Integrating sensors data in optimization methods for sustainable urban logistic

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (30^{th}cycle)

Integrating sensors data in optimization methods for sustainable urban logistics

By

Edoardo Fadda

******

Supervisor(s):
Prof. Guido Perboli

Doctoral Examination Committee:
Prof. Rei Walter, Referee, UQAM, Montreal, Canada
Prof. Mansini Renata, Referee, Universitá Di Brescia
Prof. Ricciardi Nicoletta, Università di Roma "La Sapienza"
Prof. Maggioni Francesca, Università di Bergamo
Prof. Crainic Teodor Gabriel, CIRRELT, Montreal, Canada

Politecnico di Torino
2018

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and do not compromise in any way the rights of third parties, including those relating to the security of personal data.

Edoardo Fadda
2018

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Acknowledgements

Even a barber who shaves all those, and those only, who do not shave themselves may not exists.

This thesis is the result of my PhD in the department of Computer and Control systems at the Politecnico of Torino. It has been without doubt a really important experience. In these three years I understand a lot of the academic working environment and I discover a huge amount of fascinating topics. This journey was not easy, then first and foremost, I want to thank my family, my girlfriend and all my friends for the support that they give to me.

Then, the person to who I must say thanks is my advisor Professor Guido Perboli. I truly appreciate his patient, guidance, and support through my doctorate study. His advices have been of immense value to me. I also want to thanks all the present and past members of Operations Research and Optimization (ORO) group and all the people of Laboratory 8 at Politecnico di Torino. In particular, Professor Roberto Tadei which has contribute to my formation in a strong way, Luca Gobbato who basically teach me how to program, he has built a working environment lovable and full of laughter. Finally, I also want to thanks Mario Maria Baldi, Sandro Cumani, Michele Garaffa, Stefano Musso, Mariangela Rosano, Rosario Scatamacchia and Valerio Vallesio.

Last but to least my thanks to other professors that I had the luck to work with. They are Vito Cerone, Diego Regruto and Giovanni Squillero. I am really great-full for all the advices that they give me and for the opportunity to see and to compare different ways of working.

Contents

List of Tables

Abstract

In the world of urban mobility and logistics, new technologies are spreading and are deeply changing the way in which this business is conducted. These technologies are based on three factors.

The first one is the growth of city population i.e. the increasing number of customers in the same area that decreases the delivery costs. Nowadays, UN [103] calculates that the 54 % of the world's population lives in cities and this percentage is expected to grow to 66 % by 2050.

The second factor is the development and diffusion of technologies that enable industries as well as customers to directly interact without the need of resellers i.e. smart-phones. A new business model based on this opportunity is social engagement, it consists of the request by the companies to the people to perform part of their job in exchange for a reward. Another example is e-grocer which is changing the way the grocery business is conducted. It allows people to purchase groceries online and to receive them at home or in given centers. The flow of operations of this business channel is the following: customers order groceries using websites or mobile applications, the orders are executed by the company (basically, for each order, the company has to collect in a single container all the groceries ordered by the customers), then the containers with the goods are delivered to the customers, either directly to their user house or to a collection point.

The third factor is Internet of Things (IoT), a new paradigm that enriches instruments with the capabilities of collecting data and to communicate with other devices. This technology is already spreading in several applications. In

Figure 1.1, we represent the growth and the expected growth of the number of the connected devices.

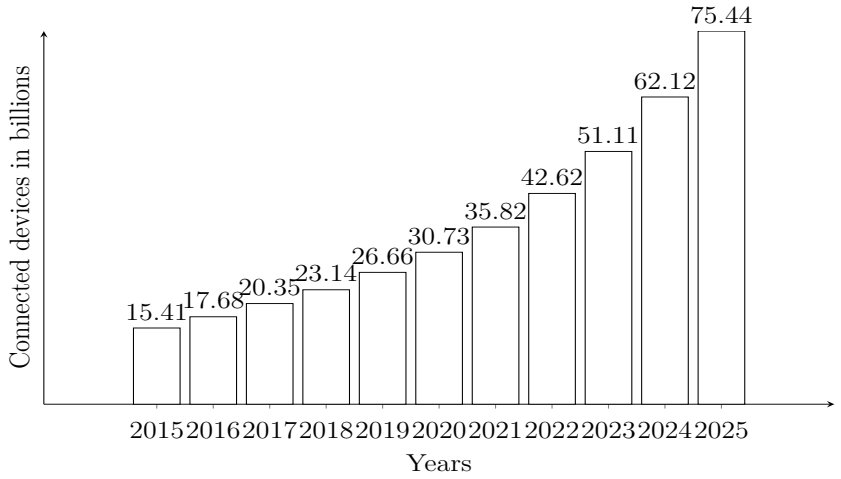

Figure 1.1 The figure shows the number of connected devices in billions and a future forecast. The data are taken from https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

The main advantage of this technology is to gather in real time local data of a wide area. Nevertheless, the main problem with this technology is that it requires a large infrastructure in order to give to all the sensors the connectivity or as an alternative to use manpower to visit all the sensors.

The main objective of this thesis is to develop new optimization methods and algorithms that use new opportunities and solves new problems generated by these factors. In particular, we focus on the exploitation of the sensor data in optimization method for sustainable urban logistics by considering two case studies: the optimization of waste management and social engagement for e-grocery and IoT data collection. In both of them, the use of new technologies modify the way by which information is gathered and communicated, hence it changes how the studied problems can be formulated and solved.

The first case study has been chosen because of the lack of information usually associated with the sector waste collection operations. To our knowledge,

the project Optimization for Networked Data in Environmental Urban Waste Collection (ONDE-UWC) [1] is the first one that exploits IoT data for smart-city applications. In this project, the data related to the evolution of the quantity of waste are collected by the vehicles used for the collection. By using these data, it is possible to develop an optimization model, able to consider the evolution of the waste and to plan in an optimal way the routing and scheduling of the waste collection. This is a very important example of how IoT data change the nature of the optimization problem and the associated business model.

Furthermore, we choose this case study because optimization of waste collection is of central importance for public health, it has an important economic component, and the efficiency of the services produces economic benefit. Moreover, waste management has political importance because it involves local administrations and it also has a social impact, if we consider emissions and pollution that can compromise the health of citizens and visual pollution.

This problem is even more important due to the expansion of urban areas and the growth of consumption increase waste production. [2] estimates that in the European Union each year more than 2.3 billion tons are produced. The 10% of this quantity is the municipal waste that is characterized by many critical factors due to the urban environment in which it is performed. [9] enlighten the political importance, because of its composition, distribution among many waste generators, and link to consumption patterns. For all these reasons, in this thesis we face the problem of optimizing the number of time shifts and the corresponding routing of the waste collection operations of a company operating near the city of Torino. The results obtained in this application are good enough to convince the companies involved to build a commercial solver based on the solution.

The model developed in this context can be used to describe every problem characterized by a network of nodes where each one of them is producing, with a different rate, a quantity that must be picked up and delivered to a given node, before too much quantity is produced. Examples of this kind of quantities are the number of people waiting for the bus at the bus stations, the

[1] www.onde.city

quantity of waste in a dumpster or the number of mechanical parts produced by several machines that need to be transported to another machine. Since in those contexts it can be difficult to have real-time information, it is possible to calibrate a statistical model by using the data collected by IoT devices. The study is innovative because it does not enforce the periodicity of the routes. Nevertheless, due to this additional degree of freedom, the mathematical model rises in complexity and it is not solvable with commercial solvers. For this purpose, we develop a math-heuristic able to compute a solution of the problem in a time compatible with the operations of the company. This solution method allows the heuristic to be used in the real field and, with the IoT architecture implemented in the project represent a breakthrough for the sector of municipal waste collection.

We choose to consider social engagement for e-grocery and IoT data collection because collecting data from the sensors and delivering packages are actions that do not require any specialization and that can be performed by every person. Furthermore, while the standard workforce has to travel on purpose to go to the location of the task, it is possible that some person passes in that area for doing other stuff. Then, it is possible to ask common people to perform tasks that otherwise would have required the standard workforce. This principle generates two positive effects: the first one is that the company can use its workforce for doing tasks that require more skilled people, the second is that the tasks are done by using travels that would have occurred in any case (this decreases the total number of travels, hence it decreases traffic and pollution).

We choose to consider both e-grocery and IoT data collection because of their similarities: both problems have an important time constraint. The information of the sensors must be gathered as soon as possible otherwise they are old and they do not describe the real scenario, while for an e-grocery retailer delivering as soon as possible increases the quality of the product and of the service (if the groceries are not delivered for a long time they go bad).

The principle of social sharing can be applied in several situations, for example of the Coiote project by TIM. This project has the goal to invite people sharing their internet connection with dumpsters in order to let each

dumpster to communicate its quantity of waste to a central unit. In this way, the central unit can organize the waste collection operations in an optimal way by voiding the dumpsters only when they are nearly full. Logically, in exchange to the internet connection, TIM gives people a reward.

The underline optimization problem is to minimize the amount of rewards and to give connection to all the dumpsters. The importance of this project relies on the fact that it applies the users engagement paradigm to activities of importance to the community.

This problem has never been tacked in the literature despite the fact a huge amount of applications can be described by such a model. For example, the same model of the Coiote project is suitable to every situation where it is possible to ask people to do some tasks by sharing their resources in exchange for a reward. Furthermore, this business model can be related not only to social engagement but also to the workforce of the company distributed in the urban context. The only difference is the costs of the reward for the execution of the task: users are cheaper than workers because they do not need to be hired. The main limitation of this model is the need for a reliable model of the flow of people. Once again, it is important to remark that the technology spread is ready to support these business models [2].

To our knowledge, this is the first time that the optimization problem derived from crowd-sourcing is considered. We formulate the deterministic and the stochastic mathematical model of the problem and we propose heuristic methods able to find a good solution in a reasonable amount of time for both types of problems. Furthermore, we define a model for the urban flow of people and we calibrate it with some real data. The innovation provided by these methods is two-fold: they are the first studies exploiting crowd-resources for performing tasks and they also develop efficient solution methods able to compute a good solution in a time interval compatible with the needs of a real field application.

[2] Data available from [101] say that the 64% of the European population has a mobile internet connection (more than 80% in Italy).

We consider the problem in a network whose nodes, in the real field, can be thought as mobile phone cells. This assumption makes the problem more realistic because we can have data regarding the number of people in each cell thanks to their mobile phones. We assume that in each cell there is demands or offers i.e. tasks to do or people available to do them (see Fig. 1.2).

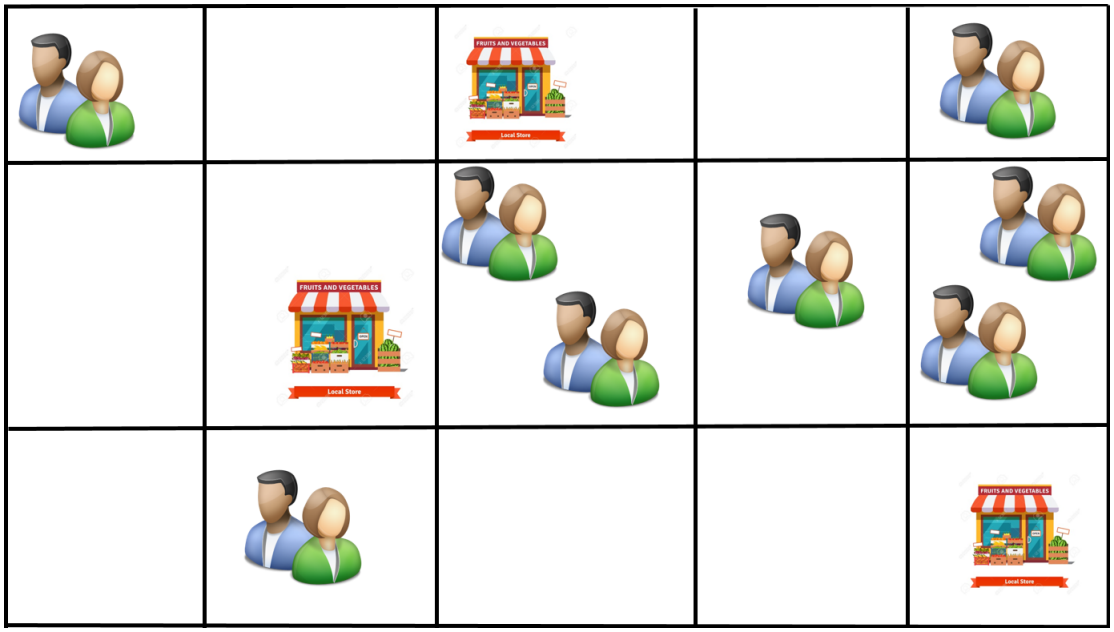

Figure 1.2 Example of network: in each node there are customers available to transport freight or there is a shop.

The reason for this assumption is due to the fact that the people in a cell can do all the operations in that particular cell with a little cost. This problem depends on two factors: the variability of the number of people in a cell and the possibility that some people could accept to do the task but then they omit to do it. The problem has been formulated as an integer program in the deterministic setting.

Since the deterministic framework does not take into account the variability in the number of available customers, the problem has been formulated as a multiperiod two-stage integer stochastic programs in the stochastic setting. The stochastic version is approximated by discrete probability distributions that generate the scenario trees. The resulting approximated problem becomes a deterministic integer problem of big dimension, beyond the reach of exact methods. Hence, heuristics and meta-heuristics are required. In this context,

we develop useful meta-heuristics for both the deterministic and stochastic versions of the problem. It is important to remark that, since this problem has not been found in the literature, our results define the present best performance. In order to make a complete analysis of the proposed solution methods, to assess the efficiency of the meta-heuristics and to evaluate the robustness of the solutions we ran several numerical experiments by mean of a generator of instances of different sizes. These experiments conclude the analysis of the optimization of social engagement.

This thesis is organized into two parts, in the first one (Part I), we describe the algorithm for computing the routing and the scheduling of the waste collection by using production information, developed during the project ONDE. In the second one, (Part II) we define a mobility model describing the flow of people in the city by using data provided from mobile phone applications and we present mathematical models and algorithms for the optimization of the delivery operations performed through social engagement.

In particular, in Chapter 2 we present the e-grocery market, its history and we analyse pros and cons of this business model by comparing the activity of different companies in the business. In Chapter 3 we describe the method and the algorithm used in order to exploit the data produced by IoT in the field of waste collection. In Chapter 4 we describe the deterministic version of the problem, we analyse similar problems in the literature and we present the performance and the limits of exact methods. In the same chapter, we also develop meta-heuristics able to obtain a good solution to the deterministic problem in a reasonable amount of time. In Chapter 5 we present the stochastic version of the problem in terms of a two-stages multi-periods stochastic model, we present the performance and the limits of exact methods as well as an analysis of the special structure of the problem. In this chapter, we also compute the value of the stochastic solution. In Chapter 6 we apply a heuristic based on the particular structure of the problem in order to find good solutions to the stochastic problem in a manageable computational time. Finally in Chapter 7 we describe the data that enable us to calibrate the urban flow model and we propose a new approach to estimate if the user is on a bus or in a car and by using these results, we propose a model able to describe the urban flow

of people and we calibrate it by using real mobile phone data in the city of Torino. This mobility model is used in the remaining of the thesis in order to generate realistic instances of the problem. The conclusions are reported in Chapter 8.

Introduction

In this chapter, we introduce the two case studies of this thesis: crowd-shipping for e-grocery and IoT application for smart-cities. It is worth noting that in both cases, the use of ITCs and in particular of IoT solutions modify the informational processes (i.e., the processes by which information is gathered and communicated) that define how the studied problems can be formulated and solved. In particular, in Section 2.1, we describe actuals models for freight transportation and in Section 2.2 we describe the e-grocery business models. Then, in Section 2.3, we introduce the business models of companies that use crowdsourcing or, more in general, social engagement. Finally, in Section 2.4, we depict some applications of IoT for smart-cities and in Section 2.5, we present the general problems that we face in this thesis.

2.1 Urban Freight Transportation

The growth of population in urban areas increases the demand for goods. This trend, together with the improvement of the living standards and with other new trends such as the Just-In-Time delivery or the home delivery service require to logistics' companies high level of efficiency. In this section, we describe the most used technique for urban logistics i.e. the two-echelon logistic model. In this model, freight is transported from outside the city to its border (in the so-called urban consolidation centers) with trucks and from the consolidation centers to the customers (the so-called last mile) with smaller vehicles (see Figure 2.1). This model has proved (see [45]) to be the most efficient logistic organization. Nevertheless, one of the main weaknesses is that the second part

of the problem has to deal with traffic conditions and, in general, it is much more inefficient than the first part. In this work, we consider an innovative way of performing last mile logistics, in particular we consider crowd-shipping. This new delivery method has positive environmental effects because it uses movement that the population would have done anyway. Furthermore, these new methods reduce congestions, traffic and a pollutions.

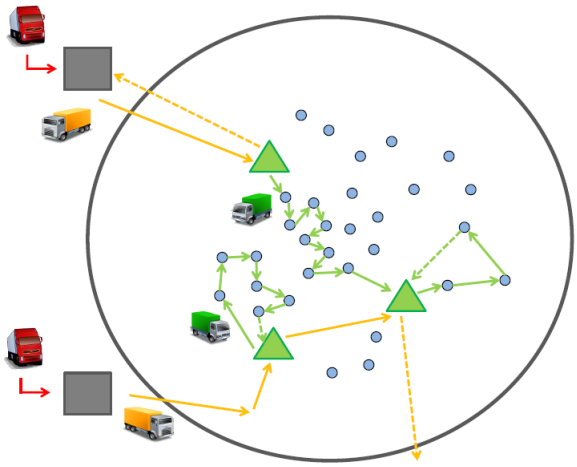

Figure 2.1 A representation of the two-echelon distribution taken from [34] with city-freighter vehicles (green vehicles). The yellow vehicles are the urban vehicles that bring freight to satellite facilities (green triangles) from UCCs (black squares) around the city. Dotted lines mean that vehicles are empty.

This method has been implemented in Padova (Italy) and in Brema (Germany) where companies have built urban consolidation centers. According to [45] this strategy has reduced by the 68% the emissions of greenhouse gas (GHG). Similar results were also found by [98], in the paper, the authors consider the city of Nijmegen (Netherlands) and they prove that with the urban consolidation centers the retailers need fewer trucks and they cover fewer kilometres than with traditional methods.

This evolution of the delivery business models allows benefits in terms of emissions and congestions, especially when the delivery is not performed by ad-hoc vehicles.

The main cons of this new business way are the additional transshipment operations. Nevertheless, these costs are compensated by the reduction of the empty trips and by a more accurate management of the resources (e.g. a higher

use of the capacity of the vehicles). Furthermore, two-echelon distribution also needs more complex cooperation and planning decisions because it needs the integration of several players, reliable routing for all the vehicles, etc. Some of these problems can be overcome by new techniques for performing the last mile delivery (e.g. crowd-shipping cancels the needs for the planning of the vehicle routing).

Other problems are due to the lack of complete information, this information can be retrieved by the use of a new paradigm called IoT that we have introduced in Section 2.4.

Since traditional distribution is already used in several sectors, in following chapters we consider e-grocery as our case study. The reason for this choice is that it is a very demanding distribution method that has not yet been deeply studied and which optimization require a large amount of resources.

2.2 The e-grocery case study

E-grocery is a sector of the e-commerce dealing with grocery and, more in general, with eat-able goods. The birth of e-grocery dates back to the 1980, with the use of phones and electronic devices[1]. The e-grocery, as we intend it today, is born after the diffusion of the internet technologies in the last years of the 20th century and the first ones of the 21st century. Companies basically operate in this sector by adopting two possible strategies: the pure e-grocery retailing strategy and the multi-channel grocery retailing strategy [42]. In the first one, the companies adopt a new operational way that totally replaces the grocery shops. In particular, the companies use the grocery shops as distribution centers and they deliver the freight directly to the customers' houses. In this model, the companies buy the grocery from the producers and they assemble the customers' orders in the distribution centers, without the need of a network of stores. For these companies, e-grocery is a disruptive technology because it completely changes the way of conducting business.

[1] For a good description of the online shopping in the 1980 see [4].

Instead, in the multi-channel grocery retailing strategy the online purchases represent an alternative business channel. The companies that implement this kind of business do not own warehouses, but they assemble the customers' orders by collecting the goods from the existing grocery shops. In both models the companies act as online intermediary between grocery producers (or manufacturer) and customers. Despite the large number of companies entering in the business, very few of them have succeeded in remaining in the business and several of them have abandoned e-grocery [21] (due to this change, also the literature related to online food retailing decreases its production). The high failure rate is due to several factors. One of them is that the gross margin in e-grocery is lower than in other activities. For this reason, the companies in the business must reach high-efficiency levels. Further, for traditional bricks-and-mortar retailers adopting the multi-channel grocery retailing strategy, online orders picking, orders execution and home deliveries represent new costs as shown in Figure 2.2.

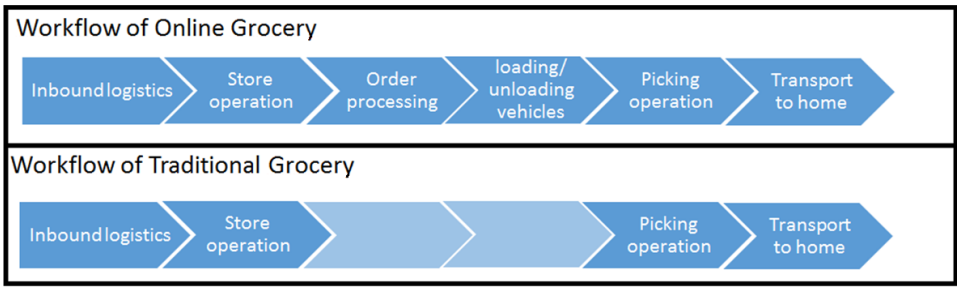

Figure 2.2 Comparison between the activities of the traditional grocery business model and the activities of the e-grocery business model.

In the traditional grocery retailing business model, the customers collect the grocery and they transport the good from the shop to their houses bearing the costs of these operations. In the e-grocery retailing business model, the company has to bear the costs of the picking and the delivery operations. From the study in [29], it emerges that the additional cost that the retailers apply for these activities is greater than the amount that the customers are willing to pay (estimated to be between 4 and 7 euros per transaction, depending on the market). From this evidence, it is possible to gather that a company, in order to make profit in the e-grocery market, must have an effective balance between services and costs. This problem is deeply analysed in [11]. Another criticality

affecting both revenues and costs is demand management. In particular, electronic demand management allows companies to directly interact with costumers and it gives the opportunity to customize in real time prices, products and offered services. For a deep analysis of this topic see [1]. Another technical problem for e-grocery companies is the difficulty to maintain perishable foods at the right food temperature all along the supply chain, until the customer obtains the package. This problem can be avoided in attended home delivery service, while it generates issues if unattended home delivery service is offered. In this last case, the only possible solution is the use of refrigerated containers.

All these possibilities produce several different packaging solutions. One possible solution is that the company installs a refrigerated reception box in the customers' garage or yard. Even if this problem is very important it is more a technological question than an operational one and for this reason we do not face it in this thesis.

All the issues that we present characterize e-grocery as a new business channel for the purchasing of food and beverages. Resuming, the major problems of this business model are the delivery in urban areas and the delivery of perishable and fresh food.

From the point of view of the literature, e-grocery has been widely analysed as an online version of supermarket. The e-grocery business models dealing only with fresh food and local food are less considered.

E-grocery has been studied in two different ways: in the first one, the authors adopt the point of view of the grocery retailer, while in the second one the authors adopt the point of view of the customer. In the first branch, some papers as [104], [12], [21] and [48] investigate on efficient operational models by comparing different existing e-grocery companies. All these articles see in the picking and delivery operations the key issues of the business and operational models. In particular, [104] proposes a framework for designing a successful delivery model, we present the main points of the framework in Figure 2.3. The interested reader can refer to [104] for more details.

In the rest of this section we follow the framework of [104] for the review of the possible methods. We enrich this dissertation with considerations from other papers. As previously said, they can be performed in-store, at local fulfilment

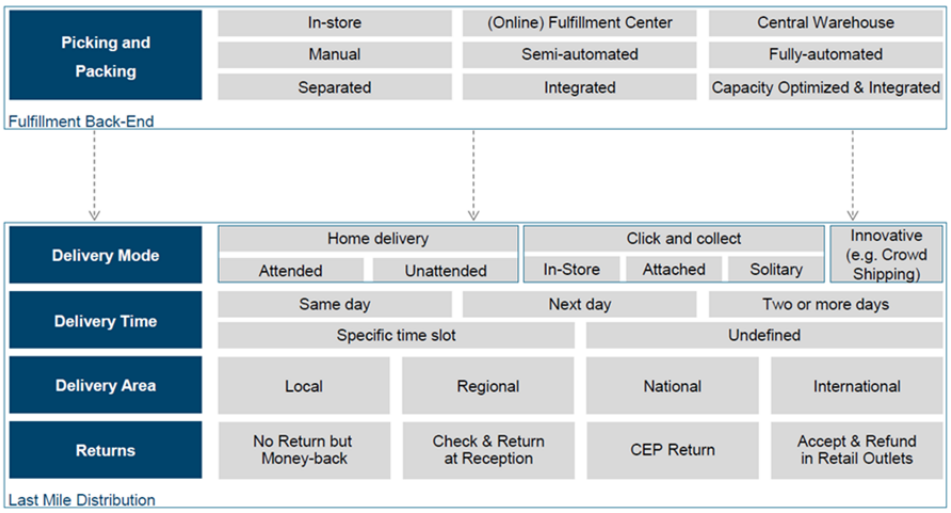

Figure 2.3 Conceptual framework for fulfilment and delivery in online grocery retailing [104].

centers (also called dark-stores) or in centralized warehouses. Furthermore, the picking operations of the online orders can be done in a manual or in automated way. Automation improves the performance of the picking activities from a speed of 80 items per hour produced in a manual way to 300 items per hour (see [1]). The main disadvantage of highly automated picking activities is that they require high initial investments, hence they have higher break-even points. Investments in automation, if not properly justified by a sufficient volume of sales, were one of the weaknesses that caused failures of the pioneering companies dealing with e-grocery. See [48] for more details.

Traditional retailers that want to use e-grocery as a new selling channel usually choose in-store picking. This choice does not need investments in new facilities. Further, the traditional retailers can exploit a network of traditional supermarkets (see[48]). The main problem of in-store picking is that the traditional stores layout is not designed in order to improve picking efficiency. For this reason, [104] claims that the picking operations are more expensive in conventional stores than in ad-hoc centers.

The picking activities can be performed by using local fulfilment centers or by using centralized warehouses. The local fulfilment centers are proximity supermarkets where retailers perform the picking activities and from where the local deliveries start. In this case, the automation of picking activities and the

layout of the freight can improve efficiency, decrease labour cost and optimize the space utilization (see [48]). A very important advantage of local fulfilment centers is a short distance between these centers and the customers' houses. This reduces transportation costs and improves the performance of the time delivery. The drawback of this choice is the cost to build the new center.

The alternatives to local fulfilment centers are centralized warehouses; they are bigger than local fulfilment centers and they serve an entire regional area. Due to the high number of orders, the centralized warehouses perform orders picking and packing in an automated or semi-automated way. As before, investments in automation are profitable only if the demand is stable and sufficiently high. If this is not the case, these investments can cause poor capacity use [48]. Further, the choice to use a single warehouse increases the delivery costs with respect to a distributed network because it increases the average distance between the customers and the depots. Finally, the high initial costs and the difficulties in the management produce a high break-even point and involve high financial risks [104].

Following [104], the delivery of the freight is the second axis of analysis. It can be characterized by delivery mode, delivery time and delivery area. We identify three main delivery types: home delivery, click & collect and innovative modes.

Definition 1. *We say that a retailer uses home delivery service if it ships freight directly to the customers' houses by using specialized drivers.*

The delivery can be done in two ways: attended home delivery or unattended home delivery. In the first one, the company delivers the freight in a fixed place during a fixed time window in order to give it directly to the consumers. While clients prefer narrow time windows, for the company this implies costs because it limits the flexibility of the delivery plans [1]. In the literature, [48] sees home delivery as a cost with a large impact in the e-grocery business. This cost can be reduced by using unattended home delivery. With this kind of delivery, the company can deliver the freights without the presence of the customer. The main issue of this strategy is that if the order contains fresh vegetables the packages must be mantained at the right temperature. As [104] reported, another issue of this strategy is that it needs high costs of investment in order

to build the infrastructure. Nevertheless, despite these issues [48] and [75] see unattended reception as the best solution for brick-and-mortar-based e-grocers because of the lower cost of delivery. Further advantage that [48] and [75] identify is that unattended reception increases the possibility of achieving more stable demand. In the paper [75], the authors prove, by means of simulations, that unattended home delivery solutions with the customer having specific reception box enable up to the 60% cost reduction compared with attended home delivery.

Definition 2. *We say that the retailer adopts click & collect service if the customers order the freight online and they pick up the freight in a particular location (it can be the store or another location).*

Hence, while in home delivery services the company bears the cost of the delivery service in the click & collect service the customer bears this cost. In [21], the authors analyse the pure e-grocery and the multi-channel models and suggest guidelines for the German market. The authors claim that for pure e-grocery models the best combination is to use a centralized distribution center and to deliver orders directly to the customers' houses. Further, due to the characteristics of the model the company can adopt dynamic pricing strategy. On the other side, multi-channel models are best suited to have click & collect service with designated areas. In this work, the authors also present different potential users groups. In particular, they suggest that the targets of pure e-grocery retailers are people between 25-44 years old while the targets of multi-channel retailer are older people.

Other modes of delivery not yet widely used are drones and crowd-shipping. Even if they are not yet widespread, they have big potential. The main advantage of the delivery by drones is that it is not influenced by traffic conditions, while the main problem is the regulation. Instead, crowd-shipping is a new paradigm suggesting to the companies to use customers that come by car to the shop to deliver the orders to other customers. This solution alone cannot be the only delivery strategy of the company but it can be proved (see Chapter 4 and Chapter5) that it is an effective way to reduce the total operational cost. Finally, an important component in the customer perception of the service is the delivery time. Same day delivery or next day delivery

increase customer satisfaction but they are difficult to organize in term of costs and planning (see [104]).

The article [104] analyses the strategy of different companies such as Colruyt (Belgium); SOK (Finland); Carrefour, Chronodrive, Intermaché and Systéme U (France); Real (Germany); Coop@home, LeShop, Migros (Switzerland); Albertsons, EfoodDepot, Freshdirect, Netgrocer, Peapod, Publix, Streamline and Webvan (USA); Asda, Leclerc, Ocado, Sainsbury's, Tesco, Waitrose (UK).

Of the aforementioned companies, the pure e-grocery companies are EfoodDepot, Freshdirect, LeShop, Netgrocer, Ocado and Peapod. They all have a central distribution center and do home delivery service. Asda is the only company of the list that uses in-store picking and home delivery service, while Chronodrive is the only example of company offering click & collect service with drive-in service, while picking orders from a central distribution warehouse.

Examples of multi-channel grocery are Albertsons, Publix and Sainsbury's. In particular, in 2014, Albertsons and Publix switch from home delivery to click & collect. Instead, Sainsbury's offers both home delivery and click & collect service. Finally, Tesco started as a pure multi-channel e-grocery company (by using the existing store network and picked orders in-store). Then, as soon as the average demand was sufficiently high, Tesco started to pick online orders in given regional fulfilment centers.

Several companies in the business fail, two of them are Streamline and Webvan. In particular, Webvan offered attended home delivery with one hour time window and it used highly automated centralized distribution warehouse. It is an example of the evidence that the high costs of automation, if not properly combined to a sufficient level of demand constitute a risk. In article [48], the authors report that Webvan went into bankruptcy while trying to reduce delivery costs, due to an expected drop-off in incoming orders. The same destiny was followed by Streamline. It adopted unattended home delivery using reception boxes with three-temperatures units. In article [17], the authors affirm that Streamline was able to built a cost-effective delivery model, nevertheless, the volume of the demand was not sufficiently high to justify the investment.

While the first branch of the literature considers the grocery retailers point of view, the second branch of the literature considers costumer preferences.

Article [67] belongs to this branch of the literature. In this work, the authors consider 243 US users of e-grocery services. They claim that the 70% of the users are women younger than 55 years old and their reasons for buying groceries online are convenience and saving time, another 15% are women with the same age but their reasons for using e-grocery are physical issues. In [77], the authors interview seven focus groups (four of them were in the UK while the others three were in Denmark). Their results agree with the ones found in [67]: they confirm that the main advantages of e-grocery are low prices and a wide product choice. In article [77] the authors find out that some disadvantages of the e-grocery are the lost of the recreational aspect of standard grocery shopping and the risk of receiving freight of inferior quality. The study in [44] suggests to the e-grocery companies to focus their advertising on the time gained by using the service because they find that travel times worry customers in the same way as delivery fees do. In article [38], the authors interview 1058 consumers and they divide them into three groups: standard consumers (people that have not bought anything on the Internet), online shoppers (people that have not yet tried e-grocery) and general online shoppers (people that have bought grocery on the Internet). From the survey, it emerges that for the group of standard consumers, e-grocery companies have to emphasize that online purchases do not revolution shopping habits. Instead, the group of online shoppers cares more about the effectiveness of e-grocery and the high degree of self-control. While, for the group of general online shoppers, e-grocery companies must focus on reducing the complexity of the service and on finding a social approved shopping mode. In article [29] the authors analyse the reasons that cause some people using e-grocery to stop. The authors claim that the main problems are reduced assortments, higher prices and additional fees. Furthermore, in the study the authors claim that users of e-grocery service prefer home delivery, while non-users prefer click & collect.

An important aspect for the last mile logistics is the environmental impact of urban freight distribution. In article [12], the authors analyse the impact of last-mile delivery models adopted by e-grocery companies. From this study, the most favourable service strategy is click & collect because it produces 9% fewer emissions of green gases than the home delivery service. The same percentage is the minimal percentage of emissions that can be saved by crowd-shipping.

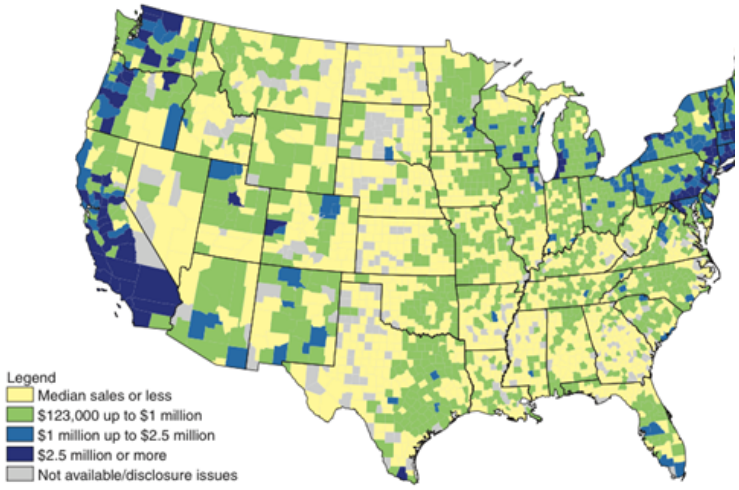

Figure 2.4 Value of direct-to-consumer sales in 2007 [96].

Nowadays, e-grocery represents an additional channel for food distribution. In article [68], the authors report that in the United States (US) discounters, hypermarkets and supermarkets own the 84% of the sales in the grocery sector. The main examples of companies in the business are Walmart and Publix. They are two of the the top five traditional grocery retailers in the US (Minister of Agriculture and Agri-Food Canada, 2013) that started to offer e-grocery service to their consumers. They also started innovative delivery methods such as crowd-shipping. In the US market, Peapod and Fresh Direct are pure e-grocery companies selling grocery online. Exactly as in US, in the European Union (UE) the biggest e-grocery retailers are supermarkets, one of the most important is Tesco.

A sub-market of e-grocey is the retailing of local food. In the US as well as in EU, the number of customers interested in this business is increased (see [68]). In particular, in 2007 local food purchases directly from producers to consumers accounted for more than 2.5 milion. In particular, the market was developed in the Northeast, on the West Coast, and around a few isolated metropolitan areas see Figure 2.4. Since 2007, the quantity of land in the farm diminishes in 31 states, while it increases in the areas where local food purchases were developed as Figure 2.5 shows.

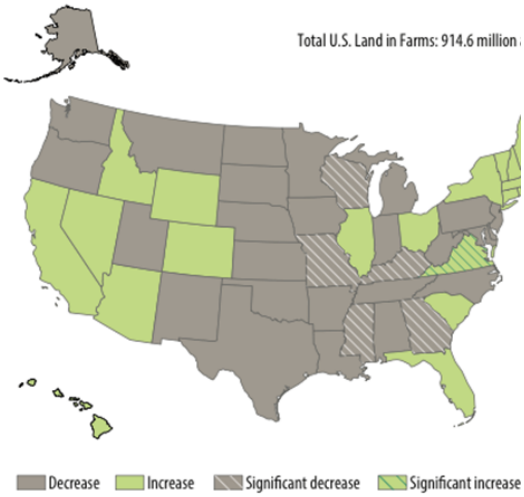

Figure 2.5 Change in Land in Farms, 2007 to 2012 [97].

The main difference between US and EU is that in Europe people are more sensitive about local food and short supply chain in the food sector and also European Commission highlights the importance of rural development[2]. Furthermore, in the Rural Development Programmes 2007-2013, short supply chains and agricultural local products are supported by several European projects and in the following program i.e. Rural Development Programmes 2014-2020, the goal is to integrate short food supply chains in the regulation. In the programmes, the European Commission defines short food supply chain as *a supply chain involving a limited number of economic operators committed to co-operation and to local economic development, and involving close geographical and social relations between producers and consumers.*

In article [56], the authors identify the main characteristics of local food in farm stores, roadside stands and community-supported agriculture arrangements. In particular, they point out that even if local food supply chain can gain a lot from the online market, to our knowledge, there is not yet such a service.

In the articles [59] and [78], the authors identify three types of short food supply chain, each one of them characterized by a different type of interface between the producers and consumers. The categories are the following:

[2] http://ec.europa.eu/agriculture/promotion/index_en.htm

- Face-to-face: customers buy freight directly from the producer using face-to-face relation.
- Spatial proximity: freight is produced and delivered in the specific region of production. Consumers are aware of the 'local' nature of the products at the shop. Specialist retailers (as delicatessens, bakeries, butchers, grocers) and the hospitality industry (as restaurants, pubs, hotels and other accommodation) that sell local foods belong to this class.
- Spatially extended: customers purchase the freight outside the region of production and they may have no personal experience of that region. Nevertheless, all the information about the places and processes of production are communicated to the consumers.

In all the classes of short food supply chain, the information about the product, the region of production, the branding and the certification are legislatively certified. According to article [81], a limitation of this supply chain is the low reliance on the Internet. Furthermore, article [52] shows that very few local food producers have their website and a proper marketing strategy. Finally, article [75] identifies the causes of this lack in the inappropriate control of the information flow, as well as the physical logistics connected to the delivery. Hence, the integration of ICT in the LFSC could improve the business and it can help small local producers to be more competitive.

2.3 Social Engagement and Crowd-shipping

In this section, we analyse the phenomenon of social engagement and in particular crowd-shipping. A business model is said to use social engagement if it uses technologies in order to ask to a large group of people to do some tasks for reaching a business goal. Crowd-shipping or crowdsourced delivery is the application of the social engagement principle applied to the delivery of goods.

While for e-grocery we have several companies in the business, the major users of social engagement strategies and crowd-shipping are still startups. The birth of these services is due to the diffusion of technologies by means

of smart-phones and internet connections. This trend opens new business channels by giving the opportunity to the companies to communicate directly to the costumers without relying on intermediaries. Further opportunity of these communication channels is to match demand and offer for services or goods directly between people. Moreover, as the importance of these channels is growing, some of these business models are becoming real alternatives to traditional models.

The economic literature related to social engagement is wide and there are different implementations of this concept. For this reason, we discriminate these implementations by using the definitions proposed by Botsman and Rogers [10].

Definition 3. *We call* collaborative economy *an economic system of decentralized networks and marketplaces that unlocks the value of underused assets by matching needs and haves, in ways that bypass traditional middlemen.*

Examples of companies implementing this business model are Etsy, Kickstarter, Vandebron, LendingClub, Quirky, Transferwise, Taskrabbit. The main examples of collaborative economy are short food supply chain and local food supply chain.

Definition 4. *We call* sharing economy *an economic system based on sharing underused assets or services, for free or for a fee, directly from individuals.*

Examples of companies implementing this business model are Airbnb, Cohealo, BlaBlaCar, JustPark, Skillshare, RelayRides, Landshare. This phenomenon is relatively new; *the term sharing economy* was introduced into the Oxford English Dictionary only in 2015. The dictionary says that sharing economy is the *economic system in which assets or services are shared between private individuals, either free or for a fee, typically by means of the Internet.* Botsman and Rogers identify 5 characteristics of the sharing economy. They are:

1. The core business idea involves unlocking the value of unused or underutilized assets (called *idling capacity*) whether it is for monetary or non-monetary benefits.

2. The company should have a clear values-driven mission and it must be built on meaningful principles including transparency, humanness, and authenticity that inform short and long-term strategic decisions.

3. The providers should be valued, respected, and empowered and the companies committed to making the lives of these providers economically and socially better.

4. The customers should benefit from the ability to get goods and services in more efficient ways: they pay for access instead of ownership.

5. The business should be built on distributed marketplaces or decentralized networks that create a sense of belonging, collective accountability and mutual benefit through the community they build.

Definition 5. *We call* collaborative consumption *the reinvention of traditional market behaviours-renting, lending, swapping, sharing, bartering, gifting-through technology, taking place in ways and on a scale not possible before the internet.*

Examples of companies implementing this business model are Zopa, Zipcar, Yerdle, Getable, ThredUp, Freecycle, eBay

Definition 6. *We call* On-Demand Services *platforms that directly match customer needs with providers to immediately deliver goods and services.*

Examples of companies implementing this business model are Instacart, Uber, Washio, Shuttlecook, DeskBeers, WunWun.

As the reader can notice, the difference between the first two definitions is that the second one considers only relationship between individuals without considering companies. Crowd-shipping can be identified as an example of collaborative economy or sharing economy in relation to the way in which it is implemented. In this thesis, we consider as case study a company using crowd-shipping in order to deliver freight. For this reason, we consider crowd-shipping to be an example of collaborative economy.

From the technical point of view, crowd-shipping can be implemented in two ways: the company can ask to one single person to do all the path from the

pick-up to the delivery or it can ask to several people to perform little parts of the path. This second way has security issues and needs a very strong control infrastructure that increases the costs and diminishes the advantages. These considerations were verified by the DHL project "bring.BUDDY" (see [102]). The tested transportation model consists in asking to smart-phone users to carry a parcel from a place to another place (a pub, a shop, etc) and leave it to the next person that would pick it along. This process is repeated until the parcel reaches the final destination. Due to the aforementioned security problems, the project was abandoned and for this reason we do not consider this business model, instead we focus on the single person delivery crowd-shipping.

This choice has several advantages: it does not require to buy a fleet of vehicles, to hire drivers, to buy fuel and to bear all the costs that traditional logistics' companies have to pay. For these reasons, crowd-shipping is cheaper than Postal Service, FedEx, and UPS. Furthermore, using crowd-shipping has also positive marketing effects for the company because more people become aware of the brand.

The benefits arising form crowd-shipping have attracted several start ups in the business, some examples are cargomatic, crowddelivery, deliv, entrusters, hitchit, jib, meemeep, nimber, packmule, piggybee, postrope, shipizy and signup.tinycarrier. Furthermore, the phenomeon is so important that even big players such as DHL and Walmart are trying to enter the business.

DHL implements a project called MyWays in Stockholm (in addition to bring.BUDDY). The project consists in an application that gives to online customers the possibility to communicate to all app users a time-window, a place and a delivery fee that they want to pay for the home delivery. The application sends this information to all the app users that can decide whether to do the delivery for a reward or not.

Instead, Walmart asks to in-store shoppers to deliver packages to online shoppers in exchange for a discount. This delivery method has the advantages to be faster than other methods and to provide same-day delivery. This application is justified by the fact that the bulk of the potential customers of the big retailers are within 5 miles from the store. Hence, by using crowd-

shipping, Walmart can turn its vast network of physical stores into distribution hubs for online customers.

These applications are still in their early stages. Nevertheless, as they mature they may have profound implications for the delivery industry.

2.3.1 Other Collaborative Economy Applications

By embracing the definition provided in the previous paragraph, other applications have to be considered. In particular, the well known line of research devoted to horizontal and vertical cooperation in vehicle routing. Indeed, collaboration is widely seen as one of the best ways to deal with increasingly complex business sectors (see [26] and [88]). In particular, companies may collaborate in goods procurement and distribution at different levels of the supply chain. Vertical collaboration is characterized by interactions among different levels of the supply chain, whereas horizontal collaboration is achieved through the cooperation of companies (even competing ones) at the same level of the supply chain (e.g. carriers). Horizontal collaboration is very well known in ocean shipping and air transport literature, whereas the application to road routing is more recent (see [16] and [99] for surveys). Collaborative transportation aims at reducing delivery times and costs by exploiting economies of scale. Most of articles refer to carrier alliances and cooperation where customers and orders are shared or exchanged within joint routes planning (see, for instance, [25] and [55]), but some work consider the sharing of other logistic assets such as vehicle capacity and depots (see [18]).

2.4 IoT for Smart-Cities

IoT is the enrichment of devices with sensors and with the capacity of exchange data. If these devices contain actuators, the possible applications encompass an enormous amount of applications (see Figure 2.6 for a quantitative estimation of the size and the market impact of IoT). The capability of collecting data is really important especially for systems characterized by high complexity, where information improves the performance and reduce costs.

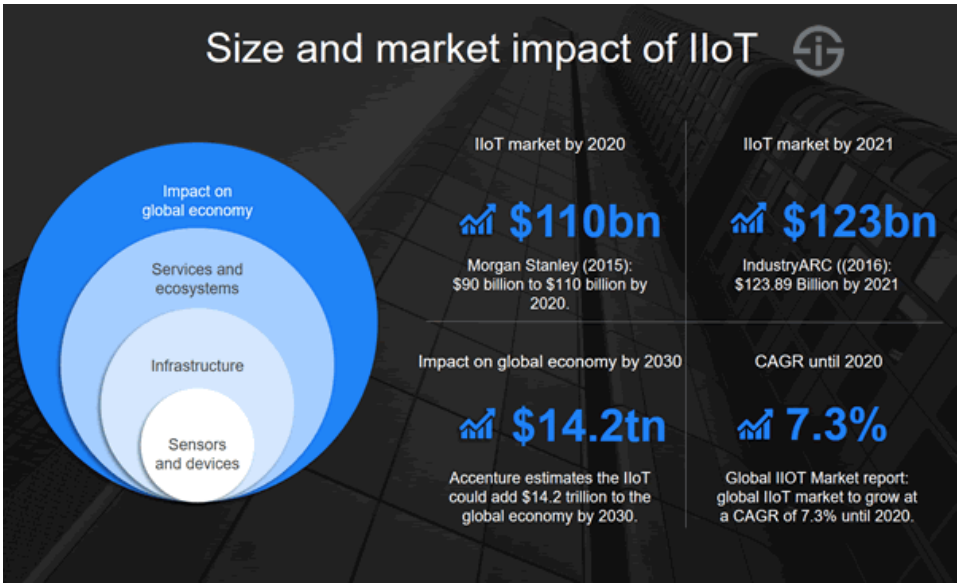

Figure 2.6 Size and market impact of IoT source: http://www.morganstanley.com/ideas/industrial-internet-of-things-and-automation-robotics.

In the context of Smart-Cities several applications are possible:

- Urban logistics, public services and Transport (vehicle localization, monitoring of people at the bus stations, monitoring of waste production, etc)
- Traffic network and management (parking management, update traffic information, public illuminations, etc).
- Environmental monitoring (measurement of the pollution in air or water, surveillance, etc.)
- Monitoring of the environment (prevention of flood, fire, landslide, etc)
- Entertainment and tourism services (information, smart-poster, cultural itinerary, etc)

In this thesis we consider the application of this paradigm in order to gather information useful for the topics related to urban logistics, public service and transportation. Since studies related to the urban logistics already exist, see for example [13] (where the authors propose an intelligent transportation system

that uses information from a network of sensors to improve route planning) we focus on waste collection. In particular, by using the data from the IoT components of the vehicles and dumpsters, we develop a statistical model of the waste growth and we use it in order to enhance the waste collection operations. If we consider the city as a network, then the same model can be applied to solve problems such that the optimization of the transportation of the freight from some points to another one fixed.

Even if IoT seems to be a possible solution to several problems it has two main weakness: the channel capacity for the information and the network coverage. Consider a city where each dumpster, each traffic light, each bus station is able to collect and to transmit information, in this context the ammount of information would be too big to be manageable. Furthermore, cover an entire city with a network infrastructure able to manage all these data is expensive. It is for these reasons that other kinds of networks such as opportunistic networks are studied. We do not consider explicitly this business model because it is equal to the crowd-shipping one.

2.5 General Problems

The literature about sharing economy, collaborative consumption, collaborative economy is wide. Nevertheless, to our knowledge there are not articles dealing with optimization and related to these topics. The main reason is that several companies involved in this business only provide a internet platform to match people requests and availabilities. In the following, we consider a company that uses people in order to do general tasks. Our goal is to use the minimum amount of rewards while convincing people to do all the tasks. This problem can be applied in several very different situations. Nevertheless, we consider as case study crowd-shipping because of the general interest rising with respect to the topic.

The following chapters consider applications similar to the aforementioned project of Walmart. The only difference is that we consider that the company has also a professional delivery service that is used in order to perform the deliveries that the crowd-shipping service does not manage to do. Note that

this solution can be used by all the companies independently by the way that the picking activities are done. Furthermore, we consider as principal users of this service the companies that have local fulfilment centers (or a network of small markets). We focus on this choice because a company that has a centralized warehouses is a particular type of the previous problem.

This problem may seem uncorrelated with the problem of the IoT but it is not the case. In fact, the main problem of IoT is the collection of the data from the sensors. In some cases, as the IoT installation in the vehicles for the waste collection, it is easy to get the data because at the end of each time shifts the vehicles have to arrive to the depot where they can exchange information with the central unit. For sensors installed in the urban context, this is not the case. For them, the only solution for these devices is to build a low-cost temporary IoT network: the so-called opportunistic IoT (o-IoT) networks. This business model tries to solve the problem of gather data from a distributed network of sensors in an urban area by using as mobile hotspots the devices of selected users. This application is critical because without the proposed opportunistic connections, retrive the data from these devices requires a large network infrastructure able to cover the whole city. O-IoT inspires the Coiote project by TIM (the largest Italian telecommunication company). The goal of this project is to develop a mobile phone application enabling TIM to ask users to do some tasks in relation to the mobile phone cell where they are located. The task that the users are asked to do is to share their mobile phone connection with smart-dumpsters. In this way, the smart-dumpsters can transmit to the central unit the data related to the amount of waste that they have collected and the company in charge of the waste collection can plan the operations in an optimal way. This architecture is shown in Figure 2.7.

One of the areas where IoT is set to make big impact is the online grocery retail sector. This comes at a time when more consumers are starting to understand the benefits of shopping online. However, there are several areas where IoT is helping to improve efficiency, reduce waste, and enhance the shopping experience for the customers. One example is the Ocado Smart-Platform (OSP) that Ocado builds with the Cambridge Consultants that manages a fleet of robots for delivering customers' groceries. The software

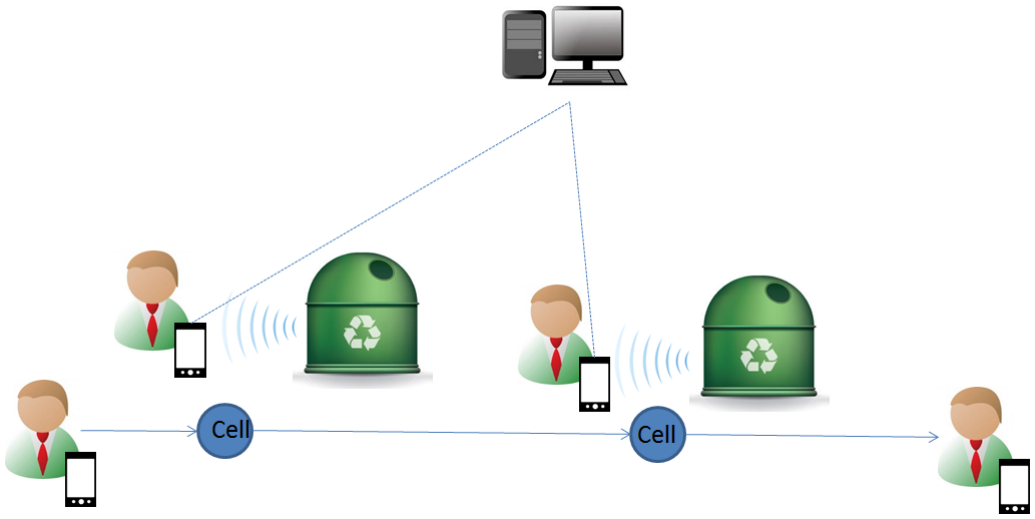

Figure 2.7 The figure shows how the Coiote project work

also manages a large fleet of vans to deliver orders from Customer Fulfilment Centres (CFCs) to Ocado customers who purchase their groceries online. In order to manage this fleet efficiently, the vans are equipped with several IoT sensors gathering valuable information such as the vehicle's location, speed, engine conditions and fuel consumption.

Figure 2.8 The figure shows some vehicles equipped with the IoT sensors http://ocadotechnology.com/blog/how-the-internet-of-things-is-changing-grocery-retail/

Part I

Models for Urban Logistics

Optimization of waste management: a case study

3.1 Introduction

One of the industries that operates in the urban logistics field is the waste collection. As aforementioned, this sector is relevant and its importance is increasing as cities growth. In this chapter, we consider the problem of routing and scheduling the waste collection operations as a case study for the use of IoT information in the context of smart cities and city logistics.

The development of this algorithm was possible in the ONDE-UWC project which aims to use in a innovative way the data generated by the waste collection activities. The innovative use of the data is not only related to the optimization phase (that reduces the total operating and environmental costs) but it is also related to the establishment of the data for the citizenship, in order for them to be aware of the costs of the waste collection and to participate in an active way to these operations by signalling if the intervention of the company is needed (e.g. when the waste is outside the dumpster). In particular, the main objectives of the project are to test a new methodology for reducing the time spent in the definition of the optimization problem; to enhance collaboration between operational actors in order to ensure the satisfaction of different stakeholders, such as public administrations, companies involved in the waste collection, citizenship, etc; to improve the awareness of citizens regarding the waste management.

In particular, we consider the activity of the company Cidiu S.p.A. (Centro Intercomunale di Igiene Urbana S.p.A.).

Cidiu S.p.A. main activity is to collect the waste from the municipalities and transport it to the dumps. It operates in the municipalities of Alpignano, Buttigliera Alta, Collegno, Druento, Grugliasco, Pianezza, Rivoli, Rosta, San Gillio, and Venaria Reale i.e. the regions shown in Figure 3.1 and it collects waste of five types: paper, solid urban refuse, plastic, metallic materials, and glass.

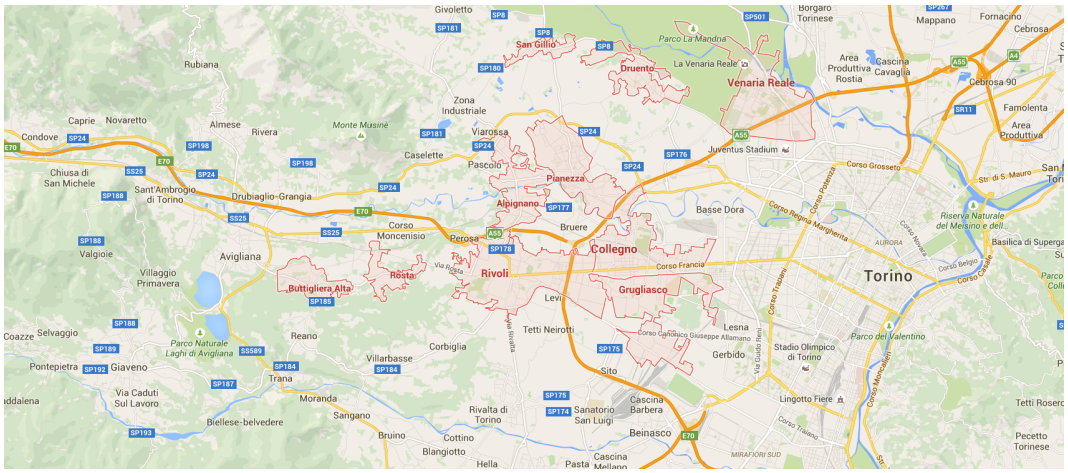

Figure 3.1 Municipalities served by Cidiu S.p.A.

Cidiu S.p.A. has two independent headquarters: one located in Rivoli and one located in Collegno. Each one of them is responsible for the collection of waste in different cities: the first is responsible for the municipalities of Alpignano, Buttigliera Alta, Pianezza, Rivoli, Rosta and San Gillio while the second is responsible for the municipalities of Collegno, Druento, Grugliasco and Venaria Reale. The collection operations are organized in time shifts (in the following shifts) i.e. periods of time during which the vehicles are dedicated only to the waste collection. The collection is performed by using one driver for each vehicle and each vehicle, during one time shift, can only collect one type of waste. The cost of each time shift is related to the salary of the driver, to the use of the vehicle and also to the opportunity cost of the vehicle (i.e. a vehicle used for a collection cannot be used for performing other operation in the same time). In particular, each day it is possible to use three time shifts (from 6:00 to 12:00, from 13:00-19:00 and from 21:00 to 03:00), the cost of

this last time shift is greater than the cost of the cost of the others due to the increased cost of the driver.

The fleet of vehicles of the company is composed by new generation trucks, capable of measuring the weight of the refuse collected, each time a dumpster is voided. This information has an enormous value for the business because it can be used to monitor the waste growth and to build a forecasting model for waste production.

The flow of operations of a vehicle is the following:

1. start form the headquarters,
2. collect the waste from the chosen dumpsters,
3. go to the dump and unload the collected waste,
4. return to the starting headquarters.

The activities of the company have to satisfy constraints imposed by two actors: the dumps' managements and the local administrations.

Cidiu S.p.A. and the dump managements have contracts related to the quantity of waste that is yearly acceptable by the plant. It is important to note that, in general, the dumps treat only a specific kind of waste.

Cidiu S.p.A. has also contracts with local administrations about the service level i.e. the emptying frequency. These contracts limit the operational freedom of the company. Nevertheless, Cidiu S.p.A. is planning to discuss the contract in order to leave the periodic constraints.

Due to the importance of the sector, the optimization of these activities creates several gains (environmental and economical) for the whole community. In this Chapter, we propose a method in order to optimize these activities developed in the context of the project ONDE-UWC [69] (founded by the Regional Council of Piedmont). The two main innovations of the project are the application of IoT in waste management and the application of the GUEST OR methodology.

The first innovative element is the information technology (IT) infrastructure which allows greater flexibility in the services by considering all types of waste

(as in [57, 80]) and by considering aperiodic routes. This is possible by using the information from historical data collected by on-vehicle weight systems and by a network of sensors built according to the IoT paradigm. The data and the model built are then used by the math-heuristic and all the information generated by the processes are sent to the municipalities and to the citizens (with the aim of improving the awareness about the performed activities).

The second innovative element, the GUEST OR methodology is a methodology of lean business dedicated to the fast development of an optimization model and to avoid possible error about the goal of the company. This methodology is a customized version of the GUEST methodology (see [71]), first introduced in [91, 72]. In order to apply this methodology, we have to define the waste collection scenario. It is defined as a multi-actor complex system describing all the stakeholders that have some influence in the process. The use of this methodology fills the gap (both in literature and in the real field) between the business perspective and the technical solutions proposed. This gap is due to the difficulties in the communications between stakeholders that generally have very different backgrounds. To fill the gap, the GUEST methodology creates a knowledge base of the needs and the values of the different stakeholders involved. This knowledge based is represented by the definition of particular canvas (see [72]) as a means of communication between the stakeholders of the company. Since the topic is not central in our thesis, we refer to [72] for further information. This chapter is organized as follows, in Section 3.3 we describe the framework of the company by using the GUEST methodology. In Section 3.4 we describe the technological architecture of the final optimization system, in Section 3.5 we describe the mathematical model and we propose a heuristic for finding a solution. Finally, in Section 3.6 we report the results of the computational experiments both in the simulations and in the real field.

3.2 IoT applications for Urban Logistics and waste collection

IoT has several domains of applications, some of them are smart-grids, smart-cities and intelligent transportation (for a survey of these applications the user

is referred to [49], [106] and [76]). In the general framework of IoT, optimization is very important. Examples of this statement can be found in [13], where the authors develop an intelligent transportation system that uses data from a network of sensors in order to compute the a heuristic routing. In this thesis, we focus on the IoT supporting the waste collection operations and for that reason, we continue this thesis by considering this stream of literature.

The collection of waste in urban areas is a difficult problem for municipalities and for the companies involved both in term of organization and in term of treatmet activities. Several authors in the operation research field consider the problem of waste collection. The authors of [32] split the literature about waste collection into two branches: one considering the time horizon of the optimization and one considering the decision making process. These two axis define three types of decisions: strategic, tactical, and operational.

Strategic decisions affect the long term evolution, tactical decisions affect short range plan and operational decisions strategic goals and objectives to tactical goals and objectives. Some examples of these decisions are the choice of the locations of the treatment sites and dumpsters, the composition of the fleet of vehicles, the choice of the technologies for treating the waste, the selection of the time shifts to use to collect waste, and the routing and the scheduling of the collection.

Of these decisions, the strategic decisions are the one related to dumpsters locations, to the construction of new facilities and the choice of the vehicles characteristics. While the tactical decisions are related to the daily activities such as when to start the waste collection, which dumpsters to visit, etc. From the point of view of the tactical field, papers as [80] and [19] compute the optimal routing, others such as [39] and [57] try to define optimal periodic routes and, for this reason, they try to define the optimal emptying frequency for each dumpster. Another important study in the field of waste optimization is [5]. In this work the authors consider an approach for scheduling the multi-period collection of recyclable materials. They investigate the conditions under which how the scheduling of emptying and transportation minimizes the operation cost, while providing high service level and ensuring that capacity constraints are not violated. They develop a heuristic composed by two steps:

in the first one they solve the problem by using a construction heuristic, in the second step the solution is re-optimize every subsequent period with a variable neighborhood search. In this study also the uncertainty about the waste production is taken into account. Finally others as [14] and others try to solve the periodic capacitated arc routing problem.

In Chapter 3, we consider the tactical problem of scheduling and routing waste collection without imposing the periodicity of the routing. This choice enlarges the solution space and produces better solutions. Furthermore, it produces more adaptive solutions to seasonal changes and to missed collections, nevertheless, this choice increases the difficulty of solving the model. In [65] the authors describe an application similar to ours, nevertheless they use the data collected from dumpsters in order to describe an inventory routing problem and to develop a heuristic to solve it. Nevertheless, even if the problem considered by [65] is similar to our (both the applications optimize the waste collection by integrating the use of sensors) there are several differences: [65] considers a fleet of homogeneous vehicles while in the proposed approach the vehicles can be different and the cost measurements of the two approaches are different.

In the literature there are also problems similar to the one of the waste collection coming from other applications. Two examples are [54] and [20]. In these works the authors consider the dairy transportation problem (DTP) i.e. a problem aiming at finding the optimal routing and scheduling of vehicles collecting milk from farms. Also in this problem the uncertainty about production is a factor strongly influencing the problem. In the first study the authors develop a generalized tabu search algorithm that integrates the different characteristics of the DTP while in the second paper the problem is enriched by considering a fleet of heterogeneous vehicles, multiple depots, and several resource constraints. Then it is solved by a branch-and-price methodology.

3.3 Framework Analysis

By using the data regarding the waste collection, we describe the project from a "as is" point of view. In this section, we describe the whole solution implemented

from the perspective of the value chain, decision process and finally we describe the operational model.

The primary decision makers are the chief operating officer (COO) and the foreman of each team of drivers. In particular, the COO has to ensure the feasibility of the plan produced by the platform, and then it forward the plan to the foreman of each group of drivers. After receiving the plan, each foreman has to take the logistics decisions (i.e. which street to use) and then, to give it to the drivers that will execute the order by collecting the waste in the considered dumpsters.

The main users of the solution computed by our approach can be divided into two groups: internal and external. The main difference between these two is that the internal users have knowledge about all the processes, while external users have only view on a subset of processes. The internal users are:

- the Cidiu S.p.A. management;
- the Cidiu S.p.A. technical staff (it includes the foremen and the drivers);
- the users of the mobile application, which permits to the drivers to exchange information and communicate problems.

The internal users are:

- the big data platform of the region;
- the citizens;
- the municipalities.

All these users are characterized by different degrees of involvement with the decision makes (e.g. the COO uses the platform to communicate the weekly plan to the foremen).

It is important to notice that the decision support system is also interacting with the big data regional platform and with the municipalities, with the goal of providing open data.

The data produced by the application are communicated by several channels, we can split them into two types:

- decision channels: they are
  - the Intranet: it is used for supporting the information flow that interconnects Cidiu S.p.A. and the workers;
  - the mobile application: it is used for supporting the information flow between the involved actors, it is based on the use of smart-objects such as tablets, smart-phones, etc;
- implementation channels: they are composed by APIs and digital reports, useful to diffuse the information related to the development and improvement of the solution.

Finally, we describe the objectives related to the solution and we divide them in two classes:

- operational goal: efficient creation of weekly shifts and consequential reduction of operational costs, minimization of total cost of the operations including environmental costs;
- tactical goal: thanks to the computation of the solution, it is possible to see the main critical resources and to perform "What if" analysis.

In order to reach these goals, we need to decide the time shift to use, for each of them the assignment vehicle - garbage type and then, to decide the routing in each used time shits.

In order to compute this solution, we need the information related to the garbage generation, the features of the shift (such as vehicle availability, time duration) and the location of each dumpsters. Furthermore, in order to implement the solution, there is the need to use the following technologies:

- geographic Information System (GIS): it permits to have information regarding the position of dumpsters;

- IoT: similarly to [65], the solution is strongly based on the IoT paradigm. The main difference between our solution and the one proposed in [65] is that they consider sensors in the underground containers while we consider vehicles equipped with sensors. This is the real innovation of our work because, by using these data, it is possible to build a statistical model of the growth of the waste production, giving insights concerning the fill levels of dumpsters at any time. The data give us the information related to the areas that produce more and, thanks to this system, it is possible to know if the collection of the waste in a dumpster fails (i.e., missed collections).

- GPS system: it is installed in the onboard units. The GPS system enables the IT solution to track the position of the vehicles. Further, it allows the company to monitor the fleet in real time;

- mobile application: it collects the warnings related to bad situations from both the citizenship and vehicle drivers.

The organization of the working activities provided by the algorithm have to satisfy several constraints in order to be implementable. These constraints can be split into two types:

- implementation constraints: they are the constraints that the solution algorithm has to satisfy. The main goal is to define a method able to provide a solution in a small amount of time. This need and the high complexity of the model imposes to implement a heuristic solution. The heuristic is composed by two steps: the first one solves the problem by considering aggregation of dumpsters called clusters and the second phase builds a feasible solution starting from the one obtained. The details of the heuristic are discussed in Subsection 3.5.1;

- problem constraints: they are the constraints that the solution must satisfy. The main constraints are:

  – each dumpster cannot contain waste for more than the 80% of its volume;

- the working day is composed by 3 shifts, each one of them is composed by 6 hours;
- each vehicle can collect only a certain type of garbage in a given time shift

3.4 Technology system

The architecture of the solution is vehicle-to-infrastructure (V2I) and it is shown in Figure 3.2. The main building blocks are:

- central unit: this is the core operative block of the IT architecture. It is responsible for the optimization and the management of the information.
- CSI database: it is responsible for containing all the data concerning the waste collection in the considered area.
- enabling technologies: they are the internal and external technologies that enhance the decision making process (e.g., providing maps, gathering data concerning the vehicles and dumpsters, etc.), and allow the exchange of information between the different components and actors. In particular, the mobile application plays an important role, allowing the drivers and customers to communicate and interact with the central unit, and by providing information about the waste collection system.
- Graphical User Interface (GUI). It allows the users of the Cidiu S.p.A. to interact with the central unit.

The central unit manages all the flow of information, it contains the optimizer and it sends the data from the optimizer to the technical board and to the CSI database. It also manages data from the waste collection coming from the vehicles, from the data transmitted by the drivers and by the customers (through mobile application). It also updates the CSI database with the data collected. When all the information is collected, the optimization algorithm is ran and the solution is sent to the technical board of Cidiu S.p.A.

The timing of the data flows are different: the CSI database is updated every time that a vehicle arrives at the headquarters, the mobile application is updated as soon as new data arrives. Finally, the optimization algorithm is ran three times every week (every two days). In this way it is possible to be responsive for possible missed collection and it is also possible to mitigate the effects of the end of the time horizon. As soon as the solutions are available they are sent to the technical board and the database of CSI is updated.

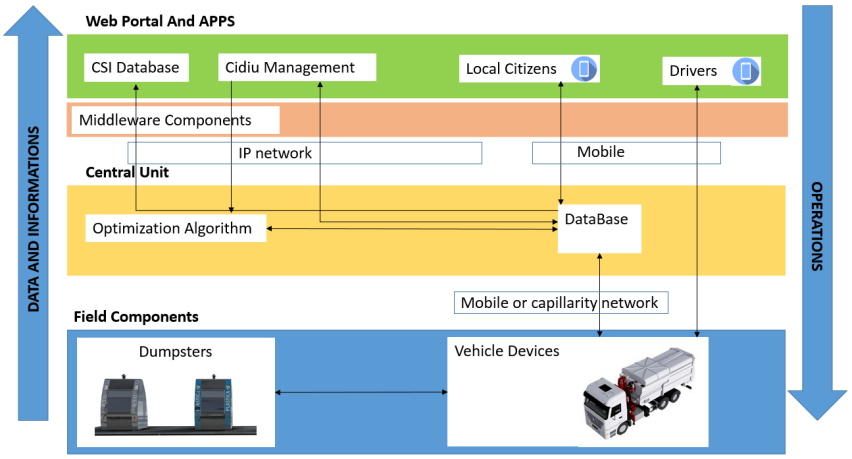

Figure 3.2 Architecture of the solution

3.5 The mathematical model

The mathematical model describing the activity of Cidiu S.p.A. is described by using the following sets:

- $\mathcal{I}$: the set of available vehicles, its cardinality is I.
- $\mathcal{S}$: the set of garbage types, its cardinality is S.
- $\mathcal{T}$: the set of time shifts, its cardinality is T.
- $\mathcal{J}$: the set of dumpsters, its cardinality is J. This set can be partitioned in the sets $\mathcal{J}^s$, each one of them containing the dumpsters collecting a particular type of waste.

For the sake of simplicity and without loss of generality, we consider that in the set of dumpster $\mathcal{J}$, $j = 1$ is the depot and $j = J - s, \ldots, J \ \forall \ s \ \in \mathcal{S}$ are the dumps. We denote the indexes of the dumps with D_s. The mathematical model is described by using the following parameters:

- c_{it} is the cost of using vehicle $i \in \mathcal{I}$ during time shift $t \in \mathcal{T}$.
- $C_{\max}$ is the maximum amount of time that a time shift can last.
- $d_{j_1 j_2}$ is the time that the vehicle uses for travelling from dumpster $j_1 \in \mathcal{J}$ to dumpster $j_2 \in \mathcal{J}$;
- $\hat{C}$ is the capacity of the vehicle;
- l_s with $s \in \mathcal{S}$ is a correction term for the capacity of the vehicle, for some kind of waste. It is used, for example, when the waste can be compressed by the press of the vehicle.
- Θ_{jt} is the estimation of the growth rate of the volume of waste in dumpster $j \in \mathcal{J}$, during time shift $t \in \mathcal{T} - 0$. In particular, Θ_{j0} is the quantity of waste in dumpster $j \in \mathcal{J}$, during the first instants of the first shift;
- a is the amount of time that drivers use to collect the waste from a dumpster.
- α is the percentage of the volume of the dumpster that the waste must not exceed.
- λ is a parameter converting the time of the routing in economical cost.
- V_j is the volume of dumpster $j \in \mathcal{J}$.

Finally, we use the following variables:

- w_{it}: binary variable, it has value one if vehicle $i \in \mathcal{I}$ is used during shift $t \in \mathcal{T}$,
- z_{ist}: binary variable, it has value one if vehicle $i \in \mathcal{I}$ collects the garbage of type $s \in \mathcal{S}$ during the shift $t \in \mathcal{T}$,

- y_{ijt}: binary variable, it has value one if vehicle $i \in \mathcal{I}$ collects the garbage of dumpster $j \in \mathcal{J}$ during the shift $t \in \mathcal{T}$,
- $r^{it}_{j_1 j_2}$: binary variable, it has value one if vehicle $i \in \mathcal{I}$ during time shift $t \in \mathcal{T}$ goes from dumpster $j_1 \in \mathcal{J}$ to dumpster $j_2 \in \mathcal{J}$,
- x_{ijt}: continuous variable describing the volume of garbage collected by vehicle $i \in \mathcal{I}$ from dumpster $j \in \mathcal{J}$ during shift $t \in \mathcal{T}$,
- V_{jt}: continuous variable describing the volume of waste present in dumpster $j \in \mathcal{J}$ at the end of time shift $t \in \mathcal{T}$

It is worth noticing that V_{jt} are decision variables, since they are influenced by the x_{ijt}. The optimization problem is then

$$\min. \sum_{i=1}^{I}\sum_{t=1}^{T} c_{it} w_{it} + \lambda \sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{j_1=1}^{J}\sum_{j_2=1}^{J} d_{j_1 j_2} r^{it}_{j_1 j_2}, \tag{3.1}$$

subject to

$$w_{it} \geq y_{ijt} \quad \forall \ \ i,j,t \tag{3.2}$$

$$My_{ijt} \geq x_{ijt} \ \ \forall\, i,j,t \tag{3.3}$$

$$\sum_{i=1}^{I} y_{ijt} \leq 1 \ \ \forall\, j,t \tag{3.4}$$

$$x_{ijt} \geq V_{jt} - V_j(1-y_{ijt}) \ \ \forall\, i,j,t \tag{3.5}$$

$$\sum_{j=1}^{J} x_{ijt} \leq \hat{C} + \sum_{s=1}^{S} l_s z_{ist} \quad \forall \ \ i,t \tag{3.6}$$

$$Mz_{ist} \geq x_{ijt} \ \ \forall\, i,s,j \in \mathcal{J}^s,t \tag{3.7}$$

$$\sum_{s=1}^{S} z_{ist} \leq 1 \ \ \forall\, i,t \tag{3.8}$$

$$V_{j0} = \Theta_{j0} \ \ \forall\, j \tag{3.9}$$

$$z_{ist} \leq y_{i(J-s)t} \quad \forall\, i,t \tag{3.10}$$

$$V_{jt} = V_{jt-1} + \Theta_{jt} - \sum_{i=1}^{I} x_{ijt} \quad \forall\, j,t \neq 0 \tag{3.11}$$

$$V_{jt} \leq \alpha V_j \quad \forall j,t \tag{3.12}$$

$$w_{it} \geq w_{i+1,t} \quad \forall\, t, i = 1 : I-1 \tag{3.13}$$

$$\sum_{j_1=1}^{J} r_{jj_1}^{it} = y_{ijt} \quad \forall\, t,i,j \tag{3.14}$$

$$\sum_{j_1=1}^{J} r_{jj_1}^{it} = \sum_{j_1=1}^{J} r_{j_1 j}^{it} \quad \forall\, t,i,j \tag{3.15}$$

$$\sum_{j=1}^{J} r_{1j}^{it} = w_{it} \quad \forall\, t,i \tag{3.16}$$

$$\sum_{j=1}^{J} r_{j1}^{it} = w_{it} \quad \forall\, t,i \tag{3.17}$$

$$\sum_{j=1}^{J} r_{jD_s}^{it} = z_{ist} \quad \forall\, t,i,s \tag{3.18}$$

$$\sum_{j_1=1}^{J} \sum_{j_2=1}^{J} d_{j_1 j_2} r_{j_1 j_2}^{it} + a \sum_{j=1}^{J} y_{ijt} \leq C_{\max} \quad \forall\, i,t \tag{3.19}$$

$$\sum_{j_1, j_2 \in S, j_1 \neq j_2} r_{j_1 j_2}^{it} \leq |S| - 1 \quad \forall S \subset \mathcal{J}, S \neq \emptyset \tag{3.20}$$

$$x_{ijt} \in \mathbb{R}^+ \quad \forall\, i,j,t$$

$$z_{ist} \in \{0,1\} \quad \forall\, i,s,t$$

$$y_{ijt} \in \{0,1\} \quad \forall\, i,j,t$$

$$r_{j_1 j_2}^{it} \in \{0,1\} \quad \forall\, i,j_1,j_2,t$$

$$V_{jt} \in \mathbb{R}^+ \quad \forall\, j,t$$

The objective function minimizes the weighted sum of the the total costs derived from the time shifts used and the total time of all the routings. It is worth noting that the cost of the use of each vehicle changes with respect to the time shift and with respect to the vehicle, in particular the variations among vehicles is explained for the heterogeneity of the fleet while the variations among time shift is due to the fact that use time shift in a different times has different economical cost. In particular the last shifts of the week are less expensive and as stated above the third shift of each day is far more expensive of the other two. Finally, the coefficient λ converts the time of the routing in cost of the total solution. The constraints (3.22) and (3.23) impose that the solution has to use a time shift in order to use a vehicle. Constraints (3.4) ensure that,

no more than one vehicle can collect the waste from a dumpster. Constraints (3.5) ensure that if a vehicle collects waste from a dumpster, then it has to void it. Constraints (3.24) impose that a vehicle cannot collect more waste than its capacity (the constraints consider that, for some types of waste, the vehicle can obtain extra capacity by using a press to reduce the volume of the waste). Constraints (3.25) and (3.26) ensure that each vehicle, in a fixed time shift can collect waste from at maximum one type of waste. Constraints (3.28) ensure that if a vehicle is used, then it has to go to the corresponding dump. Constraints (3.27) and (3.29) describe the evolution of the quantity of waste in each dumpster and constraints (3.30) limit this quantity. It is important to note that the model must react to the increase of the waste otherwise constraints (3.30) would be violated. Constraints (3.31) impose a preference to the vehicle to choose.

They break the symmetry of the solution and they are used for avoiding to spend time visiting symmetric solutions when solving the model and thus reducing the search space.

Finally, constraints (3.32), (3.33), (3.34), (3.35),(3.36), (3.37), and (3.38) define the routing problem and set a maximum time for the activity of waste collection.

Remark 1. *It is important to notice that we enforce in the model the general constraints* (3.38) *and we do not use compact formulations such as the one considered in [66] [43] or in [30]. The reason of that choice is that we implement dynamically the constraints for forbidding sub-cycles as soon as they are considering in the solution (as proposed in [70]). This is possible through the callback functions available in the solver.*

3.5.1 The solution algorithm

Theorem 2. *The mixed integer linear model* (3.1)-(3.38) *has a worst case complexity of* $\mathcal{O}(2^{IJ^2T})$*, where* I *is the number of vehicles,* J *is the number of dumpsters, and* T *is the number of time steps.*

Proof. The algorithm that we use for solving the problem is the branch and bound that simply perform a wise enumeration of all the solutions. The worst case shows itself when the algorithm explore all the possible solutions. Since in the model there are variables $r^{it}_{j_1 j_2}$, we can easily find the aforementioned complexity. □

Due to the combinatorial complexity, in order to solve real instances, we develop a heuristic applying a relaxation strategy coupled with parameter aggregation. The heuristic groups together the dumpsters in clusters, then it solves by means of a commercial solver the problem considering the clusters and finally it produces a solution considering the single dumpsters. Due to the fact that it solves the problem by using an exact approach, it is a math-heuristic. In particular, the math-heuristic is composed by the following three steps:

1. **Clusterization**. In this phase, we build a network which nodes are aggregation of dumpsters. We call them clusters. By doing so, we emulate the strategy of the company that voids all the dumpsters located in the same city. The clusterization policy that we choose aggregates dumpsters spatially closer. In order to better explain how the model (3.1)-(3.38) becomes when we consider clusters, we define the set of clusters $\mathcal{C}$ (which cardinality is C). The set of all clusters can be partitioned in disjoint sets by considering the waste type that they collect, we call them C^s. Each of the cluster can be seen as a subset of the set of dumpsters $\mathcal{J}$ ($C_c \subset \mathcal{J}$). In particular, the C_c form a partition of the set $\mathcal{J}$. Since the clusters are group of dumpsters, we have to modify several constraints, first we remove constraints (3.4) and (3.5) because it is possible to collect waste from a cluster with more than one vehicle. The distances $d_{c_1 c_2}$ represent the distances between the clusters and we set them to be $d_{c_1 c_2} = \min_{j_1 \in c_1 j_2 \in c_2} d_{j_1 j_2}$ $\forall c_1, c_2 \in \mathcal{C}$. Other parameters that change interpretation are V_c and Θ_{ct}. In particular, V_c is the capacity of the cluster, we define it to be the sum of the capacities of all the dumpsters in the cluster c i.e. $V_c = \sum_{j \in c} V_j$, while Θ_{ct} is the maximum growth rate of the dumpsters in the cluster, we define it to be $\Theta_{ct} = \max_{j \in c} \Theta_{jt}$. Finally, a becomes the time spent to empty a cluster, we fix it to be equal to the sum of the emptying times of all the dumpsters, plus the time of a

tour between all the dumpsters belonging to the cluster. We compute it by solving through the Chained-Lin-Kernighan heuristic (see [6]) a TSP problem and the by subtracting the longest arc. In the application, we consider eleven clusters one for each city and two clusters for the city of Collegno. The new model is then:

$$\min. \sum_{i=1}^{I}\sum_{t=1}^{T} c_{it} w_{it} + \lambda \sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{c_1=1}^{C}\sum_{c_2=1}^{C} d_{j_1 j_2} r_{j_1 j_2}^{it}, \tag{3.21}$$

subject to

$$w_{it} \geq y_{ict} \quad \forall \ \ i,c,t \tag{3.22}$$

$$M y_{ict} \geq x_{ict} \ \ \forall \ i,c,t \tag{3.23}$$

$$\sum_{c=1}^{C} x_{ict} \leq \hat{C} + \sum_{s=1}^{S} l_s z_{ict} \quad \forall \ \ i,t \tag{3.24}$$

$$M z_{ist} \geq x_{ict} \ \ \forall \ i,s,j \in \mathcal{C}^s, t \tag{3.25}$$

$$\sum_{s=1}^{S} z_{ist} \leq 1 \ \ \forall \ i,t \tag{3.26}$$

$$V_{c0} = \Theta_{c0} \ \ \forall \ c \tag{3.27}$$

$$z_{ist} \leq y_{i(C-s)t} \ \ \forall \ i,t \tag{3.28}$$

$$V_{ct} = V_{ct-1} + \Theta_{ct} - \sum_{i=1}^{I} x_{ict} \quad \forall \ c, t \neq 0 \tag{3.29}$$

$$V_{ct} \leq \alpha V_c \quad \forall c,t \tag{3.30}$$

$$w_{it} \geq w_{i+1,t} \ \ \forall \ t, i = 1 : I-1 \tag{3.31}$$

$$\sum_{c_1=1}^{c} r_{cc_1}^{it} = y_{ict} \ \ \forall \ t,i,c \tag{3.32}$$

$$\sum_{c_1=1}^{c} r_{cc_1}^{it} = \sum_{c_1=1}^{c} r_{c_1 c}^{it} \ \ \forall \ t,i,c \tag{3.33}$$

$$\sum_{c=1}^{c} r_{cD_s}^{it} = z_{ist} \quad \forall\, t,i,s \tag{3.36}$$

$$\sum_{c=1}^{c} r_{1c}^{it} = w_{it} \quad \forall\, t,i \tag{3.34}$$

$$\sum_{c=1}^{c} r_{c1}^{it} = w_{it} \quad \forall\, t,i \tag{3.35}$$

$$\sum_{c_1=1}^{c}\sum_{c_2=1}^{c} d_{c_1c_2} r_{c_1c_2}^{it} + a\sum_{c=1}^{c} y_{ict} \leq C_{\max} \quad \forall\, i,t \tag{3.37}$$

$$\sum_{c_1,c_2 \in S, c_1 \neq c_2} r_{c_1c_2}^{it} \leq |S| - 1 \quad \forall S \subset], S \neq \emptyset \tag{3.38}$$

$$x_{ict} \in \mathbb{R}^{+} \quad \forall\, i,c,t$$

$$z_{ist} \in \{0,1\} \quad \forall\, i,s,t$$

$$y_{ict} \in \{0,1\} \quad \forall\, i,c,t$$

$$r_{c_1c_2}^{it} \in \{0,1\} \quad \forall\, i,c_1,c_2,t$$

$$V_{ct} \in \mathbb{R}^{+} \quad \forall\, c,t$$

We then solve the model by using an exact solver that is able to find the solution in a short amount of time due to the limited dimensionality of the problem. The constrains 3.38 are enforced dynamically by using the callback function that enables to add the constrains to forbid sub-cycles during the solution of the exact optimization problem.

2. **Building of a feasible solution**. From the solution obtained by the exact solver of the model described in the previous point, the heuristic obtains a feasible solution for the original problem by applying the following pseudocode:

3. **Post optimization**. The result of the previous step is an assignment of the dumpsters to each vehicle during each time shift used. The solution obtained is then refined by solving a TSP problem for each route by using the version for asymmetric networks of the Chained-Lin-Kernighan heuristic [6].

It is important to note that the solution found by the proposed heuristic is equal to the optimal solution to the mathematical problem for small instances because, in that cases $\mathcal{J}$ coincides with $\mathcal{C}$. Furthermore, the performance of the heuristic strongly depends by the the policy of creation of the clusters. We

Algorithm 1 Outer heuristic algorithm Input: Instance, Output: $\hat{x}_{ij}^{tm}$

best_opt=$+\infty$
for $i \in \mathcal{I}$ **do**
  for $t \in \mathcal{T}$ **do**
    list_1 = []
    if $w_{it} = 1$ **then**
      insert in list_1 all the dumpsters d s.t. $d \in c$ and $y_{ict} = 1$
      remove from list_1 all the dumpsters void
      sort list_1 in decreasing order of quantity of waste
      list_2 = []
      d = list_1[1]
      while constraints (3.24) or (3.37) hold with equality **do**
        d = next(list_1, d)
        add to list_2 d
      end while
      if all the element in list_1 have been added **then**
        add in list_2 all the dumpsters near to the one in list_2 until (3.24) or (3.37) hold with equality
      end if
    end if
  end for
end for

test several different policies, nevertheless the most effective one is the policy that groups together dumpsters geographically closer.

3.6 Computational Tests

3.6.1 Data

The company wants to organize the operations for a time horizon of a week (i.e. 18 time shifts), it has to manage 8 vehicles and 525 dumpsters divided into 2 types of waste (paper and municipal solid waste). For the computational experiments we consider 100 instances, randomly generated with realistic data (from the real instance we change randomly the characteristics of the dumpsters).

The data that we need for describing the optimization problem can be divided into two kinds: geographical and technical. The geographical parameters $d_{j_1 j_2}$ are computed by using Google Maps. Since we have those values for each couple of dumpsters, we are considering a complete graph. Instead, the technical data are

1. the cost c_{it} of using vehicle i during time t. We assume, driven by the indication of Cidiu S.p.A. that the cost of the first two time shifts is the same for each day and for each vehicle while the third time shifts costs ten times more.

2. the volume of each dumpster V_j. The company has dumpsters with volumes of 2400, 3500, and 5000 litres;

3. the capacity of the vehicles $\hat{C}$. Its value is of 20000 litres;

4. the extra capacity l_s. Its value is 0 litres for the municipal solid waste because it cannot be compressed, while it is 60000 litres for the paper;

5. the maximum duration of a time shift C_{max}. Its value is 6 hours;

6. the times to void a dumpster a. We consider this value equals for all the dumpsters since the action is performed by using a mechanical arm that spend most of the time for reaching the dumpster. Then factor that most affect the execution time of the operation is the relative position between the vehicles and the dumpster that on average is always the same. For

this reason, the value is supposed to be equal for all the dumpsters and it is equal to 5 minutes (according to Cidiu S.p.A. experience);

7. the parameters Θ_{jt} are computed from the historical data of the company: for each dumpster there is a set of unevenly spaced observations of the waste in the dumpster (the registration of these data are done when a vehicle voids the dumpster). For the sake of simplicity we suppose that the waste is produced uniformly during time. Then, by calling $\hat{\theta}_{jn}$ the quantity of waste collected at time t_n (where t_n is expressed in number of time shift), we use as increment

$$\Theta_{jt} = \frac{1}{N} \sum_{n=1}^{N} \frac{\hat{\theta}_{jn}}{(t_n - t_{n-1})} \ \forall j, t.$$

8. the limit of the percentage of volume of waste in each dumpster α is set to 0.8 according to the Cidiu S.p.A. experience.

In the final version, all the technical data enumerated in points 1 to 6 are collected automatically from the Cidiu S.p.A. database.

3.6.2 Results

The benefit gained by the company from the proposed solution, can be split in two kinds: methodological and computational. The methodological benefits concern how the solution is implemented while the computational benefits concern the performance of the solution. The main methodological benefit is the use of the GUEST OR methodology that has reduced the development time of the model. For quantifying the computational benefit, we evaluate 8 KPIs:

- nTS: the average number of third shifts used by the solution, the average is obtained by considering a period of a week. The lower is this quantity the more efficient is the solution.
- nV: the average number of vehicles used daily. It is calculated by dividing the number of vehicles used during one month of activity by the number

of time shifts used. We remember to the reader that the company has 4 vehicles. The lower is this quantity the more efficient is the solution.

- WV%: the average percentage of volume occupied by the waste during the collection. During the collection of the waste, the mechanical arm of the vehicle records the weight of the waste. By the knowledge of the density we can compute the volume in the dumpster and the percentage of volume occupied by the waste. The average over all the collections and all the dumpsters, of this percentage is WV%; the greater it is, the better the capacity of the dumpster is used. It is worth noting that this value cannot exceed the 80%

- TRT: average total routing time, it is the average time of a service. This value has not a clear interpretation for the economy of the solution because the higher the travelling times, the higher the travelling cost and the environmental impact but the higher the travelling times, the higher the number of voided dumpsters and the less trip has to be done.

- FV%: average fulfilment percentage of the vehicles. The average is computed with respect to all the time shifts when the vehicle is used. The closer to one this indicator is, the better the capacity of the vehicle is exploited.

- nVD: average number of visited dumpsters for each time shift (it is computed by considering each time shifts when at least one vehicle is used). This quantity has not a clear interpretation. Nevertheless, it gives us an important data for understanding how the solutions perform.

The KPIs are computed by considering 5 months of activity of the Cidiu S.p.A. and 5 simulated month of activity by using the proposed approach. Cidiu S.p.A. nowadays perform the collection operations by organizing periodic routes organized by using the company know how and experience in the sector. Until the end of the 2018, the company works on the integration of the proposed heuristic in the information system, so at the time of the writing of this thesis we do not have information related to the performance of the heuristic in the real field. The average monthly values of the KPIs during these experiments

Table 3.1 KPIs before and after the use of the software

KPIs	Cidiu S.p.A. solution	Simulated solution
nTS	1.44 (0.5)	0 (0)
nV	3 (0)	2 (0)
WV%	0.28 (0.10)	0.70 (0.05)
TRT	4.35 (0.5)	5.24 (0.3)
FV%	54%(10%)	87%(5%)
nVD	62.3(12)	68.5(12)

are shown in Table 3.1. The first column shows the KPIs, the second one lists their value registered from the standard activities and the third one lists the values obtained by using our approach. For each cell of the Table 3.1, in brackets there is the standard deviation of each value.

The results presented in Table 3.1 shows an improvement of all the defined KPIs. The most important result is that the company has no more need to use the third shift. Furthermore, the average number of vehicles used decreases by the 33%, and the dumpsters are visited when they have more waste. Hence, also the capacity of the dumpsters is better exploited. Finally, we can claim that the number of dumpster voided in each shift is similar in the two approaches, this means that the proposed algorithm improves the productivity of the time shifts leading to a more effective solution. With the proposed approach, the benefits for the company are a reduction of the waste collection operational costs and an increase of the competitiveness.

The proposed solution has the added value to be aperiodic. To impose the periodicity of the waste collection produces an overestimation of the production of the single dumpster (the one with the higher production rate impose the periodicity). Furthermore, the better performance of the aperiodic approach with respect to the periodic solution lead the management of Cidiu S.p.A. to consider of adapting such an aperiodic solution even if it is more demanding for the drivers (each day they have to visit different sets of dumpsters). It is worth noting that nowadays, the company is using periodic scheduling for the waste collection. Hence, the comparison between the results computed by the proposed methodology and the ones implemented by the company provide an evidence of their different efficiencies. Unluckily, it is not possible to perform a

comparison between our routes and theirs because the data about their routes is protected by trade union's contracts. In order to give an idea of the final solution, we show in Figure 3.3 an example of route.

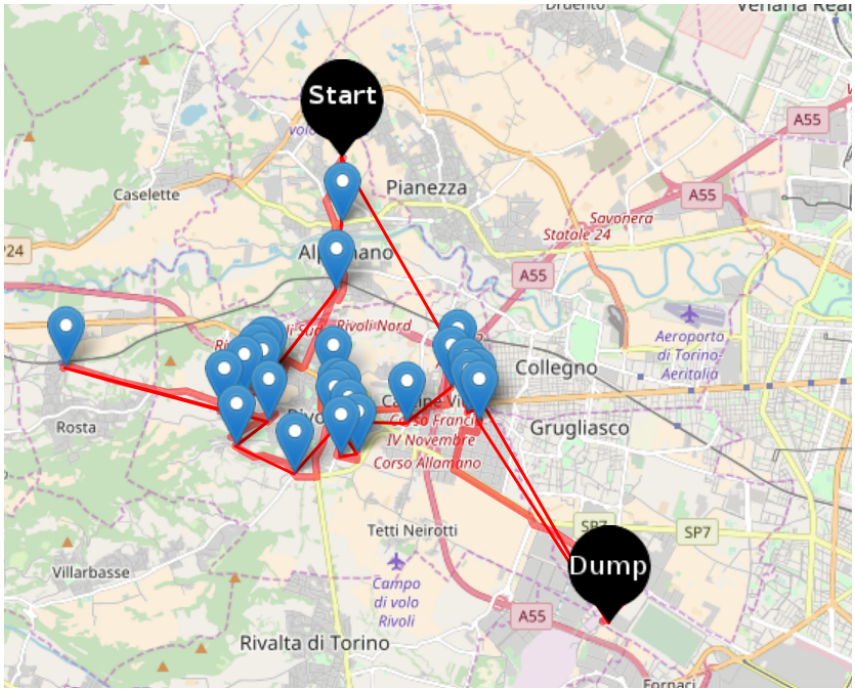

Figure 3.3 The figure shows an example of solution found by the heuristic.

Remark 3. *By changing the parameter λ (that weight of the routing time in the objective function), the solutions obtained change only slightly i.e., the number of time shift remains the same.*

During the simulations we have tested the behaviour of the model in different settings. The model has shown to be sensitive to the changes in the value Θ_{jt}, in particular, if these coefficients increase the total cost increases and, for values of Θ_{jt} high enough, the problem becomes infeasible. This infeasibility is due to the fact that for high values of Θ_{jt} the vehicles are not able to void the dumpsters before that they reach the maximum volume allowed. Another important parameter is the length of the time shits and the size of the dumpsters: an increase in these parameters generates a decrease of the total cost.

Since the constraints about the capacity of the vehicles are never active, the model is not sensitive to small changes in these values.

For testing the performance of the proposed heuristic, we compare the solutions obtained by the proposed heuristic, with the ones computed by the commercial solver Gurobi. In particular, we use instances with $10, 20, 30, 40$, and 50 dumpsters, 6 time periods and 2 types of waste. We cannot perform tests with bigger instances because the commercial solver runs out of memory.

In the instances generated for this test, the dumpsters are randomly located in clouds of points, the volume of the dumpsters, the initial quantities of waste and the increments in the waste quantity are randomly generated by considering the realistic data obtained as random changes of the statistics obtained from the historical data. The travel times are obtained by the euclidean distance between the points and the costs of each time shift are randomly generated by considering the third shift to be ten times more expensive than the first two.

The results of this comparison are shown in Table 3.2. The first column shows the number of dumpsters J, the second shows the difference in time shifts between the optimal solution and the heuristic one, the third column shows the difference between the routing cost of the two solutions and finally the last two columns report the computational time of the heuristic and the computational time of the exact solver. For each cell in Table 3.2, the mean values and standard deviations (in brackets) over 20 runs are shown.

It is important to note that the proposed heuristic is able to find the optimal number of time shifts in all the instances and the differences between the heuristic solution and the optimal one are in the routing. These differences are not critical, since the main cost of the company derives from the time shifts.

One of the most important results of this application concerns the reduction of the computational time in order to obtain a solution. While the commercial solver is not able to solve the problem due to memory constraints, the average solution time of the proposed heuristic for real instances is of 4 hours and 23 minutes with a standard deviation of 20 minutes (we computed these value by using the 100 instances used in the first simulation). The most of the computing time is required by the solution of the integer model formulated by using clusters instead of dumpsters. Once that the solution of the problem is found, then the other steps of the procedure are almost instantaneous. The duration of the process is reasonable for the management that has enough time

Table 3.2 Comparison between the optimal solutions and the heuristic ones. All the values are computed by using 6 time periods and 2 types of waste. The first column shows the number of dumpsters (J), the second shows the percentage difference in the number of time shifts (nTs), the third one shows the percentage difference in the routing cost (rC) and the last two columns report the computational times of the exact solver and of the proposed heuristic.

J	**nTs** [%]	**rC** [%]	**Time Optimal** [s]	**Time Heuristic** [s]
10	0 (0)	0(0)	43.43 (1.64)	2.76 (0.65)
20	0 (0)	0(0)	150.43 (10.89)	3.73 (0.56)
30	0 (0)	1.75 (1.38)	443.64 (15.92)	8.56 (0.84)
40	0 (0)	2.69 (1.43)	890.67 (30.53)	13.36 (0.74)
50	0 (0)	3.32 (2.34)	1842.86 (45.65)	26.27 (2.45)

to run the heuristic, validate the solution and send it to the drivers before to start the waste collection activities.

Furthermore, due to the computational time that the heuristic requires, it can be run each time shift, allowing the weekly plan to be adjusted and to consider the missed collections (e.g., when a vehicle cannot collect the waste in a dumpster because of the presence of a parked car that blocks the operations).

Qualitatively, the proposed solution generates an improvement of the working conditions of the drivers since they no more have to perform time shifts during night, with a consequential positive impact on their quality of life. Furthermore, with the freed resources Cidiu S.p.A. can offer new services and it can increase the wealth of the regions where it is working.

Part II

Collaborative Models for Freight Transportation

The Deterministic Problem Formulation of Crowdshipping

In this chapter, we present the deterministic optimization model describing the crowd-shipping activities. We propose a heuristic composed by greedy method with random multi start followed by a local search improvement that is able to find good solution to the problem in a small amount of time. All the code used in this chapter is implemented in C++ and it is freely available under the *GNU Public License*[1]. The chapter is organized as follows: in Section 4.1 we motivate the use of optimization in the context of crowd-shipping; in Section 4.3 we present the mathematical model. In Section 4.4, we analyse the main properties of the model and we develop a heuristic solution in Section 4.5. Finally, in Section 4.6, we describe the instance generation algorithm that we adopt and in Section 4.7 we present the performance of the proposed approach on a set of simulated instances. Due to the lack of literature about this problem, we define some freely available benchmark instances[2].

4.1 Motivations and Problem Definition

Nowadays, in several fields there is the request to perform easy tasks randomly located in a wide area. Some problems of this type are the deliveries of e-grocery orders, the collection of data from sensors distributed over the city (such as water meters, smart-dumpsters), etc. Due to their simplicity, these operations

[1] https://bitbucket.org/EdoFadda/Coiote/
[2] https://bitbucket.org/EdoFadda/Coiote/

can be done by every person with saving both in term of economical costs and in terms of environmental costs (the users that accept to do the task are likely to be near to the location of the accepted tasks). For this reason, companies involved in these kinds of businesses can resort to new methods as the aforementioned crow shipping and opportunistic IoT. Clearly, the company using these business models must rewards the users that perform the tasks. The main objective of this chapter is to define a deterministic mathematical model suitable to help companies that use social engagement to minimize the total amount of rewards while doing all tasks in a given time interval. To our knowledge, this is the first time that such a problem is presented. The computational experiments show that exact methods perform poorly on big instances because they require too much time with respect to the real world applications requirements. For this reason, we introduce a heuristic able to solve the test instances in a reasonable amount of time and able to find solutions close to optimal one.

4.2 The deterministic problem

The main topic of this subsection is the literature review of the applications of deterministic optimization techniques to social engagement and, in particular, to crowd-shipping. To our knowledge this is the first study in the field, hence, we consider the review of the literature of two problems similar to the one considered: the Multi Period Assignment problem (MPAP) and the Multi Period, Multi Commodity Transportation problem (MPMCTP).

Let us start our analysis with the assignment problem (see [23] and [60] for a review). All the problems under this name have in common two features: tasks (or operations) to be done and resources to be allocated to each task. In this thesis, the tasks are the deliveries of the packages from a point to another in the network, while the available resources are the customers of the company that are available to perform one or more deliveries. Since the tasks are not safety critical and they do not require particular skills, we assume that all the users can perform all the tasks. Furthermore, we assume that all the tasks have to be done during a time interval. For example, we can consider that given the

order the company has some days to deliver the freight. For all these reasons, we consider the MPAP. In the literature this problem is not deeply analysed. Nevertheless, we have found two papers dealing with this problem i.e. [107] and [7]. In the first paper, a binary multi-periods assignment problem is studied. The problem arises as a part of a weekly planning problem in mail processing operations. In the second paper, the classical MPAP is considered. In the problem, the decision variables are binary variables describing the switching of a person from a task to another. Both these papers consider binary decision variables while we consider integer decision variables.

If we consider the model to be similar to the transportation problem introduced by [35], then the model that we define is a MPMCTP. As aforementioned, the considered problem is a multi-periods transportation problem in which each type of customer that we consider (characterized by different attitude for doing tasks) is a different commodity. The literature about this problem is not very wide; to the authors knowledge the only paper addressing the argument is [74].

4.3 Mathematical Model

In this section we introduce the deterministic mathematical model in order to optimize crow shipping i.e. minimizing the total amount of rewards while performing all the deliveries that the company must do. Before presenting the model, we introduce some definitions.

Definition 7. *We call customers all people available to do one or more tasks that the company has contacted.*

Since people is extremely heterogeneous, and they have different perceptions of the value of money, we consider the different type of users. Furthermore, since the operations that have to be performed are not safety critical, then the tasks must be done before a final time T. For this reason, we consider a set of time periods $t = 1, 2, \ldots, T$ available to do all the tasks. We use the different time step do model the different availability of people during the day.

We use this terminology because we consider that they are customers of a particular service of the company.

Definition 8. *We call tasks or activities the deliveries that the company must do.*

The mathematical model describing the problem considers the following sets:

- $\mathcal{T}$ is the set of all time indexes. The cardinality of this set is T. We assume that all the tasks must be performed before time T.
- $\mathcal{I}$ is the set of all source nodes of the network. The cardinality of this set is I.
- $\mathcal{J}$ is the set of all sink nodes of the network. The cardinality of this set is J.
- $\mathcal{M}$ is the set of all user types. The cardinality of this set is M, each type of users is characterized by a number of tasks that it is willing to perform.

The model uses the following parameters:

- c_{ij}^{tm} is the cost of the reward for a customer of type m in node i at time t that goes to node j. This cost depends by the distance between the nodes i and j at time t and by the type of customer m (in relation to its perceived value of the money).
- N_j is the number of tasks that must be done in the node j before time T.
- n_m is the number of tasks that a user of type m is willing to do.
- θ_i^{tm} is the number of customers of type m in node i during time step t.

The amounts of customers of kind m that are asked to do n_m tasks in node j, starting from node i, during time step t are the decision variable of the problem. We call them x_{ij}^{tm}.

The optimization problem can be expressed as follows

$$\min \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{m=1}^{M} c_{ij}^{tm} x_{ij}^{tm} \tag{4.1}$$

subject to

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{i=1}^{I} n_m x_{ij}^{tm} \geq N_j \quad \forall \ j \in \mathcal{I} \tag{4.2}$$

$$\sum_{j=1}^{J} x_{ij}^{tm} \leq \theta_i^{tm} \quad \forall \ i \in \mathcal{I} \ t \in \mathcal{T} \ m \in \mathcal{M} \tag{4.3}$$

$$x_{ij}^{tm} \in \mathbb{N} \quad \forall \ i \in \mathcal{I} \ j \in \mathcal{J} \ t \in \mathcal{T}. \tag{4.4}$$

The objective of problem (4.1) is to minimize the total amount of rewards that the company has to pay to the customers. Constraints (4.2) impose that all the deliveries must be done before time T. Constraints (4.3) ensure a limit to the number of people that can be contacted. Finally, constraints (4.4) impose that all the decision variables are non-negative integer (i.e. natural numbers).

Remark 4. *Model (4.1)-(4.4) can be used for describing the general transportation problem of logistics' companies having a heterogeneous fleet of vehicles that transport a homogeneous good. In this interpretation, θ_i^{tm} is the amount of vehicles of type m available in depot i at time t and n_m is the number of homogeneous good transported by the vehicles of type m.*

Remark 5. *Even if the model is presented as a multi-period problem, the time index can be removed replacing every node with T nodes, one for each time step.*

Remark 6. *In order for model (4.1)-(4.4) to have a solution, then it must be satisfied that*

$$\sum_{i=1}^{I}\sum_{t=1}^{T}\sum_{m=1}^{M} n_i \theta_i^{tm} \geq \sum_{j=1}^{J} N_j. \tag{4.5}$$

4.4 Problem Analysis

In this section we study the properties of model (4.1)-(4.4). Some of these properties can be used to successfully solve the problem in particular settings.

Since in this section the development is theoretical and not algorithmical, for the sake of simplicity and without loss of generality we do not consider the time index (see 5). This section is organized as follows, in Subsection 4.4.1 we present the properties of the model if one type of customer is considered.

4.4.1 One User Type

In this subsection we state important properties of the deterministic model if one type of customer is considered i.e. $M = 1$. As we have stated in Section 4.2, model (4.1)-(4.4) is closely related to the transportation problem. One of the most important property of the single commodity transportation problem is that the solution to its continuous relaxation is integer.

Unluckily, model (4.1)-(4.4) is not a transportation problem, nevertheless we can state the following theorem.

Theorem 7. *Given the integer discrete model*

$$(P1) : \min \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} x_{ij} \tag{4.6}$$

subject to

$$\sum_{i=1}^{I} n x_{ij} \geq N_j \quad \forall \ j \in \mathcal{J} \tag{4.7}$$

$$\sum_{j=1}^{J} x_{ij} \leq \theta_i \quad \forall \ i \in \mathcal{I} \tag{4.8}$$

$$x_{ij} \in \mathbb{N} \quad \forall \ i \in \mathcal{I} \ \ j \in \mathcal{J}. \tag{4.9}$$

$(P1)$ *has the same solution of model*

$$(\overline{P2}) = \min \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} x_{ij} \tag{4.10}$$

subject to

$$\sum_{i=1}^{I} x_{ij} \geq \left\lceil \frac{N_j}{n} \right\rceil \quad \forall \ j \in \mathcal{J} \tag{4.11}$$

$$\sum_{j=1}^{J} x_{ij} \leq \theta_i \quad \forall \ i \in \mathcal{I} \tag{4.12}$$

$$x_{ij} \in \mathbb{R}^{+} \quad \forall \ i \in \mathcal{I} \ \ j \in \mathcal{J}. \tag{4.13}$$

Proof. Since $(\overline{P2})$ is the continuous version of a transportation problem, it is also the solution to the problem with integer constraints, we call it $(P2)$. We then have $(\overline{P2}) \equiv (P2)$. Furthermore, it holds that $(P1) \equiv (P2)$ because, for x_{ij} integer

$$\sum_{i=1}^{I} x_{ij} \geq \left\lceil \frac{N_j}{n} \right\rceil \longleftrightarrow \sum_{i=1}^{I} n x_{ij} \geq N_j \quad \forall \ j \in \mathcal{J}$$

□

By this theorem we have a fast way to compute the solution to problem characterized by one type of customer.

4.5 Heuristic

In this section, we develop a heuristic able to find a good solution to the problem (4.1)-(4.4), the main goal of the heuristic is to find the best possible value in a given amount of time. The heuristic is composed by two parts:

1. *Outer heuristic*: it generates a random order of visit of the nodes of the network, it runs the function *Greedy step* on the generated sequence and, if the obtained solution is better than the best found, it updates the best solution. These operations are repeated until there is enough time. The pseudo code of this part is shown in Algorithm 2.

2. *Greedy step*: it is the most important part of the heuristic. For each node in the sequence generated by the previous part, it fulfils the demand of

each sink node by considering each source cell that has available resources in an order defined by the function *Minimum_Cost*. The pseudo code of this part is shown in Algorithm 3.

Algorithm 2 Outer heuristic algorithm Input: Instance, Output: $\hat{x}_{ij}^{tm}$

```
1: best_opt=+∞
2: while there is still time do
3:     cells_sequence = random_shuffle(cells_sequence)
4:     [opt, x̂_ij^tm] = Greedy step(cells_sequence)
5:     if opt < best_opt then
6:         x̂_ij^tm = x_ij^tm ∀ i, j, t, m.
7:         best_opt = opt
8:     end if
9: end while
```

The aforementioned *Minimum_Cost* function has different behaviours with respect to the different runs of the algorithm. During the first run, it orders the sources by decreasing values of the ratio $\frac{c_{ij}^{tm}}{n_m}$. During the second run and the successive ones, it randomly changes the order of the sources in order to increase the exploration of the solution space.

The function *Try_Improve* tries to find a set of switches between sources and sinks that, if applied, generates an improvement in the objective function. In particular, starting from a feasible solution, *Try_Improve* removes one or more customers assigned to a sink and then, it tries to find other customers able to improve the value of the solution. If there exist such a customers, the recursion ends with a positive result and the solution is updated. If the customers are not available, *Try_Improve* verifies if it is possible to do it with other sinks nodes by recursively calling itself.

During several experiments we have noticed that this algorithm has good performance if there are fewer activities than resources. If instead the number of sources and sinks is equal, then a modified version of *Greedy step* can be used. The modified version is a two steps procedure. During the first step, the method does not choose customers that would lead to a waste of activities (i.e. more activities done than the requested number). While in the second step, the method relaxes this additional constraint with the goal of performing the remaining applications.

Algorithm 3 Greedy Step Input:cells sequence Output: $(\hat{x}_{ij}^{tm})$ and its value *opt*

$\hat{\theta}_i^{tm} = \theta_i^{tm}$
$\hat{x}_{ij}^{tm}$ for all i, t and m
$\hat{N}_j = N_j$
for each cell j in cells_sequence **do**
    list_m_i_t = $Minimum\_Cost(c_{ij}^{tm}, \hat{\theta}_i^{tm})$
    count = 0
    while $\hat{N}_j \geq 0$ **do**
        $[m, i, t]$ = list_m_i_t[count]
        $M_i^{tm} = min[\hat{\theta}_i^{tm}, \frac{\hat{N}_i}{n_m}]$
        $\hat{x}_{ij}^{tm} = \hat{x}_{ij}^{tm} + M_i^{tm}$
        $\hat{\theta}_i^{tm} = \hat{\theta}_i^{tm} - M_i^{tm}$
        $\hat{N}_j = \hat{N}_j - n_m M_i^{tm}$
        count = count+1
    end while
end for
$Try\_Improve(\hat{x}_{ij}^{tm} \; \forall \; i, \; j, \; t, \; m.)$

4.6 Instance Generation and Benchmark problems

In order to define the instances, first we have to set T, M, and I and then to describe the networks and the parameters θ_i^{tm} and N_j.

We choose T to be 1 or 20, $T = 1$ represents the online optimization, while $T = 20$ simulates the planning of a reasonable time horizon (if we consider one time step to represent 1 hour). M is the number of types of customers, we choose $M = 3$ in order to model three types of customers: the standard customers m_0, the business customers m_1 and the regular workers m_2. In particular, standard customers perform a small number of tasks for a low cost, business customers perform more tasks than the standard customers but they are more expensive. Finally, the regular workforce performs more tasks than the other two types but its cost is greater. Usually, we suppose that companies use the regular workforce only if they are not able to fulfil all the deliveries with the customers. The number of urban store (I) is relatively small, nevertheless,

the number of sinks J can reach high values: for cities such as Torino it can be equal to 500. Usually in big cities, we can group together several deliveries, e.g. we can group together all the deliveries in the same mobile phone cell. By using this choice, in the biggest cities we can have up to 1000 mobile phone cells.

The network considered in the instances is a complete graph (i.e. from each node, it is possible to reach every other node). Furthermore, we consider that in each node of the network, there are either customers or deliveries but not both together. For each node of the network, there is a probability ρ that it is a sources and a probability $1-\rho$ that it is a sink. For each source node, we define the number of customers in that cell (θ_i^{tm}) by taking a realization of a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with average μ and variance σ^2. The Normal distribution is used since we θ_i^{tm} is the sum of the random variables considering the place where each person is located in a certain time interval. By this consideration and by using an appropriate version of the central limit theorem, the result is provided. We postpone the discussion about the result in chapter 7, where the stochastic version of the problem is considered. Since all the variables in the problem are discrete, we round the continuous value to the nearest integer.

For each sink node, we define N_j to be the realization of a uniform distribution between $[0, N_{\max}]$.

In order to assure that the problem is feasible it is enough to assure that (4.5) holds. If it is not the case, we randomly add people in randomly selected source nodes in order to reach that condition.

The coefficients of the objective function are c_{ij}^{tm}, they describe the costs of rewards for a user of type m that in time period t is in node i and perform deliveries in node j. We define them by the following equation

$$c_{ij}^{tm} = \begin{cases} |\frac{i-j}{4} + 1| C \log(2), & \text{if } m = m_0 \\ |\frac{i-j}{4} + 1| C \log(4), & \text{if } m = m_1 \\ |\frac{i-j}{4} + 1| C \log(6), & \text{if } m = m_2 \end{cases}, \tag{4.14}$$

where C is a random number taken from a random variable uniformly distributed between $C_{\min}$ and $C_{\max}$. In the following we assume $C_{\min} = 2$ and $C_{\max} = 5$.

The parameters n_m are the numbers of tasks that each user type performs. We impose that the standard users perform 1 task ($n_{m_0} = 1$), that the business resources perform 2 tasks ($n_{m_1} = 2$) and that the regular workers perform 10 tasks ($n_{m_2} = 3$).

Since there is not literature about this problem, there are no benchmark instances as well. We fulfil this gap by proposing some instances, freely available at https://bitbucket.org/orogroup/mpap. The name of the instances is $Co\_I\_T\_n$, where I indicates the total number of nodes in the network, T indicates the number of time periods and n is the identification number of the instance. At the same link the interested reader can find the code for generating new instances.

4.7 Numerical Experiments

In this section, we compare the performance of the commercial solver Gurobi[3] and the performance of the proposed heuristic on several instances. All the following experiments are performed on an Intel R CoreTMi7-5500U CPU @2.40 Ghz with 8 GB RAM and Microsoft R WindowsTM10 Home installed.

In Tables 4.1, 4.2, 4.3 and 4.4, we show the performance of the commercial solver and the performance of the heuristic in terms of computational time and optimal solution. In Table 4.1, we consider 30 cells, 1 time period, while in Table 4.2 we consider 30 cells and 20 time periods. In these two tables, the proposed heuristic is slower than the exact solver because it needs time in order to build the knowledge base. Nevertheless, this investment enables the heuristic to outperform the exact method, as it can be seen in Table 4.3 and in Table 4.4. In all the instances considered in these two tables, the heuristic is able to find the optimal solution. Furthermore, the computational time used by the heuristic for the instances with 100 cells is 75% less than the time used by

[3]http://www.gurobi.com

Table 4.1 The table considers the Optimal Solution (Opt.Sol.), the time used by Gurobi to find it (Time[s]), the time used by the proposed heuristic (Heu.Time[s]) and the value of the heuristic solution (Heu.Sol.) for the instances with 30 cells and 1 time period.

Instance	Opt. Sol.	Time[s]	Heu. Time[s]	Heu. Sol.
Co_30_1_NT_0	1041	0.014	1.25039	1041
Co_30_1_NT_1	1756	0.013	1.25028	1756
Co_30_1_NT_2	2341	0.017	1.25023	2341
Co_30_1_NT_3	2105	0.028	1.25018	2105
Co_30_1_NT_4	1477	0.018	1.25024	1477
Co_30_1_NT_5	2996	0.021	1.25025	2996
Co_30_1_NT_6	1623	0.009	1.2502	1623
Co_30_1_NT_7	1032	0.01	1.25022	1032
Co_30_1_NT_8	2288	0.018	1.25027	2288
Co_30_1_NT_9	1562	0.018	1.25024	1563
Co_30_1_T_0	1105	0.006	1.25026	1105
Co_30_1_T_1	1796	0.013	1.25022	1797
Co_30_1_T_2	2437	0.017	1.25021	2437
Co_30_1_T_3	2073	0.008	1.25025	2073
Co_30_1_T_4	1545	0.02	1.25018	1545
Co_30_1_T_5	2888	0.009	1.2502	2888
Co_30_1_T_6	1592	0.011	1.2502	1592
Co_30_1_T_7	1394	0.008	1.25018	1394
Co_30_1_T_8	2012	0.01	1.25033	2012
Co_30_1_T_9	1820	0.011	1.25026	1820

the exact algorithm. While, for the instances with 300 cells the computational time is reduced by the 210%.

Remark 8. *It is important to note that in the real context, this problem has to be solved each time new information about the people accepting to do the task are gather. For this reason time efficiency is a problem because the solution method has to be responsive in case of missing answer, people that accept to do the task and forgot and several other cases. In the framework, the expert from TIM (the industrial partner) propose to run the optimization procedure every minutes.*

Table 4.2 The table considers the Optimal Solution (Opt.Sol.), the time used by Gurobi to find it (Time[s]), the time used by the proposed heuristic (Heu.Time[s]) and the value of the heuristic solution (Heu.Sol.) for the instances with 30 cells and 20 time periods.

Instance	Opt. Sol.	Time[s]	Heu. Time[s]	Heu. Sol.
Co_30_20_NT_0	872	0.106	1.25083	872
Co_30_20_NT_1	457	0.117	1.25093	457
Co_30_20_NT_2	706	0.142	1.25064	706
Co_30_20_NT_3	827	0.076	1.25242	827
Co_30_20_NT_4	437	0.133	1.25078	437
Co_30_20_NT_5	984	0.124	1.25076	984
Co_30_20_NT_6	937	0.104	1.25078	937
Co_30_20_NT_7	1132	0.09	1.25072	1132
Co_30_20_NT_8	719	0.115	1.25081	719
Co_30_20_NT_9	895	0.119	1.25079	895
Co_30_20_T_0	872	0.095	1.25076	872
Co_30_20_T_1	457	0.097	1.25078	457
Co_30_20_T_2	721	0.142	1.25077	721
Co_30_20_T_3	827	0.086	1.25082	827
Co_30_20_T_4	437	0.138	1.2508	437
Co_30_20_T_5	991	0.125	1.25072	991
Co_30_20_T_6	933	0.118	1.25071	933
Co_30_20_T_7	1143	0.085	1.25083	1143
Co_30_20_T_8	4453	0.127	1.25072	4453
Co_30_20_T_9	4530	0.108	1.25077	4530

Table 4.3 The table considers the Optimal Solution (Opt.Sol.), the time used by Gurobi to find it (Time[s]), the time used by the proposed heuristic (Heu.Time[s]) and the value of the heuristic solution (Heu.Sol.) for the instances with 100 cells and 1 time period.

Instance	Opt. Sol.	Time [s]	Heu. Time [s]	Heu. Sol.
Co_100_1_NT_0	5270	5.34234	1.25027	5270
Co_100_1_NT_1	3811	5.23475	1.25029	3811
Co_100_1_NT_2	4455	5.23625	1.25157	4455
Co_100_1_NT_3	4832	5.52352	1.25033	4832
Co_100_1_NT_4	4790	5.65343	1.25043	4790
Co_100_1_NT_5	6493	5.34634	1.25027	6493
Co_100_1_NT_6	4276	5.34534	1.25039	4276
Co_100_1_NT_7	4815	5.34564	1.25031	4815
Co_100_1_NT_8	4636	5.25154	1.25045	4636
Co_100_1_NT_9	4691	5.34523	1.25035	4691
Co_100_1_T_0	5350	5.43523	1.25034	5350
Co_100_1_T_1	3919	5.26234	1.25027	3919
Co_100_1_T_2	4764	5.34523	1.25031	4764
Co_100_1_T_3	5215	5.34265	1.25031	5215
Co_100_1_T_4	5012	5.29955	1.25036	5012
Co_100_1_T_5	6730	5.32334	1.25027	6730
Co_100_1_T_6	3625	5.25543	1.25027	3625
Co_100_1_T_7	3203	5.33435	1.25035	3203
Co_100_1_T_8	2196	5.34352	1.25041	2196
Co_100_1_T_9	2182	5.32345	1.25035	2182

Table 4.4 The table considers the Optimal Solution (Opt.Sol.), the time used by Gurobi to find it (Time[s]), the time used by the proposed heuristic (Heu.Time[s]) and the value of the heuristic solution (Heu.Sol.) for the instances with 300 cells and 20 time period.

Instance	Opt. Sol.	Time[s]	Heu. Time[s]	Heu. Sol.
Co_300_20_NT_0	7019	25.628	1.29306	7019
Co_300_20_NT_1	7183	28.814	1.29125	7183
Co_300_20_NT_2	8101	21.883	1.29518	8101
Co_300_20_NT_3	7638	25.531	1.29798	7638
Co_300_20_NT_4	8193	24.294	1.29795	8193
Co_300_20_NT_5	7580	21.724	1.28707	7580
Co_300_20_NT_6	7681	23.469	1.2945	7681
Co_300_20_NT_7	8546	17.117	1.29569	8546
Co_300_20_NT_8	7129	17.382	1.29748	7129
Co_300_20_NT_9	7191	19.063	1.28753	7191
Co_300_20_NT_10	8074	25.28	1.29103	8074
Co_300_20_T_0	7024	27.258	1.30792	7024
Co_300_20_T_1	7183	30.834	1.28962	7183
Co_300_20_T_2	8102	22.448	1.29249	8102
Co_300_20_T_3	7659	25.1	1.28967	7659
Co_300_20_T_4	8223	25.942	1.29351	8223
Co_300_20_T_5	7600	21.737	1.29414	7600
Co_300_20_T_6	7647	23.244	1.29524	7647
Co_300_20_T_7	8590	19.756	1.30957	8590
Co_300_20_T_8	7140	23.853	1.28804	7140
Co_300_20_T_9	7227	19.247	1.29744	7227
Co_300_20_T_10	8047	23.834	1.29261	8047

The Stochastic Problem Formulation of Crowd-shipping

In this chapter, we extend the study of Chapter 4 in order to consider uncertainty. We then consider the problem of finding the minimum amount of reward to offer in such a way to perform a set of task during a given time interval. In the crowd-shipping application, uncertainty is generated by the number of customers in the distribution centers. This chapter is organized as follows. In Section 5.2, we present the stochastic version of the problem presented in Section 4.3 and we describe the stochastic distribution more suitable for the simulations. In Section 5.3, we analyse the complexity of the proposed model. Moreover, in Section 5.5 we show that the stochastic approach must be considered since the deterministic one produces considerable inefficiencies. Finally, in order to reduce the computational time required for computing a solution we apply a novel version of the Loss of Reduced Costs-based Variable Fixing (LRCVF) heuristic and we compare, by means of computational tests, the performance of this heuristic and the performance of the commercial solver Gurobi. The results prove that the customized LRCVF heuristic is able to provide good solutions to big instances in a reasonable amount of time.

5.1 The stochastic problem

The main topic of this subsection is the literature review of the applications of stochastic optimization techniques to social engagement and, in particular, to crowd-shipping. As in the deterministic case, also in the stochastic one there

are no similar studies of the topic. For this reason, we analyse the literature of the optimization problem classes more similar to the proposed problem.

From the optimization point of view the problem is close to the assignment problem and to the transportation problem.

In particular, the proposed problem is close to the assignment one because it has the common features typical to these problems i.e. tasks to be done and resources to be allocated. In particular, we consider that the tasks are deliveries, while the resources are the customers that are available to perform deliveries. The availability of such people represent a source of uncertainty. Furthermore, since we also consider multiple time steps, we have that the problem is similar to a MPSAP.

The literature about this problem is not developed; to our knowledge the main references are [51], [92]. The two articles minimize the cost of the assignment of a fleet of vehicles to an uncertain number of tasks. While in these papers the uncertainty is in the demand, in the proposed model uncertainty is in the availability of the resources. Further differences are that both the papers consider an infinite horizon, while we consider a finite one, moreover both [51] and [92] consider a multi-stages stochastic linear problem, while we consider a two stages stochastic linear problem. Since the stochastic problem is more difficult than the deterministic one, it is common to justify the solution to the stochastic problem by comparing the deterministic solution and the stochastic one. Finally the stability of the problem is studied in in Subsection 5.5.2, as it is described in [47] and [50]. In order to solve large instances, we use the heuristic LRCVF that has been introduced in [15]. It fixes to their lower bounds all the variables having the reduced cost of the continuous relaxation problem higher than a threshold. The threshold is chosen in order to fix to zero a fixed percentage of all the variables i.e. the threshold is a quantile of the reduced costs values.

From another point of view, we can consider the problem to be a customized version of the stochastic multi-periods multi-commodities minimum flow problem. As for the assignment problem, the literature about this problem is not very developed. To our knowledge, [41] is the only paper dealing with this problem. It is an example of multicommodity two stages problem with uncer-

tain demand. Nevertheless, uncertainty is in demand while in the proposed model uncertainty relies in the availability of the resources.

5.2 Stochastic Mathematical Problem

In this section, we develop the stochastic version of the deterministic model in Chapter 4. As in the previous chapter, we consider as a case study crowd-shipping. In particular, we consider a company in the e-grocery business that has a network of supermarkets distributed in the city and it asks to people coming in the supermarket to deliver the goods ordered by online customers. This model describes also the activity of collecting data from network of sensors distributed in a city or other applications using social engagement. In exchange for the services that the company receives, it gives a reward to each person. The objective of the model is to minimize the total amount of rewards and to perform all the tasks.

It is important to notice that the stochastic formulation of the problem is important since it considers explicitly the flow of information that in the deterministic version is totally ignored. Furthermore, it is possible to associate to the flow of information variations in the amount of the rewards.

Since the model is stochastic, before to describe the model itself, we specify the flow of time and the flow of information. In the description of the model and in the following, we adopt the standard terminology saying that two states are two different stages if between the two there is a difference in the information content.

The first stage is characterized by the company that sends to each customer a request for performing tasks. Then, the company observes the amount of customers that accept to perform tasks and it reacts by sending more requests (as a recourse action). If there are more time steps available, then each one of them is characterized by those two stages. We consider that the total amount of requests is the same for all the considered time instants. This assumption is reasonable since the time of sending the messages lasts few seconds and it is not compatible with the activities of building the orders.

The model is described by using three sets of parameters:

- $\mathcal{T}$ is the set of all time steps, its cardinality is T,
- $\mathcal{I}$ is the set of all source nodes of the network (i.e. nodes where the company has a distribution center), its cardinality is I,
- $\mathcal{J}$ is the set of all sink nodes of the network (i.e. nodes where the company has tasks to perform), its cardinality is J,
- $\mathcal{M}$ is the set of all types of customers, its cardinality is M.

Set $\mathcal{M}$ is useful in order to describe the different availabilities of the customers to perform tasks and the differences in the rewards that they ask for. In particular, we consider three types of customers, a standard type, that performs one task in exchange for a low reward, a business type that is available to perform several tasks but it asks for a bigger reward and finally the standard workforce, able to perform more deliveries than the other two types, but they are far more expensive of the other two types. The model is described by using the following parameters:

- n_m is the number of tasks that a customer of type m is available to do.
- c_{ij}^{tm} is the cost of the reward for a customer of type m in the source node i at time t that performs n_m tasks in the sink node j.
- q_{ij}^{tm} is the cost of the reward for a customer of type m in the source node i at time t that during the second stage goes in the sink node j to perform n_m tasks.
- N_j is the number of tasks that must be performed in sink node j before the end of the considered time period (i.e. T).
- $\hat{\theta}_i^{tm}$ is the minimum number of customers of type m in source node i during the first stage of time step t. This number is a forecast.

Furthermore, the model is characterized by the random variables $\theta_i^{tm}(\omega)$ describing the number of customers of type m in source node i during time step

t. The information about the value of this parameter is revealed during the second stage.

Remark 9. *We assume that $c_{ij}^{tm} < q_{ij}^{tm}$ $\forall$ i, j, t, m. because if it is not the case the wait and see strategy is preferred by the solver and there is no point in sending out request a priori. This assumption is indeed reasonable since as soon as we run the optimization procedure the task are not urgent hence the company can offers a low reward in order to be satisfied. Nevertheless, the more the time pass the more the task are urgent and the more the company is willing to pay for the task.*

The model uses the following variables:

- x_{ij}^{tm} is the number of customers of type m that are in sourced node i and that are asked to do n_m tasks in sink node j, during the first stage of time t.
- $y_{ij}^{tm}(\omega)$ is the number of customers of type m that are in source node i and that are asked to do n_m tasks in sink node j, during the second stage of time t.

The stochastic model is

$$\text{minimize} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} c_{ij}^{tm} x_{ij}^{tm} + \mathbb{E}\Big[\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} q_{ij}^{tm} y_{ij}^{tm}(\omega)\Big] \tag{5.1}$$

subject to:

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{i=1}^{I} n_m (x_{ik}^{tm} + y_{ij}^{tm}(\omega)) \geq N_j \quad \forall \ j \in \mathcal{J} \quad \forall \ s \in \mathcal{S} \tag{5.2}$$

$$\sum_{j=1}^{J} x_{ij}^{tm} \leq \hat{\theta}_i^{tm} \quad \forall \ i \in \mathcal{I} \ \ t \in \mathcal{T} \ \ m \in \mathcal{M} \tag{5.3}$$

$$\sum_{j=1}^{J} (x_{ij}^{tm} + y_{ij}^{tm}(\omega)) \leq \theta_i^{tm}(\omega) \quad \forall \ i \in \mathcal{I} \ \ t \in \mathcal{T} \ \ m \in \mathcal{M} \tag{5.4}$$

$$x_{ij}^{tm} \in \mathbb{N} \quad \forall \; i \in \mathcal{I} \;\; j \in \mathcal{J} \;\; t \in \mathcal{T} \;\; m \in \mathcal{M}$$

$$y_{ij}^{tm}(\omega) \in \mathbb{N} \quad \forall \; i \in \mathcal{I} \;\; j \in \mathcal{J} \;\; t \in \mathcal{T} \;\; m \in \mathcal{M} \;\; \forall \;\; \omega$$

Even if the values of variables x_{ij}^{tm} and $y_{ij}^{tm}(\omega)$ are computed by solving the model, the company is supposed to implement only the decision variables x_{ij}^{0m} i.e. the decision variables of the first stage of the first time step. Once that these decisions are implemented, the company has to update the model and to solve it again.

Remark 10. *We assume that all the requests of the company are accepted and then performed by the customers. This hypothesis simplifies the model and it does not produce any practical problem because the company is going to implement only the first stage decision variables. Hence, if some customer does not accept or if someone accepts and then does not perform the deliveries, it is possible to rearrange the strategy in the second stage.*

Remark 11. *The proposed model is a two stages stochastic model even if the real information flow is more similar to a multi stages problem. The reason for this choice is that the company is going to implement only the decisions described by the first stage variables and the both two stages problems and multi stages problems uses the stages after the first one only for considering the future. Hence, since the multi stages problem considers many more variables than the two stages one, we use the two stages model as a reasonable approximation of the future and we postpone the study of the multi stages problem in future works. In other words, our policy is to have a more precise estimation of a wrong information flow instead of a very imprecise estimation of the real information flow.*

In order to complete the description of the model, we have to describe the distribution of the $\theta_i^{tm}(\omega)$. Then, for each time step and for each customer that the company considers (p) we define a random variable X_{ip}, such that

$$X_{ip} = \begin{cases} 1, \text{if customer } p \text{ is in the node } i \\ 0, \text{otherwise} \end{cases} .$$

For the sake of simplicity and without loss of generality, we consider these random variables to be independent. In fact, we are considering only customers that the company considers to be potential deliverers and we consider that each one of them is not influenced by the others. By this assumption, we have that for each time step, $\theta_i^{\cdot\cdot}(\omega) = \sum_p X_{ip}(\omega)$. Furthermore, $\theta_i^{\cdot\cdot}(\omega)$ is distributed as a Poisson binomial distribution because it is the sum of Bernoulli random variables not identically distributed. Since in this application we consider that the number of customers is greater than 100, we use an asymptotic result for the distribution of the sum of not identically distributed random variables. In particular, we use the Lyapunov central limit theorem (CLT) [8]. In order to introduce this theorem, we report the following definition.

Definition 9. *Given a sequence of independent random variables* $\{X_1, X_2, ...\}$ *such that* $\mathbb{E}[X_i] = \mu_i < \infty$, $\mathbb{E}[(X - \mu_i)^2] = \sigma_i^2 < \infty$ *and such that, for some* $\delta > 0$,

$$\lim_{n\to\infty} \frac{1}{s_n^{2+\delta}} \sum_{i=1}^{n} \mathbb{E}\Big[||X_i - \mu_i||^{2+\delta}, \Big] = 0 \tag{5.5}$$

where $s_n^2 = \sum_{i=1}^n \sigma_i^2$, *then we say that the Lyapunov's condition holds for the sequence* $\{X_1, X_2, ...\}$.

The Lyapunov CLT is the following.

Theorem 12. *If the Lyapunov's condition holds, then*

$$\frac{1}{s_n} \sum_{i=1}^{n} (X_i - \mu_i) \xrightarrow{d} \mathcal{N}(0,1).$$

We do not report the proof of the statement because it is not central in our discussion. The interested readers is referred to [8]. From Theorem 12, we prove the following corollary.

Corollary 13. *Given a set of Bernoulli distributed random variables* $\{X_1, X_2, ...\}$, *such that* $X_k \sim \mathcal{B}(p_k)$ *and such that* $p_k \neq 0, 1$ *for all* k, *then*

$$\frac{\sqrt{n}\frac{1}{n}\sum_{i=1}^{n}(X_i - p_i)}{\sqrt{\sum_{i=1}^{n} p_i(1-p_i)/n}} \xrightarrow{d} \mathcal{N}(0,1).$$

Proof. For each $X_k \sim \mathcal{B}(p_k)$, it holds that

$$1 \geq p_k(1-p_k) = \mathbb{E}[(X_k - p_k)^2] \geq \mathbb{E}[(X_k - p_k)^{2+\delta}].$$

Hence,

$$\frac{1}{s_n^{2+\delta}} \sum_{k=1}^{n} \mathbb{E}[(X_k - p_k)^{2+\delta}] \leq \frac{1}{s_n^{2+\delta}} \sum_{k=1}^{n} \mathbb{E}[(X_k - p_k)^2] \leq \frac{1}{s_n^{\delta}}.$$

If $p_k \neq 0$ and $p_k \neq 1$ (as requested by the corollary), then $s_n^{\delta} \to +\infty$, the Lyapunov's condition holds and Theorem 12 can be applied, from this it follows the result. □

Remark 14. *The assumption of Corollary 13 that imposes that $p_k \neq 0, 1 \ \forall k$ is not strict since by setting $p_k = 0$ or $p_k = 1$ we are imposing that a person is certainly or in node k or not in node k and both these assumptions are not good choices because of the Cromwell's rule (see [22]).*

Due to Corollary 13, we simulate the number of people in a node by using normal distributions.

Remark 15. *Corollary 13 is an important result because of the properties of the normal distribution. One of these good properties is that we can easily solve chance constraints. This opens the possibility to solve several other models. Nevertheless, we postpone this tractation in future works.*

In Chapter 7 we consider the fitting of this model by means of data coming from the mobile phone sensors.

5.2.1 Linear stochastic model

In order to solve the stochastic model (5.1)-(5.4), one possibility is to consider the associated linear problem that uses a fixed number of scenarios described by the set $\mathcal{S}$ which cardinality is S. The associated linear program is

$$\text{minimize} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{m=1}^{M} c_{ij}^{tm} x_{ij}^{tm} + \sum_{s=1}^{S} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{m=1}^{M} q_{ij}^{tm} y_{ij}^{stm} \tag{5.6}$$

subject to:

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{i=1}^{I} n_m(x_{ij}^{tm}+y_{ij}^{stm}) \geq N_j \quad \forall\ j \in \mathcal{J} \quad \forall\ s \in \mathcal{S} \tag{5.7}$$

$$\sum_{j=1}^{J} x_{ij}^{tm} \leq \hat{\theta}_i^{tm} \quad \forall\ i \in \mathcal{I}\ t \in \mathcal{T}\ m \in \mathcal{M} \tag{5.8}$$

$$\sum_{j=1}^{J}(x_{ij}^{tm}+y_{ij}^{stm}) \leq \hat{\theta}_i^{tm}+\theta_i^{stm} \quad \forall\ i \in \mathcal{I}\ t \in \mathcal{T}\ m \in \mathcal{M}\ s \in \mathcal{S} \tag{5.9}$$

$$x_{ij}^{tm} \in \mathbb{N} \quad \forall\ i \in \mathcal{I}\ j \in \mathcal{J}\ t \in \mathcal{T}\ m \in \mathcal{M}$$

$$y_{ij}^{stm} \in \mathbb{N} \quad \forall\ i \in \mathcal{I}\ j \in \mathcal{J}\ t \in \mathcal{T}\ m \in \mathcal{M}\ \forall\ s \in \mathcal{S}$$

The objective of the problem is to minimize the sum of the first stage rewards plus the expected sum of the second stage rewards. Constraints (5.7) enforce that all the tasks must be done in the considered time interval. Constraints (5.8) impose a limit to the first stage customers that can be contacted, while constraints (5.9) limit the second stage customers.

Remark 16. *With this notation we are considering that in each node there are tasks to do or customers available. If this is not the case, it is sufficient to split each node with both tasks and customers in two nodes (one with the customers and one with the tasks).*

Problem (5.6) - (5.9) is a two stages multi periods linear integer problem. In particular, the problem is a fixed recourse scenario problem since the recourse matrix does not depend on the scenario realization. Furthermore, for each first stage feasible decision the second stage problem has always at least one feasible solution. For this reason, we say that problem (5.6) - (5.9) is a complete recourse problem.

In order to set the number of scenarios to be considered, we use the concepts of in-sample stability and out-of-sample stability (for a comprehensive discussion

about these concepts we refer to [47] and to [50]). In the following, we discuss these two concepts in a general setting. We consider the general stochastic optimization problem

$$\min \mathbb{E}_{\mathcal{T}}[f(x)] \tag{5.10}$$

where $f(\cdot)$ is a general objective function and $\mathcal{T}$ is the scenario tree considered for the computation of the expected value. We denote with x^* the general optimal solution to the problem (5.10). Even if it is not explicitly reported, x^* depends by the choice of $\mathcal{T}$.

In-sample stability considers the stability of the optimal value of the objective functions obtained by solving problems characterized by different scenarios. By calling $\mathcal{T}_1$ and $\mathcal{T}_2$ two possible scenario trees we said that the problem is in-sample stable if and only if

$$\hat{f}(x_1^*, \mathcal{T}_1) \approx \hat{f}(x_2^*, \mathcal{T}_2), \tag{5.11}$$

where

$$\hat{f}(x_i^*, \mathcal{T}_i) = \mathbb{E}_{\mathcal{T}_i}[f(x_i^*)],\ i = 1, 2, \tag{5.12}$$

i.e. $\hat{f}(x_i^*, \mathcal{T}_i)$ is the expected value of the objective function computed by using the optimal solution x_i^* and the scenario tree $\mathcal{T}_i$.

In order to obtain a measure between 0 and 1 of the in-sample stability, we use the following definition.

Definition 10. *We call relative in-sample stability the ratio*

$$\frac{\hat{f}(x_2^*, \mathcal{T}_2) - \hat{f}(x_1^*, \mathcal{T}_1)}{(\hat{f}(x_2^*, \mathcal{T}_2) + \hat{f}(x_1^*, \mathcal{T}_1))/2}. \tag{5.13}$$

Out-of-sample stability checks if the different solutions to the linear problem associated to the stochastic one perform similarly when tested with the real distribution. In formula

$$\hat{f}(x_j^*, \omega) \approx \hat{f}(x_i^*, \omega), \tag{5.14}$$

where

$$\hat{f}(x_i^*, \omega) = \mathbb{E}_P[f(x_i^*)] \tag{5.15}$$

and P is the real probability measure. In order to compute $\mathbb{E}_P[f(x_i^*)]$ we have to solve a difficult integral, for this reason we approximate it by using a Monte Carlo approach (we consider a scenario tree with several more scenarios than the one used for computing the solution). As above, we define a relative measure of the out-of-sample stability by using the following definition.

Definition 11. *We call out-of-sample relative stability the ratio*

$$\frac{\hat{f}(x_j^*,\omega)-\hat{f}(x_i^*,\omega)}{(\hat{f}(x_j^*,\omega)+\hat{f}(x_i^*,\omega))/2}. \tag{5.16}$$

It is worth noting that by increasing the number of scenarios, the number of variables of the model increases too. For this reason, we have to justify the need of solving the stochastic version of the problem instead of the deterministic one. The most effective (and widely used) way to quantify the loss that the company has to face by implementing the deterministic solution is to consider the value of stochastic solution (VSS). The interested reader finds an in depth discussion of the topic in [47].

By calling $x^* = \min\mathbb{E}[f(x)]$ and $\bar{x}^* = \min f(\mathbb{E}[x])]$, we define $VRP = \mathbb{E}[f(x^*)]$ the value of the recourse problem and $EVS = \min f(\mathbb{E}[x])$ the expected value solution.

The value of the stochastic solution is computed as

$$VSS = EVS - VRP = \min E[f(x)] - \min f(E[x]).$$

If the VSS is small then the mean value problem is a good approximation of the stochastic one. Instead, if the VSS is big then it justifies the consideration of the stochastic nature of the problem. In other words, the VSS measures how much the expected value problem is worst than the solution to the stochastic problem.

As for the in-sample and out-of-sample stability, we define a relative VSS as follows

Definition 12. *We call relative VSS the ratio*

$$VSS_r = \frac{EVS - RP}{RP}. \tag{5.17}$$

In Section 5.5, we compute the in-sample stability ratio, the out-of-sample stability ratio and the relative VSS for a set of benchmark instances.

Remark 17. *This model is also suitable for companies that have a fleet of heterogeneous vehicles which availability is uncertain.*

Remark 18. *Even if the condition*

$$\sum_i n_m \theta^s = \sum j N_j$$

is not satisfied, it is possible to create an equivalent problem that respects such a condition. It is possible to do so by adding a sink node such that

$$\sum_{j \in \mathcal{J} \cup j_0} N_j \geq \sum_i n_m \theta^s$$

and by adding a source node such that its θ^s generate in all the scenario a feasible solution. Finally, we set the resources of this new sink nodes to be the most expensive with respect to all the other costs.

5.3 Complexity Analysis

In this section we analyse the complexity of Problem (5.6)-(5.9). We split this study in two parts, in the first one we consider the problem for one type of customer, while in the second one we consider to have more types of customers. In the rest of the section, for the sake of simplicity and without loss of generality, we consider the problem with $T = 1$, this assumption does not change the problem (see Remark 5).

Let us consider the problem with one type of customer, the mathematical model that we obtain is the following

$$\text{minimize} \sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij}x_{ij} + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J} p_s q_{ij} y_{ij}^{s} \tag{5.18}$$

subject to:

$$\sum_{j=1}^{J} x_{ij} \leq \hat{\theta}_i \quad \forall \ i \in \mathcal{I} \tag{5.19}$$

$$\sum_{j=1}^{J} (x_{ij} + y_{ij}^{s}) \leq \theta_i^{s} \quad \forall \ i \in \mathcal{I} \tag{5.20}$$

$$\sum_{i=1}^{I} (x_{ij} + y_{ij}^{s}) \geq \left\lceil \frac{N_j}{n_m} \right\rceil \quad \forall \ j \in \mathcal{J} \quad \forall \ s \in \mathcal{S} \tag{5.21}$$

$$x_{ij} \in \mathbb{N} \quad \forall \ i \in \mathcal{I} \ \ j \in \mathcal{J}$$

$$y_{ij}^{s} \in \mathbb{N} \quad \forall \ i \in \mathcal{I} \ \ j \in \mathcal{J} \ \ \forall \ s \in \mathcal{S}$$

Before to compute the complexity of problem (5.18)-(5.21), we first need to recall some theoretical properties and results.

Definition 13. *A matrix is totally unimodular (TU) if every square non-singular submatrix has determinant* $+1$*,* -1 *or* 0*.*

Remark 19. *By definition 13, a totally unimodular matrix must have all components* $+1, 0$ *or* -1 *because each element of the matrix can be seen as a square submatrix.*

It can be proved, see [83], that integer problems having a TU constraints matrix and an integer constraints vector have the so-called integer properties i.e. the solution to the continuous relaxation of the problem is integer. From this fact, it follows that solving the integer problem or solving its continuous relaxation provides the same solution. It is important to note that this property is preserved under some operations.

Lemma 20. *Given a TU matrix A, and a vector v with all zeros but one nonzero being ± 1, then by adding v as a row or as a column of the matrix A, the TU property is preserved.*

Proof. See [83]. □

Theorem 21. *Given a matrix A with components $0, \pm 1$, it is TU if and only if for every subset of columns $\mathcal{V} \subset \{1, \ldots, n\}$, there exists a partition $(\mathcal{V}_1, \mathcal{V}_2)$ of $\mathcal{V}$ such that*

$$|\sum_{j \in \mathcal{V}_1} a_{ij} - \sum_{j \in \mathcal{V}_2} a_{ij}| \leq 1 \quad \forall i = 1, \ldots, m. \tag{5.22}$$

Proof. See [33]. □

Theorem 22. *Given $A \in \mathbb{R}^{m \times N}$ a matrix of the form*

$$A = \begin{bmatrix} T^1 & W^1 & 0 & 0 \\ T & 0 & W & 0 \\ \ldots & \ldots & \ldots & \ldots \\ T^K & 0 & 0 & W^K \end{bmatrix}, \tag{5.23}$$

it is TU if and only if for every column subset $\mathcal{V} = \mathcal{V}_0, \mathcal{V}_1, \ldots, \mathcal{V}_K \subseteq 1, ..., N$ there exists a partition $(\mathcal{V}_1, \mathcal{V}_2) = (\mathcal{V}_0^1, \mathcal{V}_1^1, \ldots, \mathcal{V}_K^1), (\mathcal{V}_0^2, \mathcal{V}_1^2, \ldots, \mathcal{V}_K^2)$ such that

$$\begin{aligned} &|\sum_{j \in \mathcal{V}_0^1} t_{ij}^k + \sum_{j \in \mathcal{V}_k^1} w_{ij}^k - \sum_{j \in \mathcal{V}_0^2} t_{ij}^k - \sum_{j \in \mathcal{V}_k^2} w_{ij}^k| \leq 1 \\ &\forall i = 1, \ldots, m, \quad \forall \quad k = 1, \ldots, K \end{aligned} \tag{5.24}$$

Proof. See [53]. □

Theorem 23. *Let A be a TU matrix. Then, for any integral right-hand-side b, the polyhedron*

$$P = \{x : Ax \leq b, x \geq 0\} \tag{5.25}$$

has integral vertexes.

Proof. See [82] □

Theorem 24. *Given $A \in \mathbb{R}^{m \times n}$ a matrix of the form*

$$A = \begin{bmatrix} \vdots & W^1 & 0 & 0 \\ \vdots & 0 & W^2 & 0 \\ T & \dots & \dots & \dots \\ \vdots & 0 & 0 & W^K \end{bmatrix}, \tag{5.26}$$

it is TU if and only if all the matrices T and W^i for $i = 1, 2, \dots, K$ are TU.

Proof. Let us consider a set of column $J \in \{1, \dots, n\}$. We call J_T the indexes of the columns belonging to matrix T and J_{W^k} the indexes of the columns belonging to matrix W^k. By Theorem 21 and from the hypothesis of this theorem, $\exists J_T^1,\ J_T^2$ such that $|\sum_{j \in J_T^1} T_{ij} - \sum_{j \in J_T^2} T_{ij}| \leq 1\ \forall i = 1, \dots, m$. Hence, since the coefficient of $T \in \{-1, 0, 1\}$ then $\sum_{j \in J_T^1} T_{ij} - \sum_{j \in J_T^2} T_{ij}| \in \{-1, 0, 1\}\ \forall i = 1, \dots, m$. It is worth noting that the same holds for each J_{W^k} and that by simply changing the name of the sets it is possible to change the sign of the results. Hence, given a choice of the names of the sets $J_T^1, J_{W^1}^1, \dots, J_{W^K}^1, J_T^2, J_{W^1}^2, \dots, J_{W^K}$ such that

$$\begin{aligned} &\sum_{j \in J_T^1} T_{ij} - \sum_{j \in J_T^2} T_{ij} + \sum_{j \in J_{W^1}^1} W_{ij}^1 - \sum_{j \in J_{W^1}^2} W_{ij}^1 + \dots \\ &+ \sum_{j \in J_{W^K}^1} W_{ij}^K - \sum_{j \in J_{W^K}^2} W_{ij}^K \in \{-1, 0, 1\}, \end{aligned} \tag{5.27}$$

we can define a split $J^1 = J_T^1 \cup J_{W^1}^1 \cup \dots \cup J_{W^K}^1$, $J^2 = J_T^2 \cup J_{W^1}^2 \cup \dots \cup J_{W^K}^2$ such that Equation 5.24 holds. □

Theorem 25. *Problem (5.6)-(5.9) can be solved in polynomial time for $M = 1$*

In order to prove Theorem 25, for the sake of simplicity and without loss of generality, we drop the time index (by replacing the set $\mathcal{I}$ with the set $\mathcal{I} \times \mathcal{T}$). The model with this simplification and with $M = 1$ can be written as follows

$$\text{minimize} \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} x_{ij} + \sum_{s=1}^{S} p_s \sum_{i=1}^{I} \sum_{j=1}^{J} q_{ij} y_{ij}^s \tag{5.28}$$

subject to

$$\sum_{i=1}^{I}(x_{ij}+y_{ij}^{s}) \geq \left\lceil \frac{N_j}{n} \right\rceil \quad \forall\ j\ \in\ \mathcal{J} \quad \forall\ s\ \in\ \mathcal{S} \tag{5.29}$$

$$\sum_{j=1}^{J} x_{ij} \leq \hat{\theta}_i \quad \forall\ i\ \in\ \mathcal{I} \tag{5.30}$$

$$\begin{gathered}\sum_{j=1}^{J}(x_{ij}+y_{ij}^{s}) \leq \theta_i^{s} \\ \forall\ i\ \in\ \mathcal{I}\ \ s\ \in\ \mathcal{S}\end{gathered} \tag{5.31}$$

$$x_{ij} \in \mathbb{N} \quad \forall\ i\ \in\ \mathcal{I}\ \ j\ \in\ \mathcal{J}$$

$$y_{ij}^{s} \in \mathbb{N} \quad \forall\ i\ \in\ \mathcal{I}\ \ j\ \in\ \mathcal{J}\ \ \forall\ s\ \in\ \mathcal{S}$$

where $\lceil\cdot\rceil$ denote the ceiling function and n is the number of tasks performed by the considered type.

Proof. In order to prove that the problem (5.28)-(5.31) can be solved in polynomial time, it is sufficient to prove that the matrix generated by the constraints is totally unimodular (since the vector right end side of all the equations is integer). By considering the variables in the following order

$$\begin{aligned}&[x_{11},\ldots,x_{1J},\ y_{11}^{1},\ldots,y_{1J}^{1},z_{11}^{1},\ldots,z_{1J}^{1}\ldots,\\ &y_{11}^{S},\ldots y_{1J}^{S},\ldots,z_{11}^{S},\ldots,z_{1J}^{S}\\ &\ldots\\ &x_{I1},\ldots,x_{IJ},\ y_{I1}^{1},\ldots,y_{IJ}^{1},z_{I1}^{1},\ldots,z_{IJ}^{1},\ldots,\\ &y_{I1}^{S},y_{IJ}^{S},\ldots,z_{I1}^{S},\ldots,z_{IJ}^{S}]^{T},\end{aligned} \tag{5.32}$$

we have that the matrix of the constraints can be written as $A \in \mathbb{R}^{SJ+I(S+1)\times IJ(2S+1)}$

$$A = \left[\begin{array}{c|c|c|c|c} L & L & \ldots & \ldots & L \\ \hline D & 0 & \ldots & \ldots & 0 \\ \hline 0 & D & \ldots & \ldots & 0 \\ \hline \ldots & \ldots & \ldots & \ldots & \ldots \\ \hline 0 & 0 & \ldots & \ldots & D \end{array}\right], \tag{5.33}$$

where $L \in \mathbb{R}^{SJ \times J(2S+1)}$

$$L = \begin{bmatrix} -I & -I & 0 & \dots \\ -I & 0 & -I & 0 \\ \dots & \dots & \dots & \dots \\ -I & 0 & 0 & -I \end{bmatrix}, \tag{5.34}$$

with I the identity matrix of $\mathbb{R}^{I \times I}$ and $D \in \mathbb{R}^{S+1 \times J(2S+1)}$

$$D = \begin{bmatrix} 1_J & 0 & 0 & 0 \\ 1_J & 1_J & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 1_J & 0 & \dots & 1_J \end{bmatrix}, \tag{5.35}$$

with $1_J = [1 \dots 1] \in \mathbb{R}^J$

The first row of matrix A i.e. $[L, L, \dots, L]$ comes from constraints (5.29). Instead, each of the next rows considers the matrix D for different sources. In particular, the first row of matrix D comes from constraint (5.30), while the others come from constraints (5.31) for all s.

First, let us consider matrix L, it can be constructed from the upper left identity matrix by adding vectors with one non zero component equals to -1. In particular, first add the left rows to complete the first column of identity matrices of L, then add the remaining columns one by one (also these columns are made by one -1 and all zeros). Then, by applying Lemma 20 in each iteration, we can state that L is TU and with the same argument we can state that all the first row of matrix A i.e. $[L, L, \dots, L]$ is TU.

Then, let us consider matrix D. We call B the submatrix composed by the left first I columns. B has rank 1 because every square submatrix which dimension bigger than a single element has determinant zero (there exists a 1-minor which does not equal 0). Hence, the only nonsingular submatrices are composed by single elements that are all equal 1. Hence, B is TU. From this observation and by using Lemma 20 for adding to B the remaining columns (from $(I+1)$-th to the $[I(2S+1)]$-th) we prove that the matrix D is TU.

Since A is TU iff A^T is TU, all the matrices composing A are TU and A^T has the form of the matrix in Theorem 24, we can apply this theorem for proving that A is TU.

Since the coefficient matrix is TU and since for $M = 1$ problem (5.6) - (5.9) can be written as (5.28) - (5.31) hence it has integer constant terms, we can apply Theorem 23 and conclude that the continuous solution to problem (5.6) - (5.9) is integer. For this reason, it can be solved in polynomial time by using, for examples, interior point methods. □

If the problem considers more than one type customers, we have the following result.

Theorem 26. *Problem (5.6)-(5.9) is NP-Hard in the strong sense for $M \geq 2$*

Proof. Let us consider the simplified version of the problem with three sinks ($J = 3$), one scenario ($S = 1$) and two types of customers ($M = 2$). We suppose that there is not integer k such that $n_1 = kn_2$ and $c_{ij}^1 = kc_{ij}^2$ (if such a k exists, then the problem is reducible to another with $M = 1$). By relaxing constraints (5.9), we obtain the min - formulation of the multiple integer knapsack problem that is NP-hard in the strong sense (see [61]). □

5.4 The LRCVF heuristic

The LRCVF heuristic, first introduced in [15] is composed by two steps

1. solve the continuous relaxation of the problem,
2. set to their lower bounds all the variables with reduced costs greater than a threshold.

The threshold is selected to be a fixed quantile of the reduced costs obtained in the first step. In this way, it is possible to control the number of variables set to zero. In the following, we use $LRCVF_\alpha$ in order describe the $LRCVF$ heuristic, where the threshold is selected to be the α quantile of the reduced costs. During the numerical experiments we notice that by setting different

quantiles for the variables of the two stages we achieve better results. In rest of the chapter we call this new heuristic $LRCVF_{\alpha;\beta}$, where α is the quantile selected for the variables of the first stage and β is the quantile selected for the variables of the second stage.

The choice of the two quantiles add a parameter to the heuristic. During the experiments we found that $\alpha < \beta$ is the best choice. This is reasonable since the number of variables in the second stage is far greater than the ones in the first stage. In the general problem, we suggest to consider the nature of the problem or to try both the alternatives.

5.5 Numerical Simulations

In this section, we present the results of the numerical simulations. The main objectives are to justify the effort for solving the stochastic problem, to find the number of scenarios under which the solution is stable and to evaluate the performance of the LRCVF heuristic. We do it by using several benchmark instances with different ammounts of sources, sinks (I and J), costumers types (M), time steps (T) and different ratios between number of sources and number of sinks. In all the considered instances, each scenario has the sum of the available resources equals to the number of deliveries to perform.

In all the experiments, we use as exact solver the commercial solver Gurobi.

In the instances we consider 3 different kinds of customers:

- $m = 0$: the standard customers. They perform 1 delivery, their number is stochastic.
- $m = 1$: the business customers. They perform 3 deliveries, their number is stochastic.
- $m = 2$: the workforce of the company. They perform 10 deliveries and their number is known in advance.

For each kind of customer we compute the costs for the deliveries to be

$$c_{ij}^{tm} = |\frac{i-j}{4} + 1|C\log(2(m+1)), \tag{5.36}$$

where C is a realization of a uniform random variable distributed between $C_{\min}$ and $C_{\max}$. As in Chapter 4, we set $C_{\min} = 2$ and $C_{\max} = 5$.

The graph network that we consider is a complete and connected graph and each node is a sources or a sinks. In order to control the number of sources and the number of sinks we define the parameter ρ to be the ratio between the number of sources and the number of sinks. In the considered instances the networks have $\rho = 0.4$ or $\rho = 0.8$. Once that the structure of the network is fixed, we define the number of people in each node by rounding a realization of a normal distribution (as described by Corollary 13). Since we do not have data to fit the normal distributions, we arbitrarily fix their averages and their variances. In order to ensure that the problem has at least one feasible solution we check that

$$\sum_i \sum_m \sum_t n_m(\hat{\theta}_i^{tm} + \theta_i^{stm}) \geq \sum_i N_i. \tag{5.37}$$

If (5.37) does not hold, we add people to a random set of nodes until it holds. All the experiments in the following subsections are performed on an Intel R CoreTMi7-5500U CPU @2.40 Ghz with 8 GB RAM and Microsoft R WindowsTM10 Home installed.

5.5.1 Stability

In this subsection we address the problem of stability, i.e. we find the minimum number of scenarios such that the in-sample and out-of-sample stability are below a given threshold. Table 5.1 shows the minimum number of scenarios such that the relative in-sample (see Eq. (5.13)) and the relative out-of-sample stability (see Eq. (5.16)) are smaller then 0.01.

In Table 5.1, we do not have results for instances considering more than 100 nodes because the solver runs out of memory.

Table 5.1 For several combination of parameters (columns I, M, T and ρ), the table shows the average and the standard deviation of the in-sample andout-of-sample stability. Experiments have been repeated 50 times.

Instance parameters				In Sample		Out of Sample		# Scenarios
I	M	T	ρ[%]	mean-value	Std Dev	mean-value	Std Dev	
30	3	1	0.4	0.0421724	0.0479721	0.0662104	0.0261267	22
30	3	1	0.8	0.066621	0.0904609	0.0967704	0.0358688	20
30	3	10	0.4	0.00626975	0.0129024	0.0951536	0.0961721	21
30	3	10	0.8	0.0114897	0.00912324	0.0121968	0.00735419	20
30	3	20	0.4	0.0194175	0.007261	0.0139308	0.00794815	25
30	3	20	0.8	0.0983748	0.009983	0.0253745	0.00817391	24
100	3	1	0.4	0.0937461	0.001736	0.0629179	0.00026482	28
100	3	1	0.8	0.0918378	0.001232	0.0616498	0.00029837	27

It is important to note that from the results of Table 5.1 it follows that the smallest number of scenarios leading to convergence to all the instances with 30 nodes is 25 scenarios. Instead, for instances with 100 nodes 28 scenarios are needed.

As expected, theout-of-sample stabilities are greater than the in-sample ones. The confidence intervals are computed by using mean and standard deviation reported in Table 5.1. In most of the cases, 0 belongs to the interval with a low confidence. For this reason, in the following examples we consider 30 scenarios (i.e. we set $S = 30$).

5.5.2 Value of Stochastic Solution

In this subsection we study the value of stochastic solution by computing the VSS_r (see Eq. 5.17) for several instances characterized by different combinations of the parameters.

Remark 27. *For each experiment we have considered a number of scenarios such that the solution is stable i.e. the number of scenarios reported in Table 5.1.*

The results of these experiments are shown in Table 5.2. As the reader can notice, the instances with $I = 30$, $M = 3$, $T = 20$ and the instances with

$I = 100$, $M = 3$, $T = 1$ produce VSS_r greater than the 100%. Furthermore, in all the experiments the minimum value of VSS_r is 48% and the VSS_r increases as the dimensions of the problem increase.

Table 5.2 For each combination of parameters (columns I, M, T and ρ), the table shows the average VSS (column mean value) and the standard deviation (column Std Dev). Experiments have been repeated 50 times.

Instance parameters				VSS	
I	M	T	ρ[%]	mean value	Std Dev
30	3	1	0.4	0.482051	0.245137
30	3	1	0.8	0.444769	0.188483
30	3	10	0.4	0.983473	0.222917
30	3	10	0.8	0.851779	0.109818
30	3	20	0.4	3.84889	1.6708
30	3	20	0.8	3.80675	1.36582
100	3	1	0.4	4.1426	0.47008
100	3	1	0.8	5.4469	0.55

As in Subsection 5.5.1, we obtain confidence intervals for these values of VSS_r. These intervals contain the value 0 only if we consider quantile close to one. For this reason, we state that it is very unlucky to have a null VSS_r. This result is even more important since it holds for each observation, hence we for multiple hypothesis testing (see [3] for a deeper discussion) it is very unluckily that $VSS_r = 0$.

The main difference between the deterministic solution and the stochastic one is the number of costumers of type 3 used. While the stochastic solution uses them in the first stage, the expected value solution does not use them at all. Another difference between the two types of solution is that the stochastic one uses more customers during the first stage.

On average, the cost of the first stage is the 43% of the total cost. This implies that the online management of the activities is more expensive than the planning activities.

5.5.3 Heuristic Approach

The goal of this subsection is to measure the performance of the LRCVF heuristic on several instances.

Remark 28. *For each experiment we have considered a number of scenarios such that the solution is stable i.e. the number of scenarios reported in Table 5.1.*

The results in term of time and solution obtained are shown in table 5.3. As the reader can notice, in all the considered instances the heuristic is able to find the optimal solution and to reduce the computational time. On average this reduction is of the 53%.

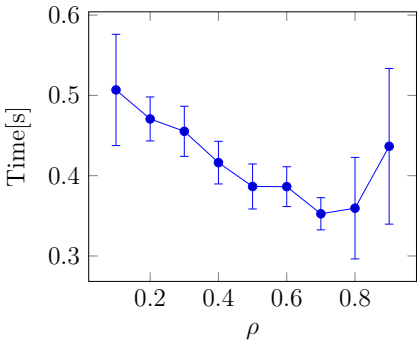

Figure 5.1 Time for obtaining the optimal solution with the exact solver Gurobi versus different values of ρ. The vertical lines represent the standard deviation of the values.

The values *n.p.* in Table 5.3 are used to indicate a value that is not present. They characterized instances which optimal solution cannot be computed because the exact solver runs out of memory.

It is important to notice that the LRCVF heuristic has good performance, This mean that to use the reduced costs of the deterministic solution leads to identify with a good accuracy the variable to keep and te ones to exclude in the stochastic framework.

As the reader can notice, the optimal value of the objective function decreases as the number of time steps increases. This characteristic is reasonable because, the more time the solver has to solve the problem, the better it can perform.

Instead, as the number of time steps increases or as the number of nodes increase, the computational time also increases. Of particular importance is how the parameter ρ influences the computational time of the algorithm. We plot the computational time for finding the optimal solution to an instance with 30 nodes versus the value of ρ in Figure 5.1. As the reader can notice, the instances generated by considering extreme values of ρ require more time to be solved.

Table 5.3 For each combination of the parameters (columns I, M, T and ρ), the table shows the computational time and the optimal objective function value. The value *n.p.* means not present and it is reported for the instances in which Gurobi produces an out of memory exception.

Instance parameters				Gurobi		Continuous		$LRCVF_{0.6;0.9}$		$LRCVF_{0.6;0.6}$	
I	M	T	ρ[%]	time[s]	sol	time[s]	sol	time[s]	sol	time[s]	sol
30	3	1	0.4	7.454	10.51	0.693	10.027	2.699	10.51	3.174	10.51
30	3	1	0.8	23.068	13.0833	0.716	12.3647	2.898	13.0833	13.27	13.0833
30	3	10	0.4	49.137	1.19	8.973	1.0725	12.489	1.19	34.675	1.19
30	3	10	0.8	40.734	2.73	8.721	2.59	16.628	2.73	29.15	2.73
30	3	20	0.4	72.794	1.05	19.843	0.981333	39.084	1.05	57.48	1.05
30	3	20	0.8	112.239	0.88	26.031	0.829667	45.653	0.88	75.553	0.88
100	3	1	0.4	511.75	22.2367	60.179	21.086	174.662	22.2167	238.919	22.2
100	3	1	0.8	514.65	64.8033	58.704	62.3223	188.209	64.8033	340.887	64.77
100	3	5	0.4	*n.p.*	*n.p.*	158.746	10.1793	379.191	10.8633	835.176	10.8533
100	3	5	0.8	*n.p.*	*n.p.*	124.496	42.3547	360.023	44.29	383.682	44.2

Progressive Hedging

In Chapter 5, we state that in real instances, the biggest networks can have 1000 nodes. With such a number of nodes, it is possible to compute a solution in a reasonable amount of time only if one type of customers is considered (i.e. $M = 1$) thanks to Theorem 24.

If, instead, $M > 1$ the LRCVF heuristic defined in Chapter 5 is effective up to instances of medium size i.e. 500-700 nodes. For this reason, we need to develop a new heuristic algorithm in order to deal with bigger instances. In particular, we propose a customized version of the Progressive Hedging (PH) heuristics (proposed for the first time in [100]).

This chapter is organized as follows: in Section 6.1, we briefly describe the PH algorithm, in Section 6.2 we describe the results of the PH as described in [100]. Finally, in Section 6.3 we proposed a customized version of the PH and we test its performance on several random instances. With the present chapter we conclude the study of the optimization of crowd-shipping.

6.1 Background

PH is a heuristic based on the augmented Lagrangian relaxation of the non anticipative constraints of the stochastic problem. Since, problem (5.6) - (5.9) has these constraints incorporated in the choice of the variables, we have to exploit them. Hence, we substitute variables x_{ij}^{tm} with variables x_{ij}^{stm} and we

add the non anticipative constraints, i.e.

$$x_{ij}^{stm} = x_{ij}^{s'tm} \ \forall \ s \neq s', i, j, t, m. \tag{6.1}$$

Constraints (6.1) impose that all the first stage variables with the same indexes i, j, t and m must be equal independently of the scenario to which they belong. If constraints (6.1) are not imposed, then the information about the scenario can be exploited by the first stage variables i.e. we allow solutions that require the knowledge of the future to be implemented.

Remark 29. *Constraints* (6.1) *are equivalent to*

$$x_{ij}^{s'tm} = \sum_{s=1}^{S} p_s x_{ij}^{stm} \ \forall s', i, j, t, m. \tag{6.2}$$

PH consider the stochastic problem with constraints (6.2) been relaxed i.e.

$$\begin{aligned} \text{minimize} & \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} c_{ij}^{tm} x_{ij}^{tm} + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} q_{ij}^{stm} y_{ij}^{stm} + \\ & + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} w_s |x_{ij}^{stm} - \bar{x}_{ij}^{tm}| \end{aligned} \tag{6.3}$$

subject to:

$$\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{i=1}^{I} n_m (x_{ij}^{tm} + y_{ij}^{stm}) \geq N_j \quad \forall \ j \in \mathcal{J} \quad \forall \ s \in \mathcal{S} \tag{6.4}$$

$$\sum_{j=1}^{J} x_{ij}^{tm} \leq \hat{\theta}_i^{tm} \quad \forall \ i \in \mathcal{I} \ t \in \mathcal{T} \ m \in \mathcal{M} \tag{6.5}$$

$$\sum_{j=1}^{J} (x_{ij}^{tm} + y_{ij}^{stm}) \leq \theta_i^{stm} \quad \forall \ i \in \mathcal{I} \ t \in \mathcal{T} \ m \in \mathcal{M} \ s \in \mathcal{S} \tag{6.6}$$

$$x_{ij}^{stm} \in \mathbb{N} \quad \forall\ i \in \mathcal{I}\ \ j \in \mathcal{J}\ \ t \in \mathcal{T}\ \ m \in \mathcal{M}\ \ \forall\ s \in \mathcal{S}$$
$$y_{ij}^{stm} \in \mathbb{N} \quad \forall\ i \in \mathcal{I}\ \ j \in \mathcal{J}\ \ t \in \mathcal{T}\ \ m \in \mathcal{M}\ \ \forall\ s \in \mathcal{S}.$$

where

$$\bar{x}_{ij}^{tm} = \sum_{s=1}^{S} p_s x_{ij}^{stm},$$

and w_s are the penalty terms for the difference between x_{ij}^{stm} and $\bar{x}_{ij}^{tm}$

Remark 30. *It can be notice that problem (6.3)-(6.6) can be split in S independent problems one for each scenario s. Hence, by relaxing constraints (6.2), the scenario tree changes as shown in Figure 6.1.*

In order to increase the speed of convergence of the Lagrangian method, we add a quadratic penalty term that yields to an augmented Lagrangian method. By removing constant terms, the problem becomes

$$\begin{aligned}\text{minimize} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} c_{ij}^{tm} x_{ij}^{tm} + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} q_{ij}^{stm} y_{ij}^{stm} + \\ + \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} w_s x_{ij}^{stm} + \frac{\rho}{2}\sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\sum_{m=1}^{M} (x_{ij}^{stm} - \bar{x}_{ij}^{tm})^2\end{aligned} \tag{6.7}$$

subject to (6.4), (6.5) and (6.6).

By exploiting Remark 30, it is possible to define the augmented Lagrangian algorithm shown in Algorithm 4, namely the PH.

The algorithm is organized in two steps: in the first one the solutions of all the scenario problems are found, in the second one these solutions are used to update $\bar{x}_{ij}^{tm}$. These two steps are repeated iteratively until convergence i.e. the difference between $\bar{x}_{ij}^{tm}$ and x_{ij}^{stm} is smaller than a threshold ϵ in all the scenario s.

Definition 14. *If it holds that*

$$x_{ij}^{stm} = \bar{x}_{ij}^{tm} \quad \forall\ s\ \in\ \mathcal{S}, \tag{6.8}$$

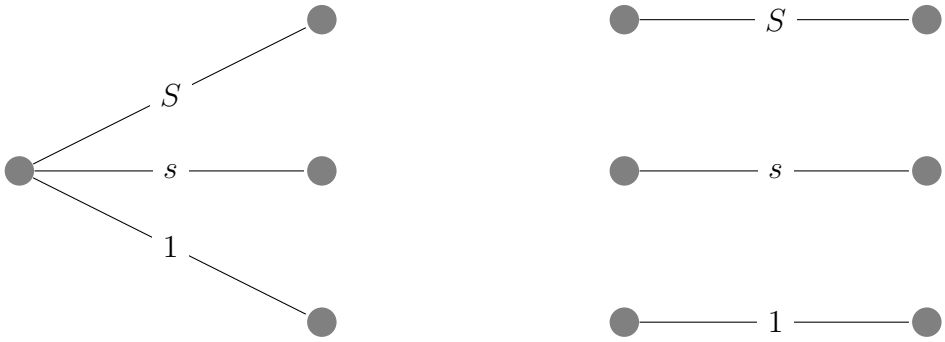

Figure 6.1 On the left the scenario tree of a three-stage problem , on the right the problem considering each scenarios based on [46] and [36] (each circle denotes a decision).

Algorithm 4 Progressive Hedging

k:=0
for each scenario $s \in \mathcal{S}$ **do**
    Solve:

$$x_s^{(k)} := \text{argmin}_{x,y_s}\, c^t x + f_s^T y_s \;:\; (x, y_s) \in \mathcal{Q}_s$$

end for
$\bar{x}^{(k)} := \sum_{s\in\mathcal{S}} p_s x_s^{(k)}$
$w_s^{(k)} := \rho(x_s^{(k)} - \bar{x}^{(k)})\ \forall\ s\ \in\ \mathcal{S}$
while $g^{(k)} < \epsilon$ **do**
    k:=k+1
    for each scenario $s \in \mathcal{S}$ **do**
        Solve:

$$x_s^{(k)} := \text{argmin}_{x,y_s}\, c^t x + w_s^{(k-1)} x + \frac{\rho}{2}\left\| x - \bar{x}^{(k-1)} \right\|_2^2 + f_s^T y_s \;:\; (x, y_s) \in \mathcal{Q}_s$$

    end for
    $\bar{x}^{(k)} := \sum_{s\in\mathcal{S}} p_s x_s^{(k)}$
    $w_s^{(k)} := w_s^{(k-1)} + \rho(x_s^{(k)} - \bar{x}^{(k)})\ \forall\ s\ \in\ \mathcal{S}$
    $g^{(k)} = \sum_{s\in\mathcal{S}} p_s \|x_s^{(k)} - \bar{x}_s^{(k)}\|$
end while

we say that the variable x_{ij}^{stm} has reached the consensus.

The PH algorithm provably converges in linear time when the decision variables are continuous. While, for the integer case some instances have proved that the convergence to the optimal solution is not guarantee (see [100]).

6.2 Exact Progressive Hedging Results

In this section, we compare the performance of the PH with the one of the commercial solver Gurobi. Since, there are not yet theoretical results on the convergence (to a general solution) of the PH algorithm in integer cases we implement a stopping criteria based on maximum limit of CPU time and number of iterations. If these limit are overcome, we then solve the original problem by fixing all the variables for which consensus has been reached. For these experiments, we set the maximum CPU time to be one hour and we set 200 to be the maximum number of iterations. Nevertheless, in all the instances that we have generated, these limits are never reached. Except for this little change, we have considered the PH as described in Algorithm 4 by setting $\epsilon = 10^{-5}$ and $\rho = 10^{-3}$ because we have found empirically that they perform well. In particular, for greater values of ρ, the PH starts cycling i.e. the objective value of the solution assume periodically the same values and the solutions found by the algorithm are periodically the same.

Remark 31. *We recall that by Remark 9 the first stage costs are smaller than or equal to the costs of the second stage. This reasonable assumption improves the speed of convergence of Algorithm 4 because the problem does not found initial solutions with all null first stage variables.*

In order to test the performance of the two algorithms, we generate several instances as described in Section 5.5. The results are shown in Table 6.1. For several combinations of number of sources (I), sinks(J), customers types (M), time periods (T) and ratios between sources and sinks (ρ), we generate and solve with the PH and with the exact method 50 instances. The result of this comparison are shown in Table 6.1.

Table 6.1 The table shows the average time of the exact method (Time Ex. [s]), the average time of the PH (Time PH [s]), the gap between the optimal solution and the one found by the PH (Gap[%]) and the average number of iterations that the PH requires in order to converge, for different values of number of nodes (I), number of customer types (M), time period (T) and ratios between sources and nodes (ρ). All the averages are computed on 50 randomly generated instances.

I	M	T	ρ	Time Ex. [s]	Time PH [s]	Gap[%]	N Iter
30	3	1	0.4	7.454	18.01	0.77	12.3
30	3	1	0.8	23.068	12.43	0.56	18.6
30	3	10	0.4	49.137	24.12	0.75	7.3
30	3	10	0.8	40.734	33.25	0.98	19.4
30	3	20	0.4	72.794	62.32	0.68	14.2
30	3	20	0.8	112.239	66.36	0.36	9.6
100	3	1	0.4	511.75	72.23	0.61	5.6
100	3	1	0.8	514.65	70.12	0.24	13.2
100	3	10	0.4	*n.p.*	208.12	*n.p.*	12.4
100	3	10	0.8	*n.p.*	210.64	*n.p.*	16.7

As the reader can notice, even if the gaps are really small, the PH does not manage to converge to the optimal solution. It is important to take in mind the entity of these gaps when we consider the performance of the customized PH, in Subsection 6.3.

As the reader can see the number of iterations until convergence has not a clear trend with respect to the dimension of the instances. The main difference between the optimal solution and the solution found by the PH is due to small increments in the number of second stage variables used by the heuristic solution. This is due to the values of the w_s that increase too much the values of the first stage variables. We tried to decrease the value of ρ in order to reach better results, nevertheless the main outcome was a delay in the reaching of the consensus. For this reason, in future work we are considering to use a ρ scenario dependent.

6.2.1 The single scenario heuristic

Most of the computational time used by the PH is spent in order to solve single scenario problems. For this reason, we develop an ad-hoc heuristic to solve those problems. Actually, since we use exact solutions of linear programs we have developed a math-heuristic. For the sake of simplicity and without loss of generality we present the heuristic without the indexes s (because we are solving a fixed scenario problem) and t (see Remark 5).

The heuristic is based on the following remark.

Remark 32. *If it is known the amount of customers of type m to use for satisfying the demand of sink j, then we have to solve m problems with one type of customer (they can be solved fast thanks to Theorem 24).*

The heuristic is composed by three steps: in the first one we guess, by using the available information the best possible values for the number of customers of type m to use for satisfying the demand of sink j, in the second step we solve m continuous linear programs, finally, in the third step we update the available information by using the solutions to the previous step.

Remark 33. *It is important to note that if we know how many customers of type m must be used to satisfy the demand of sink j we still have to decide which sources to use for each sink.*

In the following three subsection we describe the three phases of the proposed math-heuristic.

Phase I

We define variables w_j^m to be the number of customers of type m to satisfy the request of sink j. We can then write the following problem

$$\text{minimize} \sum_{j=1}^{J} \sum_{m=1}^{M} \hat{q}_j^m w_j^m \tag{6.9}$$

subject to:

$$\sum_{m=1}^{M} n_m w_j^m = N_j \quad j \in \mathcal{J} \tag{6.10}$$

$$\sum_{j=1}^{J} w_j^m \leq \sum_{i=1}^{I} \theta_i^m \quad m \in \mathcal{M} \tag{6.11}$$

$$w_j^m \in \mathbb{N} \quad \forall \; j \in \mathcal{J} \quad m \in \mathcal{M}$$

The costs $\hat{q}_j^m$ are randomly assigned in the first iteration of the algorithm, while in the following iterations are updated by using the steps described in Phase III.

As it can be notice this problem is non totally unimodular, nevertheless it has lower dimension than the original one and in the generated instances it can be solve exactly very fast.

Phase II

In the second step of the proposed math-heuristic, by using $w_j^m \; \forall \; j \in \mathcal{J}$ as the vector $N_j \; \forall \; j \in \mathcal{J}$ we can split problem (6.7) in M problems (one for each type of customer). We solve the continuous version of these problems since, due to Theorem 24 they have integer optimal solution. Since the objective function has a quadratic term, we do not consider that term in our optimization, but we substitute it with

$$\frac{\rho}{2} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{m=1}^{M} |x_{ij}^m - \bar{x}_{ij}^m|.$$

Phase III

In the third step of the proposed math-heuristic, we update the matrix of costs $\hat{q}_j^m$ used in problem (6.10) - (6.11) by using the solution found in Phase II.

The adopted rule for updating the parameter is the following

$$\hat{q}_j^m \leftarrow \left(\xi \hat{q}_j^m + \frac{\sum_{i=1}^{I} c_{ij}^m x_{ij}^m + \sum_{i=1}^{I} q_{ij}^m y_{ij}^m}{\sum_{i=1}^{I} x_{ij}^m + \sum_{i=1}^{I} y_{ij}^m}\right)(1+\zeta) \tag{6.12}$$

where ξ is set to 0.01, c_{ij}^m and q_{ij}^m are the costs in problem (6.7), x_{ij}^m and y_{ij}^m are the elements of the solution found by the algorithm in the previous phase and ζ is a random variable uniformly distributed in $[-0.1, 0.1]$. The rationale behind this formula is the following: the first term weights the knowledge acquired so far, since ρ is small the previous value is only slightly considered. The second term is the unitary resource cost of the solution that uses resources of type m in sink j.
In order to test the performance of the proposed math-heuristic, we compare it against the exact solver Gurobi. The measures of the performance that we consider are the value of the solution found and the computational time.

In particular, we compare these results by using 50 runs of a PH (as described in 4). Since in the iterations of the PH the objective function of the problem changes, we consider in each run the solution to the problem during the second iteration of the PH. The rationale behind this choice is that, if we consider the first iteration of the PH, then we do not have $\bar{x}_{ij}^m$. The results are shown in Table 6.2.

The table does not report values greater than $I = 500$, $T = 1$ because for $I = 500$ and $T = 3$ the exact algorithm runs out of memory. As the reader can see, the differences between the optimal solutions and the solutions found by the proposed heuristic are very small. By calling w_j^m the number of customers of type m that satisfies the request of sink j in the proposed heuristic and w_j^{*m} the same quantity in the optimal solutions we have that

$$\sum_{j=1}^{J} \sum_{m=1}^{M} |w_j^m - w_j^{*m}| \leq 4 \tag{6.13}$$

in all the instances. This result is acceptable since if we represent the variables w_j^m in a matrix satisfying constraints (6.10) and (6.11), then we can found another matrix satisfying constraints (6.10) and (6.11) by adding or subtracting matrices of the following form

Table 6.2 The table shows the average time of the exact method (Time Ex. [s]), the average time of the Heuristic (Time Heu. [s]), the gap between the optimal solution and the one found by the proposed heuristic (Gap[%]), for different values of number of nodes (I), number of customer types (M), time period (T) and ratios between sources and nodes (ρ). All the averages are computed on 50 randomly generated instances.

I	M	T	ρ	Time Ex. [s]	Time Heu. [s]	Gap[%]
30	3	1	0.4	0.04	0.21	0.13
30	3	1	0.8	0.02	0.19	0.26
30	3	10	0.4	0.05	0.41	0.14
30	3	10	0.8	0.07	0.34	0.42
30	3	20	0.4	0.12	0.65	0.55
30	3	20	0.8	0.13	0.64	0.25
100	3	1	0.4	1.14	1.18	0.76
100	3	1	0.8	1.14	1.12	0.18
100	3	10	0.4	2.41	1.85	0.72
100	3	10	0.8	2.42	1.86	0.27
100	3	20	0.4	9.41	3.05	0.86
100	3	20	0.8	10.02	3.16	0.36
300	3	1	0.4	69.42	7.82	0.56
300	3	1	0.8	72.35	8.10	0.27
500	3	1	0.4	129.12	9.23	0.79
500	3	1	0.8	126.23	15.34	0.86

$$\left[\begin{array}{c|c|c|c|c} 0 & -1 & \dots & 1 & \dots \\ \hline \dots & \dots & \dots & \dots & \dots \\ \hline 0 & 1 & \dots & -1 & \dots \end{array}\right]. \tag{6.14}$$

Hence, result (6.13) proves that the solutions provided by the proposed heuristic are very close to the ones found by exact methods.

These errors in the choice of the w_j^m produce errors in the number of customers of a given type in the final solution obtained by the heuristic. As the reader can notice, for small instances (until 100 nodes and 1 time period) the times used by the proposed heuristic are greater than the times used by the exact method (on average, the heuristic uses 4.5 times more than the time used by the exact method) while in the big instances the proposed approach performs better than the exact method: in the instances with $I = 500$ the

heuristic uses 10% of the time of the exact method, while for the instances with $I = 300$ this percentage is equal to 12%.

6.3 Approximated Progressive Hedging Results

Since the PH as described in Algorithm 4 does not manage to solve big instances, we customize the PH by the following changes:

- instead of solving the single scenario problem in an exact way (point 10 in Algorithm 4), we apply the heuristic described in 6.2.1;
- in each iteration we update the parameter ρ of the PH by using the following rule $\rho \leftarrow k\rho$, where k is the iteration number. In this way we increase the importance of reaching consensus between the variables.

The final algorithm that we use is then shown in Algorithm 5.

If the procedure shown in 5 does not converge, we use the same strategy of the standard PH i.e. after a fixed number of iteration we stop Algorithm 5 and we solve the mathematical model by fixing the values of all the variables that have reached consensus in the procedure.

In the following we refer to Algorithm 4 with the name exact PH and we refer to Algorithm 5 with the name approximate PH.

Since the number of scenarios that guarantees stability cannot be tested by using the solutions to the exact problem, we perform this tuning using the exact PH (Algorithm 4). Furthermore, in the instances that cannot be solved by the exact PH, we use the approximate PH (Algorithm 5) in order to determine the suitable number of scenarios. We show the number of scenarios for reaching out of sample stability in Table 6.3, with the corresponding algorithm used for the stability analysis.

For instances with more than 500 nodes, 3 customer types and 1 time period, the only algorithm that is able to solve the problem is the approximate PH.

Algorithm 5 Approximate Progressive Hedging

k:=0
for each scenario $s \in \mathcal{S}$ **do**
    Solve Heuristically:

$$x_s^{(k)} := \text{argmin}_{x,y_s}\, c^t x + f_s^T y_s \;:\; (x, y_s) \in \mathcal{Q}_s$$

end for
$\bar{x}^{(k)} := \sum_{s\in\mathcal{S}} p_s x_s^{(k)}$
$w_s^{(k)} := \rho(x_s^{(k)} - \bar{x}^{(k)})\ \forall\ s\ \in\ \mathcal{S}$
$k = 1$
while $g^{(k)} < \epsilon$ **do**
    k:=k+1
    ρ:= $k\rho$
    for each scenario $s \in \mathcal{S}$ **do**
        Solve Heuristically:

$$x_s^{(k)} := \text{argmin}_{x,y_s}\, c^t x + w_s^{(k-1)} x + \frac{\rho}{2}\left\| x - \bar{x}^{(k-1)} \right\|_1 + f_s^T y_s \;:\; (x, y_s) \in \mathcal{Q}_s$$

    end for
    $\bar{x}^{(k)} := \sum_{s\in\mathcal{S}} p_s x_s^{(k)}$
    $w_s^{(k)} := w_s^{(k-1)} + \rho(x_s^{(k)} - \bar{x}^{(k)})\ \forall\ s\ \in\ \mathcal{S}$
    $g^{(k)} = \sum_{s\in\mathcal{S}} p_s \| x_s^{(k)} - \bar{x}_s^{(k)} \|$
end while

Table 6.3 The table shows the number of scenario (n Scenarios) that produces out of sample stability and the algorithm that we use determining this number (Algo), for different values of number of nodes (I), number of customer types (M), time period (T) and ratios between sources and nodes (ρ). All the averages are computed on 50 randomly generated instances.

I	M	T	ρ	n Scenarios	Algorithm
30	3	1	0.4	22	Exact
30	3	1	0.8	20	Exact
30	3	10	0.4	21	Exact
30	3	10	0.8	20	Exact
30	3	20	0.4	25	Exact
30	3	20	0.8	24	Exact
100	3	1	0.4	28	Exact
100	3	1	0.8	27	Exact
100	3	10	0.4	21	PH_Exact
100	3	10	0.8	23	PH_Exact
100	3	20	0.4	23	PH_Exact
100	3	20	0.8	22	PH_Exact
300	3	1	0.4	28	PH_Exact
300	3	1	0.8	24	PH_Exact
500	3	1	0.4	28	PH_Exact
500	3	1	0.8	24	PH_Exact

It is worth noting that Table 6.3 highlights the biggest instances that can be solved with each one of the different algorithms.

In order to test the performance of the proposed approximate PH, we run several experiments by solving instances generated as described in Section 5.5. In Table 6.4, we report the results of these tests by showing the gap between the optimal value and the one found by the exact PH, the gap between the solution to the exact PH and the one found by the approximate PH. Furthermore, we show the time used by the approximate PH and the time used by the exact PH for several combinations of number of nodes (I), number of customer types (M), number of time periods (T) and ratio between source nodes and sink nodes(ρ). During all the experiments, we set the maximum CPU time to be one hour and the maximum number of iterations of the PH to be 200.

In Table 6.4 we do not report the number of iterations that the two PH need for convergence since both converge in less than the 200 iterations.

Table 6.4 The table shows the average time of the PH algorithm that solves each sub-problem with exact method (Time PH Ex [s]), the gap with respect to the optimal solution (Gap PH Ex [%]),the average time of the PH that uses the heuristic in 6.2.1 in order to solve every scenario sub-problem (Time PH Heu[s]) and the average gap that it reaches (Gap PH Heu[%]), for different values of number of nodes (I), number of customer types (M), time period (T) and ratios between sources and nodes (ρ). All the averages are computed on 50 randomly generated instances.

I	M	T	ρ	Time PH Ex [s]	Gap PH Ex [%]	Time PH Heu[s]	Gap PH Heu [%]
30	3	1	0.4	18.01	0.77	20.12	1.23
30	3	1	0.8	12.43	0.56	21.23	1.53
30	3	10	0.4	24.12	0.75	37.12	4.25
30	3	10	0.8	33.25	0.98	36.97	2.43
30	3	20	0.4	62.32	0.68	45.27	2.58
30	3	20	0.8	66.36	0.36	47.43	1.25
100	3	1	0.4	72.23	0.61	30.21	3.14
100	3	1	0.8	70.12	0.24	34.42	2.32
100	3	10	0.4	208.12	*n.p.*	62.23	3.21
100	3	10	0.8	210.64	*n.p.*	61.23	3.12
100	3	20	0.4	623.54	*n.p.*	125.24	2.23
100	3	20	0.8	653.53	*n.p.*	153.12	4.23
300	3	1	0.4	154.12	*n.p.*	54.64	2.76
300	3	1	0.8	124.32	*n.p.*	55.32	2.64
500	3	1	0.4	532.23	*n.p.*	123.21	3.52
500	3	1	0.8	542.12	*n.p.*	124.43	2.12

During the experiments, the approximate PH reaches consensus faster than the exact PH, due to the updated policy of the ρ parameter (without that change, the consensus was never achieved before the iteration limit).

The main difference between the optimal solution and the one found by the approximate PH consists in a greater number of first stage resources used. These errors produce an increase of the 5% of the costs, this increment is acceptable in the stochastic framework.

In conclusion, we can claim that with the proposed version of the PH it is possible to obtain good solutions for instances up to 1500 nodes, 3 customer types and 20 time periods in approximate 24 minutes with a standard deviation of 4.34 minutes (these data are obtained by using 50 runs).

Urban People Flow Model

The mathematical problem (5.6) - (5.9) needs the definition of a proper statistical model describing number of people in each node of the network during each time step considered. Furthermore, a real implementation of crowd-shipping needs that the company implementing the model knows the transportation mode selected by the customers that accepts to perform the deliveries: for example if the customer chooses to travel by bus then, the quantity of goods that he/she can carry is lower than if he/she chooses to travel by car.

For these reasons, it is important to have a model for the urban mobility of people. In particular, we can split the problem in two parts. The first one is to classify the transportation mode used by the customers by using data registered by their mobile phones. The second one is to use these data for building a mobility model (considering the choice of the transportation mode and the choice of the destination). It is important to note that the quantity of data in order to develop a model for describing the destination of each single user is huge.

Building a mobility model is important because it can help companies using social engagement to fit their reward in order to promote one type of transportation mode. Furthermore, mobility models allow the municipalities to improve public transportation and to develop policies to change people's habits in order to adopt more sustainable behaviours.

In this chapter we consider as a case study the data collected during the project Open Agora [1]. Unluckily the quantity of data in not enough for

[1] http://www.openagora.it/index-en.html

developing a model of both the transportation mode selected and the model of the destination of the people. For this reason, we just consider the model of the transportation choice. Nevertheless, this is a proof of the feasibility of the construction of model from the data coming from the mobile phone. This is another important evidence that the consumer electronics has reached very good level of accuracies.

The chapter is organized as follows in Section 7.1 we describe the approach to these problems in the literature. In Section 7.2, we describe the main data used for the development of the mobility model. In Section 7.3 we describe the classifiers used, furthermore, in this section we analyse how the performance of the method get worse as the number of available data decreases. This analysis usually is not performed in the literature nevertheless it is of central importance because it informs about the minimum requested sampling time in order to infer information with a given confidence. Thus,in Section 7.4 we describe the mobility model and we present the results obtained in the real setting.

7.1 Literature Review

The problem to classify the transportation mode use by the people with the usage of mobile phone data is known in the literature as Transportation Mode Detection (TMD). In this work, we focus on a subproblem of the general TMD: discriminate between car and bus.

The techniques used for solving this problem are two: machine learning techniques used by considering data from general sensors, without any external information and the map matching techniques that rely only on the GPS positions of the user and on the information about the position of the bus stops. The first techniques use classification algorithms (such as support vector machines, clusterization algorithms, neural networks, etc.) that use data gather by the devices. Instead, the second techniques compare the GPS positions of the device during the travel with the positions of the bus stops, train stops and so on.

The TMD problem is considered in the literature in several papers. In Table 7.1 we present the most recent papers (at the time of the writing) with the corresponding algorithm used, the reached accuracy, the data used and the transportation modes considered.

It is worth noting that the best results are achieved by the random forest, using both accelerometer data and gyroscope, followed by the same classifier using accelerations and GPS positions. These results are very good, nevertheless they are not suitable in our setting because the GPS sensor requires a lot of energy and the gyroscope is not available in a large fraction of mobile phones.

Since the map matching algorithm is not central in our dissertation we refer the interested readers to [79] and to the references therein.

The second topic of this chapter is mobility model i.e. model aiming to describe people preferences about transportation modes.

The theory behind these models started during 1931 in the paper [93] by psychologist Thurstone. Then, similar principles were presented by the Nobel price McFadden in [62] in the econometrics field, and in paper [24] in mathematics. The basic assumption of the choice models is that the benefit that a decider receives from a decision is function of the frequency that the decider makes the choice over repeated choices.

The main tools of this approach are the multinomial logit model and the multinomial probit model. In particular, we focus on the multinomial logit model that has already been used in other research fields, such as transportation [90]-[73] and has proven to be a model robust to changes in the input data. This model was introduced in the econometric literature by Mc Fadden in the 1972 and it has been applied in several settings such as the choice of a new car [94] the choice of the college [27] and the choice of the transportation mode to go to work [95] (see [64] for a comprehensive history of the topic).

The problem of transportation mode selection has a long history, the first study, to our knowledge is [63]. Nevertheless, the data needed to calibrate these models are not easy to gather and, to our knowledge, there are no extensive studies in the field. Luckily, in the present days, it is possible to use mobile phones to gather these data. In the section 7.4 we adopt the approach of the

Table 7.1 The table presents the articles that achieves the best accuracy results

Article	considered classes	Sensors considered	Accuracy	Algorithm
[86]	walk, bicycle, car, bus, train, subway	accelerometer and gyroscope	99,96%	Random Forest
[84]	walk, bicycle, car, bus,train, subway	accelerometer	94,44%	Binomial logistic regression
[31]	walk, bicycle, car, bus, tram, subway, train, unknown	accelerometer, Wi-Fi, GPS, Cell-ID	70-80%	Bayesian probability
[105]	walk, bicycle, car, bus	GPS	95%	Neural Networks and Particle Swarm Optimization
[85]	walk, bicycle, car, tram	accelerometer and GPS	99,5%	Random Forest
[87]	walking, car, train, bus, tram	accelerometer and GPS	82%	Random Forest, dynamic Bayesian Network, Hidden Markov Model
[37]	walk, static, moving slowly, train, bus, run, car	accelerometer, GPS, GIS	97%	Random Forest with GIS and GPS positions
[40]	stationary, walk, bus, train, metro, tram, car	accelerometer	85%	Hidden Markov Model and Adaptive Boost
[58]	stationary, walk, bus, train, metro, tram, car, motorcycle	accelerometer	82%	Decision Tree and FFT

article [28] by defining a multinomial logit model describing the choice of the transportation mode by the users. In particular, we consider the choice between travelling by foot, by bike, by public vehicle and by private vehicle. In the following we identify public vehicle by bus and we identify private vehicle by car.

7.2 Data

The data that can be registered by the mobile phone varies with respect to the device. Furthermore, the process of getting data is the most critical because it consumes energy. In order to better describe the problem of energy consumption we report, in Table 7.2, the energy consumptions of several actions that mobile phone can perform.

As the reader can notice, the less expensive action is to sampling the accelerations. Nevertheless, in our study we consider both data from the accelerometer and the GPS. There are several reasons for this choice. First we consider the GPS in order to quantify the efficiency of the map matching techniques, second the accelerometer and the GPS are installed almost in every smart-phone. Our goal is to produce reliable classifiers able to discriminate between car and bus without using GPS positions.

In order to collect these data we use the application TraceMe, developed in the framework of the project Open Agora. The application asks to the user the travel method adopted (car, bus, tram, metro, bicycle, ...) and it registers the accelerations on the three axis (x,y,z oriented as shown in Figure 7.1) with a sampling time of 0.05 seconds.

Furthermore, it records the GPS position every 10 seconds and the type of activity registered by the android API every time there is a change. In particular, the main activity registered are the following:

- IN_VEHICLE: the device is on a vehicle, such as a car, bus, tram, metro, etc. No information about the specific vehicle is provided.
- ON_BICYCLE: the device is on a bicycle.

Table 7.2 Consumption in mW of the most used actions.

Application	Consumption
Phone Call	680mW
Playing Music	50mW
Record Video	930mW
Play Video	660mW
Accelerometer	21mW
Magnetometer	48mW
Gyroscope	130mW
Microphone	105mW
GPS Sampling	176mW
Background	140mW
Screen	470mW

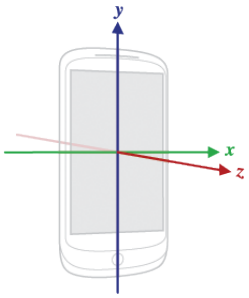

Figure 7.1 The figure shows the direction of the measured acceleration by the android applications

- ON_FOOT: the device is on a user who is walking or running. This state is considered when the device is unable to distinguish between two two sub-states:
  - RUNNING: the device is on a user who is running.
  - WALKING: the device is on a user who is walking.
- STILL: the device is still.
- TILTING: the device angle relative to gravity changed significantly. An example of activity leading to this state is when the mobile phone is took from the pocket.
- UNKNOWN: the device is unable to detect the current activity.

For more information about the Android API, the reader is referred to https://developer.android.com/reference/android/hardware/SensorEvent.html or to https://developers.google.com/android/reference/com/google/android/gms/location/DetectedActivity.

The application runs in mobile phone background and it stores the data in local folders. Then, as soon as a connection is reached it sends all the data to a server that integrates the new data in a non relational DB. By accessing the server, is then possible to download all the data. Since a more in depth description of the application is out of the scope of this thesis we refer the interested reader to [79].

Before to start the analysis of the data is it worth noting that the characteristics of the time series of the accelerations are strongly dependent by the time and by the traffic conditions. We have collected several data from the field, nevertheless the amount of data needed in order to take into account time variations is much greater than the actual one. For this reason, we postpone this analysis to future works.

7.3 Classification Methods

In order to classify if the user has used a car or a bus for moving we divide all the data that we have in travels i.e. a sequence of information such that the label given from the google API is IN_VEHICLE. Then, for each of them we compute a set of features and we classify them. The strategy is shown in Figure 7.2.

All the features that we consider are computed from the time series of the l_2 norm of the accelerations computed by means of its components along the three axis.

We then consider two types of series, the differentiated series i.e. $\breve{a}_t = a_t - a_{t-1}$ for $t = 1, ..., N$ and the l_2 norm of the vector of the acceleration minus the gravitational acceleration g (we call this time series $\tilde{a}_t$). The choice of maintaining both the time series is due to the fact that the features in the two cases contain different information. In particular, we denote with a subscript 1

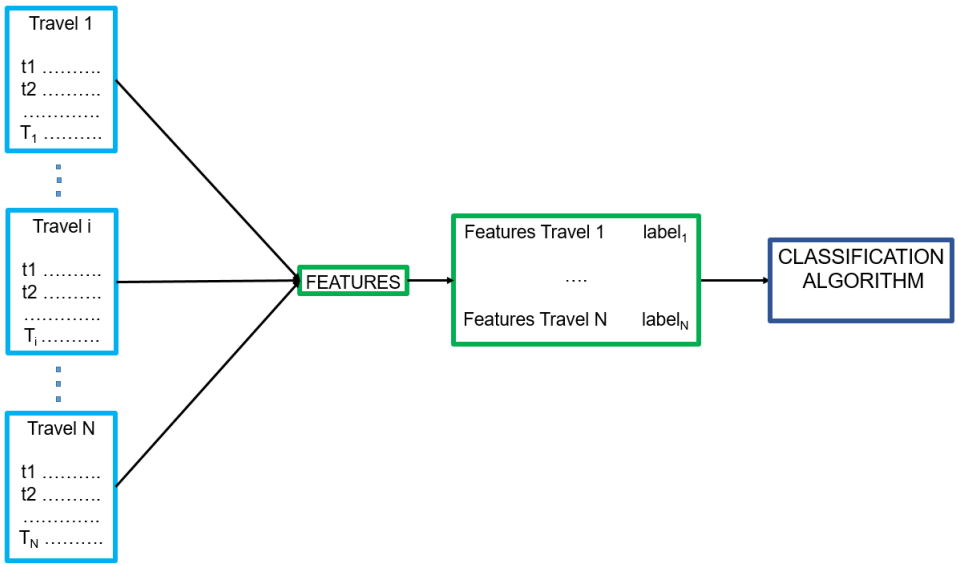

Figure 7.2 The figure shows the logical components of the classifier that we use

all the features that are computed with this second definition. The features that we consider are the following:

- Duration of the travel: we expect that long travels are more probable done by using car than by using buses.
- l_2, l_∞ and l_1 norms of the ACF: since the buses have preferential lane, then they are not subjected by the traffic flow uncertainties and, for that reason, the values of the accelerations are more correlated.
- The minimum, maximum and the average acceleration of the vehicle: we aspect that the buses and the cars have, in general, different average acceleration.
- Number of tilting: in the bus, since the user is not driving, it can use more the mobile phone, generating tilting states.
- Average time of the tilting activity: since the bus need less attention, we expect that the tilting activity lasts more on a bus than on a car.
- l_2, l_∞ and l_1 norms of the spectrum: it measures the energy in the data.

- The argmax of the spectrum: it represents the frequency that is more present in the data.
- The integral mean of the spectrum: it is another measure of the energy of the data.
- By using a sliding window we compute the time series of the variances of the accelerations. From this time series we compute its l_2, l_∞ and l_1 norms and the same norms for the ACF: we consider these features in order to investigate some ARCH structure in the series of accelerations.
- Output of the map matching algorithm: it is a number between 0 and 1 and represents how much the path followed in the travel is similar to a path of a BUS. We describe the algorithm that compute this value in Subsection 7.3.1.

It is important to notice that even if some of the aforementioned features have good discrimination power, no one of these features is enough to discriminate between car and buses.

Remark 34. *Since the values of the features can be of different magnitude, we normalize all of them between 0 and 1 before to run the classifiers.*

In order to compare the classifiers, we evaluate them by considering the following measures of performance.

Definition 15. *We define the recall r to be the ratio*

$$r_x = \frac{\#travel\ correctly\ labelled}{\#\ travel\ in\ x}, \tag{7.1}$$

where $x \in \{CAR, BUS\}$

Recall tells us the probability to retrieve all the travels in x in a sample.

Remark 35. *It is worth noting that a fake classifier that classifies all the travels to be in x has recall equal* 1.

Definition 16. *We define the precision p_x to be the ratio*

$$p = \frac{\#travel\ in\ X\ correctly\ labelled}{\#\ travel\ labeled\ as\ X} \tag{7.2}$$

where $x \in \{CAR, BUS\}$

Precision tells us the probability that a travel labelled as x is labelled correctly.

Remark 36. *It is worth noting that a fake classifier that classifies all the travels to be in X has precision equal* 1.

As KPIs of the method we then consider precisions and recalls for both car and bus (p_{CAR}, p_{BUS}, r_{CAR}, r_{BUS}).

Together, these indicators give important information about the performance of the algorithm. As the reader can notice, the best value for each one of these parameters is 1.

Remark 37. *We consider all the four indicators because the aforementioned fake classifiers can be detected only by using all of them.*

For our analysis we consider four classifiers: Decision Tree, Support Vector Machine, Random Forest and Naive Bayes. We do not consider classifiers such as neural network, nonlinear support vector machine, and other classifiers using nonlinear transformations because we already apply nonlinear transformations to the data by computing the features and we do not want to add more complexity in the classifiers.

Since collecting data is expensive in term of energy consumption, we consider what happens to p_{CAR}, p_{BUS}, r_{CAR} and r_{BUS} when the sampling frequency decreases. We develop this analysis in three axis. The first one is a reduction in the sampling frequency of the accelerometer data, the second one is a reduction of the sampling frequecy of the GPS positions while the third one is a reduction of both sampling frequencies of the GPS and in the accelerometer data.

In the following subsections, we describe the map matching algorithm (in subsection 7.3.1) and we analyse the numerical performance of several classifiers in Subsection 7.3.2.

7.3.1 The Map Matching Algorithm

The Map Matching Algorithm considers as input the time series of the GPS positions recorded during the travel and it measures the degree of similarity between these positions and the paths of the public transportation vehicles. The procedure is presented in Algorithm 6.

Algorithm 6 Map Matching Algorithm
Input: Time series of GPS records ($p.GPS\_records$)
Output: $\{BUS, CAR, METRO, TRAIN, UNKNOWN\}$

```
if p.GPS_records has not record in the central part then
    if near_METROstop(p.GPS_records(begin)) then
        return METRO
    else
        return UNKNOWN
    end if
else
    count = 0
    for GPS in p.GPS_records do
        count = count + near(GPS, Train)
    end for
    if count > 0.75length(p.GPS_records) then
        return TRAIN
    else
        count = 0
        for GPS in p.GPS_records do
            count GPS == GTT path
        end for
        if count > 0.75length(p.GPS_records) then
            if near_BUSstop(p.GPS_records) then
                return BUS
            else
                return CAR
            end if
        else
            return CAR
        end if
    end if
end if
```

Given a path, the algorithm first checks if the path was done by metro. This travel mode is the easiest to recognise because it is characterized by the lack of GPS positions in the central part of the time series. Hence, it is enough to compare the first and last part of the time series with the positions of the metro stops (with the function $near\_METROstop$) to recognise this transportation mode.

The second comparison is between the path and the coordinates of the train rail-road. If the number of GPS positions near the rail-road is grater than the 75% of the total number of observations then the algorithm returns the label TRAIN.

Finally, the algorithm compares the path followed by the user and the positions of the bus lines. If the number of positions close to the bus line is greater than the 75% of the total number of observations and the starting and ending points of the path are close to bus stations, then the algorithm returns the label BUS, otherwise it returns the label CAR.

Remark 38. *It is worth noting that the algorithm returns CAR if it does not find a match with the public lines. Hence, if the quality of data degraded the most likely output is the label CAR.*

Since this algorithm uses the GPS positions, it consumes a lot of energy from the phone. For this reason, it is interesting to see how the performance of the algorithm changes as the sampling time increase. The results of this experiment are presented in Figure 7.3.

Remark 39. *It is important to notice that we only consider the graph related to the recall of the bus because if the algorithm does not find the match, then it returns the label CAR. Hence, both precision and recall of the CAR increase as the data quality degraded.*

As the reader can notice, the performance degrade really fast.

Remark 40. *It is worth noting that in Figure 7.3 the lowest result is a recall of* 45%. *This means that the algorithm is still able to recognise the* 45% *of the total number of buses.*

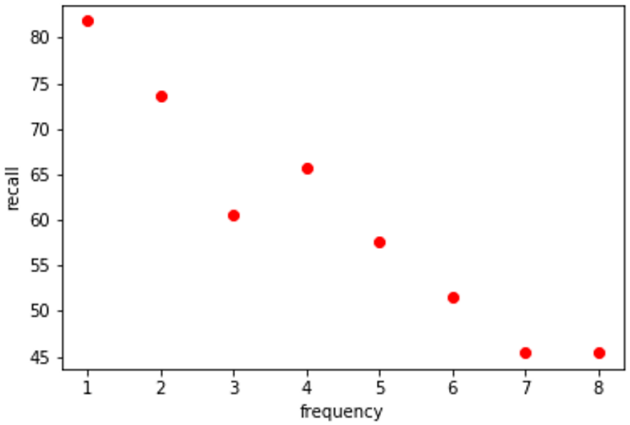

Figure 7.3 The figure shows the recall of the BUS of the algorithm 6.

7.3.2 Numerical Results

In this section we present the main results of the classification methods. We consider four classifiers: Random Forest, Decision Tree, SVM and Naive Bayes. In all the experiments we randomly divide the observations in training (70% of the data) and test (30% of the data) and we measure the precision and the recall of the labels provided by the classifiers defined as in Equations 7.2 and 7.1. We repeat these operations 100 times in order to have robust results. The classifiers are implemented in the software R a statistical software[2].

We perform two experiments. First, we test the classifiers by considering a subset of features. In particular, from the aforementioned 34 features, by using the principal component analysis (PCA), we select the 2 (the l_2 norm of the ACF of the mobile variances obtained by using the time series $\breve{a}_t$ and $\tilde{a}_t$) and the 20 most significant features (the l_2, l_1 and l_∞ norm of the spectrum, the maximum, the minimum, the average, the l_1, l_2 and l_∞ norms of the ACF of both time series, the number of tilting and the average duration of the tilting operation) and we use them in the experiments. We do not use all the features because it can result in over fitting.

Second, we study the degradation of the results when greater sampling times are considered, in this way it is possible to asses the less consuming energy setting of the sensors that permits to have a certain performance.

[2]see https://www.r-project.org

Table 7.3 Comparison between classifiers by using 2 features

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
Random forest	0.802919 (0.082677)	0.817923 (0.099179)	0.821632 (0.093882)	0.800312 (0.089966)
Decision tree	0.732830 (0.137697)	0.800060 (0.148603)	0.796688 (0.124730)	0.767942 (0.096115)
SVM	n.a.	n.a.	n.a.	n.a.
Naive Bayes	0.727190 (0.115897)	0.673944 (0.165845)	0.690267 (0.143707)	0.723270 (0.105357)

The results of the comparison made by using 2 features are shown in Table 7.3. As the reader can notice, the best performance are achieved by the random forest classifier. This result confirms the results found in the literature which, in several cases, present the random forest as the best classifier for the problem of TMD. It is important to note that also the standard deviation of the performance of the random forest is the smallest.

Table 7.4 Comparison between classifiers by using 20 features

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
Random forest	0.841150 (0.081431)	0.927028 (0.071077)	0.918942 (0.086221)	0.850839 (0.075155)
Decision tree	0.827753 (0.127197)	0.803166 (0.151295)	0.815903 (0.124632)	0.830310 (0.118881)
SVM	0.740289 (0.103344)	0.725266 (0.130770)	0.725775 (0.121683)	0.744139 (0.091791)
Naive Bayes	0.803923 (0.099138)	0.829820 (0.090858)	0.827640 (0.088811)	0.812936 (0.088797)

Table 7.4 shows the performance of the classifiers when 20 features are considered. As the reader can notice the random forest achieves results in both the precision and the recall of the 90% and it is the best classifier. Since these results are achieved by excluding the map matching those solutions have also good performance concerning energy consumption.

Table 7.5 shows the performance of the classifiers with the additional feature of the map matching. As the reader can see the performance improve as it is reasonable to expect. The most surprising fact is that while the random forest improves only slightly, other classifiers improve their performance in a sensible

Table 7.5 Comparison between classifiers by using 20 features and the map matching feature

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
Random forest	0.878969 (0.078827)	0.933454 (0.061541)	0.929486 (0.062562)	0.890928 (0.063132)
Decision tree	0.905916 (0.099972)	0.875343 (0.086921)	0.874741 (0.089588)	0.911933 (0.091483)
SVM	0.891626 (0.087533)	0.731230 (0.134008)	0.767010 (0.116983)	0.879217 (0.085336)
Naive Bayes	0.886668 (0.064695)	0.898592 (0.071131)	0.894433 (0.073784)	0.890953 (0.063997)

way. By considering more features the results improve, nevertheless we do not perform this comparison because it results in over-fitting.

In the following we analyse the performance of the classifier when the sampling time is increased. In particular, in Subsection 7.3.2 we consider the results of the decision tree, in Subsection 7.3.2 we consider the support vector machine, in Subsection 7.3.2 we consider the random forest and finally in Subsection 7.3.2 we consider the Naive Bayes classifier. In all the following tables we call ν the sampling time and in particular we call ν_a the sampling time of the accelerations and ν_g the sampling time of the GPS positions.

Decision Tree

A decision tree is a classifier that uses a tree-like graph in order to discriminate between two classes where the leaves of the tree contain the selected class. In the experiments, we use the function rpart available in the package rpart of the software R. The interested reader is referred to [89] for further information about the method.

As it is possible to see form Table 7.6, the classifier is robust with respect to the number of accelerations samples considered. In fact, the reported values are not statistically different. Furthermore, also the standard deviations of the results are similar. This is a very interesting fact. We do not continue by reducing the sampling frequency because the energy consumption of the accelerometer with sampling frequency of 2 seconds is already negligible.

Table 7.6 Performance of the decision tree for different accelerations sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_a = 100$ ms	0.918503	0.867361	0.872126	0.926246
	(0.111158)	(0.091829)	(0.085936)	(0.090842)
$\nu_a = 200$ ms	0.876750	0.867451	0.879124	0.889467
	(0.119500)	(0.107570)	(0.088882)	(0.094670)
$\nu_a = 1$ s	0.909557	0.856947	0.867557	0.917613
	(0.106081)	(0.091380)	(0.080528)	(0.087995)
$\nu_a = 2$ s	0.912739	0.874038	0.878374	0.918593
	(0.099966)	(0.075983)	(0.070908)	(0.087696)

Table 7.7 Performance of the decision tree for different GPS sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_g = 20$ s	0.912349	0.850778	0.856516	0.918002
	(0.110423)	(0.073263)	(0.070924)	(0.094680)
$\nu_g = 40$ s	0.818806	0.873158	0.873010	0.839427
	(0.104251)	(0.136120)	(0.118663)	(0.089982)

Instead, the classifier is very sensitive with respect to the sampling times of the GPS position. This is an effect of the degradation of the performance of the map matching algorithm: since the performance of the map matching are good, the most important rule of the decision tree considers the map matching feature. Once that this feature is no more reliable, the decision tree loses an important part of its discrimination power.

Table 7.8 Performance of the decision tree for different GPS and accelerations sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_a = 100$ ms	0.918315	0.854485	0.866002	0.917419
$\nu_g = 20$ s	(0.099187)	(0.081299)	(0.076350)	(0.090403)
$\nu_a = 200$ ms	0.795123	0.843348	0.845045	0.808501
$\nu_g = 40$ s	(0.125741)	(0.105955)	(0.093710)	(0.113954)
$\nu_a = 1$ s no GPS	0.710991	0.773755	0.763688	0.737258
	(0.119922)	(0.153572)	(0.137090)	(0.103011)
$\nu_a = 2$ s no GPS	0.656163	0.713462	0.706486	0.688866
	(0.150580)	(0.147354)	(0.112148)	(0.113173)

Finally, by increasing both the sampling times of the accelerations and of the GPS positions, we obtain a remarkable degradation of the performance.

Support Vector Machine

Support vector machine (SVM) is a classification method that tries to divide the two class of observations by means of a hyperplane. Since it is not central for the topic of this thesis, we refer to [89] for an introduction to the topic.

If the two classes cannot be divided, then the svm finds the hyperplane that minimizes the number of errors. A special case of support vector machine are the kernel based support vector machines. These classifiers apply a nonlinear transformation to the data before to classify them with a hyperplane. We do not consider these classifiers since we already apply nonlinear transformation when we define the features.

The implementation of the svm used in the experiment is the function svm of the package e1071 of the software R.

Table 7.9 Performance of the SVM for different accelerations sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_a = 100$ ms	0.894563 (0.077773)	0.779040 (0.098319)	0.798019 (0.091642)	0.882365 (0.085788)
$\nu_a = 200$ ms	0.894571 (0.077678)	0.742739 (0.141016)	0.770493 (0.118011)	0.883752 (0.082874)
$\nu_a = 1$ s	0.910129 (0.073855)	0.759759 (0.116162)	0.788771 (0.100550)	0.897715 (0.079853)
$\nu_a = 2$ s	0.925934 (0.068584)	0.791372 (0.103053)	0.807117 (0.100950)	0.918368 (0.072596)

The performance of the SVM for different values of accelerations sampling time are shown in Table 7.9. As the reader can notice it is possible to see that the results of precision and recall are not statistically different from a setting to the others. If we reduce the sampling time to be less than 2 seconds, the performance degraded fast. As above, we do not report further results because the use of the accelerometer with a sampling time of 2 seconds produces a negligible use of the battery.

It is worth noting that the SVM is not influenced by the GPS sampling time. This is a very important result because by using this classifier it is possible to minimize the use of the GPS, hence to consume less energy.

Table 7.10 Performance of the SVM for different GPS sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_g = 20$ s	0.907381	0.795688	0.810451	0.898031
	(0.082637)	(0.085568)	(0.080678)	(0.087732)
$\nu_g = 40$ s	0.893261	0.777434	0.795540	0.883465
	(0.078577)	(0.098020)	(0.088358)	(0.083807)

Table 7.11 Performance of the SVM for different GPS and accelerations sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_a = 100$ ms	0.918802	0.758848	0.783631	0.906523
$\nu_g = 20$ s	(0.068281)	(0.110411)	(0.104907)	(0.077221)
$\nu_a = 200$ ms	0.885911	0.735834	0.766990	0.871895
$\nu_g = 40$ s	(0.083687)	(0.111820)	(0.094961)	(0.084917)
$\nu_a = 1$ s no GPS	0.815284	0.688425	0.719517	0.790718
	(0.086382)	(0.132454)	(0.115779)	(0.099618)
$\nu_a = 2$ s no GPS	0.728778	0.649182	0.667937	0.713917
	(0.100408)	(0.114754)	(0.110435)	(0.099726)

Finally, Table 7.11 shows how the performance degraded by increasing the sampling time of both the GPS and the accelerometer. As the reader can notice, the performance strongly deteriorate when the GPS information is removed.

Random Forest

Random forest is a classifier composed by a set of decision trees. It averages multiple decision trees trained on different parts of the training set. In this way it reduces the over-fitting that a single tree can produce. In the experiments we use the function randomForest in the package randomForest of the software R. For more information about the method and its implementation see [89].

Table 7.12 shows the performance of the random forest for various accelerations sampling times. As the SVM also the random forest classifier is robust with respect to changes in the accelerations sampling times.

Furthermore, the performance of the random forest are also stable with respect to the different sampling time of the GPS positions.

Table 7.12 Performance of the Random Forest for different accelerations sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_a = 100$ ms	0.893741	0.898341	0.902069	0.894701
	(0.085337)	(0.072667)	(0.068523)	(0.082116)
$\nu_a = 200$ ms	0.875641	0.901210	0.898088	0.882643
	(0.093521)	(0.074942)	(0.076014)	(0.081416)
$\nu_a = 1$ s	0.873253	0.880434	0.874287	0.882255
	(0.074455)	(0.074519)	(0.075749)	(0.073559)
$\nu_a = 2$ s	0.870060	0.910752	0.908919	0.874263
	(0.094433)	(0.068352)	(0.066855)	(0.093479)

Table 7.13 Performance of the Random Forest for different GPS sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_g = 20$ s	0.907312	0.903385	0.902747	0.907555
	(0.075140)	(0.068760)	(0.068229)	(0.078110)
$\nu_g = 40$ s	0.873249	0.907044	0.900942	0.878677
	(0.074846)	(0.075876)	(0.079085)	(0.072259)

Table 7.14 Performance of the Random Forest for different GPS and accelerations sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_a = 100$ ms	0.912477	0.898554	0.896846	0.914673
$\nu_g = 20$ s	(0.070013)	(0.076648)	(0.075921)	(0.070514)
$\nu_a = 200$ ms	0.866037	0.874025	0.868323	0.874444
$\nu_g = 40$ s	(0.106258)	(0.075367)	(0.077646)	(0.094947)
$\nu_a = 1$ s no GPS	0.714129	0.730120	0.717892	0.730745
	(0.112619)	(0.101647)	(0.095984)	(0.101172)
$\nu_a = 2$ s no GPS	0.742227	0.743374	0.739425	0.745969
	(0.099250)	(0.101484)	(0.102906)	(0.099598)

Finally, as the above classifiers its performance degraded if no GPS information is given and the sampling frequency of the accelerations is low. Nevertheless, the results of the random forest are better than the results of the other classifiers. This confirm the random forest to be the best classifier for the TDM problem.

Naive Bayes

Naive Bayes is a classifier based on the application of the Bayes' theorem with strong (naive) independence assumptions between the features. Given $f_1, \ldots, f_n$ a set of discrete features, and by calling x the label, the Naive Bayes classifier infers $P(x|f_1)P(x|f_2)\ldots P(x|f_n)$ by fitting these probabilities in the training set. The function used for this classifier is naive_bayes of the package naivebayes of the software R. A detailed description of this method can be found in [89].

In order to use the Naive Bayes classifier, for each feature we split its values in 5 groups by using the quantiles. In this way all the conditional probabilities are discrete and it is possible to compute the conditional probabilities in an effective way.

Table 7.15 Performance of the Naive Bayes for different accelerations sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_a = 100$ ms	0.853521 (0.084850)	0.899056 (0.062433)	0.893220 (0.069047)	0.861164 (0.078875)
$\nu_a = 200$ ms	0.839676 (0.101065)	0.867174 (0.081003)	0.861987 (0.079821)	0.851404 (0.084947)
$\nu_a = 1$ s	0.862476 (0.099071)	0.889062 (0.078264)	0.887126 (0.078243)	0.866643 (0.094098)
$\nu_a = 2$ s	0.866093 (0.102977)	0.881261 (0.070670)	0.879450 (0.068689)	0.877218 (0.083346)

As the other classifiers also the performance of the Naive Bayes are not influenced by the different sampling times of the accelerations neither from the sampling times of the GPS position (see Table 7.15 and Table 7.16).

Table 7.16 Performance of the Naive Bayes for different GPS sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_g = 20$ s	0.898748 (0.071556)	0.904295 (0.063918)	0.899646 (0.063744)	0.899930 (0.074990)
$\nu_g = 40$ s	0.874193 (0.076429)	0.874069 (0.069605)	0.866415 (0.068177)	0.885243 (0.066286)

As the other classifiers, also the Naive Bayes performance degraded if the GPS positions are not considered and the accelerations are sampled with low

frequency. This behaviour is produced by the fact that if some feature degraded the classifiers can still use the others in order to have good performance. Nevertheless, if the quality of all the feature degrade, then also the performance of the algorithm degrade.

Table 7.17 Performance of the Naive Bayes for different GPS and accelerations sampling times

	r_{CAR}	r_{BUS}	p_{CAR}	p_{BUS}
$\nu_a = 100$ ms	0.849555	0.899026	0.891083	0.862926
$\nu_g = 20$ s	(0.077123)	(0.054708)	(0.061941)	(0.068126)
$\nu_a = 200$ ms	0.843246	0.821958	0.824844	0.841971
$\nu_g = 40$ s	(0.095958)	(0.086982)	(0.084238)	(0.092129)
$\nu_a = 1$ s no GPS	0.691619	0.648899	0.664730	0.680513
	(0.125581)	(0.106271)	(0.095019)	(0.119542)
$\nu_a = 2$ s no GPS	0.707868	0.672374	0.685910	0.696724
	(0.109317)	(0.118212)	(0.109138)	(0.111340)

7.4 Mobility Model

In this section we consider mobility models, in particular, we divide this section in two parts. In Subsection 7.4.1 we introduce mobility models while in Subsection 7.4.2 we define the fitting problem and in Subsection 7.4.3 we present the numerical results.

7.4.1 Introduction

Let us consider the choice X to be a random variable, and Y to be the vector of random variables describing the elements influencing the choice, we say that the model is logit if

$$\mathbb{P}[X|Y] = F(\alpha + \beta y), \tag{7.3}$$

with

$$F(z) = \frac{e^z}{1+e^z}.$$

Instead, we say that the model is probit if

$$F(z) = \int_{-\infty}^{x} \Psi(u)du,$$

where $\Psi(u)$ is the standard normal distribution.

Remark 41. *As the reader can notice, both function $F(z) \to 0$ if $z \to -\infty$ and $F(z) \to 1$ if $z \to \infty$. Moreover, for both of them, it holds that $F(0) = 0.5$.*

In the following we consider the problem of fitting a multinomial logit distribution.

7.4.2 Model

Given K transportation modes, we define the probability that a user selects the transportation mode k for path p to be

$$\pi_{kp} = \frac{e^{-c_k^T \beta - c_p^T \gamma}}{\sum_{k,p} e^{-c_k^T \beta - c_p^T \gamma}}, \tag{7.4}$$

where c_k is the vector describing the characteristics of the transportation mode k, c_p is the vector describing the characteristics of the path p, β and γ are the personal costs of the transportation mode and path characteristics. In order to calibrate the model we use the function logit belonging to the python module statsmodels.

7.4.3 Numerical Results

In this section we present the numerical results from the calibration of the logit model by using the data collected by the application TraceMe. In particular, we collect 100 travels (47 by car and 53 by bus). The travels were collected by different people in random movements of the days. In order to calibrate the model we consider two types of costs i.e. distance and economic cost. While the distances covered can be computed by using the GPS positions, computing the costs is more difficult and assumptions are needed. In particular, for the bus we consider the cost of a one way ticket and the travel cost of a person owning a subscription (this costs is obtained as the ratio between the total cost of the yearly subscription and the average number of travels in one year). For the cost of the car, we made more assumptions: we consider the perceived cost

Table 7.18 The table shows the values of the parameters β for different hypothesis with respect to the costs. In the square brackets there are the 95% confidence interval of the parameters.

	β	γ
Car perceived cost Bus Ticket	3.09 [1.87, 4.30]	-3.81 [-5.45, -2.17]
Car perceived cost Subscription	3.74 [2.25, 5.23	-3.76 [-5.30, -2.21]
Car real cost Ticket	3.34 [2.03, 4.64]	-4.47 [-6.29, -2.65]
Car real cost Subscription	2.88 [1.64, 4.11]	-3.24 [-4.65, -1.84]
Parking cost, Car perceived cost Ticket	-0.04[-0.18, 0.10]	-0.13 [-0.33, 0.08]
Parking cost, Car perceived cost Subscription	-0.1953 [0.33, 0.06]	0.0084 [-0.12, 0.13]
Parking cost, Car real cost Ticket	-0.1016 [-0.23, 0.03]	-0.04[-0.22, 0.14]
Parking cost, Car real cost Subscription	-0.2666 [0.41, 0.11]	0.11 [-0.03, 0.25]

of the car (i.e. the cost of the fuel), the real cost of the car (i.e. the cost of the fuel plus the cost of the taxes and its depreciation) and the possible addiction of the parking costs. In Table 7.18 we show for each one of these assumptions the corresponding parameters.

As the reader can notice, some parameters have negative values, this means that the cost related to the negative parameter is not considered in the choice. In the first four rows it is possible to gather that the distance is not influencing the choice of the transportation modes. This enhance the competitiveness of the car, especially because the perceived price is less than the cost of the subscription to the bus service. Instead, if the parking costs are considered there is a balance between the two transportation modes. From this observation it is possible to conclude that, in order to push people to use public transportation means, it is important to adopt wise parking cost policy.

Conclusions

In this thesis we present several case studies of optimization methods that use data coming from a network of sensors. These new opportunities of gaining data greatly improve the power of optimization by creating different new applications.

In particular, we use these data for organizing in an optimal way the waste collection operations. The performance of the developed algorithm are so good that the companies involved in the project decide to develop a commercial solver for the waste collection optimization. Future improvements of the heuristic are now tested but they are under industrial secret.

Furthermore, by using these data we have proposed a new way of conducing the last mile logistics by considering as special case the e-grocery industry. In particular, in the first chapter we have analysed the e-grocery market and we have described its big challenges. These challenges are close to the problem of gathering data from a network of IoT sensors spread all around the city. For solving some of the problems of this business model it is possible to use crowd-shipping. Nevertheless, this problem has never been faced in the optimization literature. For this reason, we define a deterministic and a stochastic model with the aim of minimizing the costs of using social engagement.

The real size instances of the deterministic model cannot be solved in a reasonable amount of time by exact solver. Hence, we propose a heuristic able to find good solutions in all the tested instances in a short amount of time.

Since the deterministic problem does not consider uncertainties, we define a stochastic model. We have proved that the gain produced by considering the

stochastic version of the problem is not negligible, hence ad hoc techniques must be developed. Then, we first develop a heuristic able to deal with the medium size instances of the stochastic problem and then we develop a heuristic, based on the PH, able to deal with the biggest instances of such a problem. Both proposed heuristics have really good performance and it is possible to state that the computational time requested for finding a good solution is compatible with a real implementation of the problem. Future line of research will try to improve even more the decomposition strategy by considering a different problem for each time instant.

Furthermore, due to the problems related to the transportation mode selected by the person involved in crowd-shipping, we develop a classification algorithm able to recognise, with high precision and recall if a user is travelling by bus or by car. The important result of this chapter is that the classification algorithm developed needs only the accelerations data. Hence, this method can be used by almost all mobile phones and it consumes a small amount of energy. By using this classification method it is possible to obtain an estimation of the user preferences and to build a mobility model. Thanks to this model, it is possible to formulate an incentive scheme for manipulating the choice of the transportation mode made by the users. This result is an innovative application of machine learning and utility theory, the main lack in the development is a proper mathematical theory of the uncertainty produced by the classifiers. In future studies, we will develop an optimization framework in order to obtain robust utility models by considering possible errors in the data collected.

In conclusion, it is possible to claim that crowd-shipping represents a new way to perform last mile logistics, furthermore the advantages provided by the sensors data in the optimization framework have huge potential.

Bibliography

[1] N. A. H. Agatz. Demand management in e-fulfillment. *Erasmus Research Institute of Management (ERIM)*, 2009.

[2] European Environment Agency. Waste - municipal solid waste generation and management. http://www.eea.europa.eu/soer-2015/countries-comparison/waste, 2015. Last Access: 27/07/2016.

[3] M. Aickin and H. Gensler. Adjusting for multiple testing when reporting research results: the bonferroni vs holm methods. *American Journal of Public Health*, 86:726—-728, 1996.

[4] M. Aldrich. Online shopping in the 1980s. *Annals of the History of Computing*, 33:57–61, 2011.

[5] M. E. Andersen and S. Wholk. A variable neighborhood search for the multi- period collection of recyclable materials. *European Journal of Operational Research*, 249(2):540–550, 2016.

[6] D. Applegate, W. Cook, and A. Rohe. Chained lin-kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15(1):82 – 92, 2003.

[7] J. E. Aronson. The multiperiod assignment problem: A multicommodity network flow model and specialized branch and bound algorithm. *European Journal of Operational Research*, 23(3):367–381, 1986.

[8] P. Billingsley. *Probability and Measure (Third ed.)*. John Wiley & Sons, 1995.

[9] K. Blumenthal. Generation and treatment of municipal waste. *Eurostat. Statistics in focus*, (31/2011), 2011.

[10] R. Botsman and R. Rogers. *What's Mine Is Yours: The Rise of Collaborative Consumption*. HarperCollins, 2010.

[11] K.K. Boyer, G.T. Hult, and M. Frohlich. An exploratory analysis of extended grocery supply chain operations and home delivery. *In Integrated Manufacturing Systems*, 14:652–663, 2003.

[12] D. Bruno and J. Gonzalez-Feliu. French e-grocery models: a comparison of deliveries performances. *Colloquium on European Retail Research-CERR Book of Abstracts*, pages 230–253, 2012.

[13] A.C. Cagliano, L. Gobbato, R. Tadei, and G. Perboli. Its for e-grocery business: The simulation and optimization of urban logistics project. *Transportation Research Proceedings*, 3:489—-498, 2014.

[14] F. Chu, N. Labadi, and C. Prins. A scatter search for the periodic capacitated arc routing problem. *European Journal of Operational Research*, 169(2):586 – 605, 2006.

[15] G.T. Crainic, F. Maggioni, G. Perboli, and W. Rei. The generalized skeleton solution: a new measure of the quality of the deterministic solution in stochastic programming. *CIRRELT-2015-2*, 2015.

[16] F. Cruijssen, W. Dullaert, and H. Fleuren. Horizontal cooperation in transport and logistics: a litterature review. *Transportation Journal*, pages 22–39, 2007.

[17] N. Dagher, D. Soumitra, and A. De Meyer. *Online Grocery Shopping*. Insead Fontaineblau, 1998.

[18] B. Dai and H.Chen. Mathematical model and solution approach for carriers' collaborative transportation planning in less that truckload transportation. *International Journal of Advanced Operation Management*, 4(1):62–84, 2012.

[19] S. Das and B. K. Bhattacharyya. Optimization of municipal solid waste collection and transportation routes. *Waste Management*, 43:9 – 18, 2015.

[20] I. Dayarian, T.G. Crainic, M. Gendreau, and W. Rei. A column generation approach for a multi-attribute vehicle routing problem. *European Journal of Operational Research*, 241(3):888–906, 2015.

[21] W. Delfmann. Concepts, challenges and market potential for online food retailing in germany. *Economics and Social Sciences*, 2011.

[22] L. Dennis. *Making Decisions(2nd ed.)*. John Wiley & Sons, 1991.

[23] U. Derigs. The shortest augmenting path method for solving assignment problems – motivation and computational experience. *Annals of Operation Research*, 4:57–102, 1985.

[24] L. R. Duncan. *Conditional logit analysis of qualitative choice behavior*. John Wiley & Sons, 1959.

[25] D. Fontana E. Fernández and M.G. Speranza. On the collaboration uncapacitated arc routin problem. *Computers & Operations Research*, 67:120–131, 2016.

[26] B. Fugate, B. Davis-Sramek, and T. Goldsby. Operational collaboration between shippers and crriers in the transportation industry. *The International Journal of Logistics Management*, 20(3):425–447, 2009.

[27] W. C. Fuller, C. Manski, and D. Wise. New evidence on the economic determinants of post-secondary schooling choices. *Journal of Human Resources*, 17:477–498, 1982.

[28] L. Gobbato F. Perfetti G. Perboli, M. Ghirardi. Flights and their economic impact on the airport catchment area: An application to the italian tourist market. *Journal Optimization Theory Application*, 164:1109–1133, 2015.

[29] N. Galante, E. G. Lopez, and S. Monroe. The future of online grocery in europe, perspectives on retail and consumer goods. *mckinsey.com*, pages 22–31, 2013.

[30] B. Gavish and S.C. Graves. The travelling salesman problem and related problems. *Operations Research Centre, Massachusetts Institute of Technology*, 1978.

[31] K. T. Geurs, T. Thomas, M. Bijlsma, and S. Douhou. Automatic trip and mode detection with move smarter: First results from the dutch mobile mobility panel. *Transportation Research Procedia*, 11:247,262, 2015.

[32] G. Ghiani, D. Laganà, E. Manni, R. Musmanno, and D. Vigo. Operations research in solid waste management: A survey of strategic and tactical issues. *Computers and Operations Research*, 44:22 – 32, 2014.

[33] A. Ghouila-Houri. Caracterisation des matrices totalement unimodulaires. *C. R. Acad. Sci. Paris*, 254:1192–1194, 1962.

[34] L. Gobbato. *Stochastic programming for City Logistics: new models and methods*. Politecnico di Torino, 2015.

[35] E. G. Gol'shtein and D. B. Iudin. Zadachi lineinogo programmirovaniia transportnogo tipa. *Moscow*, 1969.

[36] M. Guignard and S. Kim. Lagrangean decomposition: A model yielding stronger lagrangean bounds. *Mathematical Programming*, 39(2):215 – 228, 1987.

[37] R. E. Guinness. Beyond where to how: a machine learning approach for sensing mobility contexts using smartphone sensors. *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013), Nashville*, pages 2868 – 2879, 2013.

[38] T. Hansen. Consumer values, the theory of planned behaviour and online grocery shopping. *International Journal of Consumer Studies*, 32:128–137, 2008.

[39] V. C. Hemmelmayr, K. F. Doerner, F. Richard, F. Hartl, and D. Vigo. Models and algorithms for the integrated planning of bin allocation and vehicle routing in solid waste management. *INFORMS Transportation Science*, 48(1):103 – 120, 2013.

[40] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, 2013.

[41] Y. Hinojosa, P. P. Justo, and F. Saldanha-da Gama. A two-stage stochastic transportation problem with fixed handling costs and a priori selection of the distribution channels. *TOP*, 22(3):1123–1147, 2014.

[42] J. Holmstrom, K. Tanskanen, and V. Kamarainen. Redesigning the supply chain for internet shopping-bringing ecr to the households. *4th Logistics Research Network Conference Proceedings*, pages 261–267, 1999.

[43] D.J. Houck, J.C. Picard, and R.R. Vemuganti. The traveling salesman problem as a shortest path problem: Theory and computational experience. *Opsearch*, 17:93–109, 1980.

[44] Y. Huang and H. Oppewal. Why consumers hesitate to shop online: An experimental choice analysis of grocery shopping and the role of delivery fees. *International Journal of Retail & Distribution Management*, 34:334–353, 2006.

[45] J. Morana J. Gonzalez-Feliu. Are city logistics solutions sustainable? the cityporto case. *TeMA-Trimestrale del Laboratorio Territorio Mobilitá Ambiente*, 3(2):55–64, 2010.

[46] K. Jornsten, M. Nasberg, and P. Smeds. Variable splitting: A new lagrangean relaxation approach to some mathematical programming models. *Technical report, University of Link oping,Department of Mathematics*, 1985.

[47] P. Kall and S. W. Wallace. *Stochastic Programming.* John Wiley & Sons, 1994.

[48] V. Kamarainen and M. Punakivi. Developing cost-effective operations for the e-grocery supply chain. *International Journal of Logistics: Research and applications*, 5:285–298, 2002.

[49] M. Kaur and S. Kalra. A review on iot based smart grid. *International Journal of Energy, Information and Communications*, 7:11–22, 2016.

[50] M. Kaut, H. Vladimirou, S. Wallace, and S. Zenios. Stability analysis of portfolio management with conditional value-at-risk. *Quantitative Finance*, 7:397—-409, 2007.

[51] W. Klibi, F. Lasalle, A. Martel, and S. Ichoua. The stochastic multiperiod location transportation problem. *Journal Transportation Science archive*, 44:221–237, 2010.

[52] M. Kneafsey, R. Cox, L. Holloway, E. Dowler, L. Venn, and H. Tuomainen. *Reconnecting Consumers, Producers and Food: Exploring Alternatives*. Oxford: Berg, 2008.

[53] N. Kong, A. J. Schaefer, and A. Shabbir. Totally unimodular stochastic programs. *Mathematical Programming*, 138(1):1–13, 2013.

[54] N. Lahrichi, T. G. Crainic, M. Gendreau, W. Rei, and Rousseau L.M. Strategic analysis of the dairy transportation problem. *Journal of the Operational Research Society*, 66(1):44–56, 2015.

[55] R. Liu, Z. Jiang, R.Y. Fung, F. Chen, and X. Liu. Two-phase heuristic algorithms for full truckloads multi-depot capacitated vehicle routing problem in carrier collaboration. *Computers & Operations Research*, 37(5):950–959, 2010.

[56] S. A. Low and S. J. Vogel. Direct and intermediated marketing of local foods in the united states. *USDA-ERS Economic Research Report*, 128, 2011.

[57] R. Mansini and M. G. Speranza. A linear programming model for the separate refuse collection service. *Computers and Operations Research*, 25(7–8):659 – 673, 1998.

[58] V. Manzoni, D. Maniloff, K. Kloeckl, and C. Ratti. Transportation mode identification and real-time co2 emission estimation using smartphones how co2go works. 2011.

[59] T. Marsden, J. Banks, and G. Bristow. Food supply chain approaches: Exploring their role in rural development. *In Sociologia Ruralis*, 40:424–438, 2000.

[60] S. Martello and P. Toth. Linear assignment problems. *Annals of Discrete Mathematics*, 31:259–282, 1987.

[61] S. Martello and S. Toth. *Knapsack problems Algorithms and Computer Implementation*. John Wiley & Sons, 1990.

[62] D. McFadden. The revealed preferences of a public bureaucracy. 1968.

[63] D. McFadden. *Urban Travel Demand: A Behavioral Analysis*. North-Holland Publishing Co., 1975.

[64] D. McFadden. Economic choice. 2000.

[65] M. Mes, M. Schutten, and A. P. Rivera. Inventory routing for dynamic waste collection. *Waste Management*, 34:1564–1576, 2014.

[66] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulations and travelling salesman problems. *J. Ass. Comp. Mach.*, 7:326–329, 1960.

[67] M. A. Morganosky and B. J. Cude. Consumer response to online grocery shopping. *In International Journal of Retail & Distribution Management*, 28:17–26, 2000.

[68] R. A. Neff, A.M. Palmer, S. E. McKenzie, and R. S. Lawrence. Food systems and public health disparities. *In Journal of Hunger & Environmental Nutrition*, 4:282 – 314, 2009.

[69] ONDE - UWC Web Site. http://onde.city, 2017. Last Access: 13/04/2017.

[70] C.S. Orloff. A fundamental problem in vehicle routing. *Networks*, 4:35–64, 1974.

[71] G. Perboli. GUEST-OR. linking lean business and OR. *Workshop of the 28th European Conference on Operation Research*, pages Poznan, Poland, 2016.

[72] G. Perboli. The GUEST Methodology. http://staff.polito.it/guido.perboli/GUEST-site/docs/GUEST_Metodology_ENG.pdf, 2017.

[73] G. Perboli, R. Tadei, and L. Gobbato. The multi-handler knapsack problem under uncertainty. *Eur. J. Oper. Res.*, 236:1000–1007, 2014.

[74] K L Poh, K W Choo, and C G Wong. A heuristic approach to the multi-period multi-commodity transportation problem. *Journal of the Operational Research Society*, 56(6):708–718, 2005.

[75] M. Punakivi and J. Saranen. Identifying the success factors in e-grocery home delivery. *International Journal of Retail & Distribution Management*, 29:156–163, 2001.

[76] K. N. Qureshi and A. H. Abdullah. A survey on intelligent transportation systems. *Middle-East Journal of Scientific Research*, 15:629–642, 2013.

[77] K. Ramus and N. A. Nielsen. Online grocery retailing: what do consumers think? *In Internet Research*, 15:335–352, 2005.

[78] H. Renting, T. Marsden, and J. Banks. Understanding alternative food networks: Exploring the role of short food supply chains in rural development. *Environment and Planning A*, 35:393–411, 2003.

[79] L. Ruffino. *Open Data e algoritmi di Map Matching per la classificazione degli spostamenti in ambito urbano.* Politecnico di Torino, 2017.

[80] S. Sahoo, S. Kim, B. Kim, B. Kraas, and A. Popov. Routing optimization for waste management. *Interfaces*, 35(1):24 – 36, 2005.

[81] F. Santini and S. G. Paloma. Short food supply chains and local food systems in the eu: a state of play of their socio-economic characteristics. *Publications Office*, 2013.

[82] A. Schrijver. Theory of linear and integer programming. 1986.

[83] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1998.

[84] M. A. Shafique and E. Hato. Modelling of accelerometer data for travel mode detection by hierarchical application of binomial logistic regression. *Transportation Research Procedia*, 10:236,244, 2015.

[85] M. A. Shafique and E. Hato. Use of acceleration data for transportation mode prediction. *transportation*, 42:163,188, 2015.

[86] M. A. Shafique and E. Hato. Travel mode detection with varying smartphone data collection frequencies. 2016.

[87] D. Shin, D. Aliaga, B. Tuncer, S. Muller, S. Kim, D. Zund, and G. Schmitta. Urban sensing: Using smartphones for transportation mode classification. *Computers, Environment and Urban Systems*, 53:76,86, 2015.

[88] G. Stefansson. Collaborative logistics management and the role of third party service providers. *International Journal of Physical Distribution & Logistics Management*, 36(2):76–92, 2006.

[89] R. Tibshirani T. Hastie and J. H. Friedman. The elements of statistical learning : Data mining, inference, and prediction. 2001.

[90] R. Tadei, N. Ricciardi, and G. Perboli. The stochastic p-median problem with unknown cost probability distribution. *Oper. Res. Lett.*, 37:135–141, 2009.

[91] The GUEST Initiative. http://www.theguestmethod.com, 2017. Last Access: 20/04/2017.

[92] T.P. Thierry and Y. Crama. Multi-period vehicle assignment problem with stochastic transportation order availability. *Odysseus Proceedings*, 2015.

[93] L. Thurstone. The measurement of social attitudes. *Journal of Abnormal and Social Psychology*, (27):249–269, 1931.

[94] C. Train, K.; Winston. Vehicle choice behavior and the declining market share of us automakers. *International Economic Review*, 48:1469–1496, 2007.

[95] K. Train. A validation test of a disaggregate mode choice model. *Transportation Research*, 12:167–174, 1978.

[96] NASS National Agricultural Statistics Service USDA. *2007 Census of Agriculture*. Washington, D.C., 2009.

[97] NASS National Agricultural Statistics Service USDA. *2012 Census of Agriculture*. Washington, D.C., 2009.

[98] T. Van Rooijen and H. Quak. Binnenstadservice. nl–a new type of urban consolidation centre. *European Transport and Contribution*, 2009.

[99] L. Verdonck, A. Caris, K. Ramaekers, and G. K. Janssens. Collaborative logistics from the perspective of road transportation companies. *Transport Reviews*, 33(6):700–719, 2013.

[100] J. P. Watson and D. L. Woodruff. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Science*, 8(4):355–370, Nov 2011.

[101] website. http://www.itu.int/en/itu-d/statistics/pages/stat/default.aspx. *visited 26-10-2016*.

[102] website. https://www.forbes.com/forbes/welcome/?toURL=https://www.forbes.com/sites/billbarol/2010/11/26/bring-buddy-dhl-crowdsources-your-grandma/&refURL=https://www.google.it/&referrer=https://www.google.it/. *visited 26-10-2016*.

[103] website. http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html. *visited 26-10-2016*.

[104] J. Wollenburg, A. H. Hubner, and H. Kuhn. Conceptual framework for order fulfillment and delivery in online grocery retailing, 2014.

[105] G. Xiao, Z. Juan, and J. Gao. Travel mode detection based on neural networks and particle swarm optimization. *Information*, 6:522,535, 2015.

[106] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1:22–32, 2014.

[107] X. Zhang and J. F. Bard. A multi-period machine assignment problem. *European Journal of Operational Research*, 170(2):398 – 415, 2006.