, I introduce both reactive and proactive relocation strategies. I present the results of a simulation campaign conducted for the city of Turin in Chapter IV. In the end, conclusion and future expectation are proposed in Chapter V.

Chapter 2

Background

In this section, I will introduce why simulation is widely used for modelling and analysis. Then, I will introduce the python project and the data that we used for the whole thesis experiment. Other related tools have also been mentioned in a general view. Besides, the definition of specific metrics is listed and shown for analyzing the experiments results.

2.1 Simulation Modelling Background

Simulation modelling solves real-world problems safely and efficiently. It provides an important method of analysis which is easily verified, communicated, and understood. Across industries and disciplines, simulation provides valuable solutions by giving clear insights into complex systems. Simulation enables experimentation on a valid digital representation of a system. Unlike physical, such as making a scale copy of a building, simulation is computer based and uses algorithms and equations. Simulation software provides a dynamic environment for the analysis of computer models while they are running. The uses of simulation in business are varied and it is often utilized when conducting experiments on a real system is impossible or impractical, often because of cost or time. Here are the advantages of the simulation[5]:

- **Risk-free environment:** Simulation provides a safe way to test and explore different "what-if" scenarios. The effect of changing staffing levels in a plant may be seen without putting production at risk. Make the right decision before making real-world changes.

- **Save money and time:** Virtual experiments with simulation models are less expensive and take less time than experiments with real assets. Marketing campaigns can be tested without alerting the competition or unnecessarily spending money.

- **Visualization:** Simulation models can be animated in 2D/3D, allowing concepts and ideas to be more easily verified, communicated, and understood. Analysts and engineers gain trust in a model by seeing it in action and can clearly demonstrate findings to management.

- **Insight into dynamics**: Unlike spreadsheet or solver-based analytic, simulation allows the observation of system behavior over time, at any level of detail. For example, checking warehouse storage space utilization on any given date.

These features are very suitable for transportation. Cause transportation is a dynamic process, the state of the whole system changes all the time and it’s hard to get or collect data from the real world due to time and scale limitation. However, each coin has two sides. It also has some disadvantages:

- It can be expensive to measure how one thing affects another, to take the initial measurements and to create the model itself (such as aerodynamic wind tunnels).
- To simulate something, a thorough understanding is needed and an awareness of all the factors involved. Without this, a simulation cannot be created.

The process from scratch is difficult. Because the worker has to design the whole system, the structure and any part that involve in the EFFCS. They also have to collect real world data that configure into the simulation to make it more convinced. What’s more, it is hard both for understanding the scientific tool itself and the scenarios that needed to be implemented. It requires both coding ability and research ability at a relatively high level.

2.2 ICT Background

Python is chosen as the programming language because it is easy to write and understand. Due to its corporate sponsorship and big supportive community of python, python has excellent libraries that you can use to select and save your time and effort on the initial cycle of development.

- As for simulation part, **Simpy[6]** is used which is a process-based discrete-event simulation framework based on standard Python. Its event dispatcher is based on Python's generators and can also be used for asynchronous networking or to implement multi-agent systems.

- **Pandas[7]** is mainly used for data processing. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.

- Pandas is built on top of **NumPy[8]** library, which is the fundamental pack- age for scientific computing with Python, providing among other things a powerful and efficient N-dimensional array object.

- **Tableau[9]** is used for create result figure. It is a powerful and fastest growing data visualization tool. It helps in simplifying raw data into the very easily understandable format. Data analysis is very fast and clear with Tableau and the visualizations created are in the form of dashboards and worksheets.

- Other useful tools to work with Python include the IDEs **PyCharm[10]**, where I pull and modify code and do single run on local PC and the **Jupyter[11]** Notebook cluster, you can run multiple runs that consume more memory and time on the cluster by setting the appropriate port number.

2.3 Relocation Background

In this section, I focus mainly on reviewing the literature on operational aspects of the relocation problem in vehicle-sharing systems to highlight my contributions. I refer to Laporte[12] for a more comprehensive review of other relevant operational problems.

In one-way car sharing systems, relocation can be carried out either through operator intervention, e.g., using relocation personnel[13][14][15] and using a trip choice mechanism[16] or through customers by controlling their actions, e.g., through incentives[17]. The focus of relocation is to achieve certain desirable inventory levels either through manual rebalancing using trucks[18] or through incentive mechanisms designed to influence customer behavior[19][20].

In an initial conceptual paper, Weikl and Bogenberger[22] present and evaluate several user-based and operator-based relocation strategies for FFCS systems. In a subsequent paper, Weikl and Bogenberger propose a practice ready six step relocation model for a mixed FFCS system with traditional and electric vehicles. Based on historical data, the area is categorized into macro zones and an optimization model is used to achieve desired macro level relocation. Rule based methods are used for making intra zone micro-level relocation and refueling/recharging decisions. A similar model for demand-based relocation in FFCS systems is presented by Schulte and Voß and Herrmann[23]. Caggiani[24] propose dynamic clustering method to identify the size and number of flexible zones in which to perform repositioning operations. He[25] studied robust repositioning strategies in dynamic environments.

Closely related to my work, Alessandro[26] considers joint decision making for EV relocation. When the charging operations are needed, electric vehicles are relocated to the nearest available charging station instead of the nearest charging station. Although station blocking will not happen, more relocation distance will be. Different charging relocation strategies are compared in concern of the system and economic performance. One possible kind of approach is to schedule relocations at fixed times (e.g., at night), to rebalance the system[27]. In this thesis, I proposed the relocation strategies that relocation operations are happened in a given time frequency.

2.4 Related Work

2.4.1 eC2S

Alessandro Ciociola builds simulation named eC2S of the Electric Free Floating Car Sharing (EFFCS) systems to observe real problems in terms of spatiotemporal demand asymmetries. A city operating area is divided into square zones of dimension 500m*500m.Each zone is assigned an identity number. Each trip is marked from an origin zone to the destination zone. Every trip distance in the simulation in computed from a zone centroid to another zone centroid. During the trip, vehicle is not available and moves from one zone to another, its battery level corresponding decreases with the distance. After the trip ends, vehicle becomes free cars in destination zones. They defined valid zones as zones that become destination of at least one trip for the whole system duration. Charging zone is the zone with charging poles. The number of charging zones is given by a specific percentage of the whole valid zones. Charging poles are located in the area of the city with the highest probability of being destination zones. Figure 2.1 shows a kind of possible the charging zone's distribution in the city of Turin by setting the ratio between charging poles and vehicles to 0.02. We can see that trips are more likely to happened in the central of the city. Each zone has N poles that allows N vehicles charging at the same time. At the end of a rental, if the battery level of a vehicle is below a threshold β and needs to be charged to threshold α, we can have the highest probability that it does not need to be relocated and get charged. If there are no charging hubs in its own zone, it is relocated to the closest free charging point. This means if there are charging poles in zone A and B where zone A is closer to the low battery vehicle compared with zone B. The vehicle will be relocated to zone A unless all of the charging poles are been in use in zone A while poles in zone B are available. If there is no free charging pole anywhere in the city, the car queues at the nearest charging pole. When the charging operation ends, it's free to choose if the vehicle will relocate to its origin zone or just leave in the charging zone.

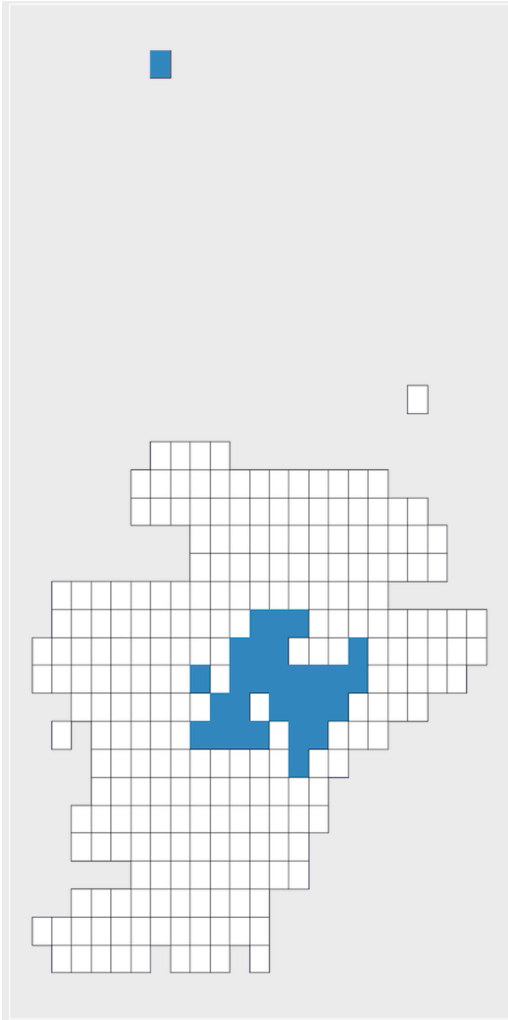


Figure 2.1 possible Charging Zone Location in Turin

2.4.2 Dataset

The input data comes from actual rentals performed by car2go users in the city of Turin [5]. Each observed rental has precise geo-spatial coordinates for origin and destination, and accurate timestamps. Data are stored in csv format file. Each line corresponds to a trip. Each booking has the following information described in Table 1:

Parameter	Description	Example values
plate	plate of the vehicle that performed the trip	245/FF124SJ
start_time	starting time of the booking	2017-10-01 02:00:37+02:00
end_time	ending time of the booking (unix time)	2017-10-01 02:37:18+02:00
start_longitude	longitude of the initial position	7.628310000000001

start_latitude	latitude of the initial position	45.0507
end_latitude	latitude of the initial position	45.07638
euclidean_distance	distance travelled in meters (line connecting initial and final position)	4413
duration	time of the bookings in second (final_time minus init_time)	2201.0
start_year	start year at the beginning of the booking	2017
end_year	end year at the beginning of the booking	2017
year	year of the booking	2017
start_month	start month at the beginning of the booking	10
end_month	end month at the beginning of the booking	10
month		10
start_hour	start hour at the beginning of the booking	2
end_hour	end year at the beginning of the booking	2
start_weekday	start day at the beginning of the booking	Sun
end_weekday	end day at the beginning of the booking	Sun
start_daytype	If start day is Sunday or Saturday, the start_daytype is weekend, otherwise is weekday.	weekend
end_daytype	If end day is Sunday or Saturday, the end_daytype is weekend, otherwise is weekday.	weekend

TABLE 2.1

DATA FORMAT OF INPUT BOOKING REQUESTS

2.4.3 Related Metrics

The intervention of the system is always accompanied by costs. The whole system should be profitable for the provider, so costs and additional benefits have to be compared. In order to monitor the system performance and economic effect, the following metrics are listed and considered:

Fraction Satisfied: Percentage of satisfied user requests. It is calculated as number of satisfied user requests divide total requests. When a new rental request is generated, the system looks for an available car with enough battery in the origin zone, or in its 1-hop neighbor zone. If the user can't find an available car close to the origin of the request, or did not find a car with enough energy to perform the desired trip. The request is regarded as unsatisfied. This metric shows the system's ability in distributing the vehicles with the changeable demand and quality of the service in terms of cars availability for users' requests.

Relocation Cost: it gives an indication of the cost of charging process in terms of time of money needed to drive cars to charge. When a car needs to be charged and no user is willing to help, the system has to physically move it to the closest charging point. Relocation is an extra activity that executes by system employed workers. They have to be paid given hourly salary. The total relocation working cost can be calculated as:

relocation worker cost = N workers * duration days * 24 * hourly worker cost

charging cost: There are charging poles infrastructure cost and charging energy cost for electric vehicles. Poles cost is defined as:

poles cost = (N charging poles) * duration months * (hardware cost / pole useful life + pole labor cost / pole useful life + pole annual maintenance cost + pole annual tax) / 12

cps zones percentage: Describe the charging zones density. For example, if it equals to 0.02, it means that the number of charging zones is 2% of the number of valid zones.

n poles n vehicles factor: Describe the charging poles density. The bigger the number is, the more charging poles the whole system has. For example, if it equals to 0.02, it means that the number of charging poles is two percentage of the number of total system vehicles.

Energy cost is described by total kWh charging energy been used and energy price 0.19 euro/kWh. It can be calculated as:

energy cost = tot charging energy * kWh price

Revenue: Revenue is the system income that users pay for renting the vehicles. Price per minute for using the vehicle is 0.2 euro. Thus

$$\textbf{\textit{revenue = tot mobility duration * price per minute}}$$

Profit: Profit describes the net income of the system. It is defined as

$$\textbf{\textit{profit = (revenue – total cost) / total cost}}$$

In order to improve the whole system performance, different relocation strategies are implemented. They are divided in two categories: *reactive* and *proactive* approaches. Reactive relocation refers to the relocation operations happen only at the end of the trip. However, proactive relocation operations happen at a given time frequency. With reactive strategies, we decide if and how to relocate only when a certain condition is triggered during the simulation. With proactive strategies, we decide how and when to relocate with a schedule. Implementation details will be discussed in Chapter III.

Chapter 3

Methodology and Tools

In this section, the structure of the simulator will be introduced first. Then, the extra metrics that used for show the experiment results will be listed. What's more, the definition and explanation of each relocation strategy will be mentioned.

3.1 Simulator

eC2S is a data-driven, discrete-event simulation software for EFFCS (Electric Free Floating Car Sharing) system. It is able to model car sharing demand from data coming from real car sharing systems and run parametric simulation campaigns, providing also analysis and visualisation tools useful to compare different charging scenarios and fleet management strategies. It is written in Python and contains the following folders:

- **Demand_Modelling**: contains code which implement the demand model for by configuring city, duration, simulation technique and so on.

- **Data:** contains raw data including booking trips, city geometric and charging stations in forms of pickle and csv.

- **Supply_Modelling**: contains code which implement the supply model for meeting the demand by configuring number of vehicle, charging poles placement policy and so on.

- **SimulationInput**: contains classes implementing the logic for managing the input of the simulation. This includes many different running configurations, statistical models and shared data structures.

- **Simulation**: contains classes implementing the simulation logic. This includes the abstraction for user requests generation, mobility, charging and relocation strategies. It is the core simulation module.

- **SimulationOutput**: contains classes for statistics collection, aggregation and visualisation.

- **SingleRun**: contains functions for running a single simulation with a specified configuration.

- **MultipleRun**: contains functions for running a set of simulations following a grid of configuration parameters. It is possible to run a set of simulations on many cores in parallel.

- **Figures**: contains charts produced in the simulation output phase organized by simulation city, scenario and configuration name. It includes booking requests and charging boxplot, event profiles, vehicle feature profile and boxplot etc.

- **Results**: contains simulation results in form of pickles and csv organised by simulation city, scenario and configuration name. It includes detailed booking requests, all system configuration and performance metrics, history status of stations, vehicles and zones. More detailed metrics about the result file are in appendix.

Each booking request has several features. Origin id is the zone which the trip start location belongs to. Destination id is the zone location which the end location belongs to. Driving distance is computed as Euclidean distance between two zones multiplied by a correction factor representing the average driving distance. Moreover, date, hour and start time and end time describe its time attribute.

3.2 Extra Metrics

In order to consider system performance and economic cost in different relocation strategies, more detailed metrics should consider. As for system performance, I consider relocation outward distance, which represents the sum of all relocation distance of the system.

As for economic cost, there will be charging poles for charging relocation strategies. Therefore, extra cost for preparation for the charging zone are needed:

zone make-ready cost = make-ready cost per zone * n charging zones /pole useful life / 12 * duration month

Thus, the total charging infrastructure cost is:

charging infrastructure cost = poles cost + zone make-ready cost

Besides, cars cost is defined as:

cars cost = n vehicles * vehicles*annual leasing cost/ 12* duration month

scenario cost = cars cost + charging infrastructure cost

Washing cost is defined as:

washing cost = disinfection cost *n charges + washing cost * n bookings / 100

The cost for the simulation process is defined as:

sim cost = relocation worker cost + energy cost + washing cost

The total cost for the whole system is sum of simulation cost and scenario cost:

total cost = scenario cost + sim cost

3.3 Reactive Relocation Strategy

E3f2s has to ensure that vehicles spread in the city have enough energy to serve the users. After the end of each trip, the simulator should check each vehicle’s battery level, if it is below a specific threshold, the vehicle needs to be charge before it serves the next trip. Charging poles are not located in every zone of the city. So here comes to the question about choosing which zones with charging poles the low battery vehicle will relocate to. Here I propose three different kinds of strategies in choosing charging zones:

- **Closest_free:** choose the nearest zone with available charging poles to charge. After relocation operation, the vehicle can be charged immediately. First I will sort the zones by the distance between the relocation starting zone and all the charging zone. Then I will check if the nearest charging zone has free poles, if it has then I choose this zone as charging relocation zone otherwise move to the second nearest charging zone and check and so on.

```
# find the nearest available station with charging poles
               for zone in zones_by_distance.index:
# check if the nearest charging zone is available
                   if self.charging_stations_dict[zone].charging_station.count < self.charging_stations_dict[
                       zone].charging_station.capacity:
                       free_pole_flag = 1
                       charging_zone_id = zone
# calculate the energy needed for relocation
                       cr_soc_delta = self.get_cr_soc_delta(
                           booking_request["destination_id"],
                           charging_zone_id,
                           vehicle
                       )
# remaining energy is not enough for relocation, mark the zone unavailable
                        if cr_soc_delta > booking_request["end_soc"]:
                           free_pole_flag = 0
# choosing the zone as the relocation charging zone
                       else:
                           charging_zone_id = charging_zone_id
                           break
```

- **Random:** randomly pick one zone with charging poles to charge. After relocation operation, the vehicle can be charged immediately. I just pick one random charging zone to check if it has free charging poles. If it has, I choose this zone as charging relocation zone otherwise I pick another zone randomly and check again and so on.

```
# find a random station to charge
                while True:
# randomly pick one zone
                    random_zone_id = random.choice(zones_by_distance.index)
# remove the zone from picked zone list
                    zones_by_distance.pop(random_zone_id)
# check if the charging zone is available
                    if self.charging_stations_dict[random_zone_id].charging_stati
on.count < self.charging_stations_dict[
                        random_zone_id].charging_station.capacity:
                        free_pole_flag = 1
                        charging_zone_id = random_zone_id
# calculate the energy needed for relocation
                        cr_soc_delta = self.get_cr_soc_delta(booking_request["des
tination_id"], charging_zone_id, vehicle)
# remaining energy is not enough for relocation, mark the zone unavailable
                        if cr_soc_delta > booking_request["end_soc"]:
                            free_pole_flag = 0
                        else:
                            charging_zone_id = charging_zone_id
# if the charging zone is available or all the zones have been picked,finish the
picking loop procedure
                    if free_pole_flag == 1 or zones_by_distance.empty :
                        break
```

- **Closest_queueing:** choose the nearest zone with charging poles to charge. After relocation operation, the vehicle should wait in the queue until other charging operation ends and charging pole become available again. First I will sort the zones by the distance between the relocation starting zone and all the charging zone. Then I will set the nearest charging zone as charging relocation zone.

```
# find a nearest station to charge
# sort the other charging zones by distance with the giving vehicle zone id
zones_by_distance = self.simInput.supply_model.zones_cp_distances.loc[
                    int(booking_request["destination_id"])
                ].sort_values()
                free_pole_flag = 0
                for zone in zones_by_distance.index:
                    free_pole_flag = 1
                    charging_zone_id = zone
# calculate the energy needed for relocation
                    cr_soc_delta = self.get_cr_soc_delta(
                        booking_request["destination_id"], charging_zone_id, self
.vehicles_list[vehicle]
                    )
# remaining energy is not enough for relocation, mark the zone unavailable
                    if cr_soc_delta > booking_request["end_soc"]:
                        free_pole_flag = 0
```

```
# choosing the zone as the relocation charging zone, finish the picking loop procedure
                    else:
                        charging_zone_id = charging_zone_id
                        break
```

Relocation operation are described as a dictionary:

```
charge_dict = {
            "charge": charge,
            "resource": resource,
            "vehicle": vehicle,
            "operator": operator,
            "zone_id": charging_zone_id,
            "timeout_outward": timeout_outward,
            "timeout_return": timeout_return,
            "cr_soc_delta": cr_soc_delta,
            "charging_outward_distance": charging_outward_distance
        }
```

Zone id is the charging zone id I choose for relocation. Timeout outward describes the time duration of doing the relocation operation. If relocation flag equals to true, the vehicle has to return back to its origin zone before changing, timeout return will be equal to timeout outward otherwise 0. Cr_coc_delta is the battery consumption of doing the relocation operation and charging outward distance is the distance between relocation starting zone and relocation ending zone.

The relocation cost and system performance will differ within the above three strategies, cause the relocation path and distance is totally different. The closest_*queueing strategy definitely has the minimum relocation distance however closest*_free strategy saves time for waiting in queue, thus will make vehicle become available again more quickly. Random strategy is hard to tell its pros and cons. The overall performance will depend on their experiment results.

3.4 Proactive Layer Relocation Strategy

The above part is reactive relocation, which means that the system decides whether to do relocation operations at the end of each trip. This kind of relocation strategy will be very time consuming and lack of intelligence. In order to avoid this problem, I have proposed a proactive layer relocation strategy. This kind of relocation is not happened by trips but on a regular time duration. We set the relocation operation execute at the end of each hour. When trips end in the scheduled time slot, we trigger relocation. In other word, proactive relocation happens 24 times a day. Given a number of relocation workers N, each worker will remove a car from one choosing zone to another destination zone. So, there will be N electric vehicles changing their position separately at the end of each hour. Which strategies we will use to choose relocation starting zones and ending zones?

3.4.1. Zone Selection Techniques

- **Aggregation:** By choosing starting(ending) zone, we choose cars in the most(least) aggregated and crowded zones. We sorted all the valid zones by the number of vehicles in its zone by descending order. Then we selected top N zones as our relocation starting(ending) zone.

- **KDE[28][29]:** By choosing the destination zones, we first to make sure that the ending zone should be the valid zones. Then the end zones list should not overlap with the starting zones. Otherwise the relocation work is useless, one relocation moves vehicle from zone A to zone B, the other moves it back from zone B to A. Then we use the next hour's **KDE** distribution to generate ending zone for N times in order to get N ending zones. In statistics, kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable. Kernel density estimation is a fundamental data smoothing problem where inferences about the population are made, based on a finite data sample. KDE is largely used as a general tool in spatial analysis. For example, parameters of traffic accident prediction models have been estimated mainly based not on KDE but on raw count data in Japan. Yu et al. (2014) recently reported that KDE outperformed other hazardous road segment identification methods. We generalize over space using KDE. For this purpose, we leverage Kernel Density estimator from *scikit-learn*, with a Gaussian kernel and a 2 x 2 identity matrix as bandwidth. First, we divide the city into

500m x 500m squares, generating a matrix of city zones. Then, we fit a four-dimensional KDE on origin-destination zone couples, where each zone is represented by its two indexes inside the matrix. We do this fitting 48 times, one for each time slot for weekdays and for weekends. Thus, we have a spatial representation of mobility patterns between different zones during each hour of the day.

- **Delta[30]:** It is the most complex technique that we propose. It uses the fraction of current available vehicles in each zone, as a proxy for current state S. It uses approximated counts of origins (O) and destinations counts (D), to calculate a prediction of the the total outcoming or incoming flow of vehicles in a zone at a given hour of a given type of day (i.e., weekday or weekend). O and D are derived directly from the trace, computing the average out-flow and in-flow of vehicles from a zone at a given hour of a given day type, and they type for zone z is them computed as the difference between O(d,i;z) and D(d,i;z). A positive flow means that we predict that the number of vehicles depart from a zone at a given hour, will be higher than the number of vehicles arrive. The strategy selects as starting (ending) zone the one with the lowest (highest) *delta* (Δ), which is the difference between predicted flow for next hour(s) and current state, for a given time t (in hours) and a given zone z, as can be seen in (3.1). Thus, higher delta means that a shortage of vehicles is more probable. For example, it can mean that we predict high positive flow and we know from S that there are not enough vehicles.

$$\Delta(d,t;z) = \frac{\sum_{i=t+1}^{t+W} O(d,i;z) - D(d,i;z)}{W} - S(d,t;z) \qquad (3.1)$$

This is the only strategy for which we can specify a window width W, to be able to take into consideration more than just one hour in the next future. This is also the only strategy that allows us to relocate more than one vehicles at a time, with a number of relocated vehicles that is given by Δ itself.

Chapter 4

Case Study of Turin

In this chapter, various relocation strategies are implemented and tested in the simulation by given different configurations. After getting the digital results, I draw different plots in Tableau in order to visualize the result and analyze them.

4.1. Reactive Relocation Strategy Results

In order to compare system performance and economic performance of three different post charging relocation strategies. I run the simulator in the city of Turin from October to November of 2017. The parameters grid for this set of simulations is in table 4.1.

Parameter	Description	Values
α	the charging threshold	20
β	the charging upper bound	100
cps zones percentage	charging zones density with regards to total zones	0.2
N poles/N vehicles factor	number of charging poles with regards to vehicles	(0.01,0.19), step 0.01
n vehicles sim	total vehicles run in the simulation	414
relocation	the flag to show if the car after charging will bring to its origin zone before charging	False
relocation worker	The number of charging relocation workers	1000
charging relocation strategy	charging relocation strategy used for experiments	["closest_free", "random", "closest_queueing"]
annual leasing cost	annual leasing cost per vehicle	4000
disinfection cost	disinfection per charging operation	15

washing cost	washing cost per vehicle	8
pole labor cost	labor cost for building the pole	2200
pole hardware cost	hardware cost for building the pole	1700
pole useful life	how many years a pole can use	10
pole annual maintenance cost	pole maintenance cost per year	5000
cosap annual tax	pole tax fee per year	355
zone make-ready cost	building cost per zone	1500
kWh cost	energy price per kWh	0.19
price per minute	price the when a user use the vehicle per minute	0.2
request rate factor	the ratio between the number of the real booking request and input booking request	1

TABLE 4.1

PARAMETER FOR CHARGING RELOCATION EXPERIMENTS

4.1.1. System Performance

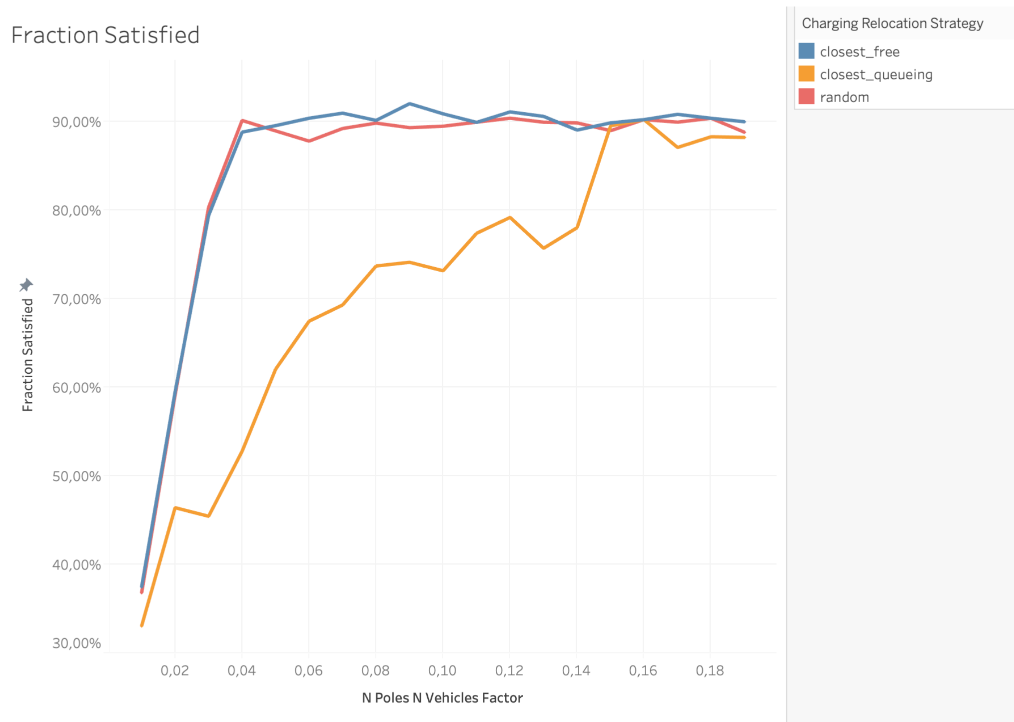


Fig. 4.1.1: Fraction Satisfied with respect to N poles/N vehicles Factor. Curves show the performance with different charging relocation strategies.

Figure 4.1.1 shows the fraction satisfied varying different charging poles and relocation strategies. We can see that the more charging poles the system has, the more booking requests that can be satisfied. Because when the trip ends, if the system has as many poles as possible, the vehicle is easily to find charging poles and can reach to the charging zones very quickly, at the same time it consumes little battery for relocation, thus charging duration also decreases. As for relocation strategy, closest_free and random has the overall better performance compared with closest_queueing. When the N poles/N vehicles Factor is greater than 0.04, the satisfied fraction stays in stable rate at around 90% for the two better strategies. But for N poles/N vehicles Factor greater than 0.15, the performance doesn’t vary a lot. Therefore, it is not necessary to increase charging poles as much as possible in order to improve system performance. The reason can be that as for closest_queueing, the vehicle may need to wait after it reaches to the charging zone, so the vehicle is not available both at charging time and waiting time, this strategy will reduce the number of available vehicles for the whole system, thus influence the system performance in satisfy booking request.

In Figure 4.1.2 shows the charging outwards distance varying different charging poles and relocation strategies. We can see that there is a clear order that random is greater than closest_free, and closest_free is greater than closest queueing for charging outwards distance. It’s easily to understand because in closest_queueing strategy, vehicle relocates to its nearest zone however closest_free’s relocation zone may a little further than closest_queueing because the zone should be not only charging zone but also currently free for charging. As for random relocation, it is the most distance consuming. The relocation zone selection is completely random regardless of the relocation starting zone of the vehicle, therefore the sum of relocation distance is much higher than the other two strategies. With the more density of poles in the operation zone, vehicle is more easily to find a charging zone in its neighbor, therefore the total charging outwards distance decreases when then N poles/N vehicles Factor is greater than 0.05 both in closest_queueing and closest_free.

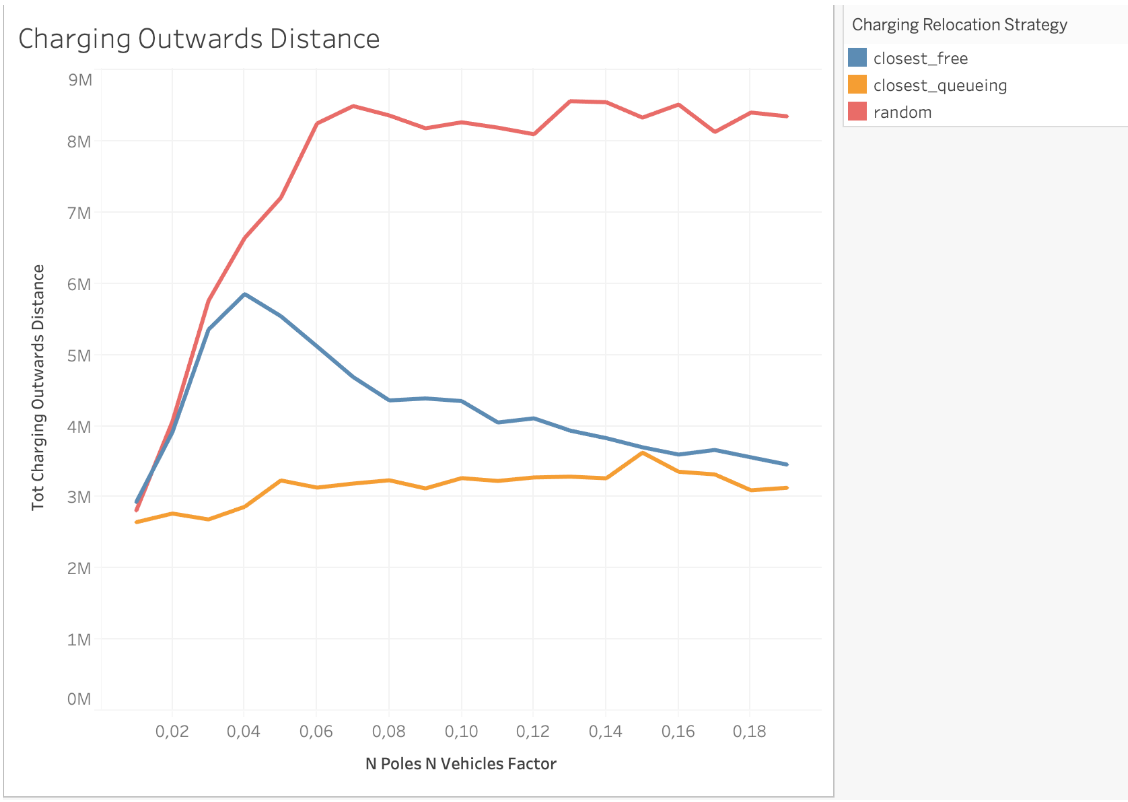


Fig. 4.1.2: Charging outwards distance with respect to N poles/N vehicles Factor. Curves show the performance with different charging relocation strategies.

4.1.2. Economic Performance

Then we move to analyze economic metrics for relocation and the whole system. In Figure 4.1.3, I show the cost related metrics including simulator cost and total cost. Firstly, it is easy to tell that the more charging poles the system has, the more infrastructure cost for building the charging zone and charging poles will have. Therefore, the total cost will increase by increasing charging pole density. Next, as for simulation cost, we can see that cost of random relocation strategy is greater than cost of closest_free strategy. Closest_queueing strategy has the lowest cost. This result is accordance with the order of charging outwards distance and fraction satisfied above. The more satisfied fraction is, the more booking request will be. Therefore, there will be more vehicles need to charge so the charging cost increase. Closest_queueing strategy has the least satisfied fraction, hence, the simulation cost is the lowest. Closest_free and random have relatively equal satisfied fraction, but closest_free's relocation distance lower than random's, so the cloest_free's simulation cost is lower than random's. Since total cost is the sum of simulation cost and scenario cost and scenario cost doesn't change with different kinds of relocation strategy, the shape and tend of total cost is accordance with the simulation cost.

Finally, in Figure 4.1.5, I analyze the revenue and profits by giving different charging relocation strategies. Revenue is strongly linear related to the total mobility duration. So, we can get a conclusion that the more satisfied fraction is, the more mobility duration it will be. In consequence, revenue will be higher with more satisfied fraction. Cloest_queueing has the lowest system performance, therefore, it causes in the lowest revenue. Closest_free and random has the relatively same system performance so the revenue they get is relatively the same. Profits refers to net income of the whole system, so both cost and revenue will influence the profit rate of the car sharing system. We can see from the figure that by the N poles/N vehicles Factor is smaller than 0.15, closest_queueing has the lowest system profits. The reason is that although closest_queueing cost least in relocation and total system, it also brings the least revenue for satisfying booking request. It both earns the least and cost the least. However, for the other two relocation strategies, both the revenue and cost is relatively high, thus the profit don't vary a lot. But in general, closest_free has the highest profit because its relocation distance is much smaller than random's.

Through comprehensive consideration of the above data, we can get the final conclusion that in the simulation case of city of Turin with the specific configuration, closest_queueing relocation strategy has the worst system performance. What's more, closest_free is the best solution both for system performance and economic performance. Besides, as for the charging pole density, too low or too high is not a good choice. Too low density leads to bad system performance, however, too much poles cost a lot. Hence, N poles/N vehicles Factor between 0.04 and 0.15 performs good in the Turin case study.

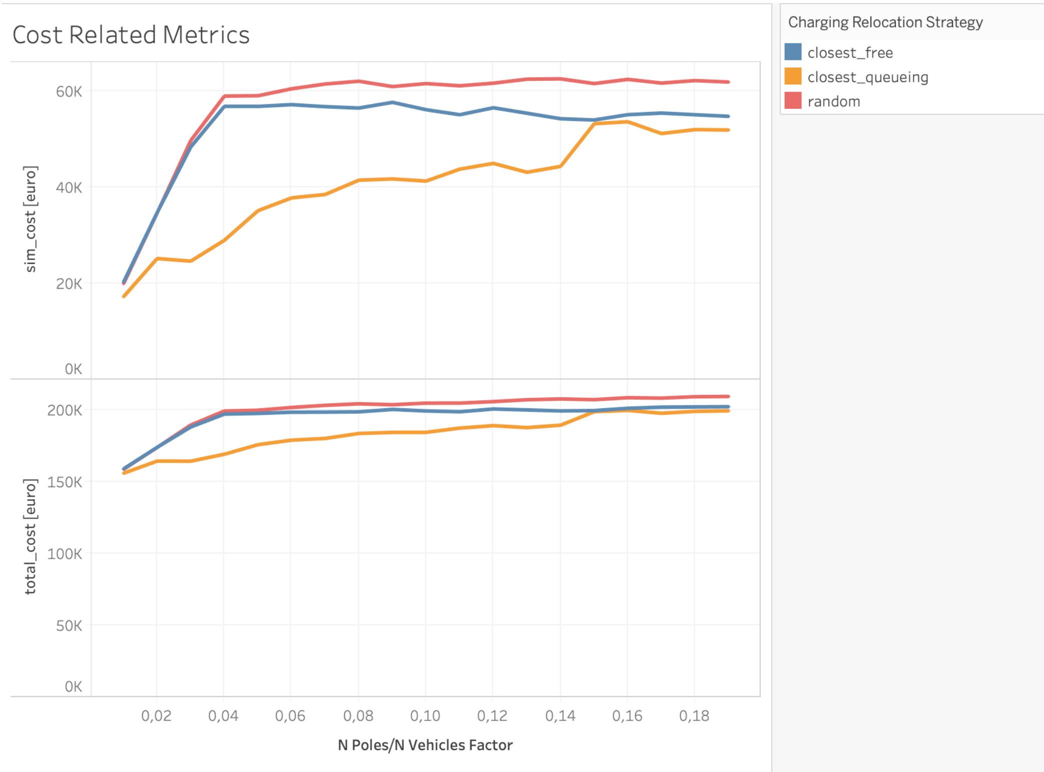


Fig. 4.1.3: Cost Related Metrics with respect to N poles/N vehicles Factor. Curves show the performance with different charging relocation strategies.

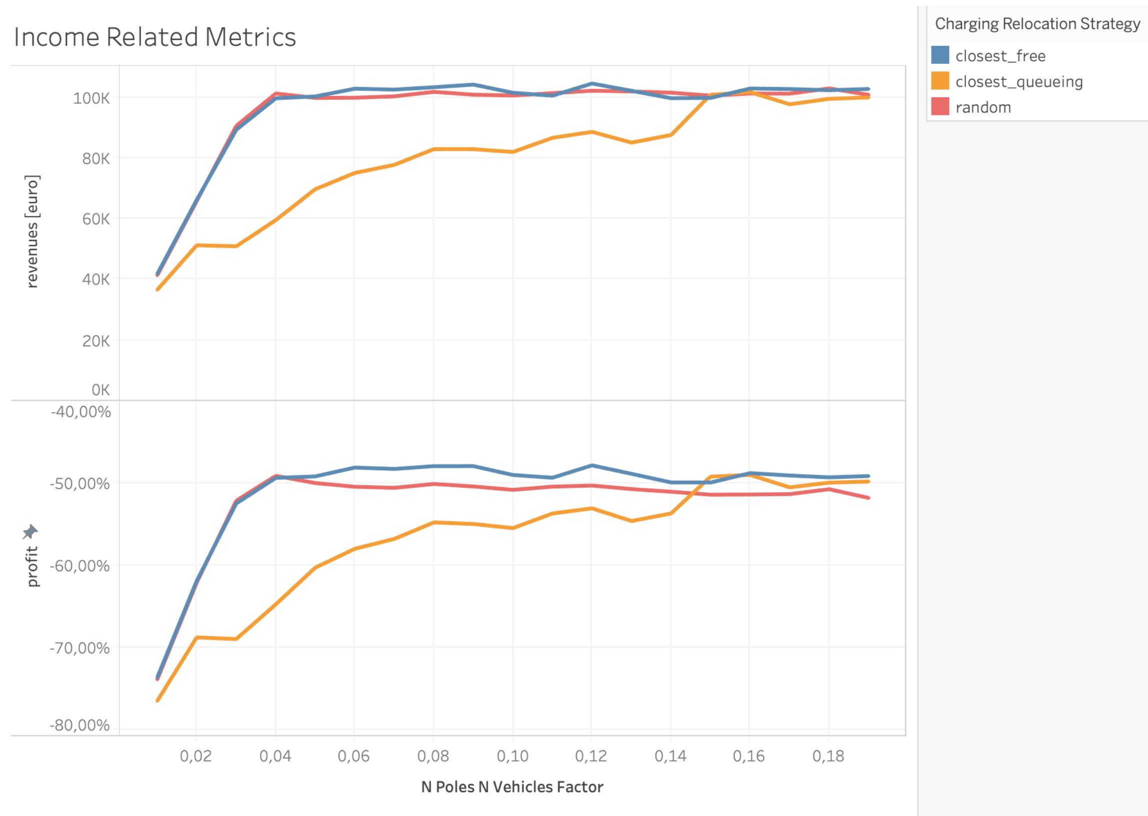


Fig.4.1.4: Income Related Metrics with respect to N poles/N vehicles Factor. Curves show the performance with different charging relocation strategies.

4.2. Proactive Relocation Strategy Results

Still, in order to see how different proactive layer strategies will affect the whole car sharing system. I implement the proactive layer strategy in the original simulator. The number of relocation workers will influence both the relocation throughput and relocation cost. Therefore, different number of relocation workers are set and tested. I run the simulator in the city of Turin from October to November of 2017. The parameters grid for this set of simulations is in table 4.2:

Parameter	Description	Values
α	the charging threshold	20
β	the charging upper bound	100
n vehicles	total vehicles run in the simulation	(100,400), step 10
n requests	number of booking requests	10000
cps zones percentage	charging zones density with regards to total zones	0.2
N poles/N vehicles fator	number of charging poles with regards to vehicles	0.2
charging relocation strategy	charging relocation strategy used for experiments	"closest_free"
annual leasing cost	annual leasing cost per vehicle	4000
disinfection cost	disinfection per charging operation	15
washing cost	washing cost per vehicle	8
pole labor cost	labor cost for building the pole	2200
pole hardware cost	hardware cost for building the pole	1700
pole useful life	how many years a pole can use	10
pole annual maintenance cost	pole maintenance cost per year	500

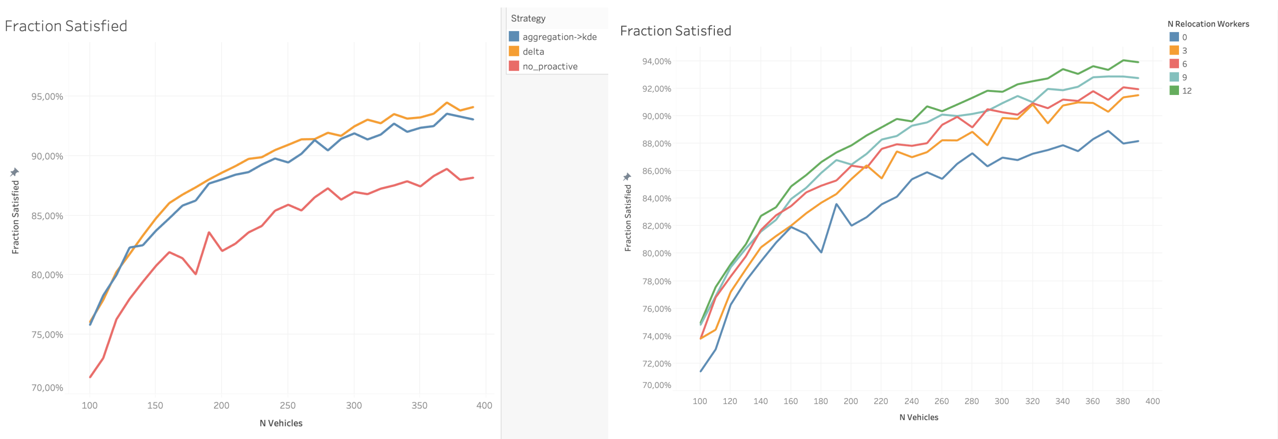
cosap annual tax	pole tax fee per year	355
zone make-ready cost	building cost per zone	1500
kWh cost	energy price per kWh	0.19
vehicle relocation	the flag to show if proactive relocation will be execueted	[True,False]
vehicle relocation strategy	vehicle relocation strategy chosen for experiments	"only_scheduled"
vehicle relocation technique	how to choose starting zones and end zones of relocation	[{"start": "aggregation", "end": "kde_sampling",}, {"start": "delta", "end": "delta",}]
n relocation workers	the number of proactive relocation workers	[0,3,6,9,12]
worker hourly salary	price for hiring the relocation workers	18
price per minute	price the when a user use the vehicle per minute	0.2
request rate factor	the ratio between the number of the real booking request and input booking request	1.69546

TABLE 4.2

PARAMETER FOR PROACTIVE RELOCATION EXPERIMENTS

4.2.1. System Performance

In the Figure 4.2.1.1, I visualize the satisfied fraction of the whole car sharing system in the above configuration. Part (a) shows the system performance among different proactive relocation strategies. It is clear that proactive layer strategy improves the satisfied demand. What’s more, **delta** relocation strategy performs better than other strategy. I change the total number of vehicles in the whole system and the number of relocation workers in part (b). The more workers, the more relocations are executed hourly. When the number of worker equals to 0, it means that no system operator operates vehicle relocation, therefore, proactive layer relocation doesn’t trigger. We can see that when compared with no proactive relocation, proactive relocation has an average 2%~10% improvement in fraction satisfied metric when the number of vehicles is in an abundant level. However, when the number of vehicles is too little, in this case less than 160, fraction satisfied doesn't vary a lot. The reason is that the number of vehicle itself can't meet the demand of system, instead of unbalanced distribution of vehicles leads to the unsatisfied booking request. When the number of relocation workers increases, the number of satisfied booking requests also increases. The reason is that more vehicles are relocated to the region which more requests are generated. Another obvious phenomenon is that the more vehicles the whole system owns, the more booking request can be satisfied.



(a)various strategies *(b) various number of workers*

Fig 4.2.1.1: Fraction Satisfied with respect to the number of vehicles.

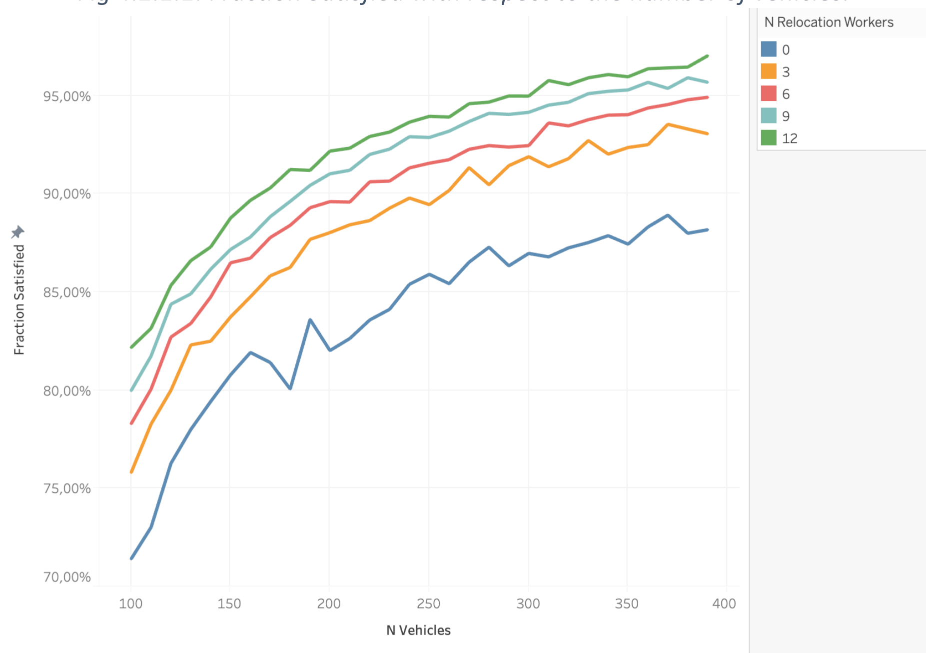


Fig. 4.2.1.2: Fraction Satisfied with respect to the number of vehicles for new proactive relocation .Curves show the performance with different number of relocation workers.

Meanwhile, I also analyze vehicle relocation distance by varying the number of relocation workers. The results are shown in Figure 4.2.1.3. It is easy to understand that with more number of workers, more relocation operations are executed, so the total relocation distance is positively correlated with the number of workers. When the number of vehicles increases, total vehicle relocation distance also increases slightly. We can see from Figure 4.2.1.4 that the number of relocations also increases slightly with the number of vehicles increases. Why the number of relocation operation increases? The reason is that although our proactive relocation is triggered hourly, it selects vehicles which end their trips recently. That means that only if a trip ends in a zone included in the schedule, we relocate. So, we do less relocation than planned, until the number of vehicles is high enough to trigger all scheduled relocation for each hour. In order to avoid this issue, I have developed a new proactive relocation strategy. The relocation operation is only triggered at the end of each hour, instead of the end of trip. The results are shown below:

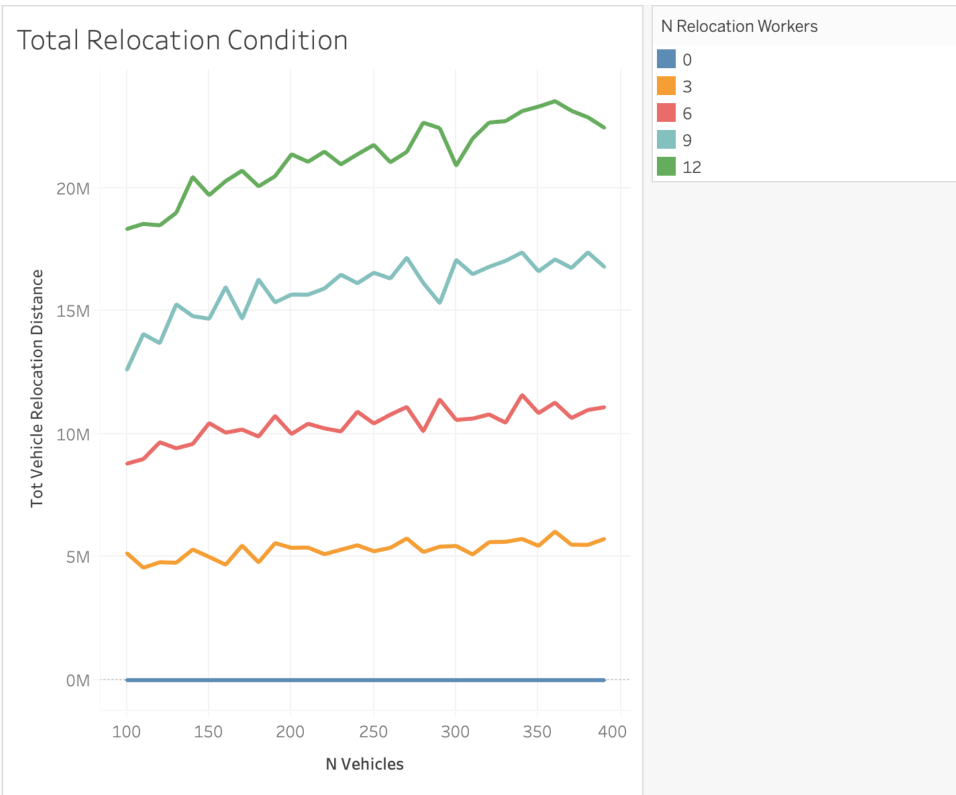


Fig. 4.2.1.3: Total Vehicle Relocation Distance with respect to the number of vehicles. Curves show the performance with different number of relocation workers.

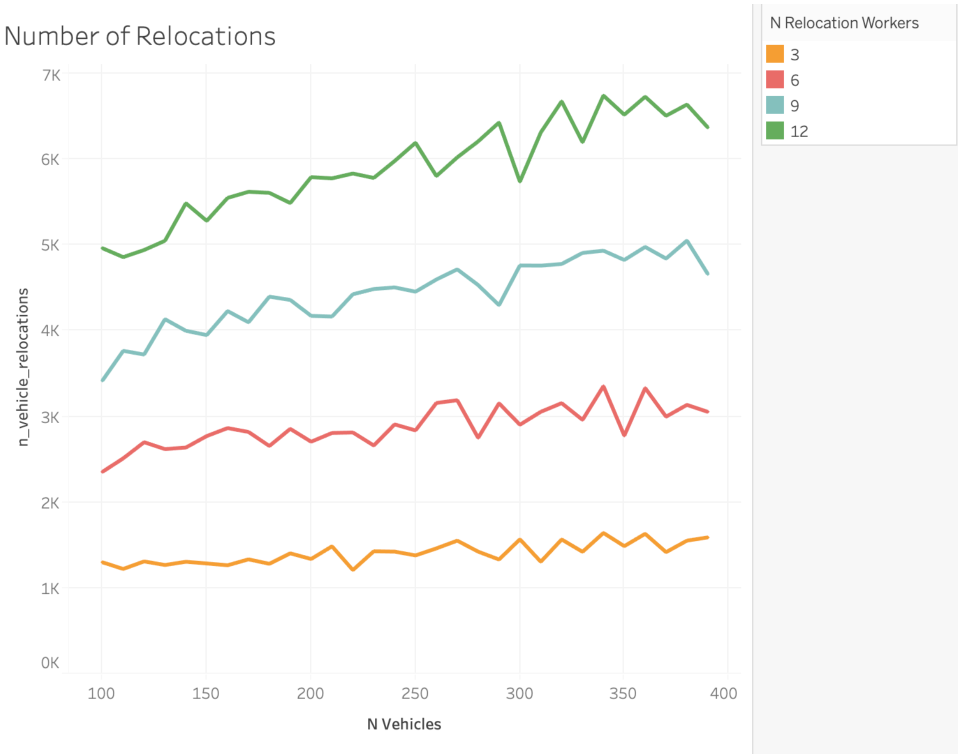


Fig.4.2.1.4: Number of Relocations with respect to the number of vehicles. Curves show the performance with different number of relocation workers.

We can see from the lower part of Figure 4.2.1.5 that the new proactive relocation strategy keeps the same amount of relocations no matter the number of vehicles changes or not. Meanwhile, with higher car density, shorter distance between the starting zone and ending zone will be. For this reason, the total relocation distance decreases slightly with the number of vehicles increases.

Besides, satisfied fraction is also updated and plotted in Figure 4.2.1.2. With more relocations, more requests will be satisfied. The system performance gap between no relocation and relocation becomes bigger.

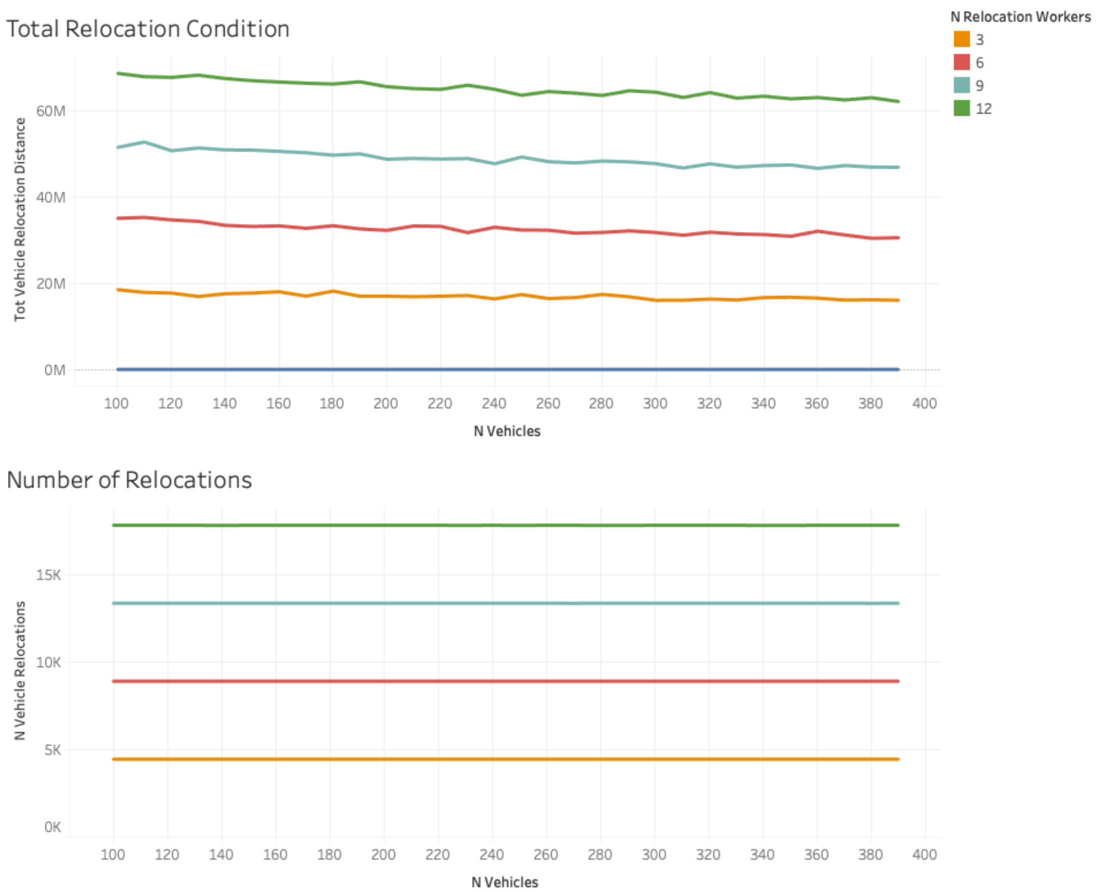


Fig.4.2.1.5: Number of Relocations & Relocation Distance of new proactive relocation with respect to the number of vehicles. Curves show the performance with different number of relocation workers.

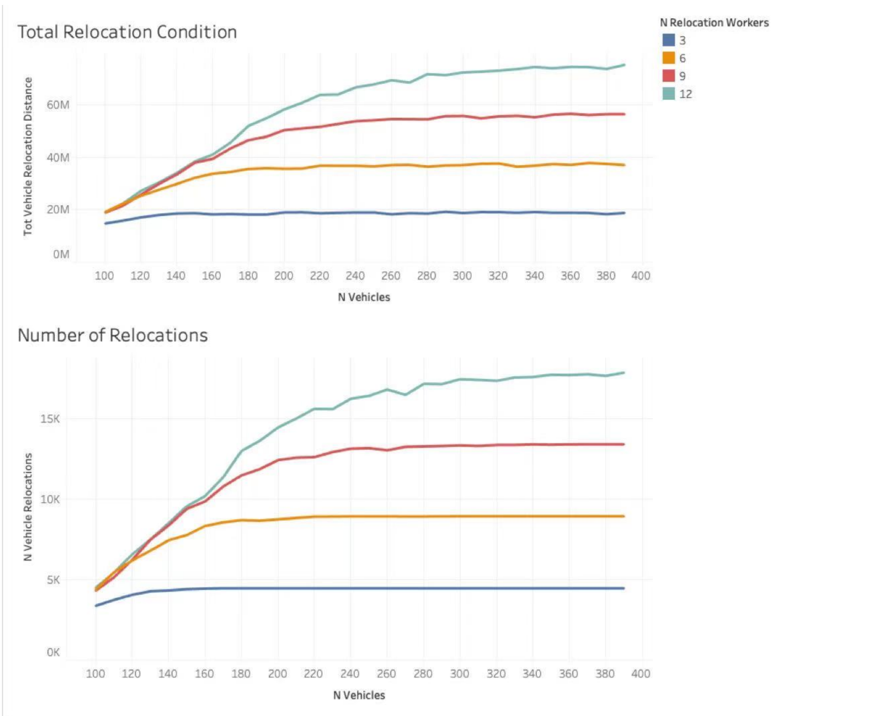


Fig. 4.2.1.6: Number of Relocations & Relocation Distance of delta strategy with respect to the number of vehicles. Curves show the performance with different number of relocation workers.

4.2.2. Economic Performance

Now we move to see the economic change that the proactive layer strategy brings. Firstly, let's see the index of revenue in Figure 4.2.2.1. The trend and shape of revenue are similar with the satisfied fraction. When the number of vehicles is smaller than 160, revenues doesn't vary a lot by different number of workers. When the number of vehicles is enough, proactive layer strategy actually improves the revenue of the whole system. With more number of workers, the more money the system can earn. However, that's not absolute regulation. We can see that when the number of vehicles equal to 310, three relocation workers brings higher revenue compared with six and nine workers. Then reason is that revenue reflects the total mobility duration. Total mobility duration is the product of number of booking request and average duration per trip. With more requests satisfied, that doesn't mean the total mobility duration will 100% increase. But in a general view, increasing the number of relocation workers has a positive effect in increasing the whole revenue. When considering new proactive relocation strategy in Figure 4.2.2.2, revenues becomes higher with the same number of relocation workers and vehicles because of higher satisfied fraction for the car sharing system.

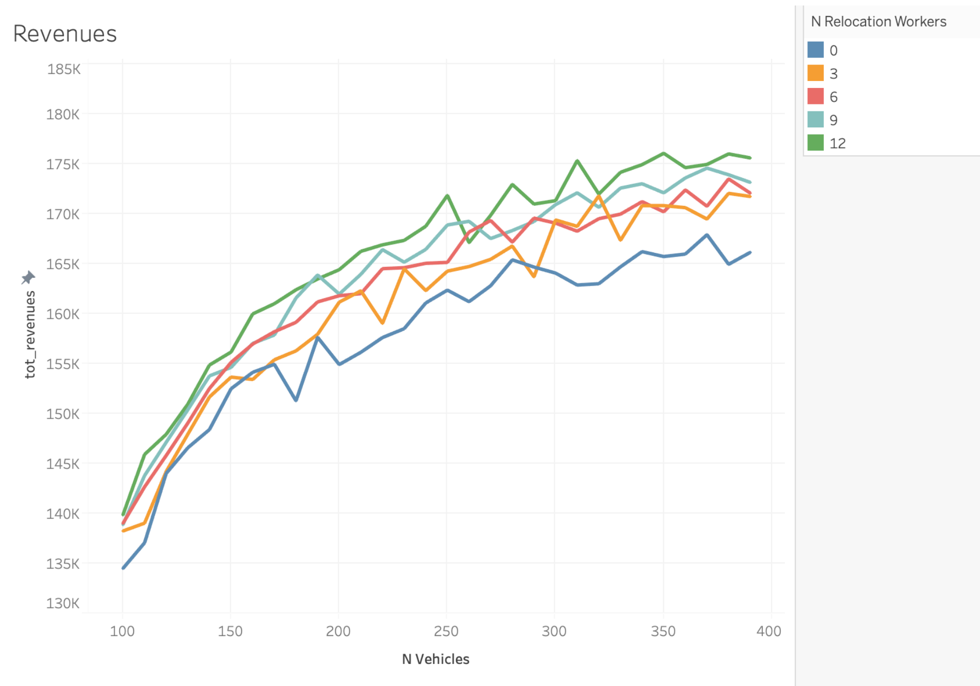


Fig. 4.2.2.1: Total Revenues with respect to the number of vehicles. Curves show the performance with different number of relocation workers.

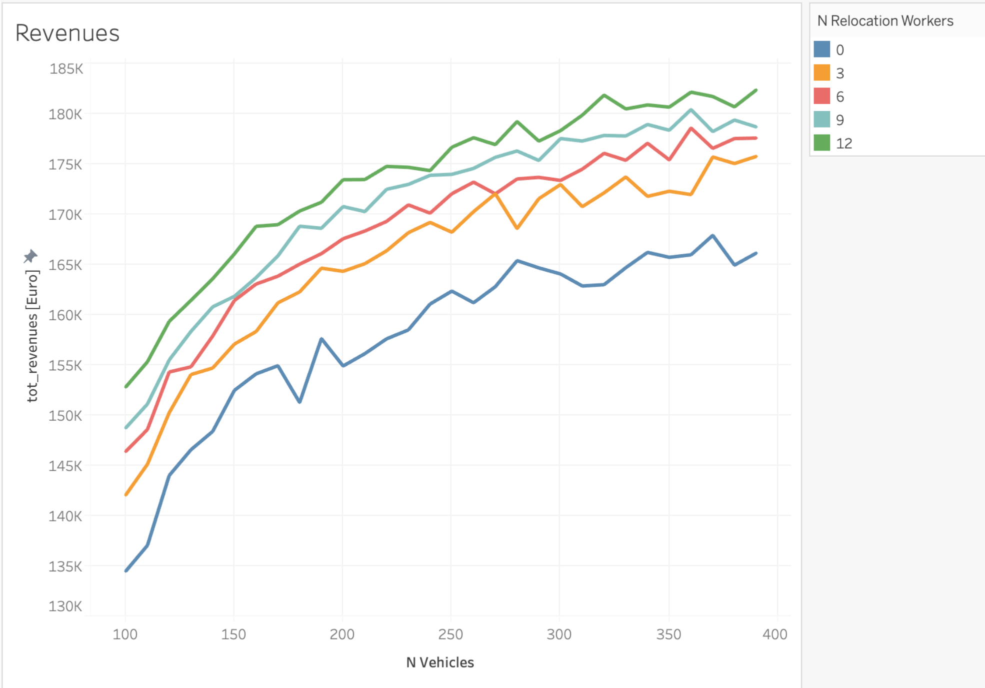


Fig. 4.2.2.2: Total Revenues of new proactive relocation with respect to the number of vehicles. Curves show the performance with different number of relocation workers.

Figure 4.2.2.3 shows the cost related metrics including total cost and proactive relocation cost. Relocation cost is highly correlated with the total relocation distance. Both the shape and trend is similar with upper part of Figure 4.2.1.6. As for total cost, it is easy to figure out that the more vehicles and the more number of relocation workers, the higher total cost will be.

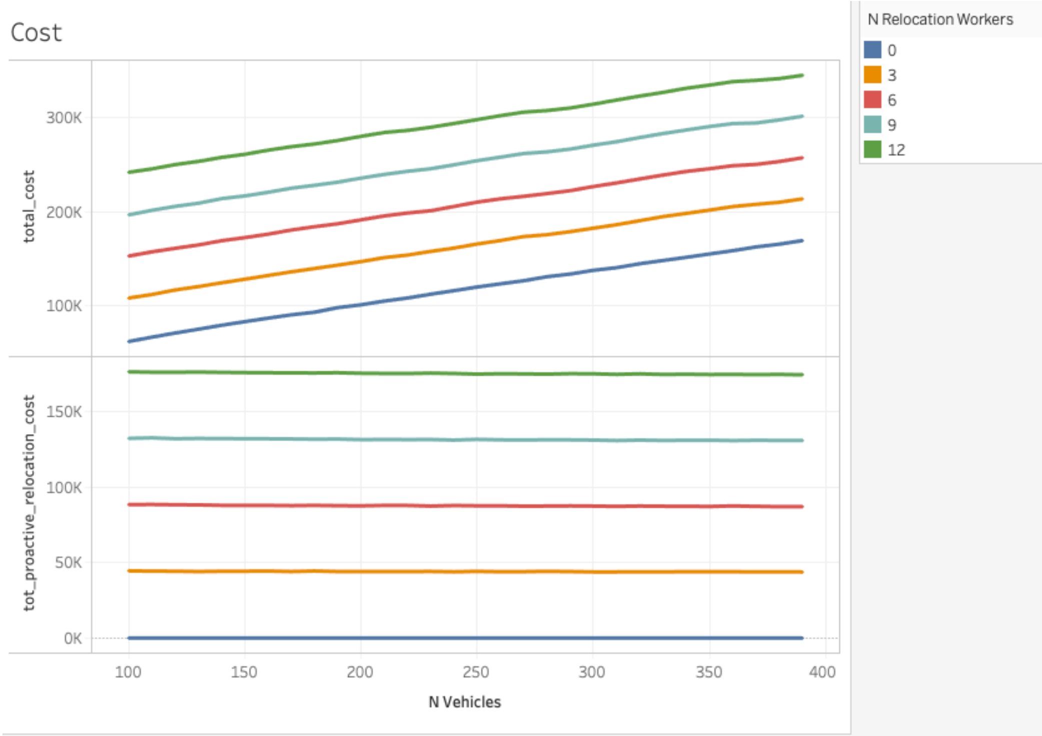


Fig.4.2.2.3: Total Cost & Proactive Relocation Cost of new proactive relocation with respect to the number of vehicles. Curves show the performance with different number of relocation workers.

Last but not least, profits are shown in Figure 4.2.2.4. Profits are in a reverse order compared with revenues. No proactive relocation scenario gets the highest profit. The more relocation workers employed, the lower profit system performs. It's not difficult to find out the reason. Although proactive relocation improves the system performance, which means more requests are satisfied thus brings more revenue, both relocation worker cost and energy cost are not evitable increasing at the same time. Profits should be influenced not only by the revenue, but also by the cost. In this case, relocation operation costs more than the extra revenue it can bring for the whole system. What's more, extra cost with more vehicles also exceeds the extra revenue more vehicles can bring. In consequence, profits show a downward trend with the number of vehicles increases.

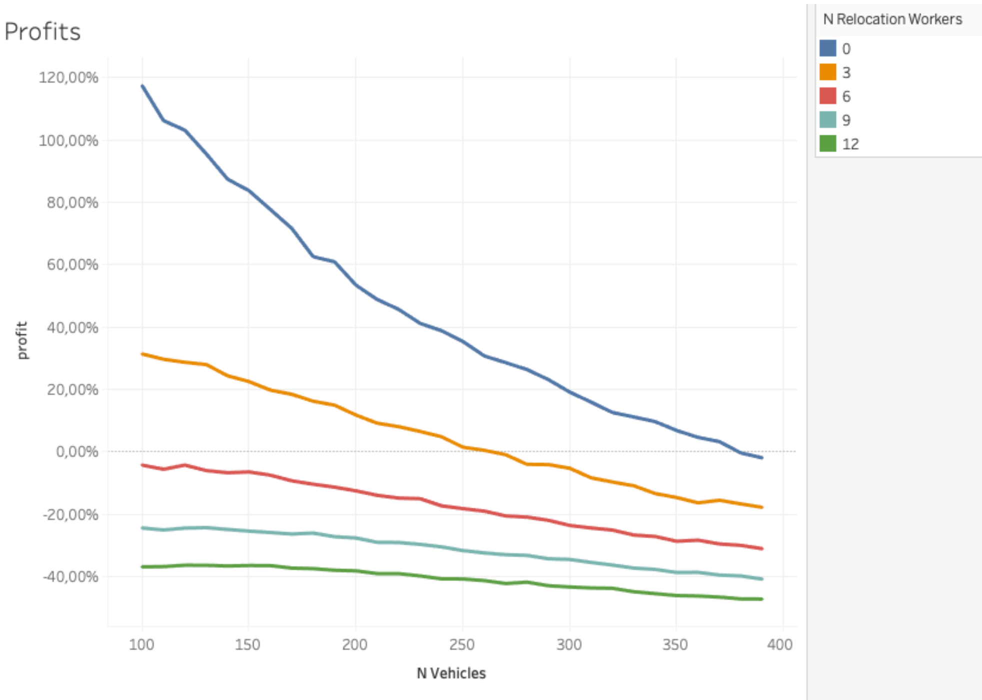


Fig 4.2.2.4: Profits of new proactive relocation with respect to the number of vehicles. Curves show the performance with different number of relocation workers.

In order to show the above conclusion more clearly, I have drawn Figure 4.2.2.5. Given two scenarios of no proactive relocation and proactive relocations with three relocation workers. I calculated the extra revenues relocation operation brings and extra cost it needed. We can see that extra revenue is far less that extra cost that relocation operation. Improving system performance comes at the cost of additional overhead.

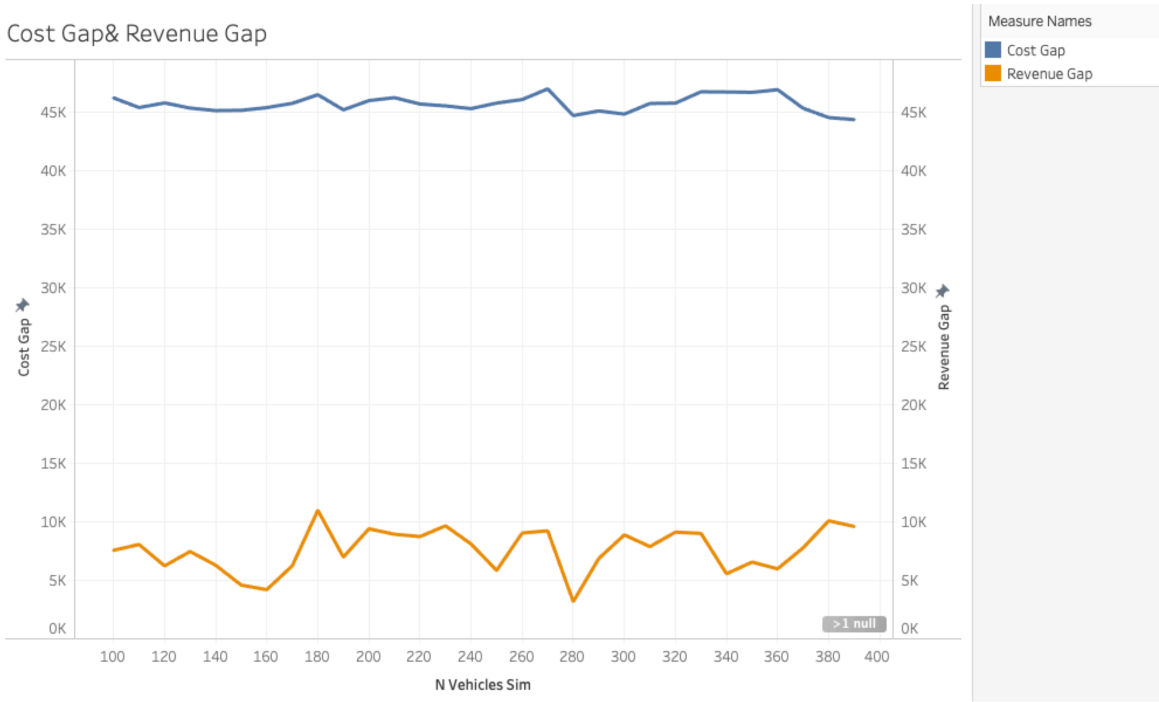


Fig 4.2.2.5: Cost Gap and Revenue Gap with respect to the number of vehicles.

Chapter 5

Conclusions and Future Work

The main content of the thesis is that modify and run the simulation based on previous e3f2s work. Both reactive and proactive relocation strategies have been implemented. The simulation model has a number of input parameters that allow for the evaluation of numerous scenarios. By using the simulation in the case of Turin, I focus on plotting the results and digging out the hidden reason under the system performance and economic performance by changing fleet size, the number of relocation workers, the number of charging poles and varying different relocation strategies. In the result of Turin, it was found that when considering post charge relocation strategy, closest-free has the overall best performance whereas closest-queueing is worse considered with the other two. When it comes to proactive relocation, it improves the system performance however it leads to extra cost. With the more number of relocation operations, the more requests can be satisfied. Among different relocation strategies, delta performed best in proactive relocation scenario. The balance between more cost and more earnings should be considered carefully.

There are still many things that can be considered besides the thesis:

- Improve economic performance in proactive strategy, for example, add relocation operations frequency.
- Implement other kinds of reactive relocation strategy, for example, add relocation scenario besides charging relocation.
- Implement other kinds of proactive relocation strategy, for example , add relocation operations strategy using machine learning .
- Add more meaningful metrics for system and economic analysis.
- Expand the simulation scenarios in other cities and make comparison among cities.
- Improve the simulation with hybrid energy transportation. For example, add scooter, oil cars in the simulation. Consider more complex scenarios.

Bibliography

[1]. https://ourworldindata.org/co2-emissions

[2]. https://www.statista.com/outlook/502/100/carsharing/worldwide#market-onlineRevenueShare

[3]. Sheheen, S., Cohen, A.P., 2007. Growth in worldwide carsharing, an international comparison. Transportation Research Record 1992, 81–89.

[4]. Alessandro Ciociola,2019,Modelling, simulation and design of Electric Car Sharing systems with eC2S

[5]. https://www.bbc.co.uk/bitesize/guides/zvxp34j/revision/3

[6]. https://pypi.org/project/simpy/

[7]. http://geopandas.org/.

[8]. https://numpy.org/.

[9]. https://www.tableau.com/.

[10]. https://www.jetbrains.com/pycharm/

[11]. https://jupyter.org/

[12]. Laporte, G., F. Meunier, R. W. Calvo. 2015. Shared mobility systems.

[13]. Kek, A. G.H., R. L. Cheu, Q. Meng, C. H. Fung. 2009, A decision support system for vehicle relocation operations in carsharing systems.

[14]. Jian, S., D. Rey, V. Dixit. 2016. Dynamic optimal vehicle relocation in carshare systems.

[15]. Bruglieri, M., A. Colorni, A. Luè. 2014. The vehicle relocation problem for the one-way electric vehicle sharing.

[16]. de Almeida Correia, G. H., A. P. Antunes. 2012. Optimization approach to depot location and trip selection

[17]. Pfrommer, J., J. Warrington, G. Schildbach, M. Morari. 2014. Dynamic vehicle redistribution and online price incentives in shared mobility systems.

[18]. Raviv, T., M. Tzur, I. A. Forma. 2013. Static repositioning in a bike sharing system: Models and solution approaches.

[19]. Fricker, Christine, Nicolas Gast. 2016. Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity.

[20]. Haider, Zulqarnain, Alexander Nikolaev, Jee Eun Kang, Changhyun

[21]. Kwon. 2018. Inventory rebalancing through pricing in public bike sharing systems.
[22]. Weikl, S., K. Bogenberger. 2015. A practice-ready relocation model for free-floating carsharing systems with electric vehicles–mesoscopic approach and field trial results.
[23]. Herrmann, S., F. Schulte, S. Voß. 2014. Increasing acceptance of free-floating car sharing systems using smart relocation strategies: A survey based study of car2go hamburg.
[24]. Caggiani, L., R. Camporeale, M. Ottomanelli. 2017. A dynamic clustering method for relocation process in free-floating vehicle sharing systems.
[25]. He, L., Z. Hu, M. Zhang. 2019a. Robust repositioning for vehicle sharing.
[26]. Alessandro Ciociolab, Michele Coccab, On Scalability of Electric Car Sharing in Smart Cities Michelangelo Barullia
[27]. Tal Raviv and Ofer Kolka. «Optimal inventory management of a bike-sharing station».
[28]. https://scikit-learn.org/stable/modules/density.html#kernel-density-estimation.
[29]. https://en.wikipedia.org/wiki/Kernel_density_estimation/
[30]. Leonardo Tolomei, Relocation strategies for e-scooter system optimization