**UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA**

**Dottorato di Ricerca in Lavoro, Sviluppo e Innovazione**

XXXV Ciclo

# Integrated Optimization and Decision Support Systems for Attended Home Delivery and Service Problems

**Candidato**
Dario Vezzali

**Relatore**
Prof. Manuel Iori

**Correlatore**
Prof. Carlo Alberto Magni

**Coordinatrice del Corso di Dottorato**
Prof.ssa Tindara Addabbo

Settore Scientifico Disciplinare: MAT/09 - RICERCA OPERATIVA

*To “zia” Paola*

# Abstract

This research addresses the wide class of attended home delivery and service problems, which have been widely studied in the last two decades. In addition, the recent COVID-19 pandemic has boosted the interest in attended home delivery and service, with a significant increase in terms of global demand and a still lasting effect on people's habits. From an operational perspective, such problems require the customer to be present at home when the goods are delivered or the service is executed. Typically, the service provider and the customer agree on a particular time window for the delivery of goods or the execution of a service. The purpose of this research is to review the state of the art for attended home delivery and service problems, and study specific real-world applications in gas and water distribution, as well as in the context of global service providers. In particular, the solution of real-world optimization problems through integrated decision support systems, which rely on mathematical formulations plus additional modules, is investigated. This requires a careful problem definition, to clearly state the objective function and the main constraints of the application at hand, followed by the implementation in an exact or heuristic fashion, and ended with several computational experiments aimed at producing valuable solutions. This iterative process implies a preliminary real-data collection and preparation, which has to be performed thoroughly, to compute all the relevant information that occurs in the decision-making process. The proposed methodology integrates classic techniques of Operations Research with machine learning, to predict missing information for future periods, with multi-criteria decision analysis, to define and weight the multiple factors that determine a complex decision, with the principles and models provided by Engineering Economics, for evaluating a project from a financial perspective. The resulting methodology has been applied to a number of real-world applications.

**Keywords:** Attended Home Delivery, Attended Home Service, Integrated Optimization, Decision Support Systems, Real-world Applications.

# Abstract (in italiano)

La presente tesi di ricerca analizza la classe dei problemi di consegne e servizi a domicilio, che sono stati ampiamente studiati nell'ultimo ventennio. Inoltre, la recente pandemia da COVID-19 ha aumentato enormemente l'interesse in tema di consegne e servizi a domicilio, con un consistente incremento della domanda globale e un evidente cambiamento nelle abitudini delle persone. Da un punto di vista operativo, tali problemi richiedono la presenza del cliente per la consegna dei beni o l'esecuzione del servizio a domicilio. Lo scopo principale di questa tesi di ricerca riguarda l'analisi approfondita dello stato dell'arte sui problemi di consegne e servizi a domicilio, e lo studio di specifiche applicazioni reali nel settore della distribuzione dell'acqua e del gas, e nell'ambito delle aziende "global service". In particolare, viene studiato come risolvere problemi reali di ottimizzazione per mezzo di sistemi a supporto delle decisioni, basati su formulazioni matematiche del problema e moduli aggiuntivi. Questo richiede un'attenta definizione del problema, attraverso la formulazione della funzione obiettivo e dei principali vincoli, seguita dall'implementazione esatta o euristica e conclusa con una serie di test computazionali volti alla generazione di soluzioni efficaci ed efficienti. Questo processo iterativo richiede anche un'accurata fase di raccolta dati, e relativa preparazione, in modo da processare tutte le informazioni rilevanti che incorrono nel processo decisionale. La metodologia proposta integra tecniche classiche proprie della ricerca operativa con il *machine learning*, per la previsione di informazioni mancanti relative a periodi futuri, l'analisi decisionale multicriterio, per la definizione e il calcolo dei pesi dei molteplici criteri da considerare nel prendere una decisione complessa, e con gli strumenti offerti dall'*Engineering Economics*, per la valutazione di un progetto da una prospettiva finanziaria.

**Parole chiave:** consegne a domicilio, servizi a domicilio, ottimizzazione integrata, sistemi a supporto delle decisioni, applicazioni reali.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Acknowledgements

First of all, I would like to thank my tutor, Prof. Manuel Iori, for giving me the opportunity to start a Ph.D. after many years from the MSc. He was key to inspiring my interest for Operations Research since I was a graduate student. Thanks also for all the support in this three-year journey as a Ph.D. student.

Many thanks also to Prof. Carlo Alberto Magni for being such a great co-tutor and initiating me into the field of Engineering Economics. I appreciated his intellectual honesty and his rigorous approach, and I enjoyed being part of his General and Financial Mathematics course at the Department of Economics “Marco Biagi”.

Heartfelt thanks to Prof. Jean-François Cordeau for the warm welcome that I received as a visiting student at HEC Montréal and CIRRELT. His collaboration, constant support and tutorship were greatly appreciated.

Many thanks also to Prof. Bruno Petrato Bruck, for being such a great colleague and friend to work with in all these years, and to my co-authors that helped me in developing my research project and writing this thesis. In particular, I would like to thank Prof. Daniele Pretolani.

I would like to thank also the “Marco Biagi” Foundation for all the organizational effort, before, during and after the pandemic. In particular, many thanks to Jenny Rivas and Gabriele Benassi.

Many thanks also to my colleagues for the authentic collaboration, friendship and patience. In particular, I would like to thank Stefano Novellani, Giulia Caselli, Mirko Mucciarini and Mirko Cavecchia.

Heartfelt thanks also to my closest friends with whom I shared many important moments in these years. Their discreet support was very helpful to obtain this achievement.

Special thanks to my family, and in particular my mother Patrizia, my father Andrea and my sister Serena, for having supported me in the crazy idea of quitting a good job to chase the dream of working in the field of knowledge and education.

Finally, but not least importantly, I would like to thank Vera for encouraging me everyday to give my best and focus on what really matters in life.

Dario Vezzali

# Introduction

In the last two decades, *Attended Home Delivery* (AHD) and *Attended Home Service* (AHS) business models have experienced a fast growing rise, with positive implications on flexibility and quality of service level offered to final customers. At the same time, costly externalities linked to these last-mile operations (e.g., additional congestion in transportation systems, increased environmental pollution, and lack of labor policies to regulate platform work) have emerged. Some of these challenges and opportunities in AHD and AHS were already addressed in the seminal work by [2]. Recently, with the *COVID-19* pandemic being one of the amplifying factors of the even more rapid growth of AHD and AHS, further challenges, opportunities and shortcomings have arisen in this research field.

For this reason, the main purposes of this thesis are to provide an extensive review on the state of the art for AHD and AHS problems, to set the theoretical framework for this important class of problems, and present different applications on the development of *Decision Support Systems* (DSS), to help companies in tackling their real-world AHD and AHS problems in a more automated, effective, efficient and sustainable way.

The methodology used to model such real-world AHD and AHS problems is the typical *Operations Research* (OR) modeling approach described by [90], which consists of the following phases: (i) problem definition and data gathering, (ii) mathematical model formulation, (iii) mathematical model implementation, (iv) mathematical model testing and validation, (v) DSS prototyping and integration, and (vi) DSS implementation. The innovative characteristic introduced by this work is the integration of the OR modeling approach with other well-established quantitative techniques, like *Machine Learning* (ML) and *Multi-Criteria Decision Analysis* (MCDA), and with principles and models of *Engineering Economics* (EE).

The thesis is structured as follows. In Chapter 1, a survey on AHD and AHS problems is provided. Given the multi-stage nature of these problems, the main articles on demand management and routing in AHD and AHS are reviewed. In line with the rest of the work, a focus on practical applications is kept.

A DSS for solving a specific three-stage AHS problem arising in the context of public tenders for the distribution of gas in minimum territorial areas is presented in Chapter 2. In this real-world AHS problem, ML is applied before optimization and simulation methods to recreate unknown information, based on available historical data and additional open data. Such an application builds upon the work of [33]. The DSS, implemented as a modular system, is currently used by *IRETI*, an Italian multi-utility company, to design and fine-tune the organizational models proposed in public tenders.

AHD and AHS problems also occur in a *business-to-business* environment, where companies provide services to other companies. In particular, one may find service providers that subcontract the execution of services to external qualified suppliers. This is the case of those general players competing in the facility management industry, named Global Service Providers (GSP). In their business model, the selection of the best supplier for a facility management contract represents a complex decision. Such a decision typically depends on multiple conflicting criteria. In Chapter 3, a DSS for a multi-criteria supplier selection in the facility management industry is described. In this study, MCDA is applied to group and weight the multiple criteria that occur in the problem of selecting the best supplier for a given contract. A DSS prototype has been implemented and tested with *H2H Facility Solutions SpA*, a real GSP company.

AHS problems may as well regard the monitoring of water distribution networks to detect potential leakages or sources of contamination. In Chapter 4, we present a real-world application encountered in the city of Mashhad (Iran), where a complex water distribution network comprising households/shops, reservoirs/tanks, wells and treatment plants is daily inspected by a group of technicians. In this particular application, precedence constraints and multiple visits arise, thus requiring an accurate mathematical formulation of the problem.

Finding an effective and efficient solution for real-world AHD and AHS problems in a more automated and sustainable way is not restricted to the operational side, but it may be translated in a suitable accounting and financial model to provide a reliable valuation of the technical project, as suggested by [123]. In Chapter 5, another AHS problem in the context of water distribution is addressed by integrating the typical OR modeling approach with the EE perspective. In particular, a thoroughly defined Net Present Value is set as the objective function of the mixed integer linear programming formulation that describes the specific smart-meter installation scheduling project faced by *IRETI* in the province of Reggio Emilia.

# Chapter 1

# A Survey of Attended Home Delivery and Service Problems with a Focus on Applications

Cordeau, J.F., Iori, M., Vezzali, D. (2022). A Survey of Attended Home Delivery and Service Problems with a Focus on Applications. *Working paper*.


## Abstract

The research field on Attended Home Delivery (AHD) and Attended Home Service (AHS) problems has experienced fast growing interest in the last two decades, with the rapid diffusion of online platforms and e-commerce transactions. The COVID-19 pandemic has just fostered that interest, raising further challenges, opportunities and shortcomings that have to be tackled to answer the need for innovative methodologies as well as new policy actions. The aim of this work is to provide an extensive literature review on the state of the art for AHD and AHS problems, with a particular focus on real-world applications. A discussion of possible future research directions is also provided.


## 1.1 Introduction

Attended Home Delivery (AHD) and Attended Home Service (AHS) are demanding last-mile operations, where the customer must be present at home for the delivery of goods, the execution of a service or, in some cases, both the delivery of goods and the execution of an additional service [2], [56]. Examples of AHD and AHS are, among others, the delivery of groceries directly at home, the delivery and installation of large furniture and appliances, or the provision of home healthcare therapies. By definition, they differ from Unattended Home Delivery (UHD) and Unattended Home Service (UHS) operations, which can be fulfilled without the customer being present at home. Examples of UHD and UHS are the delivery of parcels right in front of the door or inside a nearby parcel locker, or the reading of a meter installed outside an apartment. To limit the research area, in this work we focus only on those operations that are *attended* by the customers. For a detailed review on last-mile delivery concepts we refer the interested reader to [28]. We are neither interested in surveying the class of Same-Day Delivery (SDD) problems, for which we refer to [177], or in recent trends in last-mile delivery, such as the use of drones and autonomous delivery robots or crowdshipping, which are described in the detailed review by [28].

AHD problems originated in the context of e-grocery (see, e.g., [143] and [118] for seminal ideas) and, more generally, e-fulfillment (see, e.g., [6] for an in-depth introductory review). Since the first definition found in the work by [39], they have seen a continuous increase not only in terms of interest in the research community, but also in terms of importance in many business sectors. The COVID-19 pandemic has just fostered the demand for AHD services, as confirmed by a report released by the *Organisation for Economic Co-operation and Development*. In particular, during the first and second quarters of 2020 online retail sales have registered a worldwide increase of 14.8% to 16% in the United States and 30% in the 27 member countries of the European Union, with a similar trend in the Asia-Pacific countries [133]. How long this growth will last and whether we will ever return to the pre-pandemic levels is still matter for debate [179]. In the meantime AHD has already triggered irreversible changes in the logistics of our cities [167], and new trends are emerging in large metropolitan areas calling for further challenges [27]. Among these trends, we mention the delivery of building materials to contractors directly on site and the recent phenomenon of ultra-fast delivery of groceries in as little as 15 minutes. A further indication that AHD and AHS problems are drawing increasing attention is represented by the following analysis performed on Scopus. In particular, we looked for the number of documents per year where the entries "attended home delivery", "attended home service", "attended home deliveries", or "attended home services" appeared between 2006 and 2021. The results show a slightly yet constantly growing trend between 2006 and 2017, followed by a notable increase between 2017 and 2021. The detailed results are reported in Figure 1.1.

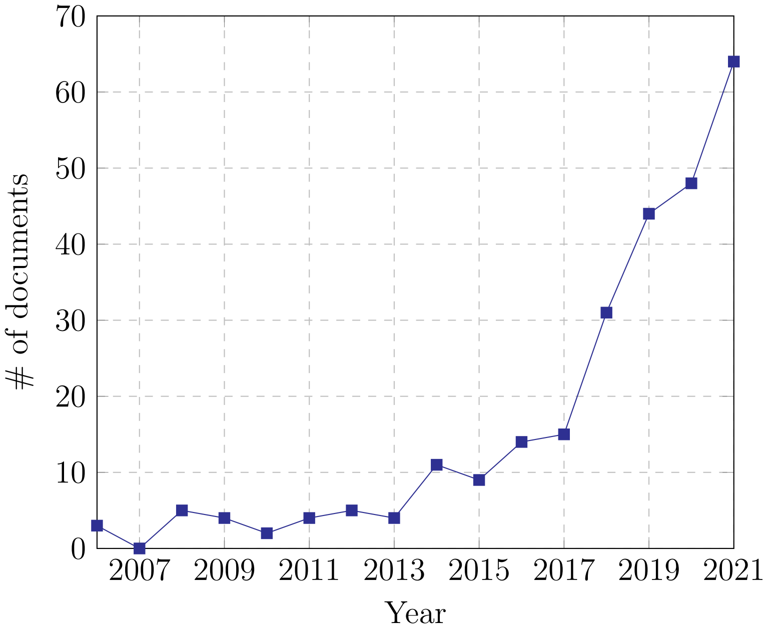



Figure 1.1: Documents per year on AHD and AHS published between 2006 and 2021

As mentioned before, AHD problems are directly linked to the growth of the *e-grocery* business model, where a fierce competition has arisen around the logistical challenges offered by this particular sector, like the perishability of goods, the unpredictability of demand, the narrow time windows made available to customers for the delivery, and the low profit margins. Even more challenging is the practice of *food delivery*, which has become increasingly popular in the last years. Another sector that is commonly associated with AHD is the *online retail* of so-called "dry" goods, where the perishability is not an issue, but the parcels may be fragile and require a careful handling, the demand volume can be very high and unpredictable, the goods need to be moved rapidly along the supply chain, and, lastly, the customer might not be at home during the delivery, thus causing additional routing costs and further congestion in city road networks. More traditional sectors are those of *large appliances and furniture*, which usually combine the delivery of goods to the additional installation service. In this sense, we can insert them at the intersection of AHD and AHS problems. Typically, these operations might require a careful handling due to the fragility of some appliances and furniture, but they usually benefit from a larger planning horizon.

The field of AHS itself has received less attention from the research community compared with AHD, but still includes some essential activities like *home care services*, that are important not only to efficiently manage the capability of hospitals but especially to guarantee high-quality therapies to patients who cannot move from home. In this context, we should distinguish between *ordinary* and *extraordinary* care services. The first can be planned over a larger planning horizon, while the latter deal with emergencies and must provide an immediate response. This leads to different problems from an operational research perspective. AHS problems typically arise also in the context of *multi-utilities* (e.g., electricity, gas and water distribution companies, internet and telecommunications service providers, and so forth), where companies might be committed by local authorities (see, e.g., [32], [33]) to give customers the opportunity to book their installation or maintenance services within publicly available time slots. As for home care services, we should distinguish these ordinary activities from extraordinary ones (e.g., a gas leakage) that require an immediate response. So far, we have mentioned only business-to-consumer sectors, but many observations also hold in a business-to-business environment. Indeed, *on-site maintenance and repair services* present similar characteristics to many AHS operations, including the distinction between ordinary and extraordinary services.

Facing real-world AHD and AHS problems is challenging, as it typically implies solving a multi-stage problem: firstly, a *demand management problem*, and consequently, a *routing problem*, where the decisions taken in the previous stage can greatly affect the feasibility as well as the profitability of the following decisions.

As clearly described in the recent surveys by [132], [158] and [183], on the *demand-side* companies must be able to find effective ways to efficiently leverage the demand of customers by putting into action principles of Revenue Management (RM). Initially borrowed from the airline industry, the practice of RM has become increasingly popular for AHD and AHS problems. Examples of RM decisions in the context of AHD and AHS problems might regard the basic offering and pricing of time slots, their length, the choice of overlapping versus non-overlapping time slots, or the capacity allocated to each of them. These are typically static decisions. More complex decisions are required in a dynamic environment, where a company might be willing to frequently adjust the offering and pricing of time slots, or increase/decrease the capacity allocated based on the actual demand of customers. The complexity of these decisions is also affected by the immediate responsiveness they typically require.

On the *supply-side*, companies seek to limit the operational costs by applying traditional routing techniques, which have been widely studied in the Vehicle Routing Problem (VRP) literature. The degree of complexity of these techniques is affected by the decisions taken at the demand management stage. However, in recent years stochastic and dynamic routing aspects are receiving increasing attention from the research community. In addition, AHD and AHS problems require considerable "back-end" activities in terms of inventory management and order assembly, which are out of scope of this work.

Finally, a meet-in-the-middle approach that is worth mentioning is to integrate demand management and vehicle routing, as discussed in the recent survey by [72].

AHD and AHS problems can also be classified according to the planning horizon of the decisions that must be taken. Long-term decisions typically dealing with the setup of business (i.e., with lasting effects from months to years), like the opening of new facilities or the creation of demand clusters given an extended geographical area, are taken at a *strategic* level. Medium-terms decisions typically dealing with the sizing of business (i.e., with lasting effects from weeks to months), like the design of basic model-weeks for each demand cluster or the allocation of capacity to each single time slot, are taken at a *tactical* level. Finally, short-term decisions typically dealing with the management of business (i.e., with lasting effects on a few days), like the dynamic adjustment of the basic time slot offering and pricing or the definition of detailed routing plans for the delivery of goods or the execution of services, are taken at an *operational* level.

Our work makes a number of valuable contributions and implements the recent surveys of [132], [158], [72], and [183]. In particular:

- it extensively reviews the academic literature by distinguishing for the first time between AHD and AHS problems;
- it looks at this relevant class of problems through the lens of real-world applications, with the aim of highlighting the main managerial leverages to stay in business in a profitable way.

The remainder of the paper is organized as follows. Mathematical models and solution methods for demand management, routing, and integrated demand management and routing problems in AHD and AHS are reviewed in Sections 1.2 and 1.3 , respectively, with a focus on real-world applications. Finally, in Section 1.4 we draw some conclusions on the state of the art of AHD and AHS problems and we discuss possible future research directions.

## 1.2 Demand Management Problems in AHD and AHS

The practice of Demand Management (DM) refers to those structural, price and quantity decisions that need to be taken in a business context. Synonymous with previously mentioned RM, DM has its origin in the early 1980s, when Robert Crandall, then American Airline's vice president of marketing, introduced the first principles of DM in the airline industry [166]. Since then, other industries adopted (and adapted) techniques of DM. Among others, we cite many service industries, like hospitality, transportation, and energy. As clearly explained by [166], all of these industries share similar conditions that motivate the adoption of DM: *customer heterogeneity*, *demand variability and uncertainty*, *production inflexibility*, *data and information system infrastructure*, and *management culture*. Many of these conditions may well be found in AHD and AHS systems, which probably explains why in recent years the practice of DM has become common in this industry.

A widely accepted classification of demand management decisions in AHD and AHS is the one proposed by [4]. On one dimension, the authors distinguish between *slotting* and *pricing* decisions, that deal with the proposal of time slots to customers and the definition of prices for each time slot, respectively. On the other dimension, they distinguish between *differentiated* (or *static*) and *dynamic* decisions, where the first are taken off-line and are usually based on forecasts, while the latter are taken in real time and all the available information is updated after each order.

The main difference between DM in traditional industries, where costs are generally supposed to be fixed, and DM in AHD and AHS, is that decisions taken at this level greatly affect the resulting routing costs. Therefore, even at early stage, it is necessary to seek a trade-off between revenue maximization and cost balance, which is not trivial.

In this section, we review several demand management models proposed in the literature on AHD and AHS problems, where the routing part is not the core of the work. An overview of the main characteristics of the reviewed articles is provided in Table 1.1. A particular emphasis is put on real-world applications. In addition, we highlight that column "Opportunity Cost Estimation" includes both rather simple methods, used to compute the additional routing cost while accepting an incoming request, and more sophisticated methods, used to estimate the opportunity cost of accepting an incoming request and forgoing a potentially more profitable future request.

For a more detailed study on DM/RM we refer the interested reader to the reviews by [163] and [105], where in the latter a specific section is dedicated to innovative applications of RM in AHD.

### 1.2.1 Slotting Problems

Although the authors do not refer directly to the problem of slotting, the paper of [26] may be considered a pioneering work in this area, as it anticipates the idea of using stochastic information in the decision to accept or reject a request. Indeed, the Multiple Scenario Approach (MSA) to dynamic stochastic Vehicle Routing Problem with Time Windows (VRPTW) they proposed fits well with the ordering phase of AHD problems that precedes the cutoff time, when the order requests arrive and must be accepted or rejected. Also, the MSA may be applied for practical implementations of maintenance and repair services, where it is not known a priori when the next call will arrive. The basic principle of MSA is to keep in memory a set of routing plans (and, among them, a distinguished plan whose selection is guided by a consensus function) that are updated at each execution step. These routing plans are generated by

Table 1.1: Overview of the main characteristics of demand management problems in AHD and AHS

| Context | Sector | Real-World Application | Degree of Dynamism | Problem | Planning Horizon | Objective | Main Framework | Customer-Choice Model | Opportunity Cost Estimation | Reference |
|---|---|---|---|---|---|---|---|---|---|---|
| AHS | Maintenance services | No | Dynamic | Slotting | Operational | Max AR | MSA | – | IH | [26] |
| AHD | E-grocery | No | Dynamic | Slotting | Operational | Max PR | SIM | RP | IH | [38] |
| AHD | E-grocery | Yes | Differentiated | Slotting | Tactical | Min RC | CA | – | CR | [3] |
| AHD | E-grocery | Yes | Differentiated | Slotting | Tactical | Min RC | ILP | – | SB | [3] |
| AHD | Online retail | Yes | Dynamic | Slotting | Operational | Max AR | SIM | – | IH | [57] |
| AHD | Large appliances | No | Differentiated | Slotting | Tactical | Min RC | MILP | – | – | [86] |
| AHS | Multi-utilities | Yes | Differentiated | Slotting | Tactical | Min RC | LNS | SS | ILP | [33] |
| AHD | E-grocery | No | Dynamic | Slotting | Operational | Max PR | LP | GAM | SB, MILP | [120] |
| AHD | E-grocery | Yes | Dynamic | Slotting | Operational | Max AR | SIM | SP | IH | [110] |
| AHD | E-grocery | Yes | Dynamic | Slotting | Operational | Max RV | SIM | MNL | ADP | [114] |
| AHD | E-grocery | No | Dynamic | Slotting | Tact./Oper. | MULTIPLE | SIM | MNL | IH | [115] |
| AHD | E-grocery | No | Dynamic | Pricing | Operational | Max PR | LP | SP | IH | [39] |
| AHD | E-grocery | No | Dynamic | Pricing | Operational | Max PR | DP | MNL | – | [14] |
| AHD | E-grocery | Yes | Dynamic | Pricing | Operational | Max PR | DP | MNL | IH | [189] |
| AHD | E-grocery | Yes | Dynamic | Pricing | Operational | Max PR | ADP | MNL | CR | [188] |
| AHD | E-grocery | No | Dynamic | Pricing | Operational | Max PR | ADP | MNL | IH, SB | [106] |
| AHD | E-grocery | Yes | Differentiated | Pricing | Tactical | Max PR | MILP | GNR | SB | [107] |
| AHD | E-grocery | No | Dynamic | Pricing | Operational | Max PR | ADP | MNL | IH, RB | [109] |
| AHD | E-grocery | No | Dynamic | Pricing | Operational | Max PR | QP | SP | ADP | [176] |
| AHD | Large appliances | No | Differentiated | Pricing | Operational | Min TC | DP | AP | – | [191] |
| AHD | E-grocery | No | Dynamic | Pricing | Operational | Max PR | LP | MNL | CR | [162] |

List of abbreviations: Number of Accepted Requests (AR), Profit (PR), Routing Costs (RC), Revenue (RV), Total Cost (TC), Multiple Scenario Approach (MSA), Simulation (SIM), Continuous Approximation (CA), Integer Linear Programming (ILP), Large Neighborhood Search (LNS), Linear Programming (LP), Dynamic Programming (DP), Approximate Dynamic Programming (ADP), Mixed Integer Linear Programming (MILP), Quadratic Program (QP), Realization Probabilities (RP), Simulation Strategies (SS), General Attraction Model (GAM), Selection Probabilities (SP), Multinomial Logit (MNL), General Nonparametric Rank-Based (GNR), Acceptance Probabilities (AP), Insertion Heuristics (IH), Cluster-first, Route-second (CR), Seed-Based (SB), Route-Based (RB).

considering information on already known requests as well as possible future requests. The experimental results presented by the authors show that the MSA performs well compared to less sophisticated methodologies (e.g., a state-of-the-art greedy algorithm for the dynamic VRP and its Multi Plan Approach generalization) in terms of number of customers served and number of vehicles used.

Among the first to see a potential in the integration between order promise and order delivery phases, [38] proposed several insertion-based heuristics for AHD problems, which were tested on purposely created instances through different rounds of simulation. In particular, the authors developed a number of probability-based heuristics where the information on potential future orders (i.e., corresponding to the opportunity cost of accepting an order before the cutoff time) is considered in the decision to either accept or reject an order request. Compared to the common practice of accepting a fixed number of orders per time slot and to more simple dynamic insertion heuristics, the proposed probability-based heuristics are constantly more efficient in capturing the profitability of incoming requests. The authors extensively tested such heuristics by varying some experimental characteristics (e.g., the sparseness of requests, the probability of generating a request before the cutoff time, the number of preferred time slots, the number of vehicles, the revenue per delivery request, and the time slot length). In many cases, the probability-based heuristics were able to come very close to the results obtained in the presence of perfect information and, except in one case, they showed computational times that are compatible with practical implementations (which, indeed, represents a valuable achievement).

A milestone in the field of AHD is the work of [3], where the Time Slot Management Problem (TMSP) in AHD was defined for the first time. The authors studied the particular TMSP arising at Albert.nl, the leading Dutch e-grocer at the time, and proposed two alternative formulations for the problem, in which the expected delivery costs are minimized. The first extends the Continuous Approximation (CA) approach found in [48]; in particular, the authors start from a base schedule (e.g., the one adopted by the company) and they iteratively improve it until the expected routing costs decrease or a maximum number of iterations is reached. In this formulation, a "cluster-first, route-second" strategy is used to approximate the delivery costs. The second formulation is an Integer Linear Programming (ILP) model, which explicitly states the TMSP at hand and relies on the seed-based scheme originally proposed by [71] to approximate the routing costs. As shown by the computational experiments both formulations produce high-quality schedules, resulting in a slight reduction of delivery costs compared to the schedule used by the company (which was already good). But the greatest potential generated by the two formulations is that of automating the schedule design process; in this sense, the CA approach is better than the ILP model as it requires shorter computational times (even though it is important to remark that a tactical problem like the TMSP does not necessarily need to be solved in a matter of seconds at the expense of solution quality). Further remarkable findings are presented in the what-if analyses conducted by the authors, where the effects of potential changes (increase of demand, increase or decrease of vehicle capacity, increase or reduction of service level, and use of alternative time slot templates) are investigated. Among them, they remark the existence of a trade-off between the time slot length (the narrower the length, the higher the service level offered to the customers) and the routing efficiency (with an increase of up to 25% in delivery costs going from an entire shift length to a two-hour length). Also, they highlight the idea that introducing a demand clustering may have a beneficial effect of approximately 10% reduction in terms of delivery costs, as well as a demand growth may generate economies of scale thanks to the increased number of stops in the delivery areas.

Building upon the work of [38] as well as the results previously found by [59], [60], [57] developed and compared novel customer acceptance mechanisms for AHD applications in metropolitan areas. The innovative idea behind their work is represented by the introduction of time-dependent and stochastic travel time information in the decision-making process of accepting or rejecting an incoming order request. In particular, to take care of possible lateness, due to variable travel times in rush hours, and the so-called *lateness propagation* effect, which depends on accumulated travel time variations during the execution of delivery routes, the authors included a thorough computation of individual buffer times. Such computation was integrated in a time-dependent variant of the I1 insertion heuristic algorithm originally developed by [159]. The results obtained from several rounds of simulation show that the proposed acceptance mechanism generally outperforms alternative approaches, both static and dynamic, in terms of the number of accepted requests and potential to avoid lateness. The authors also investigated the effect of changes in some input parameters (e.g., distribution of customer locations between downtown and suburban areas, service times, time window length, lateness avoidance, and confluence of requests in popular time slots) and provided meaningful practical insights.

A different interpretation of the Tactical Time Slot Management Problem (TTSMP) was given in the work of [86], where the authors defined the TTSMP through a Mixed Integer Linear Programming (MILP) formulation and solved it heuristically. In particular, two alternative heuristics were proposed. The first heuristic relies on a three-phase decomposition, that initially solves a Periodic Vehicle Routing Problem (PVRP), in which the time slots in the TTSMP correspond to the periods in the PVRP, subsequently merges the routes obtained from Phase 1 over each day, and, finally, solves a VRPTW for each day in the planning horizon (i.e., optimizes the routes merged during Phase 2). The second heuristic interprets the TTSMP as a Periodic Vehicle Routing Problem with Time Windows (PVRPTW), in which the days in the TTSMP correspond to the periods in the PVRPTW while the time slots correspond to the time windows. Both problems were solved using a Unified Tabu Search algorithm that has proven to be efficient for these problems (see, e.g., [46], [47]). Although the first heuristic is competitive for being more generic and tractable with state-of-the-art techniques and available software, it is generally outperformed by the second heuristic both in terms of computational times and solution quality.

A very interesting real-world application of differentiated slotting in the context of multi-utilities was studied in the work of [33]. Here the authors addressed a particular problem arising from an Italian gas distribution company, named *IRETI*, in which the required quality of service level is exogenously fixed by the public authority that regulates the market, so there is no opportunity to influence the demand of customers using RM principles. As a consequence, the design of good quality time slot tables is fundamental to limit the routing costs generated after the actual demand is revealed. For doing so, the authors developed a three-step approach having at its core a Large Neighborhood Search (LNS) algorithm that iteratively improves an initial set of time slot tables by means of destroy and repair methods. Remarkably, the customer-choice behavior in the process of booking the preferred time slot for the execution of a service was reproduced using four alternative simulation strategies (evenly horizontal, evenly vertical, rescheduling based, and popularity based). The cost of the solutions computed by the LNS algorithm is evaluated through a Multidepot multiple Traveling Salesman Problem (MmTSP), which relies on a *time-extended network*. Note that a different MmTSP is solved for each day in the booking horizon. The results obtained on real-case instances showed an expected reduction of routing costs in the order of 5% to 15% compared to the company's solution. Another beneficial contribution is represented by the development of an automated approach that may be helpful in case the public authority decides to update the required quality of service levels.

Following the research avenue opened by [14], [189], [188], and [106], which is discussed in the subsequent section on pricing problems, [120] proposed a new approach for the dynamic TMSP in AHD. In particular, the author was the first to introduce a customer-choice model in the context of slotting problems; namely, he used a General Attraction Model (GAM) (see, e.g., [75]), of which the Multinomial Logit (MNL) largely found in the stream of literature on pricing problems is a special case. The advantage of using the GAM, instead of the MNL, is to avoid a potential overestimation of the choice probabilities in particular settings. Another noteworthy contribution of this work is the definition of a novel MILP model to approximate the value function, hence the opportunity costs, of the Dynamic Program (DP) underlying the slotting problem. In doing so, the author built upon the work of [106], combining insertion heuristics, for the computation of the routing costs associated to already accepted orders, and a dynamic seed-based scheme, to estimate the delivery costs of expected future orders. The resulting online slotting problem is solved through a Linear Programming (LP) formulation derived from a Non-Linear Binary Program. In the computational experiments performed using relaxed versions of the proposed MILP model to favor real-time decisions, the results show a potential increase of 4 to 7% in terms of average profit compared to benchmark policies (where the opportunity costs are estimated using insertion heuristics and a myopic approximation, respectively). Also, the proposed methodology demonstrates a more stable DM of the incoming requests from the early phases in the booking horizon, especially in scenarios in which the capacity is tight.

The idea of adding flexibility to the slotting problem was introduced in the work of [110], where the authors presented four alternative algorithmic approaches to derive the time slot offering for each incoming customer request. Their main contribution was to investigate the effect of proposing both long time windows (i.e., of 4 hours), to preserve a certain flexibility in building the tentative routing plan during the booking horizon (especially in the early phases), and short time windows (i.e., of 30 minutes), which are commonly used in the e-grocery business sector. The results obtained on different demand scenarios (one derived from a German e-grocer) were greatly affected by the customers' willingness to accept long time windows, but they show a clear potential in terms of increased number of accepted orders compared to the benchmark approach in which only short time windows are offered. Additionally, the information on proximity between already accepted orders and incoming requests (that is included in two of the proposed algorithmic approaches) has proved to be a further key in the decision of which type of time window (i.e., long or short) to offer.

In the first of a series of papers on dynamic slotting, [114] studied incremental modular approaches that rely on the idea of anticipating, through simulation during an offline phase preceding the booking horizon, the information on delivery schedules and opportunity cost. In particular, the authors solve a Team Orienteering Problem with Multiple Time Windows to build anticipatory schedule patterns, while they apply an Approximate Dynamic Programming (ADP) to estimate the opportunity cost (taking inspiration from the work of [188] on dynamic pricing that is reviewed in the following section). During the online booking phase, an Assortment Optimization Problem is solved to derive the set of time slots proposed for each incoming request, adding a Theft-based mechanism to dynamically adjust delivery capacity by "stealing" extra capacity from neighboring areas of the previously determined schedule patterns.

In their following work, [115] were the first to introduce the Multi-Criteria Dynamic Slotting Problem, where they seek to (i) maximize revenue, (ii) maximize the visibility of branded trucks, and (iii) maximize the social influence produced by the most influencing groups of customers, using a scalarized objective function. The last two objectives are in line with marketing principles, but the proposed approach is flexible and adaptable to other sets of criteria. Extending the ideas of [45], another important contribution of this work is represented by the swap of the typical stages that compose an AHD problem. Indeed, the authors first solve a multi-objective a priori routing problem, which is based on forecasted requests and whose aim is to predict the information on available time slot capacities, and later apply RM techniques to dynamically determine the offering of time slots as requests arrive. Also, we highlight that a strong assumption made by the authors is to consider delivery resources as fixed, therefore they do not insert the routing cost minimization among the multiple objectives of the a priori routing problem. Such an assumption is justifiable given the scope of their work.

### 1.2.2 Pricing Problems

Building upon their previous work (i.e., [38]), [39] addressed the use of incentive schemes to steer customer behavior in AHD services. In particular, the authors propose two alternative LP formulations to solve the Home Delivery Problem with Time Slot Incentives and the Home Delivery Problem with Wider Slot Incentives, respectively, that do not incorporate a proper customer-choice model but use, instead, simple selection probabilities. In both formulations, an estimation of the delivery costs of accepted orders, performed using a combination of insertion heuristics (see, e.g., [159] and, later, [37]) and randomization, is inserted in the objective function. In addition, the feasibility of the routes under construction is checked. Interestingly, the results show that companies could take advantage from the use of incentive schemes (preferably incorporating intelligence to enhance their performance, especially when a larger number of time slot is involved) to reduce delivery costs and, consequently, increase profits even in the early stages of the decision process (i.e., when few orders have been processed). The authors also demonstrate that developing incentives schemes for wider time slots is easier and has the potential to produce an increase in profits as well (additionally determining a benefit in terms of flexibility in building efficient routes).

[14] developed a dynamic pricing model that dynamically adjusts the delivery prices of multiple delivery options over a discrete booking horizon according to the remaining time (in the booking horizon), the residual capacity, and the affinity of customers with a particular class (which characterizes their arrival probability, expected profit,

predictable utility for each delivery option and price sensitivity). The authors adopt a Logit-based model to reproduce the customer-choice behavior and a discrete-time, discrete-state Markov Decision Process to set the pricing decisions of the e-grocer. Using simple examples, they demonstrate how optimal prices may change over time and how an increase or decrease in terms of capacity can influence them, even in the case when more than one class of customers is considered.

[189] defined a dynamic programming framework for the dynamic pricing of delivery time slots based on a thorough demand model, where the arrival of customers for a single delivery day is estimated using a time-dependent Poisson process, while the selection of time slots within a given delivery day is modeled through an MNL model. The dynamic program is defined to gain insights for the development of good pricing policies, as is not solvable in short computing times due to the curse of dimensionality and the VRP with Time Windows that must be solved at each stage. To overcome this problem, during the online booking phase (i.e., when the decision on the dynamic pricing of time slots has to be taken in the order of milliseconds) an approximation of the routing costs is computed based on the insertion heuristics found in [39] and an online pricing problem is solved. As a valuable result, the authors demonstrate that a dynamic pricing policy that includes an estimation of the delivery costs for expected future orders, instead of focusing only on already accepted orders, is preferable. Moreover, they show how a similar policy produces a remarkable increase in terms of total profits (i.e., 3.8% on average) compared to the common industrial practices of using static prices or order-based prices for time slots. This effect is even more evident when capacity is scarce. The work was motivated by an industrial partnership with a major e-grocer in the United Kingdom that provided anonymized booking data that were used to train the models and perform different runs of simulation. Building upon their previous work and using the same sample data provided by a major e-grocer operating in the Greater London area, [188] developed an APD procedure. In particular, the proposed approach adopts a dynamic pricing policy that incorporates both approximated delivery costs (obtained by applying the "cluster-first, route-second" approach originally proposed by [48]) and estimated revenues to compute the opportunity costs from expected future orders. Remarkably, the results show an average total profit increase of more than 2% compared to base policies where no opportunity cost is considered, and a computational time compatible with real-world applications.

[106] presented a novel MILP formulation to approximate the opportunity costs in dynamic pricing problems. In the proposed approach, which is repeated in an iterative way for each customer request received within a discrete booking horizon, the authors combine insertion heuristics (to compute the delivery cost for already accepted orders), an MNL model (to anticipate expected customers' reactions to future pricing decisions and, consequently, estimate future revenues), a dynamic seed-based approximation (to estimate the delivery costs of expected future orders), and the MILP formulation (to approximate the value function of a customer request in an dynamic programming framework). The results show an average increase in terms of total profits compared to common policies (e.g., fixed price and order value-based), as well as the "Foresight Policy" by [189], which is considered as a benchmark policy by the authors. The so-obtained total profit is on average 5.5% higher in the first case, and 2.3% higher in the latter case. In addition, they find that a regular recalculation of the opportunity costs is preferable (if computationally compatible with the requests arrival rate and the

requested response time) rather than a periodic, less frequent recalculation. However, in the worst case the total profit is 1.66% lower than the best result, thus confirming the robustness of the proposed methodology even in practical scenarios where the pricing decisions need to be taken instantly and the opportunity costs cannot be recomputed continuously.

[107] were the first to address the problem of pricing from a tactical perspective, proposing different variants of an exact MILP formulation for the Differentiated Time Slot Pricing Problem (DTSPP). In their work, motivated by an industrial partnership with a German e-grocer, the customer-choice behavior is modeled using a general non-parametric rank-based approach where the preferences of customers (assuming that all customers in a particular segment share the same preferences) are expressed through simple preference lists of slot-price tuples (named *time slot price point combinations*). The restrictions imposed by the demand management problem are embedded into the MILP formulation in a first group of constraints, while the restrictions imposed by the routing problem are embedded into a second group of constraints. Here we can find typical route construction, demand and capacity, and time window constraints. Given the NP-hardness of the DTSPP, the authors proposed two alternative model-based approximations for the routing constraints, one seed-based [71] while the other adapting and extending the approach found in [3]. After an extensive computational study, the authors demonstrate that at a tactical level it is preferable to adopt model-based approaches that embed routing constraints. In fact, an early approximation of the delivery costs results in higher profits compared to diffused practical pricing approaches. In this sense, a trade-off between more accurate formulations, where the delivery cost approximation is more elaborate at the expense of integrality gap, and less accurate formulations, where the delivery cost approximation is particularly rough but they can be solve to optimality, needs to be found.

Extending previous works and combining them with ideas from recent streams of literature on the VRP, [109] were able to propose a route-based ADP approach for dynamic pricing, where the opportunity cost due to the displacement of potential future orders (which is a function of the particular system state of the dynamic program) is carefully estimated through a route-based formulation borrowed from the Stochastic Dynamic VRP literature (see, e.g., [170]). In particular, the authors used artificial skeletal routes to improve the estimation of future routing costs and introduced a time window budget approach to better evaluate the idle time of vehicles within the time windows. These features serve as an input for the online pricing problem, which is solved using an efficient heuristic algorithm (compatible with the required limited computational time of real-world applications). The proposed simulation study shows that the performance of the route-based ADP approach with time window budget is superior compared not only to another ADP approach with waiting time (proposed by the same authors), but also to other policies adapted from the literature (among which the one found in the work by [188]). Such superiority is expressed both in terms of average profit and number of served customers. Interestingly, the authors further demonstrate how the use of artificial skeletal routes determines a general improvement for all the policies compared to the case where routes are built starting from an empty route plan. Another valuable change that the authors introduced in this work, compared to the previous literature, is represented by the use of a finite-mixture MNL model as the customer-choice model.

[176] developed an Incentive-Routing Optimization framework for solving the dynamic pricing problem in AHD, where the pricing problem itself is formulated as a Quadratic Programming (QP) model with the objective of maximizing the total expected profits. As in [39], the authors adopt a simple model to shape the customer-choice behavior, based on selection probabilities and a linear response to incentives. The QP formulation receives as an input the marginal fulfillment cost of each incoming order, which is computed through an ADP mechanism. The boundary condition for the ADP is obtained by solving an independent VRP with Service Choice for each time slot; to reduce the computational time (which is essential in real-world applications) this particular sub-problem is solved using a minimum-regret construction heuristic [139]. Compared to a "Free Choice" policy, where the customers are free to choose their preferred time slot, and a "Myopic Incentive" policy, where the incentives are set based only on the QP model (with a myopic marginal cost anticipation), the "ADP Incentive" approach proposed by the authors shows better results in terms of total costs and fulfilled orders. The results are confirmed by a sensitivity analysis on some parameters (e.g., order density, arrival probability, and number of vehicles).

[191] studied the Pricing for Delivery Flexibility Problem where, unlike in other reviewed articles, they seek to minimize the total expected cost (which comprises both the delivery costs and the discounts offered to customers for changing the delivery day). The idea is to increase the delivery flexibility by proposing a discount to those customers that accept a different delivery day than the preferred one, with the objective to reduce the delivery costs. To solve the problem the authors implemented an exact dynamic programming algorithm where the customer-choice behavior is modeled through acceptance probabilities. Several computational experiments were performed to evaluate the potential of cost reduction in presence of different properties (i.e., distribution of customers' preferred delivery day, distribution of customers' delivery locations, customers' willingness to accept discounts, revenue generated by the orders and vehicle capacity). The results show an expected cost reduction of more than 30% in the best cases, albeit a similar approach may result applicable only to those applications where the level of detail is the delivery day and the demand volume is not so high (e.g., large appliances).

The opportunity of proposing flexible time slots (i.e., adjacent or non-adjacent) compared to single standard time slots is investigated in the work by [162], where a dynamic pricing approach based on a LP formulation is developed. The authors show how the offering of flexible time slots to customers may be beneficial for companies in reducing delivery costs, as it gives them more flexibility to build their routes. An additional and interesting insight regards the composition of the proposed flexible time slots. Indeed, a combination of more popular and less popular non-adjacent time slots is able to generate higher total profits compared to adjacent time slot, especially when the capacity is tight relative to the demand.

A promising work that is worth mentioning and might open new directions for dynamic pricing implementations is the one by [117], where the authors studied several mathematical properties of the pricing problem, in the context of AHD, that can be used to find closer approximations of the value function in dynamic programming algorithms.

## 1.3 Routing Problems in AHD and AHS

In the broad sense, the VRP consists in determining a set of minimum-cost routes for a set of customer requests, given a starting depot, a fleet of vehicles, and specific constraints depending on the application at hand. A rich body of literature on the family of VRPs is available, as these problems have been widely studied for more than 60 years and represent one of the leading domain in combinatorial optimization. We refer to [168] for an extensive review on the VRP and its main variants, and to [180] for a recent survey.

Given that they are associated with last-mile delivery operations, AHD and AHS problems are strongly related to city logistics, as the majority of deliveries is naturally condensed in populated urban areas. A detailed overview of VRPs arising in city logistics is provided by [42]. In recent years, we have also seen the emergence of new VRP variants in line with the increasing complexity and variety of real-world applications; a brief overview of this topic can be found in the survey of [175], where the authors focus on emerging metrics to evaluate VRP solutions (which may give several hints for novel multi-criteria formulations), integrated approaches where the VRP is linked to upstream decisions and sometimes conceived as an evaluation tool for these decisions (which, to some extent, can be the case of AHD and AHS applications), and refinements of existing models.

When we consider the routing stage of AHD and AHS problems, we are interested in solving a VRPTW, in which capacity constraints are typically not binding if compared to time window constraints. For state-of-the-art works on the VRPTW we refer to [29] for route construction methods and local search algorithmic techniques, [30] for metaheuristic algorithms, [98] and [22] for exact solution approaches, [174] for an efficient hybrid genetic algorithm, and [50] for mathematical formulations, as well as exact and heuristic methods. Recently, new VRPTW extensions have emerged, by considering stochastic service times [61], multiple trips per vehicle and time-dependent travel times [135], as well as synchronized visits [140]. In addition, the Electric VRPTW has received much attention for its practical implications (see, e.g., [154], [49], [89], [101], [102], [103], [104], [53], and [113]).

In the previous section, we have seen that the VRPTW may be used as a boundary condition in a DP framework, where the selected customer-choice model most of the times is an MNL model and a VRPTW must be solved for each state to update such boundary condition. However, this makes the problem intractable due to the NP-hardness of the VRPTW (see, e.g., [152]). We have also seen that this drawback can be partially overcome, at the expense of optimality, by applying approximate techniques, like insertion heuristics (see, e.g., [159] and [37]), "cluster-first, route-second" strategies (see, e.g., [48]), and seed-based schemes (see, e.g., [71]).

The anticipation of the routing costs during the demand management stage is another critical aspect in AHD and AHS problems. We have already introduced the idea that an early approximation of the routing costs leads to higher profits compared to pure revenue management approaches that are still diffused in practice. This idea was further investigated in the work of [36], where the authors proposed four MILP models, all based on the Set Covering formulation for the VRP. The four formulations are conceived to be integrated into more developed demand management models as "plug-in" modules to anticipate the estimation of the routing costs. The results show that the proposed models, decremental in terms of decision variables and constraints, approximate well the routing costs (i.e., the overestimation is no more than 10% compared to benchmark exact models, and slightly less than 3% compared to benchmark heuristics) in an acceptable computational time, thus resulting promising for real-world applications and suitable for decision supporting at a tactical level. In the aforementioned work by [107], the authors built on these preparatory findings, by introducing a routing module into their MILP formulation for the DTSPP.

Since a detailed review of routing problems would be too ambitious, we limit the scope of this section to the main routing models developed to solve specific AHD and AHS problems. An overview of the main characteristics of the reviewed articles is provided in Table 1.2. We remark that a particular emphasis is put on real-world applications.

### 1.3.1 Routing Problems in AHD

In the first work of a series of articles on VRPTW variants for AHD problems, [16] defined the Single-Vehicle Routing Problem with Time Windows and Multiple Routes (S-VRPMTW), where during a typical workday a single vehicle performs multiple routes of short duration for the delivery of perishable goods. The problem is solved using a two-phase solution approach based on the exact algorithm for the Elementary Shortest Path Problem proposed by [67].

In their second paper, [17] defined a multiple-vehicle generalization of the S-VRPMTW, named Vehicle Routing Problem with Time Windows and Multiple Routes (VRPMTW), solving it via branch-and-price. In particular, the primary problem is a Set Packing formulation solved through progressive linear relaxations of the restricted primary problem, while the pricing subproblem is an elementary shortest path solved using the aforementioned algorithm by [67].

[19] presented an Adaptive Large Neighborhood Search (ALNS) algorithm to solve the static version of the VRPMTW. Interestingly, the authors demonstrate the advantage of applying destruction and insertion operators at different levels (customer, route, and workday) instead of using only customer-based operators.

Building upon the problem definition presented by [17] and the ALNS algorithm implemented by [19], [18] solved the dynamic VRPMTW, where the source of dynamicity is given by the arrival of new customer requests during the operational horizon (i.e., while planned routes are executed). Note that such requests are inserted in future routes, as the current ones are fixed. Compared to the previously mentioned ALNS, a dynamic environment (in which the acceptance rule is slightly modified to take care of dynamicity) and an event management mechanism (to handle different types of event) were added. The results show that the proposed non-myopic approach (i.e., where future requests are considered) outperforms the myopic approach (i.e., where future requests are not considered) in terms of profit, percentage of served customers, number of routes per day, and number of customers per route, at the expense of considerably higher computational times (however acceptable and compatible with the response time required by an offline real-world application). Such results are confirmed by two sensitivity analyses in which the authors evaluate the impact of increasing the number of scenarios during the simulation and increasing/decreasing the number of customers, respectively.

Table 1.2: Overview of the main characteristics of routing problems in AHD and AHS

| Context | Sector | Real-World Application | Source of Stochasticity | Source of Dynamicity | Problem | Planning Horizon | Objective | Model Structure | Constraints | Solution Method | Reference |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AHD | E-grocery | No | – | – | S-VRPMTW | Operational | MULTIPLE | MILP | CP, TW | FDGG | [16] |
| AHD | E-grocery | No | – | – | VRPMTW | Operational | MULTIPLE | MILP | CP, TW | BP | [17] |
| AHD | E-grocery | No | – | – | VRPMTW | Operational | MULTIPLE | MILP | CP, TW | ALNS | [19] |
| AHD | E-grocery | No | – | NR | DVRPM | Operational | Max PR | MILP | CP, TW | ALNS | [18] |
| AHD/AHS | Cross-sectoral | No | TT | – | VRP-SITW | Operational | MULTIPLE | – | CP, TW | TS, LP | [97] |
| AHD | Online retail | No | NP, SD | – | SDVRPTW, AS | Operational | MULTIPLE | MILP, SDP | CP, TW | TS, ADP | [84] |
| AHD | Online retail | Yes | OR | NR | ISSLAP | Tact./Oper. | Min TC | 2-SP | CP, TW, DT, NB | MLM | [147] |
| AHD | Online retail | No | CA | – | VRSPTDC | Operational | MULTIPLE | MILP | TW | ALNS | [134] |
| AHD | Food delivery | Yes | – | – | MDRP | Operational | Min CC | MILP | S-CS, T-CS, CtD | CR | [190] |
| AHD | Food delivery | Yes | – | – | MDRP | Operational | Min CtD | MILP | S-CS, T-CS | CR | [190] |
| AHD | Food delivery | Yes | – | – | MDRP | Operational | Min RtD | MILP | S-CS, T-CS, CtD | CR | [190] |
| AHD | Food delivery | Yes | – | – | MDRP | Operational | Min CtDO | MILP | S-CS, T-CS | CR | [190] |
| AHD | Food delivery | Yes | – | – | MDRP | Operational | Min RtP | MILP | S-CS, T-CS, CtD | CR | [190] |
| AHD | Food delivery | Yes | OR, RT | NR | RMDP | Operational | Min ESD | MDP | DD | ACA | [171] |
| AHS | Home healthcare | No | – | – | VRSPTW | Operational | MULTIPLE | MILP | TW, PC, SYN | LBH | [31] |
| AHS | Home healthcare | Yes | – | – | PHCP | Tactical | Maxmin | MILP | SK, CC, WL | PGP | [40] |
| AHS | Home healthcare | Yes | – | – | PHCP | Tactical | Minmax | MILP | SK, CC, WL | PGP | [40] |
| AHD/AHS | Large appliances | Yes | – | – | DIRP | Operational | Min TC | MILP | CP, TW, PC, SYN | ALNS | [7] |

List of abbreviations: Travel Time (TT), No-show Probability (NP), Service Duration (SD), Order Requests (OR), Customer Availability (CA), Ready Time (RT), New Requests (NR), Single-Vehicle Routing Problem with Time Windows and Multiple Routes (S-VRPMTW), Vehicle Routing Problem with Time Windows and Multiple Routes (VRPMTW), Dynamic Vehicle Routing Problem with Multiple Routes (DVRPM), Vehicle Routing Problem with Self-Imposed Time Windows (VRP-SITW), Single-Depot Vehicle Routing Problem with Time Windows (SDVRPTW), Appointment Scheduling (AS), Integrated Shift Scheduling and Load Assignment Problem (ISSLAP), Vehicle Routing and Scheduling Problem with Time-Dependent Costs (VRSPTDC), Meal Delivery Routing Problem (MDRP), Restaurant Meal Delivery Problem (RMDP), Vehicle Routing and Scheduling Problem with Time Windows (VRSPTW), Palliative Home Care Problem (PHCP), Delivery and Installation Routing Problem (DIRP), Profit (PR), Total cost (TC), Courier Compensation (CC), Click-to-Door Time (CtD), Ready-to-Door Time (RtD), Click-to-Door Overage (CtDO), Ready-to-Pickup Time (RtP), Expected Sum of the Delay (ESD), Mixed Integer Linear Programming (MILP), Stochastic Dynamic Programming (SDP), Two-stage Stochastic Programming (2-SP), Markov Decision Process (MDP), Capacity (CP), Time Window (TW), Distance Traveled (DT), Neighboring (NB), Spatial Consistency (S-CS), Time Consistency (T-CS), Delivery Deadline (DD), Precedence (PC), Synchronization (SYN), Skill Constraints (SK), Continuity of Care (CC), Workload (WL), Feillet-Dejax-Gendreau-Gueguen Algorithm (FDGG), Branch-and-Price (BP), Adaptive Large Neighborhood Search (ALNS), Tabu Search (TS), Linear Programming (LP), Approximate Dynamic Programming (ADP), Multicut L-shaped Method (MLM), Column- and Row-Generation (CR), Anticipatory Customer Assignment (ACA), Local Branching Heuristic (LBH), Pattern Generation Policy (PGP).

An interesting characteristic introduced by the work of [97] is the use of self-imposed endogenous time windows rather than the exogenous ones typically considered in the VRPTW literature. Those self-imposed time windows are assigned to the customers by the company which, in turn, is committed to respecting them. A similar approach may be applicable to sectors like online retail, large appliances and furniture, as well as multi-utilities (especially for installation services). Another important feature included in this work is the presence of stochastic travel times that are dependent on a random variable representing a non-negative delay. Such delay is added to the base travel time. To solve the problem, the authors proposed a collaborative two-stage hybrid algorithm. First, the routing part is solved via tabu search using three alternative criteria for choosing a move. Second, the scheduling part, which takes as an input the solution found at the previous stage, is solved through an LP formulation that includes buffer times to handle the uncertainty given by the adoption of stochastic travel times. From a practical perspective, the use of self-imposed time windows may represent an unconventional policy (compared to the common practice of letting customers select their favorite time windows) to lighten the time windows constraints, thus reducing the operating costs (both in terms of traveled distance and number of required vehicles) while keeping a certain service level.

Inspired by the work of [153], [84] developed an integrative approach for solving the appointment scheduling and routing problem in the context of AHD. What characterizes this work is the inclusion of random customer behavior in the proposed model by considering no-show probabilities and random response times during the delivery phase. Such randomness typically represents a remarkable issue in real-world applications, frequently causing inefficient re-routing, potential disruptions, and extra costs. To solve the problem, the authors implemented a hybrid heuristic algorithm, which iteratively combines a tabu search metaheuristic, for solving the routing part, and an approximate dynamic programming algorithm, for solving the scheduling part. The results show how the proposed integrative approach outperforms a traditional hierarchical approach (in which the routing part is solved first, followed by the scheduling part). However, the computational times obtained on large instances warn against a potentially low compatibility with real-world cases, as the developed algorithm took almost 20 hours to solve instances with up to 5 vehicle and 50 customers. In addition, two sensitivity analysis were performed to evaluate the effect of increasing/decreasing the number of vehicles and, more interestingly, of using hard time windows rather than soft time windows (which were included in their problem definition).

In their work at the border between AHD and SDD, [147] introduced for the first time the Integrated Shift Scheduling and Load Assignment Problem. The problem, originating from a real-world start-up company offering last-mile delivery services in many cities of France, is formulated as a two-stage Stochastic Programming (SP) model. In particular, the first stage aims at designing tactical schedules for couriers, which are allocated to a restricted number of geographic areas (in such a way that the traveled distance is constrained and some neighboring conditions between "origin-destination" pairs are verified), while the second stage defines the assignment of customer orders to couriers. In this work, we have a co-presence of stochasticity (given a portion of stochastic orders generated using a Poisson distribution) and dynamicity (given a portion of orders that must be fulfilled according to a same-day delivery policy). To solve the problem, the authors implemented a multicut L-shaped method with some additional algorithmic refinements to generate initial cuts and derive valid inequalities. The main idea underlying this work is represented by the opportunity of using the tactical model to compare alternative policy offerings and to evaluate their impact on total cost and solution quality. In addition, the results show the advantage (in terms of optimality gap and potential for cost reduction) of including uncertainty when generating tactical solutions.

Resuming the idea originally proposed by [136] of using customer-related data to improve the effectiveness of AHD systems, [134] defined the Vehicle Routing and Scheduling Problem with Time-Dependent Costs (VRSPTDC). The problem is a variant of the VRPTW, as it adds a time-dependent penalty cost to the objective function. Such penalty cost is directly linked to the so-called customer availability profiles (introduced for the first time by [73]) that identify, for each customer, the probability of being present at home when the delivery is performed (given some historical data from which this information can be learned). In case the customer is absent during the first attempt of delivery, the authors assume that the next attempt is outsourced to an external courier, thus causing additional costs. From a practical perspective, the issue of low hit rates (i.e., frequent unsuccessful deliveries due to the absence of customers) is still one of the most significant problem in last-mile delivery. The VRSPTDC is solved using an ALNS-based metaheuristic algorithm with several removal and insertion operators. The results indicate the existence of a trade-off between the minimization of travel costs and the increase of hit rates. However, by taking advantage of customer-related data, it is possible to reach relevant cost savings. In particular, introducing the information on customer availability, in combination with the practice of waiting before serving a customer, may generate up to 40% in cost savings. Last but not least, the ALNS-based algorithm produced good results in comparison with a state-of-the-art MILP solver, and showed short computational times, which is desirable for a potential real-world application.

### A Focus on the Meal Delivery Routing Problem

Given the outstanding expansion of the food delivery sector in the last few years, a necessary exception from the main scope of our work is required by the Meal Delivery Routing Problem (MDRP). Such problem is part of AHD (in the sense that the customer must be present at home for the delivery of food), but it also comprises typical elements of SDD (with new requests coming during the operational horizon) as well as the use of innovative practices arising in last-mile logistics, like crowdshipping and bundle generation. For an overview on last-mile delivery challenges and, in particular, routing problems with crowdshipping we refer to [10], while for a recent work on routing with bundle generation and occasional drivers we refer to [124].

Among the first to study the MDRP, [190] introduced a mathematical formulation which is adaptable (with small adjustments) to different objectives (or metrics) that may be worth considering for an online food ordering and delivery platform (e.g., courier compensation, click-to-door time, ready-to-door time, click-to-door overage, and ready-to-pickup time). Interestingly, their work is based on the concept of work package, which is a possible way to serve a bundle of orders. To solve the problem, the authors implemented a column- and row-generation algorithm, enhanced by a selective column inclusion scheme, that proved to be effective on the MDRPLIB instance set publicly made available by Grubhub (an American online ordering and delivery platform and a subsidiary of Just Eat Takeaway). In addition, a noteworthy analysis reported by the authors demonstrates that by guaranteeing a minimum-pay to couriers does not cause a dramatic increase in terms of total cost (i.e., 9% in the worst case); quite the opposite, it ensures a large availability of couriers (which is desirable to offer better response time to customers). In our opinion, such an analysis may well contribute to the wide debate on policies for platform workers.

The Restaurant Meal Delivery Problem (RMDP) was also addressed by [171]. Inspired by the previous work of [170], the authors defined the RMDP as a route-based Markov Decision Process, solving it by means of an Anticipatory Customer Assignment (ACA) heuristic algorithm. Such an approach was strengthened by the use of time buffering and postponement to soften the effects of stochasticity (orders arrive following a Poisson distribution, while meal ready times are generated according to a gamma distribution) and dynamicity (new orders continue to arrive while deliveries are executed). The proposed policy was tested in an extensive computational study on real-world data from Iowa City. In comparison with the common-sense benchmark policy of assigning an incoming order to the driver that is able to deliver it as fast as possible (based on the information available at that moment), which is typically used in current practice, the results show that the ACA, relying on both time buffering and postponement, achieves dramatic improvements in terms of total delay. In particular, the use of time buffering itself produces significant improvements, as it decreases the effects of uncertain events (e.g., more time is needed to prepare a meal compared to the average time). With the addition of postponement, it is also possible to take advantage of newly collected information which favor the assignment, as well as the bundling, of orders. From a practical perspective, the proposed algorithm proved to be robust in presence of variability (e.g., different workloads and increasingly uncertain meal ready times) and suitable to solve real-world problems, being able to satisfy the objectives of multiple stakeholders.

### 1.3.2 Routing Problems in AHS

In this section, we are interested in reviewing some recent articles on routing models for AHS. However, given their practical implications, we cannot forget to mention seminal works, in the context of home care services, on service planning and patient-to-nurse assignment. Among these, we refer to [65], where the authors described *LAPS CARE*, a decision support system developed for the Swedish healthcare system, which is based on a set partitioning formulation and a repeated matching algorithm for optimizing the generation of attended home visiting schedules; another noticeable work is that of [54], where the case of *Landelijke Thuiszorg*, a Belgian home care service provider, is described. For what concerns the assignment of patients to traveling nurses, [88] developed an assignment algorithm to solve a real-world problem arising in a small area of Montréal (Québec), while [116] proposed a structural policy to guarantee the continuity of care (i.e., which means that a patient is visited by a restricted group of caregivers). For more references on routing and scheduling problems in home healthcare we refer the interested reader to the survey by [69] and to the recent survey by [62].

Starting from the real-world application described by [65], [31] defined a novel MILP formulation for the Vehicle Routing and Scheduling Problem with Time Windows (VR-

SPTW). The peculiarity of the VRSPTW is given by the presence of pairwise temporal precedence constraints and pairwise synchronization constraints. As discussed by the authors, similar constraints may be found in homecare staffing and scheduling problems, where different staff members are required to visit a patient one after the other or simultaneously (depending on the task that must be executed). The problem was solved using a local branching heuristic inspired by the "Diversification, Refining, and Tight-refining" method proposed by [70]. This solution method was tested by considering alternative objective functions (i.e., minimization of preferences, minimization of traveling time, minimization of maximal difference in workload among staff members, or minimization of a weighted sum of multiple objectives).

[40] addressed the Palliative Home Care Problem (PHCP), an important problem arising in home healthcare that refers to the provision of palliative therapies to terminal patients. The authors modeled the PHCP through a MILP formulation where assignment, scheduling and routing decisions are taken in an integrated fashion. Two alternative objective functions, *maxmin* (i.e., which balances the operator workload by maximizing the minimum utilization factor) and *minmax* (i.e., which balances the operator workload by minimizing the maximum utilization factor), were defined and used to guide the solution process. The MILP formulation was strengthen with the addition of symmetry breaking constraints and valid inequalities; moreover, possible model extensions were also discussed. To solve the PHCP, the authors implemented three alternative pattern generation policies (a greedy heuristic procedure, a realistic procedure based on current practice, and a flow-based model). The so-generated a priori patterns are given as an input to the MILP formulation that solves the original PHCP. Such a solution approach proved to be effective on different sets of realistic instances. From a practical perspective, it is worth highlighting that the selection of *maxmin* as the objective function of the MILP formulation produces more balanced solutions in terms of workload among operators. On the contrary, the selection of *minmax* as the objective function of the MILP formulation produces less costly solutions, as the total travel time for the operators is minimized.

A particularly interesting problem at the intersection between AHD and AHS is the Delivery Installation and Routing Problem (DIRP) investigated by [7]. The motivation for which we decided to review this work here is the similarities that the problem has with potential applications in the context of home care services. The DIRP is inspired by a real-world application encountered in the sector of large appliances and furniture, where the deliveries and the installations are performed by two heterogeneous fleet of deliverymen and installers, respectively. This particular application requires the synchronization of worker skills and is characterized by the presence of temporal precedence constraints (i.e., an installer must wait for a deliveryman to complete the delivery service before reaching the location of a customer and starting the installation service). In some cases, the installation may be directly performed by the deliveryman (with a lower efficiency as such figure is less specialized than an installer). The authors defined the DIRP using a flexible MILP formulation, from which specific variants of the VRP can be easily derived (i.e., in case all the installations are performed only by deliverymen we refer to the VRP with time windows and driver-specific times, while in case all the installations are performed only by installers we refer to the VRP with multiple synchronization constraints). In addition, a variant of the DIRP, where more than one worker is allowed to perform an installation, was discussed. To solve the problem, the authors implemented an ALNS metaheuristic algorithm and compared its performance with a branch-and-bound algorithm used to solve the MILP formulation. The results show that the ALNS algorithm is able to find good-quality solutions in short computing times both for test instances, as well as for real-world instances obtained from an industrial partner. Two noticeable insights emerged from the sensitivity analysis performed by the authors. The first is represented by the evidence that using two heterogeneous fleet of deliverymen (who may perform the installation service, if necessary) and installers has a positive impact in terms of total routing cost reduction. The latter demonstrates the existence of a correlation between the deliverymen's efficiency and the percentage of installations performed by the installers; indeed, the more the deliverymen are skilled, the less installation services are executed by the installers, and the less the installers must wait for the deliverymen to complete their deliveries before starting their services.

## 1.4 Conclusions and Future Research Directions

This work has provided a detailed literature review on the state of the art for Attended Home Delivery (AHD) and Attended Home Service (AHS) problems, a research field that is experiencing increasing attention, as confirmed by the fast growing number of documents published each year on this class of problems. Given its strong practical relevance, a particular focus has been put on real-world applications with the purpose of gaining useful managerial insights. Indeed, AHD and AHS problems owe their popularity to the rapid diffusion of online platforms, where a particularly high demand is registered for e-grocery and online retail transactions.

Since the seminal works in this topic, an increased awareness of the multi-stage nature of AHD and AHS problems, where the decisions taken at the first level (i.e., which typically requires to solve a demand management problem) greatly affect the feasibility as well as the profitability of the decisions taken at the second level (i.e., which typically requires to solve a routing problem), has emerged. Demand management and routing are well-established research fields per se, but the integration of demand management and routing decisions represents the complex part of solving real-world AHD and AHS problems, as these subdecisions are affected by uncertainty.

Many authors have proposed several sophisticated methods to solve alternately demand management problems (where the information related to the routing subproblem is estimated or forecast) or routing problems (where information related to the demand management subproblem is oversimplified and used as an input or, once again, forecast), but the search for a more effective integration of these two stages may represent one of the most significant future research directions in AHD and AHS.

In this sense, a promising approach may be that of using Dynamic Programming as the main framework, but great efforts are needed to overcome the issues of dimensionality and complexity of solving a Vehicle Routing Problem with Time Windows as the boundary condition for each state. An alternative approach may be that of borrowing some ideas from the Stochastic Dynamic Vehicle Routing Problem literature to roughly solve the online demand management problem by anticipating some routing aspects that must be fine-tuned offline.

The sustainability of AHD and AHS systems is another relevant topic having received little attention as compared to the wide literature on AHD and AHS problems. The recent work of [5] presents an interesting discussion on the effectiveness of using "green" incentives to steer customer choices, along with traditional price incentives. As sustainability may represent for AHD and AHS problems an additional objective, which may be conflicting with profit maximization or cost minimization, the benefit from introducing multi-criteria problem formulations is worth exploring. Also, further objectives may emerge and be considered in the future. For this reason, the introduction of Multi-Criteria Decision Analysis for solving AHD and AHS problems may represent another future research directions in this field.

Finally, we have seen that real-wold AHD and AHS applications may be encountered in heterogeneous business sectors, although the problem at its core maintain a similar structure (with some exceptions). In upcoming years we expect a denser transfer of ideas and technologies among different sectors as well as the emergence of innovative areas of application.

# Chapter 2

# A Decision Support System for Attended Home Services

Bruck, B.P., Castegini, F., Cordeau, J.F., Iori, M., Poncemi, T., Vezzali, D. (2020). A Decision Support System for Attended Home Services. *INFORMS Journal on Applied Analytics*, 50(2), pp. 137-152.

## Abstract

This paper describes a decision support system developed to solve a practical attended home services problem faced by Iren Group, an Italian multi-utility company operating in the distribution of electricity, gas and water. The company operates in several regions across Italy and aims to optimize the dispatching of technicians to customer locations where they perform installations, closures or maintenance activities within time slots chosen by the customers. The system uses historical data and helps operations managers in performing a number of strategic decisions: grouping municipalities into clusters; designing sets of model-weeks for each cluster; evaluating the obtained solutions by means of a dynamic rolling horizon simulator; and providing as output several key performance indicators as well as visual optimized technician routing plans in order to analyze different scenarios. The system uses mathematical models and heuristic algorithms that have been specifically developed to take into account different quality of service levels. Computational experiments carried out on data provided by the company confirm the efficiency of the proposed methods. These methods also constitute a powerful tool that can be used by the company not only to reduce costs, but also to help them in their strategic evaluation of existing and potential market opportunities.

## 2.1 Introduction

*Attended Home Services* (AHS) are service delivery systems in which a supplying company and a customer agree on a time window during which the customer will be home and the service will be performed. AHS systems are common in many fields, such as the distribution of perishable goods, the delivery of furniture or kitchen appliances and pharmaceutical products, and the provision of repair or technical services of different types [2], [6]. Many AHS companies schedule the deliveries by hand and spend considerable resources on schedule evaluation. Their process becomes even more complex because of the dynamics of the activities performed and the presence of demand variability [38]. Even when problems are solved with optimization tools, it is not obvious that the trade-off between *Quality of Service* (QoS) level and cost is perceived as optimal by the customers. Indeed, to fulfill customer intervention requests, the development of an efficient system, providing a high QoS level while balancing the service cost, is essential.

Typically, the optimization of AHS requires solving a two-stage problem, combining appointment scheduling and vehicle routing [84]. In the specific case of IRETI, the optimization of appointment scheduling consists in designing a set of *time tables*, defined by five working days and eight daily time slots of one hour, that are associated to a specific group of municipalities, called *cluster*. Logically, the configuration of clusters is a decision that has to precede the creation of time tables.

Essentially, a time table is a matrix of time slots in which customer intervention requests are booked by the consumption of a given capacity of allocated resources, corresponding to a certain amount of working hours of technicians available to perform the services. The initial configuration of a time table is called *model-week*, and may change on a seasonal basis according to the expected demand profile. The available time slots are gradually filled with services and, during the booking process, IRETI might dynamically change the model-weeks by moving or adding resources to meet a peak demand. Once the demand is known, the design of routing plans on the basis of customer locations and selected time slots is performed.

As imposed by the authority that regulates the market, IRETI has to respect minimum QoS levels, which may concern the maximum lead time, in terms of working days, between the customer intervention request and the execution of service, or the maximum delay from the assigned time slot, in terms of hours.

This paper describes a *Decision Support System* (DSS) designed and implemented to support IRETI in the set-up and refinement of operations in a specific territory. In particular, the aim of the DSS is to support IRETI in (i) determining optimal cluster configurations for a given territory, (ii) designing an efficient set of model-weeks, by determining the capacity allocated for each time slot, and (iii) simulating detailed routing plans for the technicians.

To solve this three-stage problem, we propose an integrated approach consisting of a series of optimization methods. In particular, the first stage is formulated as a simple *mixed integer linear programming* (MILP) model, based on the well-known *P-Median Facility Location Problem* (P-MFLP), the second stage builds upon the heuristic algorithm proposed by [33], while the third stage is based on a model that dynamically simulates customer intervention requests and their fulfillment by using a rolling-horizon simulation approach in the creation of routing plans.

The introduction of the DSS has dramatically decreased the effort required by IRETI to identify the optimal cluster configuration and create an efficient set of model-weeks for a given territory. Furthermore, by using the DSS, IRETI can evaluate alternative scenarios in terms of strategic *Key Performance Indicators* (KPI) over predetermined time periods (e.g., a week, a month, a year) and visualize the simulated routes of each technician on a real road network. Lastly, the DSS was initially conceived to optimize the operations of IRETI in a specific territory. However, it could be easily extended to other territories in which IRETI has no historical data or in other contexts, such as electricity and water distribution, thus representing a powerful strategic tool to set up the operations over any territory and context.

The computational results presented in [33], obtained without the clustering optimization, show that it is possible to save, on average, 10% in the routing costs considering the most realistic scenario. As expected, these results also showed that imposing QoS constraints has a negative impact on the solution cost, even though such constraints are very important to provide customers with a more substantial set of options and improve the overall satisfaction. This fact is particularly interesting for companies that have to compete with each other to get public contracts, where every feature that improves the QoS levels counts. In the present paper, we build upon this prior work and extend it to a more realistic setting which yields a powerful tool for strategic planning.

## 2.2 Context Description

IRETI is a division of Iren Group, an industrial holding operating in the Italian market of multi-utilities, which distributes electricity, gas and water in several Italian regions, such as Piemonte, Liguria and Emilia-Romagna, as shown in Figure 2.1. The development of the DSS was justified by a public tender issued by the Italian government to renew the concession of gas distribution in an area currently served by IRETI. To clarify the procedure of public tenders in the market of multi-utilities, a brief overview is provided.

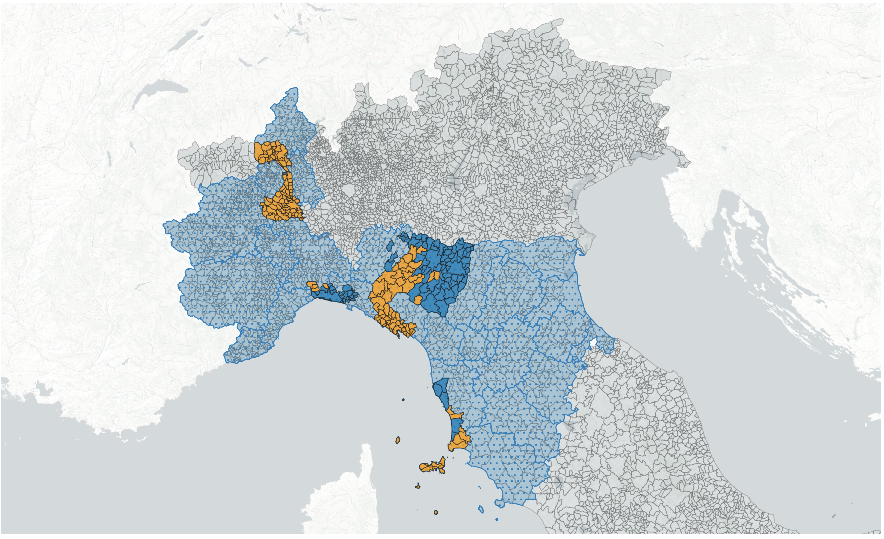


Figure 2.1: Regions served and municipalities in which IRETI operates as a gas supplier

### 2.2.1 European Market of Multi-Utilities

The European Union is the institution responsible for the definition of directives to which every member state should conform, concerning the regulation of the European gas market.

Every member state accepts the directives and adopts national laws in order to implement them. Local authorities employ national laws and adopt specific deliberations to guarantee the observance of the regulations. The Italian market of gas distribution is regulated by the *Authority for Regulation of Energy Networks and Environment* (ARERA). Through the "Accounting Unbundling Obligation" [11], called TUIC, the authority defines two main actors: the independent network manager, also denoted as the *distribution company*, and the gas seller, also denoted as the *trading company*. The former operates in the territory assigned through a specific public tender and is responsible for managing and maintaining the gas distribution network. The latter stipulates commercial contracts and is responsible for managing the relationship with customers, while competing with other trading companies in a free market. To sum up, several trading companies compete against each other to sell contracts to customers, while a single distribution company, selected by means of a public tender, is responsible for operating the gas distribution network.

Until 2012, customer appointments for AHS were defined by mutual agreements (e.g., phone calls) between the distribution company and the customers [121]. Nowadays, as a consequence of changes in the institutional and regulatory framework [64], the distribution company must provide an *on-line agenda* which consists of a portal containing the model-weeks for all clusters in a served territory. Trading companies book the appointments for their customers in the available time slots directly on the portal, and the distribution company (e.g., IRETI) receives the requests through the on-line agenda.

### 2.2.2 Public Tenders

The public tenders published by ARERA concern the regulation of gas, water and electricity distribution. The principle behind these tenders is to encourage improvements in the QoS levels, by positively evaluating those proposals in which companies commit to reach advances in one or more of these levels. Usually, the QoS is evaluated on the basis of *service time*, defined as the number of working days between the customer intervention request and the execution of service, and *punctuality*, defined as the maximum time at which the technician can arrive at customer's location and start executing the service, once a service is booked in a time slot.

Specific QoS levels are defined in the "Regimentation of Quality for Distribution and Measurement of Gas" [13], called RQDG and currently valid for the period from 2014 to 2019. The AHS regulated by the RQDG include:

- Installing a new meter;
- Re-opening a meter after closure due to a situation of potential risk;
- Re-opening a meter after closure due to being in arrears;
- Closing a meter at the request of a customer;
- Checking a meter at the request of a customer;
- Making available technical data from a meter.

A public tender refers to a so-called *minimum territorial area* (ATEM), a definition introduced by ARERA to represent a cluster of municipalities supplied by the same distribution company. Distribution companies applying to the public call for an ATEM submit their proposals, which are evaluated both in terms of financial evaluation and expected QoS levels compliance. Regarding the latter, the authority usually identifies one or more criteria for which improvements in relation to the current QoS levels are requested. For example, the authority might ask for a 50% reduction in the service time required for installing a new meter, from 10 to 5 days, or in the service time for re-opening a meter after closure due to being in arrears, from 2 days to just 1 day. If an applicant distribution company declares to fulfill the tightened QoS levels, a higher technical evaluation is obtained but the achievement of these performances should be guaranteed in order to avoid penalties in case of delays in the execution of services. In this sense, the trade-off between cost and QoS level is crucial, and hence the optimization of operations is a lever to maintain low costs while increasing QoS levels.

The distribution company reaching the highest overall score undertakes the contract in the ATEM for a certain interval of years. This type of tender is very common in public procurement [137] and has a large number of real-world applications, not only regarding gas distribution, but also in subcontracting other commodities and services.

## 2.3 Company Description

Iren Group is a large Italian corporate group established in 2010 through the merging of Enìa and Iride. Iren operates in the market of multi-utilities with approximately 6 200 employees, and achieved a revenue of 3.8 billion euros in 2018. The company is listed in the FTSE Italia Mid Cap index of the Milan Stock Exchange and is one of the leaders in its sector. The group consists of an industrial holding company, Iren S.p.A., and four fully controlled business units operating in their specific sectors either directly or through controlled companies in which they hold a share:

- *Iren Energia* operates in electricity and heat supply, managing some district heating networks and providing technological services;
- *Iren Mercato* is a trading company that stipulates commercial contracts with customers for the trade of commodities such as electricity, gas, water and district heating;
- IRETI is a distribution company specialized in gas, electricity and water distribution networks management;
- *Iren Ambiente* operates in the field of waste collection, treatment and disposal, and in the design and management of renewable energy systems.

Based on the aforementioned "Accounting Unbundling Obligation", IRETI is an independent network manager operating as a distribution company, while Iren Mercato is a trading company. The regulation settled by ARERA [11] imposes administrative and accounting separation for companies operating in gas or electricity distribution markets. In other words, even if distribution and trading companies can be part of the same group, they must be completely independent, both from the operational and the accounting point of view. To encourage fair competition, efficiency and high QoS levels, they cannot share commercial or sensible information.

Iren Group is the first player in Italy in terms of volume heated in district heating systems, the third in terms of volume of water supplied and collected waste, the fifth in terms of gas and electricity supplied to final customers.

This study was developed as part of a collaboration between the *Operations Research Group* of the University of Modena and Reggio Emilia and the *External and Metering Operations* unit of IRETI, both based in Reggio Emilia. With almost 8 000 km of distribution network, IRETI provides gas supply to about 750 000 customers located in 95 municipalities in Emilia-Romagna, Piemonte and Liguria, for a total volume of 1.2 billion cubic meters of gas per year. Such distribution consists in transporting gas from the pipelines of *Snam Rete GAS* (the company that manages the Italian national gas transportation system) to the local distribution networks. This includes a number of external operations such as filtering, preheating and pressure regulation in order to provide safe and timely services to final users.

## 2.4 Brief Literature Review

AHS are commonly designed as the combination of two problems: (i) booking process and (ii) service execution [84]. During the booking process, the customers book a service (either directly or by means of their trading company) in one of the available time slots. Then, the distribution company has to perform the services by sending technicians to customers' locations. In most AHS problems, the generation of the routing plans for the technicians can thus be represented as a variant of the *Vehicle Routing Problem With Time Window* (VRPTW).

Among the works that focus on the booking process, we cite [143], who study different success factors in attended home delivery of grocery, and [38], [39], who propose techniques to determine when to accept or reject requests and to influence customer behavior towards low-demand time slots. For the literature that concerns service execution, and thus the routing component of the problem, we refer the reader to the detailed reviews by [2] and [58] on the VRP for AHS, [22] on the VRPTW, and [168] on VRP variants in general. For the specific VRP faced in the routing phase of our DSS, we refer instead to [33], who proposed a heuristic optimization method based on a *Large Neighborhood Search* (LNS).

A number of studies (see, e.g., [118]) analyze the correlation between decreasing time window size and increasing delivery costs. A loss in profit when offering shorter time windows is observed also by [38], who highlight that a significant cost reduction can be reached with longer time windows but obviously at the expense of the QoS level provided. In contrast, the problem considered in this paper combines the necessity of increasing the QoS level and at the same time considering the possibility of decreasing the time window size (as this could be imposed by the authority). Because of these specific characteristics, a cost increase might be unavoidable.

It must be noted that many models in the literature are intended to support the decision makers during the booking period, while the final vehicle routes are planned, for instance, by means of commercial routing software. In this sense, a valid contribution of our study is to provide an integrated tool capable of managing both these aspects, analyzing and simulating different scenarios with different assets in order to guarantee the maximum QoS level and minimizing resource deployment.

A related interesting problem that combines booking and routing is the *Time Window Assignment Vehicle Routing Problem* (TWAVRP), originally proposed by [161] and later generalized by [160]. In the TWAVRP, time windows have to be assigned to a fixed set of customers before the demand is known. Once the demand is revealed, a routing plan is constructed with the objective of minimizing the travelling costs. Our problem is, however, different from the TWAVRP because the time windows are not assigned by the company but chosen by the customers.

We are not the first to propose an integrated approach for an AHS problem, as relevant methods have been presented in, e.g., [121] and [3], among others. Other integrated approaches have been proposed for other multi-utility activities, as the recent DSS developed by [66] for urban waste collection.

However, our research may provide a number of innovative and interesting contributions:

- In most of the literature on AHS problems, costs are related to routing [164]. IRETI, instead, puts a high emphasis on costs related to failures in reaching the required QoS, determined by compensations in favor of customers in the case precision range or service time constraints are not satisfied. This does not mean that IRETI ignores routing costs, but that they put more efforts in the time table creation, to avoid exceeding service time and thus compensations imposed by the authority. In view of this, the time table creation process takes a very strategic role for the company;

- Most of the literature has focused on heuristic algorithms (see, e.g., [86]) in an attempt to automate tasks normally performed by decision makers. Our DSS is equipped with a combination of simulation and MILP models, and has been developed with the intent of finding an optimal configuration of assets for a given set of inputs, not just to replace operations. In this respect, the simulation aspect is particularly important, especially for making and evaluating predictions on new territories;

- Our simulation is based on real geographical and historical data, providing the decision makers with meaningful information. This is a relevant component of the first stage, that is, the division of an ATEM into clusters;

- Due to ARERA regulations, we cannot influence customer choices, as suggested, e.g., in [39] and [189], nor reject them as all customers must be served under the same conditions. Thus, demand uncertainty, a typical component of many real-life AHS applications [41], must be fully considered in the DSS.

To the best of our knowledge, no such integrated strategic tool with the same conditions exists in the AHS literature.

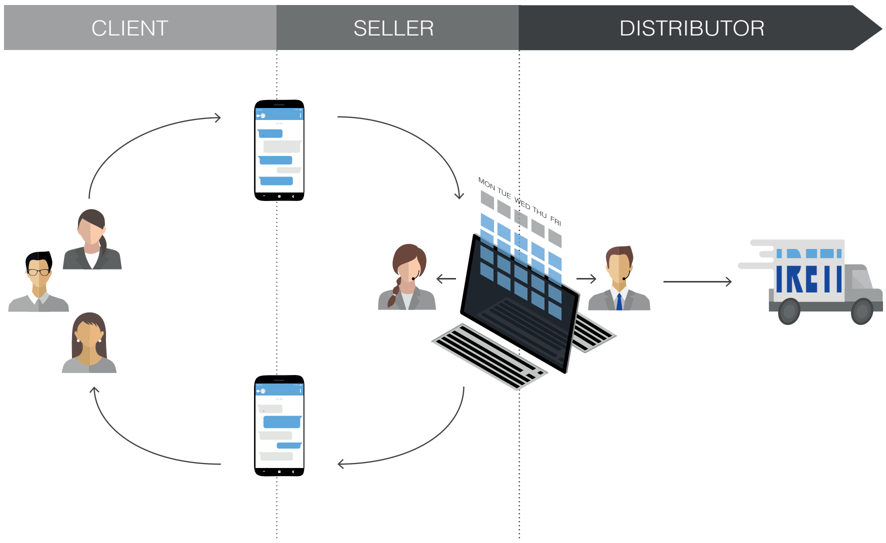



Figure 2.2: Flowchart representing the booking process for a service request

## 2.5 Problem Description

The AHS problem managed by IRETI is a three-stage problem, consisting of: (i) a clustering problem that aims at grouping the municipalities of a certain ATEM into clusters; (ii) a model-week design problem concerning the definition of a time table for each cluster; and (iii) a routing problem that aims at creating optimal routing plans for technicians appointed to perform the services. An in-depth description of the three stages is given in the following sections.

Solving this specific AHS problem means considering several interconnected decisions that increase the complexity of the problem. Therefore, designing an integrated approach capable of proposing cost-effective and balanced clusters, creating good-quality and efficient time tables, and simulating technicians routing plans is a complex task.

In order to better understand the whole process, we start by describing the booking process (see Figure 2.2). First of all, when a customer needs to book a service, she must contact her trading company and agree on a date and time slot for the execution. Then, the service request is forwarded to the independent network manager (e.g., IRETI), through the on-line agenda. At this point, the independent network manager might contact the customer to anticipate the appointment. In case the anticipation is rejected, the date and time slot of the appointment are confirmed. Eventually, the customer might ask to postpone or cancel the appointment. In this case, she must contact her trading company to reschedule the appointment. If the request is made more than 24 hours before the previously arranged appointment, the entire booking process is repeated by searching for a new date and time slot. Only in the case where the request is made less than 24 hours before the previously arranged appointment, can the independent network manager reject it. Once an appointment is confirmed, it is assigned to a technician and, if the independent network manager does not respect the required QoS level for the requested service, a compensation in favor of the customer is due. The only situation in which IRETI is relieved of this compensation is when the customer is absent on the execution date.

### 2.5.1 Stage 1: Clustering

The clustering is the first stage of the AHS problem managed by IRETI and precedes the creation of the time tables. This stage is solved at a strategic level because the clustering of municipalities is seldom redefined. Until now, this stage has been performed manually by IRETI, based on experience and common sense. For instance, due to the weekly scheduling of outdoor markets or to changes in the road network, decision makers might decide not to group particular municipalities in the same cluster. The proposed method adopts and incorporates these best practices.

### 2.5.2 Stage 2: Model-week Design

The model-week design is the second stage of the problem and is another strategic activity that is performed only a few times over the year, usually on a seasonal basis depending on the demand profile of service requests in a specific ATEM. The definition of a model-week implies deciding which days and time slots to open for each cluster and how many resources to allocate for each time slot, given the availability of technicians. The objective of this stage is to minimize the unbalanced distribution of resources in time slots, between morning and afternoon and among the different days of the week, so as to provide a high QoS level to customers. IRETI is also responsible for managing the booking process, performing continuous adjustments to the time tables, by adding or moving resources in order to fulfill peak demand.

At the beginning of each week, the time table is reset to the original model-week for each cluster, except for the resources already booked from the previous weeks. Note that IRETI assumes that a resource corresponds to 30 working minutes of a technician. Given the fact that the length of a time slot is one hour, then two resources can be allocated per technician available. Thus, time table adjustments refer to the additions or movements of technicians among clusters and time slots. Many types of services require only 30 minutes (i.e., one resource), while others require 60 minutes (i.e., two resources). Therefore, a technician could execute a single 60-minute service or two 30-minute services in a time slot. An explanation of how these aspects are managed during the simulation is given in the following sections.

From a practical point of view, the addition or movement of technicians from one cluster to another is performed to fulfill the required QoS level. Currently, the adjustment of time tables is performed manually, but handling a scheduling process in this way is a complex task (see, e.g., [111]).

### 2.5.3 Stage 3: Simulation of Detailed Routing Plans

The third stage of the problem concerns the building of routing plan for each technician, which must be done on a daily basis. In the proposed method, the routing is also used to evaluate and compare the cost of a given set of model-weeks. Technicians are routed from the depots to customers' locations and, if necessary, additional technicians might be employed from subcontractors which provide outsourcing services. Ideally, technicians are assigned to a determined depot and perform most of the services in the clusters served by that depot. In practice, technicians might be required to perform services outside their area of competence. This might happen in case the allocated capacity is not sufficient to completely fulfill the demand.

## 2.6 Solution Method

We developed a suite composed by three modules, each solving one of the stages defined in the previous section.

The first module aims at supporting IRETI in the definition of the geographical clusters (Stage 1). In this module, the evaluation of travel distances and times between municipalities is essential to determine the best cluster configuration. To that aim, we use the *Open Source Routing Machine* (OSRM), a routing engine from OpenStreetMap, which is able to efficiently evaluate the shortest path distance on the real road network between a pair of geographical coordinates received in input. These distances are then used for the generation of clusters, which are obtained by implementing a simple MILP model based on the P-MFLP. Basically, the clustering stage is characterized by several real constraints that could increase the complexity of the model. Nevertheless, we decided to implement a simple and flexible model, whose constraints could eventually embed some of the recommendations drawn from IRETI. The detailed P-MFLP formulation that we developed is reported in the Appendix.

The second module is the so-called *static solver*, which aims to create a model-week for each cluster, thus solving Stage 2. Note that, to estimate the cost of a model week, the second module employs non-trivial algorithms, including a one-week simulation of the booking of time slots and the consequent construction of routing plans for the technicians. We implemented this module building upon the heuristic approach by [33]. In particular, their formulation has been replaced by what we call *Model-Week Generator* (MWGen), that is a MILP model that introduces additional constraints that were not addressed in the original paper. A distinction between clusters containing large and small cities has been introduced, and, according to what the company suggested, different QoS constraints have been adopted. Indeed, as large cities (e.g., the regional county seats) give rise to strong imbalances in the demands, we have decided to adopt specific constraints to better mitigate uneven distribution of resources among time slots. Furthermore, the resource constraints are now expressed in terms of number of technicians instead of number of resources. This not only reduces the problem size, but also removes some symmetry from the formulation.

Based on the results of the computational experiments, the MWGen formulation has proved to be an effective tool to generate initial feasible model-weeks. However, it uses a simplified objective function with respect to the real problem, so a further effort is required to understand what would be the resulting operational costs derived from the use of such model-weeks. Therefore, the set of model-weeks produced by the MWGen is given as an input to an LNS algorithm, which evaluates them and possibly modifies them by means of destroy and repair methods. Once a set of model-weeks has been generated, the actual demand registered in a one-week time horizon is revealed, the booking of time slots is simulated by assigning the customer intervention requests to the time slots, and then the routing plan for each technician is created. As in [33], the detailed technician routing plan creation is obtained by solving a variant of the *Multi-depot multiple Traveling Salesman Problem* (MmTSP), in which the total traveled distance is minimized. The LNS continues for a certain number of iterations. At each iteration, the current set of model-weeks is randomly modified and then rebuilt by invoking once more model MWGen. To avoid cycling, we introduced a family of *no-good cuts* into the model. These impose that the configuration of open and closed time slots at a given iteration is different with respect to the configurations obtained in the previous iterations. All these changes helped improve the original model and fulfill the requirements of the new tenders. Model MWGen is described in detail in the Appendix (objective function (9) and constraints (10)–(31) are used to generate the initial solution, whereas constraints (32)-(35) are added to avoid LNS cycling).

The static solver returns a feasible set of model-weeks, that is given to a third module, the *dynamic solver*, which dynamically simulates how customer intervention requests would be satisfied in practice for a long time horizon (typically a few months), thus solving Stage 3. Similarly to the second module, the creation of routing plans for each technician and each day is modeled as a variant of the MmTSP. Nevertheless, the swapping of resources through different model-weeks, in case of capacity excess, and the use of additional technicians, in case of demand peaks, are introduced. Again, as proposed by [33], a *time-extended network* is considered in order to ensure the respect of time window constraints and, consequently, define the specific schedule of each technician. Based on historical data, we have identified three demand scenarios: low, medium and high. By analyzing the output of this module, the specific KPIs logged during the simulation and the adjustments required in certain time horizons to fulfill the QoS levels imposed by the authority are evaluated. Furthermore, the tool also allows us to visualize and inspect the routes of each technician, providing information on average speed and route duration, among other KPIs.

## 2.7 Implementation of the DSS

We embedded the proposed methods into a DSS that consists of three macro modules: data-processing, optimization and simulation (Figure 2.3). The data-processing module represents the interface between the DSS and IRETI historical data. The optimization module is responsible for creating instances, generating cluster configurations by running the P-MFLP model, and designing sets of model-weeks by means of the static solver. The simulation module then simulates under different scenarios the proposed solution, given a set of previously designed model-weeks. The results of these simulations can be visualized in a web app, allowing decision makers to verify strategic KPIs. In the following, we analyze in detail each module.

### 2.7.1 Data Processing

The data processing module consists of a set of methods, tools and scripts that are used to preprocess and cleanse all of the data extracted from the IRETI database, which contains the list of intervention requests, performed services and georeferences over an extended interval of years. Additional information might be integrated from the open data published by the *National Institute of Statistics* (ISTAT) and the *Ministry of Economic Development* (MISE). In particular, demographics and geographical data on administrative borders are extracted from the previous source, while technical data

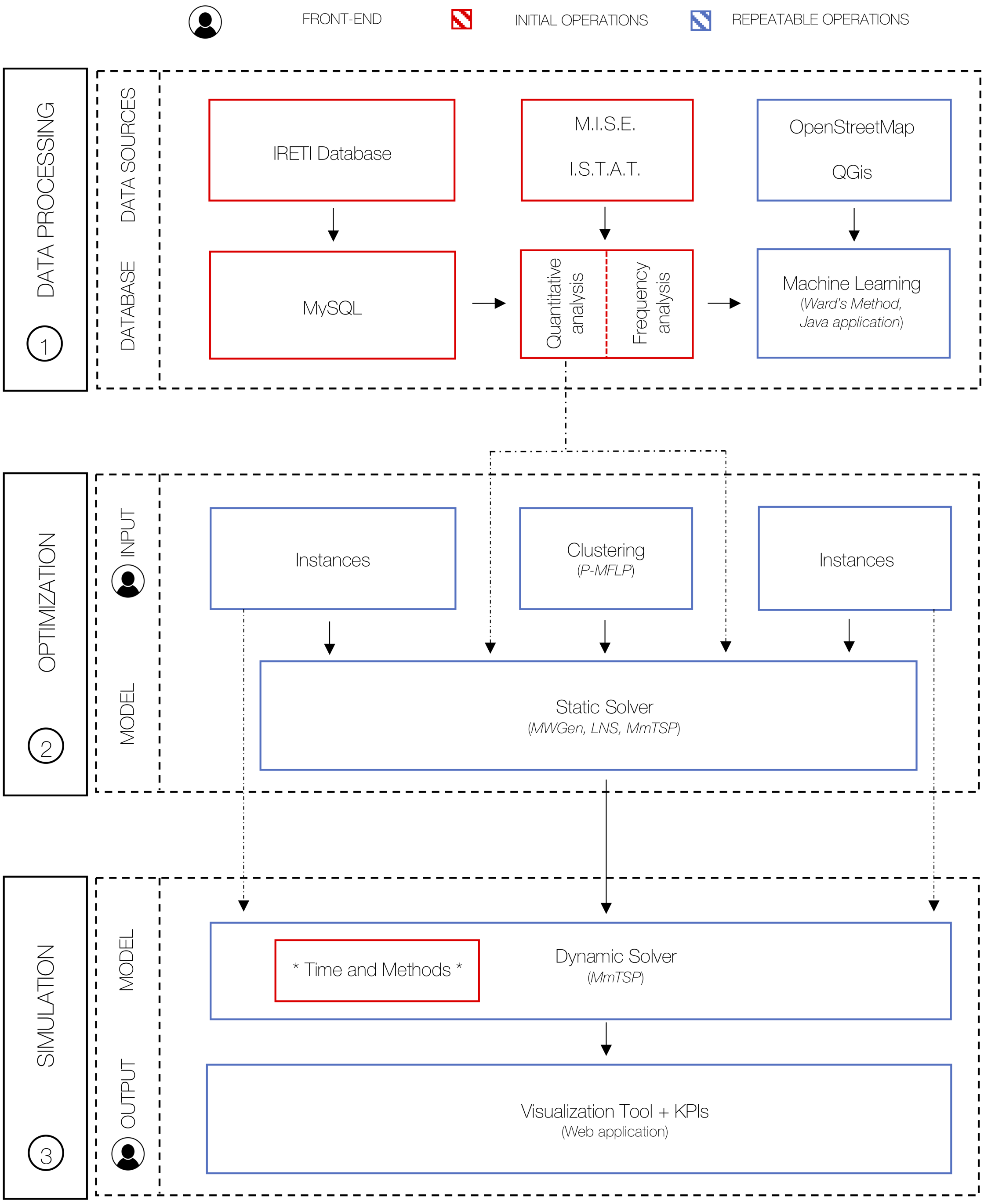



Figure 2.3: The diagram shows the three main components of the DSS and their inner parts

(e.g, users per municipality, volume of distributed gas, network distribution length and category of supply) are obtained from the latter.

In case of a public call referred to a particular ATEM, that is completely or partially not served by IRETI, and for which historical data are not available or incomplete, we implemented simple machine learning methods that rebuild or predict missing information by processing determined features and historical available data, both from other served ATEMs and from eventually already served municipalities in the ATEM aim of the public tender. The idea of this module is to create dummy intervention requests for all of the unknown municipalities, based on similarities with already served and known territories.

In the following, a brief description of the implementation of the aforementioned machine learning methods is provided.

Initially, the additional features obtained from the open data published by ISTAT and MISE are given as an input to a hierarchical clustering algorithm, known as *Ward's Method* (WM) [128], and integrated in the software RStudio. That constitutes the first part of the machine learning algorithm, for which the output is a dendogram successively converted into a similarity matrix where the element $a_{ij}$ is a parameter that takes the value 1 if municipality $i$ is similar to municipality $j$, 0 otherwise. After that conversion, the similarity matrix is given as an input to a Java application, along with a series of additional inputs such as:

- the calendar year in which the rebuilding is performed;
- the frequency analysis of historical customer intervention requests;
- the set of real building coordinates in the ATEM aim of the public tender, previously extracted using a query of the QGis open-source software on an OpenStreetMap layer.

The Java application output is a set of dummy intervention requests that, together with historical customer intervention requests, are added to a local database used to create sets of instances that are given as an input to the following modules.

### 2.7.2 Optimization

The optimization module contains the tool responsible for generating optimized cluster configurations and the static solver. Both tools rely on the database for input data. However, they do not have direct access to it. Instead, we generate instance files that comprise data from specific periods of time that we consider in the simulations. These files are created by Python scripts that search through the database and select only the requested data. The database can be populated either with real data or with the machine learning algorithm.

The static solver requires additional input information, such as the cluster configuration, which can be either given by the company or generated using our clustering algorithm. Note that, if we choose to use the latter approach, it is important to consult the company experts to ensure that the proposed cluster configuration is indeed viable. In addition, it is necessary to input the demand scenario for the static solver, which basically defines the amount of demand that will be covered during the generation of the model-week tables. At this point, the decision makers are ready to run the static solver through a command line and select some customization elements in order to reach different QoS levels.

It is important to mention that the solver might not always be able to find a feasible solution. This might happen due to several reasons. For instance, the given number of technicians for each depot may be insufficient, or the number of available time slots may be too low to reach the specified QoS levels. In these cases, the decision maker can analyze input and output data and correct any inconsistency or underestimation of the workload to overcome the problem, and then run the solver again.

### 2.7.3 Simulation

The simulation module refers to the dynamic solver and to the simulations that can be performed to fine tune the solutions found by the optimization module. Instead of considering a single week of data, the instance files for the dynamic solver may contain information from an arbitrary period of time. In our experiments, we usually run simulations for an entire year to test the efficiency of a given solution in the seasons of low, medium and high demand.

In the dynamic solver, decision makers can customize specific parameters of the simulation such as the maximum execution time of each type of service, and decide whether or not to allow the swap of resources between time tables, in order to fulfill requests in peak demand scenarios. In particular, the former parameter is crucial to ensure that the company is able to simulate possible scenarios that might be presented in future tender roles. Furthermore, to perform more realistic simulations, the dynamic solver considers that the execution times of services from the same type may vary by a certain degree, which can be specified by the decision maker. Furthermore, during the simulation, a delay in the schedule of a technicians is tolerable and logged as a KPI. Consequently, an evaluation of delays is given as an output to the decision maker who can assess whether the total amount of delay is acceptable or not in the simulated time horizon.

## 2.8 Usage and Benefits

The DSS is a strategic tool for IRETI that substantially decreases the decision making process, providing more efficient solutions and exploring different demand scenarios that would be hard and time consuming to compute manually. The system also allows the team to consider different QoS levels in the simulations, which is important when trying to establish a reasonable trade-off between the QoS offered to clients and the costs incurred by the company.

Prior to employing our DSS, most of the decisions made by IRETI relied on analysis performed manually, with the aid of several spreadsheets. However, due to the sheer amount of data involved, the number of constraints and possible scenarios, finding a good solution was a difficult task. Moreover, due to the complexity of the decisions that had to be made and considering that the process was mostly manual, the team was not able to reliably plan for a longer time horizon, having to limit their forecast to only a few weeks ahead. Besides being able to automate most of the process of generating efficient solutions, the DSS also provides easy access to the results of the simulations (i.e., graphs and tables) so that alternatives can be evaluated and more informed decisions that do not rely solely on experience ca be taken.

Currently, IRETI is using the DSS in preparation for tenders that would give the opportunity to the company to operate on new territories or confirm its position in already served ones. The simulations performed using the DSS enable the decision makers to experiment and analyze how they can provide a high QoS level to customers while minimizing the cost. The DSS might also represent a tool to analyze past decisions and review them either to improve the current KPIs or to prepare for critical issues that might happen, being able to react quickly and accurately. For instance, in a currently served ATEM, the evaluation of alternative cluster configurations would be performed, aiming at reducing costs while maintaining or increasing the QoS level.

Another benefit of the system is that it is very flexible and can be adapted to changes in the regulations imposed by the market or to new strategies that the company might be interested to adopt. In a sector that is continuously changing, this sort of flexibility is crucial to the success of such an application. For example, in anticipation of possible changes in the regulations of water distribution, the company could already start up a tailored analysis by simply modifying the input data and performing some small adjustments to the constraints of the models.

### 2.8.1 A Realistic Instance

Due to an agreement to confidentiality we cannot present the results of a specific scenario that was optimized by the company. However, to illustrate and show the flexibility of the DSS, in the following we present the results that were found by using the system to analyze and find a solution for a possible tender in the ATEM of Verona, which might be of interest to the company. The data for the instance were generated using the machine learning approach described in the Data Processing Section. In order to assess the efficiency of the proposed approach we performed a range of experiments using both the static and dynamic solvers to simulate different demand scenarios and varying the execution times required by each type of service. In all cases, the DSS obtained efficient solutions even when considering tightened QoS constraints than what is required in real tenders.

The cluster configuration created for this ATEM is depicted in Figure 2.4, where the depot is represented by a black triangle. The most efficient set of model-weeks is reported in Table 2.1. Note that, in this solution, Cluster 1 corresponds to the county seat of Verona and due to its predicted high demand, all time slots are opened, offering a high QoS level with an average number of technicians per time slot equal to five, whereas all other clusters have an average of one or two technicians. Two other interesting features with respect to QoS are that all clusters have time slots opened in at least two days of the week and there is a balance between the distribution of time slots in the morning and afternoon. In Table 2.2, we show the sum of all resources allocated to each time slot. This table is useful to analyze the distribution of resources per day and per time slot. Note that a total of 664 resources are provided, corresponding to 332 working hours per week, with well balanced values per day (ranging from 110 to 154 resources) and per hour of the day (between 72 and 92).

In practice, the model-weeks are gradually populated as service requests are made

| Cluster 1 | | | | | |
|---|---|---|---|---|---|
| | Mon | Tue | Wed | Thu | Fri |
| 08:30-09:30 | 10 | 10 | 12 | 12 | 12 |
| 09:30-10:30 | 10 | 10 | 12 | 12 | 12 |
| 10:30-11:30 | 10 | 10 | 12 | 12 | 12 |
| 11:30-12:30 | 10 | 10 | 12 | 12 | 12 |
| 12:30-13:30 | - | - | - | - | - |
| 13:30-14:30 | 10 | 10 | 8 | 8 | 8 |
| 14:30-15:30 | 10 | 10 | 8 | 8 | 8 |
| 15:30-16:30 | 10 | 10 | 8 | 8 | 8 |
| 16:30-17:30 | 10 | 10 | 8 | 8 | 8 |

| Cluster 2 | | | | | |
|---|---|---|---|---|---|
| | Mon | Tue | Wed | Thu | Fri |
| 08:30-09:30 | 0 | 2 | 0 | 0 | 0 |
| 09:30-10:30 | 0 | 2 | 0 | 0 | 0 |
| 10:30-11:30 | 0 | 2 | 0 | 0 | 0 |
| 11:30-12:30 | 0 | 2 | 0 | 0 | 0 |
| 12:30-13:30 | - | - | - | - | - |
| 13:30-14:30 | 0 | 0 | 0 | 0 | 2 |
| 14:30-15:30 | 0 | 0 | 0 | 0 | 2 |
| 15:30-16:30 | 0 | 0 | 0 | 0 | 2 |
| 16:30-17:30 | 0 | 0 | 0 | 0 | 2 |

| Cluster 3 | | | | | |
|---|---|---|---|---|---|
| | Mon | Tue | Wed | Thu | Fri |
| 08:30-09:30 | 0 | 2 | 2 | 0 | 2 |
| 09:30-10:30 | 0 | 2 | 2 | 0 | 2 |
| 10:30-11:30 | 0 | 2 | 2 | 0 | 2 |
| 11:30-12:30 | 0 | 2 | 2 | 0 | 0 |
| 12:30-13:30 | - | - | - | - | - |
| 13:30-14:30 | 0 | 2 | 2 | 0 | 0 |
| 14:30-15:30 | 0 | 2 | 2 | 0 | 2 |
| 15:30-16:30 | 0 | 2 | 2 | 0 | 2 |
| 16:30-17:30 | 0 | 2 | 2 | 0 | 2 |

| Cluster 4 | | | | | |
|---|---|---|---|---|---|
| | Mon | Tue | Wed | Thu | Fri |
| 08:30-09:30 | 2 | 2 | 2 | 2 | 0 |
| 09:30-10:30 | 2 | 2 | 2 | 2 | 0 |
| 10:30-11:30 | 2 | 2 | 2 | 2 | 0 |
| 11:30-12:30 | 2 | 0 | 2 | 2 | 0 |
| 12:30-13:30 | - | - | - | - | - |
| 13:30-14:30 | 0 | 2 | 2 | 2 | 0 |
| 14:30-15:30 | 2 | 2 | 2 | 2 | 0 |
| 15:30-16:30 | 2 | 2 | 2 | 2 | 0 |
| 16:30-17:30 | 2 | 2 | 2 | 2 | 0 |

| Cluster 5 | | | | | |
|---|---|---|---|---|---|
| | Mon | Tue | Wed | Thu | Fri |
| 08:30-09:30 | 2 | 2 | 4 | 2 | 2 |
| 09:30-10:30 | 2 | 2 | 4 | 2 | 2 |
| 10:30-11:30 | 2 | 2 | 2 | 2 | 4 |
| 11:30-12:30 | 2 | 2 | 2 | 2 | 4 |
| 12:30-13:30 | - | - | - | - | - |
| 13:30-14:30 | 2 | 2 | 2 | 2 | 2 |
| 14:30-15:30 | 2 | 2 | 2 | 2 | 2 |
| 15:30-16:30 | 2 | 2 | 2 | 2 | 2 |
| 16:30-17:30 | 2 | 2 | 2 | 2 | 2 |

| Cluster 6 | | | | | |
|---|---|---|---|---|---|
| | Mon | Tue | Wed | Thu | Fri |
| 08:30-09:30 | 0 | 2 | 0 | 4 | 2 |
| 09:30-10:30 | 0 | 2 | 0 | 4 | 2 |
| 10:30-11:30 | 0 | 4 | 0 | 2 | 2 |
| 11:30-12:30 | 0 | 4 | 0 | 2 | 2 |
| 12:30-13:30 | - | - | - | - | - |
| 13:30-14:30 | 0 | 2 | 0 | 2 | 2 |
| 14:30-15:30 | 0 | 2 | 0 | 2 | 2 |
| 15:30-16:30 | 0 | 2 | 0 | 2 | 2 |
| 16:30-17:30 | 0 | 2 | 0 | 2 | 2 |

Table 2.1: Model-week tables for ATEM of Verona

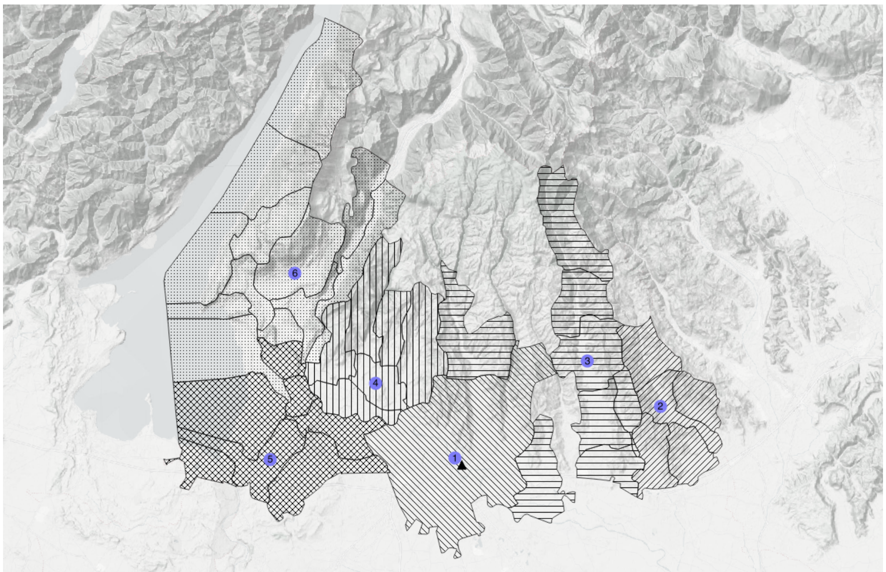


Figure 2.4: Clusters configuration for ATEM of Verona

| **All clusters** | | | | | | |
|---|---|---|---|---|---|---|
| | Mon | Tue | Wed | Thu | Fri | TOTAL |
| 08:30-09:30 | 14 | 20 | 20 | 20 | 18 | 92 |
| 09:30-10:30 | 14 | 20 | 20 | 20 | 18 | 92 |
| 10:30-11:30 | 14 | 22 | 18 | 18 | 20 | 92 |
| 11:30-12:30 | 14 | 20 | 18 | 18 | 18 | 88 |
| 12:30-13:30 | - | - | - | - | - | - |
| 13:30-14:30 | 12 | 18 | 14 | 14 | 14 | 72 |
| 14:30-15:30 | 14 | 18 | 14 | 14 | 16 | 76 |
| 15:30-16:30 | 14 | 18 | 14 | 14 | 16 | 76 |
| 16:30-17:30 | 14 | 18 | 14 | 14 | 16 | 76 |
| TOTAL | 110 | 154 | 132 | 132 | 136 | 664 |

Table 2.2: All clusters table: sum of all model-week tables

by customers through the trading companies. Especially during high demand profile periods, the time tables might be altered to face demand peaks that would otherwise not be completely fulfilled with the resources that are normally allocated. In these scenarios, the decision makers usually have two options: (i) diverting technicians from one cluster to the other, or (ii) employing third party technicians to compensate for the demand peak. Note that, although the former alternative does not lead to additional costs for the company, it might not always be feasible. Moreover, given that both alternatives lead to changes in the model-weeks, there is always the risk that poor decisions are made, leading to a higher increase in costs. Therefore, it is very important to simulate different scenarios in terms of demand profile when designing the configuration of model-weeks, not only to have a good initial solution but also to understand system behaviour and to estimate which dynamic changes are expected to be performed. This is the purpose of the dynamic solver that is integrated in the DSS.

In the particular case of the ATEM of Verona, we present the results of a two-month (eight-week) simulation based on a realistic instance, related to July and August. This is a typical medium demand profile scenario and each intervention request is revealed to the dynamic solver in the day in which it was supposedly requested by the customer. The appointment is booked in the first available time slot of the cluster online agenda to which the customer belongs. In case no time slot allows to satisfy the imposed QoS levels, the dynamic solver simulates the response from IRETI, providing another date, time slot or both. The dynamic solver gives an evaluation of the performed simulation reporting different strategic KPIs such as: (i) delays within 1 hour and 2 hours, (ii) driving times and (iii) distance evaluations. An example of output provided by the DSS and containing the aforementioned KPIs is presented in Figure 2.5. On the basis of the obtained KPIs, a decision-maker could conclude that no additional technicians are required during the simulation. This means that the number of technicians provided as an input is consistent. By contrast, comparing this scenario with an identical one except for the number of available technicians, reduced to 7 instead of 10, would return that three technicians should be added during weeks 2 and 6. Furthermore, in the current solution, week 6 could be characterized from a significant demand variation, due to the high number of technician swaps. In addition the percentage of delays is always under 3%, highlighting that they are minimal with the reduction of the precision range. Note that a variable number of appointments (between 210 and 420), corresponds to a variable number of resources (between about 295 and 635).

Contextually, the system provides detailed information on the performance of technicians, analyzing their daily routes and giving evidence to the compliance with the QoS levels. The minimum routes on which the operators can move and the relative travel times are computed. The covered distance is about 4100 km per week on average and it is consistent with the number of appointments, resulting in 12 km per appointment, with a minimum of 10.8 and a maximum of 12.1. Considering the dimensions of the ATEM, a KPI of about 12 km per appointment is plausible. Even the travelled time is compatible with the activities performed, as technicians spend an average of 16 minutes driving to move from one appointment to another. The respect of the imposed QoS levels has been achieved through continuous dynamic changes in the model-weeks, moving an average of 10 resources per week, that corresponds to about two technicians per day. A Visualization Tool has been developed in order to test the set of routes with the company. Figures 2.6 and 2.7 show, respectively, a screen-shot where all of the routes used by technicians during a week are depicted, and another screen-shot in which the detail of one of these routes in a particular day is displayed.

For what concerns the consultation of service times, another table is created. Figure 2.8 reports the comparisons between different service times for each activity (e.g., average simulated service time, current QoS level and hypothetical new QoS level). From this report, IRETI can verify if, on average, some activity is exceeding the hypothetical new QoS level. In our simulation, with the chosen inputs, all of the activities respect the QoS levels. The simulated service time shows considerable improvements, especially concerning the activities D01 and M02.

## 2.9 Conclusions

In this paper, we presented a DSS that was developed as a strategic tool to support IRETI in the achievement of new QoS levels for possible tenders. A major benefit

| Simulation period | | | Demand | | Dynamic manipulation | | Lateness [One-hour precision range] | | | Driving distance [km] | | Driving time [hh:mm] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Weeks | From | To | #appointments | #resources (30 min.) | #technicians movements | #technicians additions | #late appointments | % on total week appointments | Avg. Lateness [hh:mm] | Total driving km | Km per appointment | Driving time per appointment |
| 1 | 04/07 | 08/07 | 211 | 296 | 3 | 0 | 3 | 1% | 00:02 | 2954 | 14,0 | 00:18 |
| 2 | 11/07 | 15/07 | 360 | 532 | 7 | 0 | 11 | 3% | 00:20 | 4365 | 12,1 | 00:16 |
| 3 | 18/07 | 22/07 | 417 | 637 | 13 | 0 | 14 | 3% | 00:31 | 4809 | 11,5 | 00:15 |
| 4 | 25/07 | 29/07 | 418 | 632 | 9 | 0 | 14 | 3% | 00:23 | 4885 | 11,7 | 00:16 |
| 5 | 01/08 | 05/08 | 415 | 639 | 22 | 0 | 5 | 1% | 00:16 | 4981 | 12,0 | 00:16 |
| 6 | 08/08 | 12/08 | 376 | 578 | 12 | 0 | 5 | 1% | 00:26 | 4379 | 11,6 | 00:15 |
| 7 | 15/08 | 19/08 | 369 | 575 | 6 | 0 | 11 | 3% | 00:24 | 4424 | 12,0 | 00:16 |
| 8 | 22/08 | 26/08 | 240 | 401 | 5 | 0 | 2 | 1% | 00:10 | 2598 | 10,8 | 00:15 |
| **Avg Values** | | | **351** | **536** | **10** | **0** | **8** | **2%** | **00:19** | **4174** | **12** | **00:16** |

Figure 2.5: KPIs provided in output

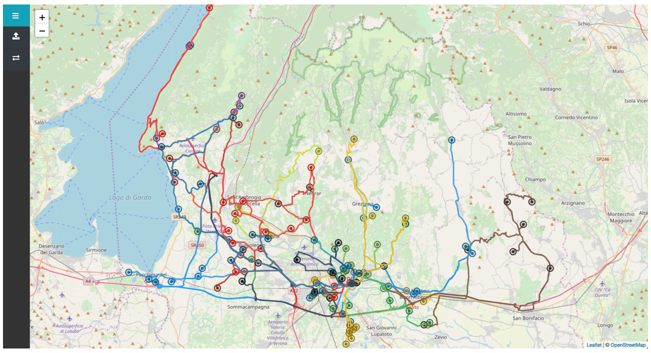


Figure 2.6: Visualization interface: view of routes for all technicians during a week

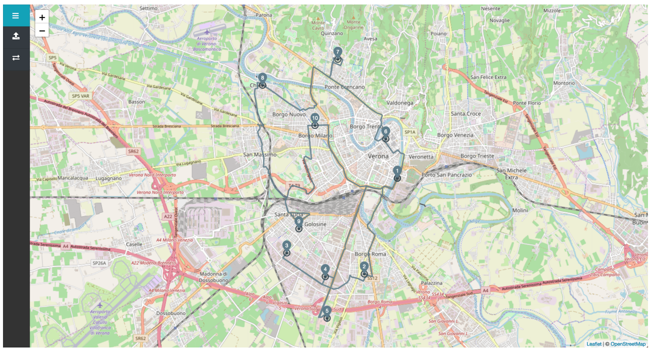


Figure 2.7: Visualization interface: detailed daily route plan for a single technician

| Activity code | Avg simulated service time (working days) | Current QoS level | Hypothetic new QoS level |
|---|---|---|---|
| A01 | 4 | 10 | 5 |
| D01 | 2 | 5 | 2 |
| A40 | 4 | 10 | 5 |
| V01 | 10 | 20 | 10 |
| M01 | 4 | 10 | 5 |
| M02 | 3 | 15 | 8 |

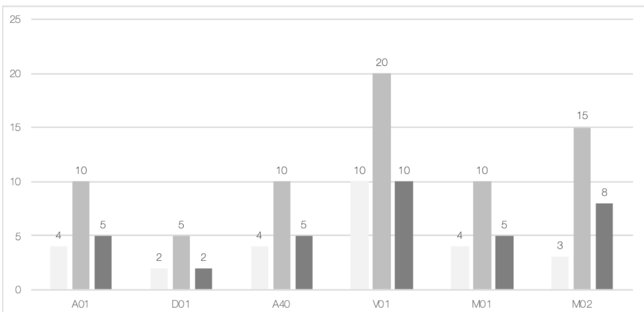



Figure 2.8: Service time output chart

of using the DSS is the possibility to simulate several different scenarios, trying to identify the best solutions for a given territory. The DSS is flexible, easy to replicate and allows the company to simulate service performance in territories that are not currently served. As an example, IRETI is using the system to prepare technical documents for new tenders. The current version of the DSS is a prototype that requires specialized knowledge, given the multiple information technologies used. We do not, however, foresee any technical difficulty to implement a version of the system comprising all its current features into a more user-friendly package (e.g., [85]).

Furthermore, as suggested in [26], another contribution for this study could be to develop new strategies to handle situations in which demand exceeds capacity. At the moment, in these scenarios we consider that IRETI is capable of diverting resources from other sectors to make sure that all services are fulfilled in time, which obviously increases the cost for the company. We plan to modify the system to analyze the consequences of such planning strategy and compare with a scenario where it is possible to postpone services at the cost of a penalty per service that would be imposed to the company. The aim would be to find a trade-off between diverting resources and accepting that paying the penalty sometimes might be less costly.

# Appendix. Mathematical Models

This appendix contains the details of the two mathematical formulations that are used for optimization purposes within the DSS.

## Formulation P-MFLP

Formulation P-MFLP is the core of the clustering part of the solution approach described in the *Problem Solution* section, and is used for dividing the territory managed by the company (ATEM) into clusters of municipalities that will share the same time table. Let $n$ be the number of municipalities in the ATEM, and $c_{ij}$ be the traveling distance from the center of municipality $i$ to the center of municipality $j$, for $i, j = 1, \ldots, n$. Let also $q_i$ be the expected demand of $i$, and $F$ be a set of incompatible pairs of municipalities that cannot be part of the same cluster. Set $F$ is useful to impose additional constraints possibly required by the company on the basis of practical experience and knowledge of the territory. The aim of model P-MFLP is to assign all municipalities to exactly $p$ clusters, in such a way that the sum of the distances between each municipality and the centroid of its cluster is minimized and a maximum unbalance among the total demands of the clusters is limited by an input ratio $\alpha$. In our model, cluster centroids have to be chosen in the centers of the municipalities, and $p$ is an input parameter that can be varied to test different solutions. In other words, we need to select $p$ municipalities, among the $n$ available ones, whose centers will serve as cluster centroids.

Let $y_i$ be a binary variable that takes the value 1 in case cluster $i$ is selected, 0 otherwise, for $i = 1, \ldots, n$, which means that a cluster with centroid located in the center of municipality $i$ is opened. In addition, let $x_{ij}$ be a binary variable that takes the value 1 if municipality $j$ is assigned to cluster $i$, 0 otherwise, for $i, j = 1, \ldots, n$. We obtain the following formulation.

$$\text{(P-MFLP)} \quad \min \; z_{\text{(P-MFLP)}} = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\; x_{ij} \tag{2.1}$$

subject to

$$\sum_{i=1}^{n} y_i = p \tag{2.2}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1, \dots, n \tag{2.3}$$

$$x_{ij} \le y_i \qquad i, j = 1, \dots, n \tag{2.4}$$

$$\sum_{j=1}^{n} q_j\; x_{ij} \le \left\lceil \sum_{j=1}^{n} q_j / p \right\rceil (1+\alpha) \qquad i = 1, \dots, n \tag{2.5}$$

$$x_{ij} + x_{it} \le 1 \qquad i = 1, \dots, n\; ;\; (j,t) \in F \tag{2.6}$$

$$y_i \in \{0,1\} \qquad i = 1, \dots, n \tag{2.7}$$

$$x_{ij} \in \{0,1\} \qquad i, j = 1, \dots, n \tag{2.8}$$

The objective function (2.1) minimizes the total distance between each municipality and the centroid of its cluster. Constraints (2.2) ensure that exactly $p$ clusters are created, constraints (2.3) specify that each city must be assigned to a cluster. Constraints (2.4) impose that municipality $j$ can be assigned to cluster $i$ only if $i$ has been selected (note that each municipality is a potential centroid of a cluster, so both indices $i$ and $j$ vary from 1 to $n$). Constraints (2.5) aim at balancing the distribution of resources among the clusters by ensuring that the demand of each cluster does not exceed the average demand per cluster by more than $\alpha$. Note that smaller values of $\alpha$ lead to clusters where the demand is well balanced, while larger ones neglect balancing but allows to obtain lower cost solutions. In our experiments, we found good results with $\alpha \in [0.2,\ 0.6]$. Finally, constraints (2.6) specify that any pair of incompatible municipalities cannot be assigned to the same cluster, and (2.7) and (2.8) impose integrality on the variables.

## Formulation MWGen

The mathematical model described in this section is based on the formulation originally proposed by [33], but has been modified to better fit the needs of the project and generate more robust solutions. As described in the *Problem Solution* section, model MWGen is used for generating initial good-quality model-weeks for the clusters obtained by formulation P-MFLP, that are then passed to the LNS for further optimization. Before presenting the model, we first need to introduce some notation.

We are given a set $M$ of depots, which are spread out in the territory managed by the company (ATEM). The territory is divided in a set $R$ of clusters. Each cluster is associated with a depot, in such a way that $R = \cup_{i \in M} R_i$, with $R_i$ being the subset of clusters associated with depot $i \in M$. Recall from the *Introduction* section that we use the term *resource* to specify a time interval of half an hour, given that in our case study services require either half an hour or one hour. Each cluster $r \in R$ has an expected demand of $q_r$ resources. Clusters that have significantly high demand values usually contain large cities, which require different QoS levels. These clusters are grouped in a subset $R_0 \subseteq R$. Furthermore, each depot $i$ has a certain number $Q_i$ of technicians that are available to perform services. Each technician is able to provide at most $\sigma$ resources per time slot, with $\sigma = 2$ in our case study, and its route must always start and end at the same depot.

The time horizon is divided into a set $D$ of days, each of which is further split into a set $T$ of non-overlapping time slots. In our case study, $|D| = 5$ (corresponding to the working days in a week, from Monday to Friday) and $|T| = 9$ (corresponding to intervals of one hour each from 8:30 to 17:30, where the interval [12:30-13:30] is reserved for the lunch break). In the following, a *time slot* is defined by a pair $(d, t)$, with $d \in D$ and $t \in T$, a *time table* is an assignment of resources to the time slots, and a *solution* to the problem is a collection of time slot tables, one per cluster.

Let $u_{rdt}$ be an integer variable that specifies the number of technicians assigned to the time slot $(d, t)$ of cluster $r$. Note that, differently from [33], instead of resources, we assign technicians to time slots. This is a simple optimization based on the observation that each technician is always able to perform $\sigma$ resources per time slot, and thus any variable $u$ that is not a multiple of $\sigma$ could be rounded up to the next multiple. Furthermore, let $z_r$ be an integer variable specifying the number of working hours that could not be allocated to the available technicians in a certain cluster $r \in R$. These working hours will be assigned to a third-party logistics provider. In addition, let $v_{rdt}$ be an integer variable that determines the difference in the number of allocated technicians between two consecutive time slots. These variables are used to measure how well the technicians assigned to a certain cluster $r$ are distributed among consecutive time slots along the day. To penalize uneven distribution of technicians per day, we introduce variables $\rho_{min}$ and $\rho_{max}$, that evaluate, respectively, the minimum and maximum number of technicians assigned per day to large clusters, i.e., to any cluster $r \in R_0$. We also define variables $\eta_{min}$ and $\eta_{max}$ for the same purpose, but this time we assign them to any cluster $r \in R \setminus R_0$. We are now ready to introduce the MWGen formulation.

$$\text{(MWGen)} \qquad \min z_{(\text{MWGen})} = \sum_{d \in D} \pi \; (\rho_{max} - \rho_{min}) + \sum_{d \in D} \pi \; (\eta_{max} - \eta_{min}) + \sum_{r \in R} \sum_{d \in D} \sum_{t \in T} \gamma \; v_{rdt} + \sum_{r \in R} \Omega \; z_r \tag{2.9}$$

subject to

$$\sum_{d\in D}\sum_{t\in T} u_{rdt} = \frac{q_r}{\sigma} - z_r \qquad r \in R \qquad (2.10)$$

$$z_r \leq \lambda_1 \, \frac{q_r}{\sigma} \qquad r \in R, d \in D, t \in T \qquad (2.11)$$

$$\sum_{r\in R_i} u_{rdt} \leq Q_i \qquad i \in M, d \in D, t \in T \qquad (2.12)$$

$$\sum_{t\in T} u_{rdt} \geq \rho_{min} \qquad d \in D, r \in R_0 \qquad (2.13)$$

$$\sum_{t\in T} u_{rdt} \leq \rho_{max} \qquad d \in D, r \in R_0 \qquad (2.14)$$

$$\sum_{t\in T} u_{rdt} \geq \eta_{min} \qquad d \in D, r \in R \setminus R_0 \qquad (2.15)$$

$$\sum_{t\in T} u_{rdt} \leq \eta_{max} \qquad d \in D, r \in R \setminus R_0 \qquad (2.16)$$

$$v_{rdt} \geq u_{rdt} - u_{rd,t+1} \qquad r \in R, d \in D, t \in T \setminus \{\ell - 1, \ell, |T|\} \qquad (2.17)$$

$$v_{rdt} \geq u_{rd,t+1} - u_{rdt} \qquad r \in R, d \in D, t \in T \setminus \{\ell - 1, \ell, |T|\} \qquad (2.18)$$

$$v_{rdt} \geq u_{rdt} - u_{rd,t-1} \qquad r \in R, d \in D, t \in \{\ell - 1, |T|\} \qquad (2.19)$$

$$v_{rdt} \geq u_{rd,t-1} - u_{rdt} \qquad r \in R, d \in D, t \in \{\ell - 1, |T|\} \qquad (2.20)$$

$$u_{rd\ell} = 0 \qquad r \in R, d \in D \qquad (2.21)$$

$$u_{rdt} \geq 0, \text{integer} \qquad r \in R, d \in D, t \in T \qquad (2.22)$$

$$z_r \geq 0, \text{integer} \qquad r \in R \qquad (2.23)$$

$$v_{rdt} \geq 0, \text{integer} \qquad r \in R, d \in D, t \in T \qquad (2.24)$$

$$\rho_{min}, \rho_{max}, \eta_{min}, \eta_{max} \geq 0, \text{integer} \qquad (2.25)$$

The objective function (2.9) minimizes the sum of four penalties. The first two penalize solutions that have an uneven distribution of technicians among the time slots of each day for, respectively, large and small clusters. The third one penalizes uneven distribution of technicians among consecutive time slots and the last one aims at minimizing the number of working hours that could not be assigned to the available technicians. Parameters $\pi$, $\gamma$ and $\Omega$ represent the weights associated with each penalty. Constraints (2.10) ensure that the number of resources assigned to each cluster does not exceed its demand, while (2.11) impose an upper bound to the number of technicians that could not be assigned to fulfill the demand of a certain cluster. Thus, constant $\lambda_1$ specifies the minimum ratio of the demand that must be fulfilled. Constraints (2.12) ensure that the capacity of each depot is not exceeded, whereas (2.13)–(2.16) determine the minimum and maximum number of resources assigned per day. Constraints (2.17)–(2.20) evaluate how balanced is the distribution of technicians among the time slots of a day by connecting $u$ and $v$ variables. For an in depth explanation of these constraints we refer the reader to [33]. Finally, constraints (2.21) impose that all time slots associated with the lunch break (time slot $l \in T$) are unavailable for services, and (2.22)–(2.25) require integrality of the variables.

Although the aforementioned model is able to generate complete solutions for the problem, it does not consider some key QoS elements that are important for the company. To improve the level of QoS offered in the solutions designed by the model, we introduce some additional variables and constraints. Let $y_{rd}$ be a binary variable that specifies whether there is at least one time slot opened in day $d \in D$ for cluster $r \in R$. Note that, a time slot $(d,t)$ of cluster $r$ is considered opened if $u_{rdt} > 0$. A higher QoS level is then imposed by including the following constraints.

$$\sum_{t \in T} u_{rdt} \geq y_{rd} \qquad r \in R, d \in D \qquad (2.26)$$

$$\sum_{l \in \phi(d,g)} y_{rl} \geq 1 \qquad r \in R, d \in D \qquad (2.27)$$

$$u_{rdt} \leq y_{rd}\, Q_i \qquad r \in R_i, d \in D, t \in T \qquad (2.28)$$

$$\left| \sum_{t=1}^{\ell\ -1} u_{rdt} - \sum_{t=\ell\ +1}^{|T|} u_{rdt} \right| \leq \lambda_2 \sum_{t \in T} u_{rdt} \qquad d \in D, r \in R_0 \qquad (2.29)$$

$$\left| \sum_{d \in D} \sum_{t=1}^{\ell\ -1} u_{rdt} - \sum_{d \in D} \sum_{t=\ell\ +1}^{|T|} u_{rdt} \right| \leq \lambda_2 \sum_{d \in D} \sum_{t \in T} u_{rdt} \qquad r \in R \setminus R_0 \qquad (2.30)$$

$$y_{rd} \in \{0,1\} \qquad r \in R, d \in D \qquad (2.31)$$

As in [33], constraints (2.26)–(2.28) impose a limit $g$ on the maximum number of consecutive days without any time slot opened in a cluster, which is particularly important for clusters with low demand. In (2.27), function $\phi(d,g) = \{(i \bmod |D|) + 1 : i = d-1, d, \ldots, d+g-1\}$ is used to switch indices from one week to the next one. Constraints (2.29) and (2.30) state that there must be a certain balance in the distribution of technicians between time slots in the morning and in the afternoon, by forcing the difference in number of assigned technicians to be smaller than or equal to $(\lambda_2 \times 100)\%$ of the total number of assigned technicians. In our case, the company specified that larger clusters (i.e., any cluster $r \in R_0$) should have even higher QoS levels, which explains why constraints (2.29) are defined for each day and cluster, whereas (2.30), in contrast, are only imposed for each cluster. In our experiments, we observed that good results were obtained by setting $\lambda_2 = 0.15$.

One of the main goal of formulation MWGen is to repair solutions in the iterations of an LNS procedure (see [33]). To this aim, it is important to allow the model avoiding finding very similar solutions, or even the same one over and over again. Let $s \in S$ represent a feasible solution found at a given iteration, with $S$ being the set of solutions explored. Let also $w_{rdt}$ be an additional binary variable that specifies whether time slot $(d,t)$ from cluster $r$ is opened. Let $\Upsilon = |R||D||T|$ be the total number of $w_{rdt}$ variables, and, in addition, let $W_s^0$ and $W_s^1$ denote the sets of $w_{rdt}$ variables that take value 0 and 1, respectively, in solution $s$, for $s \in S$. The following set of constraints can be used to remove from the search space solutions that are very similar to any solution $s \in S$.

$$u_{rdt} \leq w_{rdt}\ Q_i \qquad r \in R_i, d \in D, t \in T \tag{2.32}$$

$$u_{rdt} \geq w_{rdt} \qquad r \in R, d \in D, t \in T \tag{2.33}$$

$$\sum_{(r,d,t) \in W_s^0} (1 - w_{rdt}) + \sum_{(r,d,t) \in W_s^1} w_{rdt} \leq \Upsilon - 1 \qquad s \in S \tag{2.34}$$

$$w_{rdt} \in \{0, 1\} \qquad r \in R, d \in D, t \in T \tag{2.35}$$

Constraints (2.32), (2.33) and (2.35) define the $w$ variables and link them with the $u$ variables, while constraints (2.34) are *no-good cuts* that ensure that the new solution found by formulation MWGen will have a different configuration of opened time slots with respect to the previous solutions found during the LNS search.

# Chapter 3

# A Decision Support System to Evaluate Suppliers in the Context of Global Service Providers

Extended version of "Bruck, B.P., Vezzali, D., Iori, M., Magni, C., Pretolani, D. (2021) A Decision Support System to Evaluate Suppliers in the Context of Global Service Providers. In *Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 1: ICEIS*, pp. 420-430" (to be submitted to an international journal).

## Abstract

In this paper, we present a decision support system (DSS) developed for a global service provider (GSP) and aimed at solving a real-world supplier selection problem. The GSP operates in the Italian market of facility management, supplying customers with a variety of services. These services are subcontracted to external qualified suppliers spread all over Italy and chosen on the basis of several criteria, such as service quality, capacity and proximity. Selecting the best supplier is a complex task due to the large number of suppliers and the great variety of facility management services offered by the GSP. Here, we formulate the supplier selection problem as a multi-objective generalized assignment problem, where we maximize quality and proximity of the selected suppliers and minimize penalties produced by overcapacity assignments. In the proposed DSS, the choice of the best supplier for a certain service is made according to a thorough multi-criteria decision analysis (MCDA). The weights for the criteria are generated by implementing both a simplified Analytic Hierarchy Process and a revised Simos' procedure, later validated by the decision makers at the GSP. The quality score of each supplier is computed by applying an accurate weighted sum method, based on the MCDA. The DSS provides quick access to historical performance data, visual tools to aid decisions, and a rolling horizon algorithm to perform the assignment of contracts to suppliers. The effectiveness of the proposed system is assessed by means of an extensive computational evaluation on a seven-year period of real data.

## 3.1 Introduction

The supplier selection problem (SSP) is a well-known strategical problem in supply chain management. Many authors agree on the idea that a careful selection of suppliers leads to long-term competitive advantages [79]. To perform this careful selection, it may be convenient to adopt a multi-criteria evaluation that takes into account different characteristics of suppliers. According to [93], quality, delivery and cost are the most popular criteria, but several other criteria might be equally important depending on the context. Grouping and weighting these multiple criteria is not an easy task though, and a careful analysis is usually required to obtain the best results.

Such a careful selection is particularly critical in the facility management industry, where the term Global Service Provider (GSP) is used to identify general players which compete to supply their customers (e.g., banks, hotels, offices and shop chains) with facility management services (e.g., air-conditioning, heating, electrical and fire protection system maintenance and cleaning services), by subcontracting their execution to external qualified suppliers. The definition of a comprehensive multi-criteria evaluation is key to supporting GSPs in the selection of the most adequate partners in their business.

Multi-criteria decision analysis (MCDA) is a well-established research field which deals with decision problems, such as ranking and sorting, where the decision process must consider multiple criteria ([81], [96], [141]). In this sense, applying MCDA to the problem of selecting the best supplier for a requested service is of particular interest. As reported by [78], [43] and [93], integrated approaches that combine MCDA and other methods, like optimization and simulation, are to some extent diffused in the SPP literature.

This paper presents a real-world study on the implementation of a DSS for a multi-criteria and multi-objective SSP in the facility management industry. In particular, the DSS was developed to support *H2H Facility Solutions SpA*, an Italian GSP based in Zola Predosa (Bologna), in the process of supplier selection. *H2H Facility Solutions SpA*, as a GSP, supplies its customers with a series of facility management services, which can be classified as planned preventive maintenance, corrective maintenance, or extraordinary maintenance. The categories of service provided vary from air-conditioning and heating systems to water supply systems, electrical systems, elevator systems, fire protection systems, cleaning services, alarm systems and security, and so forth. *H2H Facility Solution SpA* faces a decision problem any time a facility management contract for a category of service has to be subcontracted. To help the company solve this problem, we developed a DSS, which relies on a particular multi-objective version of the generalized assignment problem and consists of determining the optimal assignment of contracts to suppliers by (i) maximizing the suppliers' quality score, (ii) minimizing the suppliers' distance score, and (iii) minimizing the suppliers' penalty score induced by overcapacity assignments.

The quality score was determined in partnership with the company by carefully defining a comprehensive hierarchical tree of criteria. We implemented a simplified Analytic Hierarchy Process (AHP) and a revised Simos' procedure to compute the weights of the identified criteria. Following an accurate data preparation process, the quality score of suppliers for each category of service was then obtained. The distance score is, instead, intended to take account of the distance between the customers' facilities and the appointed supplier's location. Indeed, proximity between customers and suppliers is desirable as it guarantees a better compliance with service level agreements and, consequently, a greater customer satisfaction. The penalty score aims at penalizing the assignment of contracts which exceed the suppliers' service capacity.

It is important to note that in the SSP application faced in this work, all suppliers that intend to collaborate with the GSP sign a long-term framework agreement, which includes detailed determination of the cost of each category of service provided. As a consequence, we do not seek to minimize such a cost and we intend to find the best supplier only in terms of quality score, distance score and penalty score.

To formally describe and solve the decision problem, a multi-objective mixed integer linear programming (MILP) model and a heuristic algorithm were developed. They were both implemented and integrated in a rolling horizon framework to perform the assignment of facility management contracts to suppliers. In addition, a web application with a user-friendly interface was developed and tested with potential users on a real-data set obtained from historical data provided by the company and from additional data collected with an online survey sent to a sample of suppliers. Furthermore, several computational experiments over a seven-year period were performed to assess the effectiveness of the proposed methods and gain practical insights.

The remainder of this paper is structured as follows. Section 3.2 presents a brief literature review on integrated approaches for supplier selection. In Section 3.3, the SSP in the context of GSPs is formally defined. The proposed multiple criteria evaluation, the computation of weights and the computation of quality scores are provided in Section 3.4. Section 3.5 describes the DSS implementation, while the computational experiments are reported in Section 3.6. Finally, in Section 3.7, we draw conclusions and formulate possible future research directions. A preliminary version of this work, reporting a limited set of experiments with a simplified approach (no penalties for over capacity, no mathematical model and no company configuration implementation) was presented as [35].

## 3.2 Literature Review

Integrated approaches, optimization and evaluation methods based on multiple criteria for supplier selection have been widely studied since the early 1990s. For relevant seminal works we refer the interested reader to [184], [79] and [77], [78]; for a more in-depth overview of this field of research, we refer to the surveys by [93], [182] and [44]. Furthermore, in the latest years, the topic of sustainability is drawing increasing attention in supply management due to its high applicability. For an overview of the problem of green supplier selection we refer to the survey by [80].

The AHP is a multi-criteria decision method developed in the early 1970s, whose purpose is to break down a decision (e.g., a selection or ranking problem) into factors, arranged in a hierarchic structure from an overall goal to criteria, subcriteria and alternatives in successive levels [148]. The AHP can be applied as an individual method or integrated with other techniques, due to its simplicity, ease of use and flexibility. Among the multi-criteria decision making approaches for supplier evaluation and selection surveyed by [93], integrated AHP approaches were proved to be the most commonly used. In addition, from the recent survey by [92] it also emerges that integrations of the AHP are widely applied in manufacturing and logistics areas, and the most commonly studied problem is supplier evaluation and selection.

The integrated approach that most concerns our work is AHP-mathematical programming [77]. Among other techniques used in conjunction with AHP for supplier evaluation and selection, one finds Lexicographic Goal Programming [43], Goal Programming [112], Preemptive Goal Programming [178], and Dynamic Programming [122]. See also [91] and [92] for a review of these papers.

Recently, several authors have successfully developed DSSs based on MCDA to help decision makers in selecting the best suppliers. An interesting work that resembles ours is [55], where an integrated AHP-based DSS for supplier selection in automotive industry is developed. In this implementation, AHP is applied to rank automotive suppliers in Pakistan, identifying four main criteria (price, quality, delivery and service) broken down into subcriteria (e.g., lead time, error, and on-time delivery to assess delivery, order update, warranty, and geographical location to evaluate service). The relative weights of criteria and subcriteria are computed using an AHP, based on the opinions of sourcing experts collected through a survey. The DSS is then tested on a simplified case study consisting of 3 suppliers.

In contrast, our DSS was implemented in the context of GSPs and tested on a broader database consisting of 158 suppliers and 12,412 contracts. The identification of the main criteria determining the quality score, and their relative subcriteria, was performed in partnership with the company in an early stage of our work. The computation of the weights was performed using AHP and data from a survey performed with experts from the company.

Remarkably, our work provides a series of valuable contributions, as compared with the reviewed literature:

- The choice of criteria and their relative subcriteria, performed jointly with an extended working group from the company, is consistent with the most popular evaluating criteria found in the literature on supplier selection.

- We use the AHP to compute the weights of a complex and multilevel tree of criteria and the obtained results are compared with and validated by a revised Simos' procedure [68]. Our pairwise comparisons are based on a simplified 1-3 scale instead of the fundamental 1-9 scale for AHP preference originally proposed by Saaty, to simplify the surveying process that precedes the definition of comparison matrices. The proposed methodology is highly repeatable and can be reiterated at regular intervals in accordance with the *desiderata* of the company.

- The specific SSP of *H2H Facility Solutions SpA* is formally defined as a multi-objective MILP model and solved both exactly and heuristically.

- Our case study is built on a broad database of 158 suppliers and 12,412 contracts, which makes it particularly relevant in terms of problem dimension.

- Extensive computational experiments on a seven-year period of real data were performed using a simulator with three different configurations: a *company* configuration that recreates and evaluates the choices made by the company, a *greedy* configuration that performs the assignments of contracts to suppliers based on a weighted utility function, and a *MILP-based* configuration that has at its core the aforementioned multi-objective MILP model. All configurations are implemented and integrated in a rolling horizon framework.

- Finally, a web application with a user-friendly interface and interactive visual tools (to favor the evaluation of suppliers and support the user in the decision-making process) is proposed.

To the best of our knowledge, no analogous strategic and operational tool exists in the supplier selection literature. Furthermore, because of the emerging role of GSPs in many different markets, our study constitutes a valuable real-world application of AHP, MCDA and optimization.

## 3.3 Problem Definition

In our case study, a facility management contract is related to a service and concerns a particular facility. Every time the GSP formalizes a contract with a customer, the contract is subcontracted to an external qualified supplier that is capable of providing the required service, in accordance with a predefined service-level agreement. Being fixed a priori, cost is independent from the solution of the SSP and is thus not considered as an objective in the problem definition nor as a criterion in the multi-criteria evaluation.

Formally, given a set $C$ of contracts and a set $F$ of suppliers, the SSP in the context of GSPs is to subcontract a series of facility management contracts to the best suppliers with the multiple objective of (i) maximizing the total quality score of the selected suppliers, (ii) maximizing the total distance score, and (iii) minimizing the total penalty score due to the assignments of contracts exceeding a supplier's capacity (defined as the maximum number of contracts that can be assigned to a supplier). The quality score and the distance score are multiplied by coefficients $\alpha$ and $(1-\alpha)$, where $\alpha$ lies in $[0,1]$, which control the relative importance of these two terms of the objective function and can be customized by the decision maker. The third term is instead multiplied by coefficient $\beta$, which corresponds to the penalty generated by each contract assigned over capacity. In Section 3.6, we test several combinations of these coefficients.

For each supplier, we define a normalized quality score $S_f \in [0,100]$ obtained by scaling a quality score $s_f$ derived from the MCDA that is described in Section 3.5. Then, we define $D_{cf} \in [0,100]$ as a normalized distance score derived from the geographical distance $d_{cf}$ between the facility to whom contract $c$ is related and the branch of supplier $f$ that requires the facility management service. In Sections 3.4 and 3.5, we describe in detail how $S_f$ and $D_{cf}$ are computed. Further, we define $q_f$ as the capacity of supplier $f$ in terms of number of contracts that the supplier can serve at the same time.

Let $x_{cf}$ be a binary variable that takes the value 1 if contract $c$ is subcontracted to supplier $f$ and 0 otherwise, and let $y_f = \max\{\sum_{c\in C} x_{cf} - q_f, 0\}$ be a continuous variable reporting the number of contracts assigned over the capacity of supplier $f$, if any. The SSP can then be modeled as follows:

$$\text{(SSP)} \qquad \max z_{(\text{SSP})} = \left( \alpha \sum_{c \in C} \sum_{f \in F} S_f x_{cf}; (1-\alpha) \sum_{c \in C} \sum_{f \in F} D_{cf} x_{cf}; -\beta \sum_{f \in F} y_f \right) \tag{3.1}$$

subject to

$$\sum_{f \in F} x_{cf} = 1 \qquad c \in C \tag{3.2}$$

$$\sum_{c \in C} x_{cf} \leq q_f + y_f \qquad f \in F \tag{3.3}$$

$$x_{cf} \in \{0, 1\} \qquad c \in C, f \in F \tag{3.4}$$

$$y_f \geq 0 \qquad f \in F \tag{3.5}$$

The objective function (3.1) maximizes the total quality score of the selected suppliers and the total distance score, and minimizes the total penalty score for contracts assigned over the capacity of the suppliers. The minimization of overcapacity is particularly important to guarantee the assignment of contracts to several suppliers, instead of using always the same ones. Constraints (3.2) impose that each contract $c$ has to be assigned to exactly one supplier, whereas constraints (3.3) are soft capacity constraints that links variables $x_{cf}$ and $y_f$. Finally, constraints (3.4) and (3.5) define the domain of the variables. Note that an independent SSP is solved for each category of service, as capacity $q_f$ varies depending on the category of service. This aspect could have been highlighted using an additional index in the mathematical model, but we decided to omit it for better readability.

It is worth mentioning that the rolling horizon algorithm proposed in Section 3.5 adds a dynamical aspect to the problem, which is the daily update of quality score $s_f$ (and, consequently, $S_f$) due to the assignment of new contracts to suppliers. Such a dynamical evaluation should avoid the issue of saturating a few suppliers with most of the contracts, which has the potential of gradually deteriorating their performance in the long-term.

## 3.4 Multiple Criteria Evaluation

In this section, we describe how the quality score $s_f$ is evaluated for each supplier $f \in F$. We obtain this value as the solution of an underlying *multiple criteria group decision* problem, which involves a hierarchy of criteria as well as a plurality of decision makers. Due to the large number of criteria and alternatives, this problem bears strong resemblance to the computation of a *composite index*. Composite indexes are a powerful and widespread tool for obtaining a numerical synthesis of multiple assessments from different perspectives. The European Commission created the Competence Centre on Composite Indicators and Scoreboards (COIN) [63] to provide guidelines and tools for building robust composite indexes. Similarly, the United Nations Environment Programme developed the Sustainability Assessment of Technologies (SAT) Methodology [172] to support the assessment process in the context of sustainable development.

Taking into considerations the features mentioned above, we solved the multiple group decision problem by partitioning it into two distinct subproblems. In the first

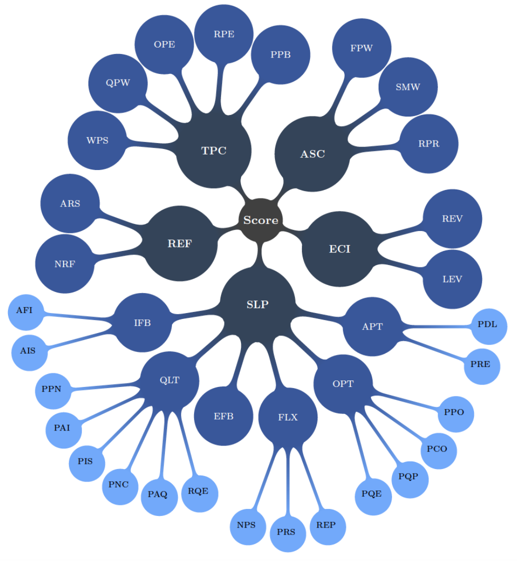



Figure 3.1: Tree of criteria.

one, we aggregate the evaluations issued by several experts to assign a weight to all the indicators in the hierarchy of criteria. In the second subproblem, we exploit the criteria weights to compute a quality score for each alternative; in this phase, we apply a *weighted sum* method where we pre-process the suppliers' evaluations according to the guidelines recommended for composite indexes (see, e.g., [141], Ch. 4.1).

In the following subsections, we show how to derive the quality score $s_f$. In particular, we (i) build the criteria hierarchy, (ii) assess the criteria weights (in two ways: a simplified AHP and revised Simos' procedure), and (iii) compute the suppliers' quality scores.

### 3.4.1 Definition of the Criteria Hierarchy

The multiple criteria setting on which the supplier evaluation is based is the result of an analysis performed in partnership with the company. This analysis was conducted through several rounds of interviews, which led to the definition of the multi-level tree of criteria depicted in Figure 3.1.

In particular, three levels of criteria were identified. The *macro* criteria directly contribute to define the quality score $s_f$ for each supplier $f$. This first level is broken down into a second level of *micro* criteria, which, in a few cases, are further split into a third level of *nano* criteria. We selected five macro criteria that describe the main dimensions of supplier evaluation in the context of GSPs:

- economic indicators (ECI);
- technical and professional capability (TPC);
- additional saturation capacity (ASC);
- service level performance (SLP);
- references (REF).

We here present in depth only the five macro criteria. The ECI aims to give an evaluation of suppliers in terms of dimension and economic soundness, based on the last year's financial statements. The TPC evaluates the organizational structure, the competencies and the extensiveness of suppliers over the territory. The ASC gives the residual capacity of suppliers in terms of possibility of accepting new contracts. The SLP aims to carefully evaluate the suppliers across several performance indicators, on the basis of their historical data. The REF is particularly important to qualify suppliers, given that it is based on the references of customers with whom the suppliers have already worked.

Next, we report the list of micro criteria for each of the aforementioned macro criteria:

- ECI: revenue (REV) and leverage (LEV);
- TPC: workers per service (WPS), qualifications per worker (QPW), office workers per employee (OPE), revenue per employee (RPE), and number of provinces per branch (PPB);
- ASC: facilities per worker (FPW), square meters per worker (SMW), and revenue produced with *H2H Facility Solutions SpA* per total revenue (RPR);
- SLP: operational punctuality (OPT), administrative punctuality (APT), flexibility (FLX), quality (QLT), internal feedback (IFB), and external feedback (EFB);
- REF: number of references (NRF) and average reference segment (ARS).

These micro criteria are very context-specific and, among them, the micro criteria regarding SLP are further broken down into a series of nano criteria, which are listed in the following:

- OPT: percentage of planned preventive maintenance services performed out of service-level agreement (PPO), percentage of corrective maintenance services performed out of service-level agreement (PCO), percentage of quotes presented late (PQP), and percentage of quotes executed late (PQE);
- APT: percentage of requested documents presented late (PDL) and percentage of maintenance reports erroneously filled out (PRE);
- FLX: ratio of extraordinary maintenance to planned preventive maintenance (REP), percentage of rejected corrective maintenance services (PRS), and assigned but not performed services (NPS);
- QLT: ratio of quoted extraordinary maintenance to extraordinary maintenance (RQE), percentage of accepted quotes (PAQ), percentage of notifications from customers (PNC), percentage of incomplete maintenance services (PIS), percentage of additional information sent by means of the maintenance app (PAI), and percentage of planned preventive maintenance services not performed (PPN);
- IFB: average internal score (AIS), and affordability index (AFI);
- EFB: this micro criterion is not further defined.

### 3.4.2 Assessing Criteria Weights

After defining the multi-level tree of criteria, the computation of weights was performed. In particular, the weight of each item in the criteria hierarchy is computed as follows: at the *first* level, the weights of macro criteria are determined; at the *second* level, for each macro criterion the weights of the related micro criteria are determined; at the *third* level, for each micro criterion, the weights of the related nano criteria (if defined) are determined.

Note that the sum of macro criteria weights must be equal to one, and the same holds for the sum of micro (respectively, nano) criteria weights for each macro (respectively, micro) criterion. The weight assessment was performed following a rather conservative approach: we employed a simplified AHP procedure and validated the set of weights by calculating a distinct set of weights with the revised Simos' procedure. As a matter of fact, the results obtained in the two cases turned out to be remarkably similar, although some small fluctuations in the values were detected. Overall, the weights obtained with the AHP procedure, which we used in our computational experiments, can be considered sufficiently robust.

#### Weight Computation with Simplified AHP

Criteria weights are the result of a group decision procedure, where the answers from 20 decision makers at the GSP were collected through an online survey. For each respondent, a simplified AHP was performed, based on the three levels of the *reduced scale* reported in Table 3.1. The rationale behind the use of a reduced scale, instead of the fundamental scale originally proposed by Saaty, is to simplify the collection of pairwise judgments, possibly minimizing inconsistencies[1].

Table 3.1: Reduced scale.

| Relative Importance | Comparison Value |
|---|---|
| Strongly less | 1/5 |
| Moderately less | 1/3 |
| Equal | 1 |
| Moderately more | 3 |
| Strongly more | 5 |

The respondents were asked to use the reduced scale to answer standard questions such as "*What is the relative importance of criterion A compared to criterion B?*". For each respondent and for each level of the criteria hierarchy, pairwise comparison judgments were converted into numerical values and recorded in a reciprocal comparison matrix A. Let $n$ be the number of criteria. Each entry $a_{ij}$ of A gives the comparison value of criterion $i$ with respect to criterion $j$ $\forall\ i, j = 1, ..., n$. In addition, $a_{ji} = 1/a_{ij}$ $\forall\ i, j$, and $a_{ii} = 1$ $\forall\ i$. Given the comparison matrix, the corresponding vector of weights $p$

[1]Note that the use of small size evaluation scales is a rather common practice in the computation of composite indexes. For example, a three-level scale was adopted (within a weighting procedure simpler than ours) for the 2016 European Digital City Index [23].

was derived by applying the so-called "mean of row" method (see, e.g., [95]), which is based on the following three steps:

1. Sum the elements of each column $j$: $S_j = \sum_{i=1}^{n} a_{ij} \ \forall\, j$;

2. Divide each element $a_{ij}$ by the relative column sum: $a'_{ij} = \frac{a_{ij}}{S_j} \ \forall\, i, j$;

3. Compute the mean of each row $i$: $p_i = \frac{\sum_{j=1}^{n} a'_{ij}}{n} \ \forall\, i$.

Finally, for each node in the criteria tree the *aggregated* weight was obtained computing the geometric mean of the weights assigned by all the respondents (the geometric mean is the standard aggregation method in group decision making contexts like ours; see, e.g., [1]). The weights obtained for the macro criteria are reported in Table 3.2, where $\overline{p}_i$ denotes the weight for macro criteria $i$.

Table 3.2: Aggregated weights for macro criteria using the AHP.

| $i$ | IEC | TPC | ASC | SLP | REF |
|---|---|---|---|---|---|
| $\overline{p}_i$ | 0.1527 | 0.2672 | 0.1794 | 0.2394 | 0.1614 |

#### Weight Computation using the Revised Simos' Procedure

To verify the results obtained using the simplified AHP, we applied a different weight assessment method, namely the revised Simos' procedure by [68]. In this case, the experiment was restricted to a group of 8 decision makers at the GSP, whose answers were collected during individual interviews. The motivation for this group restriction lies in the fact that the interviews took considerable time and were performed in person.

The experiment followed a four-step procedure, which was repeated for each level of the criteria tree. The first three steps correspond to the original Simos' procedure, while the fourth step was introduced in the revised methodology proposed by [68] to improve a few drawbacks of the original work. The whole procedure is described in the following:

1. Given a set of $n$ criteria that have to be weighted, give the respondent a first set of $n$ cards with the name of each criterion written on them. Then give the respondent a second set of white cards, having the same size;

2. Ask the respondent to rank the criteria in ascending order, from the least important to the most important. If some criteria have the same importance, they must be grouped together;

3. Ask the respondent to insert white cards between successive criteria (or subsets of *ex aequo* criteria) if a difference in terms of importance needs to be highlighted. The principle of white cards insertion is simple: the greater the difference, the greater the number of white cards that must be inserted;

4. Finally, ask the respondent to estimate the relative importance of the last criterion (or one in the last subset of *ex aequo* criteria) compared to the first.

For each decision maker, the normalized weights of criteria were obtained applying the algorithm by [68]. Again, the aggregated weights were obtained computing a geometric mean; the results for the macro criteria are reported in Table 3.3.

Table 3.3: Aggregated weights for macro criteria using the revised Simos' procedure.

| $i$ | IEC | TPC | ASC | SLP | REF |
|---|---|---|---|---|---|
| $\overline{p}_i$ | 0.1571 | 0.2608 | 0.1971 | 0.2676 | 0.1175 |

The results are consistent with those obtained using the AHP: the largest relative difference from the corresponding AHP weight arises for the REF criterion and is still below 30%, which can be considered an acceptable variation.

### 3.4.3 Computing the Scores

Once the weight of each criterion has been determined, we applied a weighted sum method to compute the score of each supplier. The weighted sum procedure consists of three steps:

1. statistical treatment of outliers (*winsorization*);
2. normalization;
3. aggregation.

Let us denote by $I$ the set of nodes in the criteria tree (i.e., macro, micro and nano criteria) and by $L \subset I$ the set of leaves of the criteria tree. The set $L$ contains the first-order criteria, that is, the nano criteria for macro criterion SLP, the micro criterion EFB of SLP, and the micro criteria for the other macro criteria (see also Figure 3.1). For each supplier $f \in F$ and each leaf $i \in L$ we are given an evaluation $e_{if}$, expressed on a criterion-specific cardinal scale. In what follows, we describe the above three steps separately and finally discuss some theoretical properties of the resulting weighted sum procedure.

#### Statistical Treatment of Outliers

Outliers detection is quite relevant in our context, where a small number of suppliers may be characterized by uncommon features. As an example, consider the micro criterion *revenue* (REV), where a couple of larger companies showed a much larger evaluation compared to the other suppliers. As a consequence, after normalization most of the suppliers (except the two larger ones) would receive an evaluation close to zero, which means that most of the discriminating power of the criterion would actually be lost. As we now show, a suitable treatment of outliers avoids this kind of loss of information.

Outliers are detected applying a rather simple *box plot* method. Given a criterion $i \in L$: we find values $Q_1$ and $Q_3$ of the first and third quartiles of the evaluations $e_{if}$, respectively; we compute the Inter Quantile Range $\text{IQR} = Q_3 - Q_1$; and then we define the lower threshold $T^l = Q_1 - 3 \cdot \text{IQR}$ and the upper threshold $T^u = Q_3 + 3 \cdot \text{IQR}$.

A value $e_{if}$ larger than $T^u$ or smaller than $T^l$ is identified as an outlier for criterion $i$. Note that in many case we have $T^l < 0$, while the evaluations are restricted to non-negative values. Outliers are then treated by applying the following winsorization process:

- each evaluation $e_{if} > T^u$ is replaced by the value $e_i^{\max} = \max\{e_{if} : f \in F, e_{if} \leq T^u\}$;
- each evaluation $e_{if} < T^l$ is replaced by zero.

This rather simple treatment of outliers is sufficient in our context, but clearly more sophisticated methods exist. For example, an iterative process based on higher moments is suggested by COIN ([63]; see, in particular, [130]).

### Normalization

In this phase, each evaluation $e_{if}$ for $i \in L$ is mapped onto a *normalized evaluation* $E_{if} \in [0, 1]$. For every criterion $i$, we distinguish between *direct* and *reverse* normalization:

- For a maximization criterion $i$ (i.e., the better $f$, the greater $e_{if}$) direct normalization gives $E_{if} = e_{if}/e_i^{\max}$, where, after winsorization, we have $e_i^{\max} = \max\{e_{if} : f \in F\}$. Note that outliers previously falling over the upper threshold $T^u$ take value $E_{if} = 1$;
- For a minimization criterion $i$ (i.e., the better $f$, the smaller $e_{if}$) reverse normalization gives $E_{if} = 1 - e_{if}/e_i^{\max}$; outliers previously falling below $T^l$ take value $E_{if} = 1$.

As a result, the better $f$, the greater $E_{if}$.

Since $e_{if} \geq 0$ in our context, we have $e_i^{\max} = \|e_{i\cdot}\|_\infty$, where $e_{i\cdot} \in \mathbb{R}^{|F|}$ is the vector of evaluations for criterion $i$. Normalization based on the infinity norm has been often advocated in MCDA, together with other norms such as $\|\cdot\|_1$ and $\|\cdot\|_2$; here $\|\cdot\|_\infty$ was chosen also because it is not sensitive to the number of outliers. Note that $E_{if}$ does not necessarily attain the extremes of the interval $[0, 1]$ for each criterion, because it may be $\min_f E_{if} > 0$ for direct normalization and $\max_f E_{if} < 1$ for reverse normalization. This fact is acceptable in our context, even if it could be prevented by a slightly more complex normalization step (see, e.g., [141], Ch. 4.1).

### Weighted Sum Aggregation

The aggregation phase can be seen as a three-step bottom-up recursive process. At the first step, for each micro criterion $i$ of SLP (the only macro criterion divided up to nano ones), except EFB, we compute

$$E_{if} = \sum_{j \in S_i} \overline{p}_j E_{jf} \qquad \forall f \in F \tag{3.6}$$

where $S_i$ is the set of nano criteria for $i$, and $\overline{p}_j$ is the weight of nano criterion $j$. Since we have $\sum_{j \in S_i} \overline{p}_j = 1$, it follows that each value $E_{if}$ is normalized between zero and one. Thus, at the end of this step, we have a normalized evaluation $E_{if}$ for each $f \in F$ and each micro criterion $i$.

In the second step, we define for each macro criterion $i$ the values $E_{if}$ as in Equation (3.6), where, in this case, $S_i$ is the set of micro criteria for $i$ and $\overline{p}_j$ is the weight of micro criterion $j$. Again, at the end of this step, we have a normalized evaluation $E_{if}$ for each $f \in F$ and each macro criterion $i$.

In the last step, we obtain the quality score $s_f$ for each $f \in F$ as

$$s_f = \sum_{i \in M} \overline{p}_i E_{if} \qquad \forall f \in F$$

where $M = \{\text{ASC}, \text{ECI}, \text{REF}, \text{SLP}, \text{TPC}\}$ denotes the set of macro criteria and $\overline{p}_i$ is the weight of $i \in M$.

**Independence, Rank Reversal and Stability**

Due to the winsorization and normalization steps, weighted sum lacks the *independence* property, that is, a quality score $s_f$ is not uniquely determined by the evaluations $e_{if}$, but depends on the evaluations of the whole set of suppliers $F$. This implies, in particular, that our scores are exposed to *rank reversal*: given two suppliers $f, g \in F$, their relative ranking as determined by $s_f$ and $s_g$ may be reversed if another supplier is added to or removed from $F$, or if its evaluations change. The rank reversal phenomenon is almost ubiquitous (and often debated) in MCDA methods; see, for example, [181], [76] for discussion, explicative examples and further references.

Observe that our scores have been conceived to be computed repeatedly throughout a wide time horizon, during which the set $F$ and the suppliers' evaluations are assumed to evolve. Thus, we may question the *stability* of our scores over time: a similar issue has been discussed in [142] for the SDEWES Index [94], a composite index that resembles our scores in many aspects. As shown in [142], the combination of winsorization and normalization may occasionally lead to rather unexpected outcomes. However, stability is not a very significant issue in our context, for at least two reasons. First of all, the actual occurrence of rank reversals is rather unlikely, also due to winsorization. Most importantly, our scores should not be considered as an absolute measure of the "quality" of a supplier, but, rather, as a relative measure of attractiveness with respect to a particular time instant. This aspect is further clarified in the next section, which presents our DSS for supplier selection.

## 3.5 DSS Implementation

The DSS consists of three main modules. The first is a MySQL relational database that stores data regarding all suppliers available to the company and all the necessary information about contracts.

The second module is responsible for evaluating the quality score of each supplier according to the hierarchy of criteria presented in Section 3.4. As previously mentioned, when evaluating a given supplier, the quality score $s_f$ is derived by means of a bottom-up recursive process, preceded by the winsorization and the normalization phases. These processes take place in the second module of the DSS.

The third module is a simulator that performs the assignment of contracts to suppliers. In particular, three alternative configurations were implemented in this module: a *company* configuration, a *greedy* configuration and a *MILP-based* configuration. All these configurations can be selected in the rolling horizon algorithm described in Section 3.5.1. Such an algorithm decomposes the whole problem into narrower periods in a way that, for each period, the DSS is able to retrieve updated information from the database, recompute the quality score of each supplier, perform the assignments of contracts to suppliers, and store the results in the database. Note that, with this structure, we are able to run simulations for any period of time based on real-data from the company.

More specifically, the *company* configuration recreates and evaluates the choices made by the company during each period; such a configuration is used exclusively to set a benchmark for the other two configurations. The *greedy* configuration performs the assignments of contracts to suppliers based on a weighted utility function that recalls the objective function of the MILP model defined in Section 3.3. All configurations use the same weighted utility function to evaluate the assignment of contracts to suppliers. In particular, each assignment is evaluated through the following *assignment score*:

$$\zeta_{cf} = \alpha\ S_f + (1 - \alpha)\ D_{cf} - \beta\ y_f,$$

where $c \in C$ and $f \in F$ are, respectively, the contract that we want to assign and the supplier to whom the contract is assigned. The normalized quality score of supplier $f$ is expressed by $S_f = 100 \cdot (s_f/s_{\max})$, given the previously defined quality score $s_f$ and the maximum quality score $s_{\max}$, whereas $D_{cf} = 100 \cdot (1 - (d_{cf}/d_{\max}))$ defines the normalized distance score, given the geographical distance $d_{cf}$ between the nearest branch of supplier $f$ and the facility of customer associated with contract $c$, and the maximum distance $d_{\max}$. Both $S_f$ and $D_{cf}$ are thus scaled in the interval $[0, 100]$. The total number of contracts over capacity for supplier $f$ is represented by $y_f$.

For each contract, the *company* configuration simply replicates the assignments made by the company. The *greedy* configuration computes *assignment score* $\zeta_{cf}$ for all available suppliers and assigns the contract to the one with the highest score. In the *MILP-based* configuration, the decision on the assignments of contracts to suppliers is guided by the multi-objective MILP model (1)-(5) defined in Section 3.3.

Note that, once fixed coefficient $\alpha$, the optimization problem defined by (1)-(5) becomes trivial, as it corresponds to a minimum cost flow problem with a single objective. In addition, we highlight that the *greedy* configuration solves the problem to optimality when $\beta = 0$, as the capacity of suppliers becomes irrelevant. Indeed, without penalty for contracts assigned over capacity, the soft capacity constraints of the MILP model are relaxed and both the *greedy* configuration and the *MILP-based* configuration solve the same problem of finding a minimum cost assignment for each contract. This is confirmed by the results reported in Section 3.6.3.

The overall DSS architecture is depicted in Figure 3.2. The second and the third modules were coded in C++ and CPLEX 12.9 was used as MILP solver in the *MILP-based* configuration. An additional user-friendly interface, described in Section 3.5.2, allows the decision maker to easily interact with the system and make use of simple visual tools.

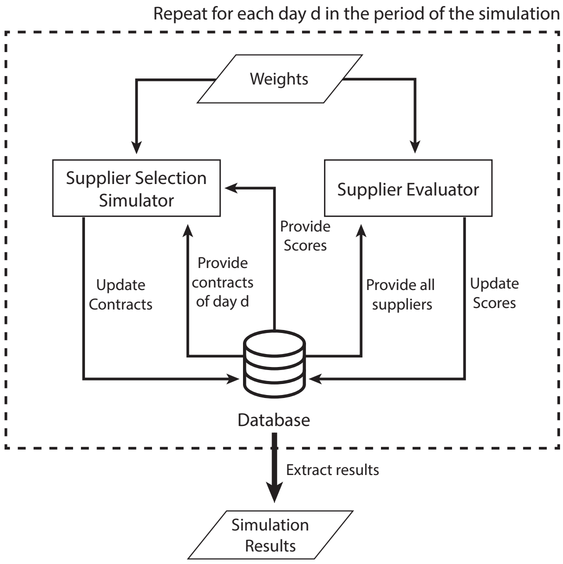



Figure 3.2: DSS architecture.

### 3.5.1 The Rolling Horizon Algorithm

In Algorithm 1, we report the pseudo-code of the proposed rolling horizon framework, where $t \in T$ is a period of $\lambda$ days in the simulation horizon $T$.
For each period, the results are stored in the database, so that the scores are recomputed and updated accordingly whenever new contracts must be assigned. Because the evaluation of suppliers depends on the (past and present) information stored in the database, the score of a certain supplier may well change over time. For instance, as more contracts are assigned to the same supplier, it might become saturated, potentially reducing its score for future contracts. This dynamical aspect is particularly important, as it replicates a typical characteristic of the real problem. However, it is worth mentioning that these simulations are intended to fine-tune the system and validate the results together with the company.

In practice, the DSS is designed to provide decision makers with the necessary tools to make an informed decision without automating the complete process, and it is meant to be integrated as a decision-making component within an Enterprise Resource Planning system.

### 3.5.2 User-friendly Interface

With the aim of further testing and validating the DSS with the decision makers from the company, we developed a set of simple visual tools that are briefly described in this section. The interface was coded in HTML, CSS and JavaScript.

The first is a *radar chart* tool that allows a decision maker to manually select and compare a restricted group of suppliers. This tool visualizes on a spider graph the evaluation obtained by the selected suppliers on the macro criteria. An example of a comparison created using this visual tool is reported in Figure 3.3-(a).

The second is a *ranking* tool that allows the decision maker to query the system and obtain a ranking of suppliers based on their evaluations on the macro criteria. In addition, the decision maker may decide to manually modify the weights of the macro

**Algorithm 1** Rolling Horizon Algorithm

**Require:** $\alpha \in [0,1]$, $\beta \geq 0$ (integer), $\lambda$, $|T|$ ▷ Set parameters
$Config \leftarrow \texttt{Select}\{company, greedy, \textit{MILP-based}\}$ ▷ Select a configuration
**for** $t \leftarrow 1$ **to** $|T|$ **do**
    Get subset $\overline{C} \subseteq C$ of contracts to assign during period $t$
    **if** $\overline{C} \neq \emptyset$ **then**
        **for** $f \leftarrow 1$ **to** $|F|$ **do**
            Update ASC macro criterion
            Recompute supplier quality score $s_f$
        **end for**
        Scale quality score $S_f$ in $[0, 100]$
        **for** $c \leftarrow 1$ **to** $|\overline{C}|$ **do**
            **for** $f \leftarrow 1$ **to** $|F|$ **do**
                Evaluate the branch of supplier $f$ having the shortest distance $d_{cf}$
            **end for**
        **end for**
        Scale distance score $D_{cf}$ in $[0, 100]$
        **if** $Config = company$ **then** ▷ If the *company* configuration was selected
            Recreate and evaluate the choices made by the company during period $t$
        **else if** $Config = greedy$ **then** ▷ If the *greedy* configuration was selected
            **for** $c \leftarrow 1$ **to** $|\overline{C}|$ **do**
                **for** $f \leftarrow 1$ **to** $|F|$ **do**
                    Compute *assignment score* $\zeta_{cf}$
                **end for**
                Assign contract $c$ to supplier $f$ having the highest *assignment score* $\zeta_{cf}$
            **end for**
        **else** ▷ If the *MILP-based* configuration was selected
            Solve the SSP for subset $\overline{C}$ of contracts to assign during period $t$
        **end if**
        Update the simulation statistics
    **end if**
**end for**

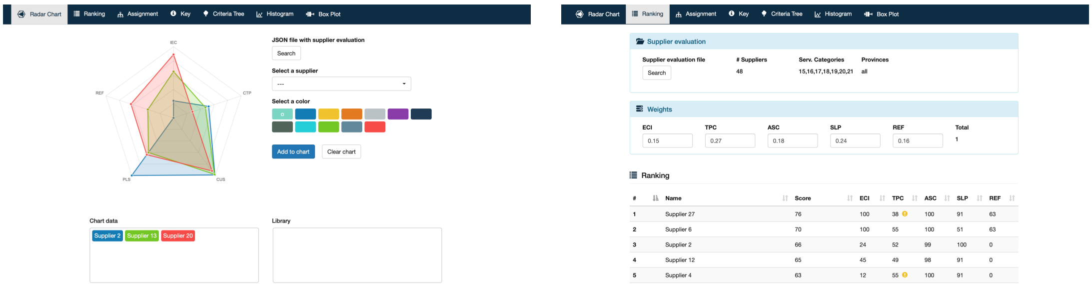
criteria and see how the ranking changes. An example of ranking created using this visual tool is reported in Figure 3.3-(b).

The third is an *assignment* tool that helps the decision maker in selecting a supplier for a new contract. In particular, given a new contract relative to a specific facility, the decision maker can query the system, filter the returned ranking of suppliers, and use his/her experience to select the most appropriate supplier from a reduced list of candidates.

Other simple visual tools, not described here, allow to visualize the statistical series and the box plots for each criterion described in Section 3.4.

## 3.6 Computational Evaluation

In this section, we present the results of the computational experiments performed to test the rolling horizon algorithm presented in Section 3.5.1. The experiments were



(a) *Radar chart* tool (b) *Ranking* tool

Figure 3.3: Screenshots of the DSS interface

run on a PC equipped with an Intel Core i5 dual-core CPU processor @ 2.70 GHz and 8 GB of RAM. We recall that the rolling horizon framework was coded in C++ and CPLEX 12.9 was used as MILP solver with the default configuration.

### 3.6.1 Database of Suppliers and Contracts

Here, we briefly describe the process of data collection and organization that preceded the execution of the experiments.

First, we identified a sample of suppliers together with the company. An online survey was sent to this sample of suppliers to collect several data on their organizational structure (e.g., headquarter and branch positions, number of workers, and number of office workers), technical capabilities (e.g., categories of service offered to customers, type of qualifications, and number of qualifications per type), and economic soundness (e.g., revenue from last year's financial statement). In this way, we collected data for 158 suppliers.

Second, we obtained from the company historical data on 12,412 contracts assigned over seven years from January 2008 to December 2015. Note that each contract is associated with a single category of service required by a customer for a specific facility (with a position and a surface area), and it has a planned duration and a fixed cost. In addition, we obtained from the company detailed performance data registered on all contracts.

Given this large volume of data, we designed a relational database (using MySQL), where the so-collected data were loaded after an accurate process of data cleaning. This database corresponds to the first DSS module described in Section 3.5.

#### Capacity per Category of Service

To configure the rolling horizon algorithm, we performed a preliminary analysis on the database. In particular, we identified 20 categories of service and, for each of them, we computed the median of contracts per worker. This is an important input for the rolling horizon algorithm, as it expresses a different tendency of accepting more or less contracts per worker depending on the category of service.

In particular, at the beginning of the simulation horizon, for each category of service and for each supplier offering that particular category of service, the corresponding median of contracts per worker is multiplied by the number of workers and the result

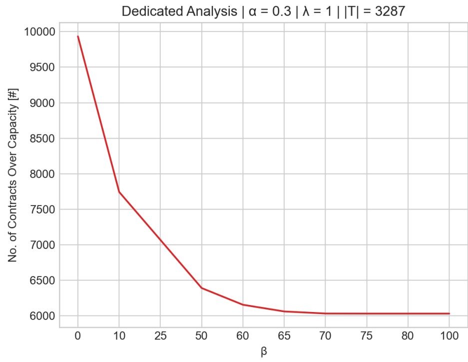



Figure 3.4: Dedicated analysis on coefficient $\beta$

is rounded up to the nearest integer number to obtain the initial capacity $q_f$ for each supplier $f$. Then, this capacity is dynamically updated during the simulation horizon due to the assignment of new contracts to suppliers.

The complete list of service categories is: air-conditioning and heating systems, alarm systems and security, automatisms, cleaning services, construction, consulting and support services, deratting and disinfestation, electrical systems, elevator maintenance, elevator systems, facility management, fire protection systems, furniture and equipment, green services, mechanical systems, porter services, reception desk services, special systems, technological presidium, and water supply systems.

### 3.6.2 Parameter Setting and Dedicated Analysis on Coefficient $\beta$

In the following, we recall the list of parameters that must be given as an input to the rolling horizon algorithm:

- coefficient $\alpha \in [0, 1]$, which controls the relative importance of quality score $S_f$. As a consequence of defining $\alpha$, also coefficient $(1-\alpha)$, which controls the relative importance of distance score $D_{cf}$, is automatically defined;

- coefficient $\beta \geq 0$ (integer), which corresponds to the penalty generated by each contract assigned over capacity. A dedicated analysis to fine-tune this coefficient is reported below;

- length $\lambda$ of each period $t$ in the simulation horizon, expressed in number of days.

Using the rolling horizon framework with the *MILP-based* configuration and parameters $\alpha = 0.3$, $\lambda = 1$ and $|T| = 3,287$ (corresponding to the minimum number of periods to consider all contracts loaded into the database), we performed a dedicated analysis to evaluate the number of contracts assigned over capacity given different values of coefficient $\beta$. From the results reported in Figure 3.4, we see that the number of contracts over capacity reaches an asymptotic value when parameter $\beta$ is between 70 and 75. For this reason, in the computational experiments we decided to choose $\beta \in \{0, 10, 100\}$ to represent different scenarios in terms of penalty on over capacity.

### 3.6.3 Experimental Results

In this section, we illustrate the experimental results obtained while testing the three configurations (i.e., *company*, *greedy*, and *MILP-based*) of the rolling horizon algorithm. We recall that the *company* configuration, which recreates and evaluates the decisions made by the company over the simulation horizon, serves exclusively as a benchmark for the other two configurations.

To limit the computational burden, as the SSP is multi-objective, we avoided computing the entire Pareto front. Instead, using a weighted sum scalarization method, we generated a discrete set of solutions. In particular, these solutions were obtained by varying $\alpha \in \{0.0, 0.1, 0.2, \dots, 1.0\}$.

For what concerns the other parameters, we chose $\beta \in \{0, 10, 100\}$, $\lambda = 1$ and $|T| = 3,287$. In other words, we solved a *daily* SSP for all the days in the simulation horizon. Additional experiments with $\lambda = 7$ (i.e., solving a *weekly* SSP) and $\lambda = 30$ (i.e., solving a *monthly* SSP) were performed but no significant improvements were noticed. This may be due to the particular structure of the real data used for the experiments. Also, note that the distances between the facilities of customers and the branches of suppliers were evaluated using the *haversine formula*.

In the following, we report the experimental results of three alternative scenarios: a first scenario without penalty for contracts assigned over capacity ($\beta = 0$), a second scenario with a limited penalty for contracts assigned over capacity ($\beta = 10$), and a third scenario with a high penalty for contracts assigned over capacity ($\beta = 100$).

#### Scenario without Penalty on Over Capacity

The results that were obtained for the experiments with $\beta = 0$ are reported in Table 3.4, where columns "$\alpha$" and "$(1 - \alpha)$" give the relative weights of the quality score and the distance score, respectively, columns "*company*", "*greedy*", and "*MILP-based*" give the objective function value obtained by each configuration of the rolling horizon algorithm, respectively, and columns "$\%\text{gap}_{company-greedy}$" and "$\%\text{gap}_{greedy-MILP\text{-}based}$" give the percentage gap between the *company* configuration and the *greedy* configuration, and the *greedy* configuration and the *MILP-based* configuration, respectively.

In this scenario, the *greedy* configuration and the *MILP-based* configuration obtained the same results, because, as noticed in Section 3.5, they both solve to optimality the SSP when $\beta = 0$.

From the results reported in Table 3.4, we see that on average the *greedy* configuration and the *MILP-based* configuration improved the result of the *company* configuration by 25%. This behavior is more evident for higher values of $\alpha$, proving that the *company* configuration tends to favor proximity towards quality.

Additional results are reported in Figure 3.5. In Figure 3.5-(a) we plot the number of contracts over capacity. Here, we observe that for $\alpha < 0.7$ the *greedy* configuration and the *MILP-based* configuration assigned less contracts over capacity than the *company* configuration. The opposite holds for $\alpha \geq 0.7$. In Figure 3.5-(b) we plot the average quality score values. On average, the *greedy* configuration and the *MILP-based* configuration outperformed the *company* configuration by 38.4%. In Figure 3.5-(c) we plot the average distance score values. Here, we observe that the *greedy* configuration and the *MILP-based* configuration obtained higher average distance score values than

Table 3.4: Objective function values for $\beta = 0$. Best values in **boldface**

| $\alpha$ | $(1-\alpha)$ | *company* | *greedy* | *MILP-based* | %gap$_{company-greedy}$ | %gap$_{greedy-MILP\text{-}based}$ |
|---|---|---|---|---|---|---|
| 0.0 | 1.0 | 1024615 | **1145267** | **1145267** | 11.8 | 0.0 |
| 0.1 | 0.9 | 983772 | **1103407** | **1103407** | 12.2 | 0.0 |
| 0.2 | 0.8 | 942930 | **1065234** | **1065234** | 13.0 | 0.0 |
| 0.3 | 0.7 | 902088 | **1030535** | **1030535** | 14.2 | 0.0 |
| 0.4 | 0.6 | 861245 | **997898** | **997898** | 15.9 | 0.0 |
| 0.5 | 0.5 | 820403 | **969193** | **969193** | 18.1 | 0.0 |
| 0.6 | 0.4 | 779560 | **950321** | **950321** | 21.9 | 0.0 |
| 0.7 | 0.3 | 738718 | **951462** | **951462** | 28.8 | 0.0 |
| 0.8 | 0.2 | 697876 | **950596** | **950596** | 36.2 | 0.0 |
| 0.9 | 0.1 | 657033 | **953763** | **953763** | 45.2 | 0.0 |
| 1.0 | 0.0 | 616191 | **969082** | **969082** | 57.3 | 0.0 |
| | avg | 820403 | **1007887** | **1007887** | 25.0 | 0.0 |

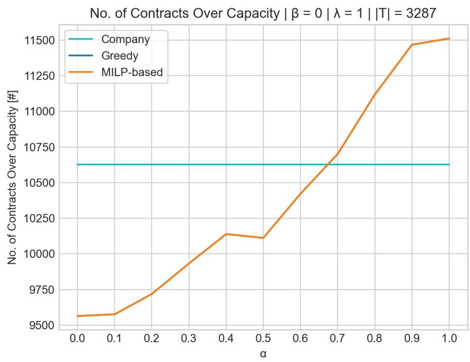
the *company* configuration for $\alpha < 0.8$. Finally, the graphical representation of the Pareto sets is reported in Figure 3.5-(d). Here, on the x-axis we report the average quality score, while on the y-axis we report the average distance score. Note that, in this scenario, the points corresponding to the solution values found by the *greedy* configuration and the *MILP-based* configuration overlap.

### Scenario with a Limited Penalty on Over Capacity

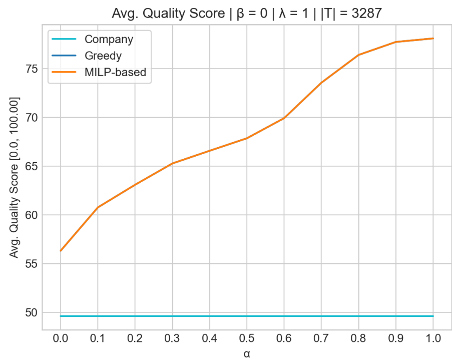
The same experiments were repeated for $\beta = 10$ and the results that were obtained are reported in Table 3.5.

We can notice that, on average, the objective function value of the *greedy* configuration is higher than the objective function value of the *company* configuration by 28.3%, while the objective function value of the *MILP-based* configuration is higher than the objective function value of the *greedy* configuration by only 0.2%. This indicates, on one hand, that with a limited penalty on over capacity there is a remarkable difference between the *greedy* configuration and the *company* configuration (which is greater than the one observed in the previous scenario), and, on the other hand, that the *greedy* configuration and the *MILP-based* configuration perform similarly for $\beta = 10$.

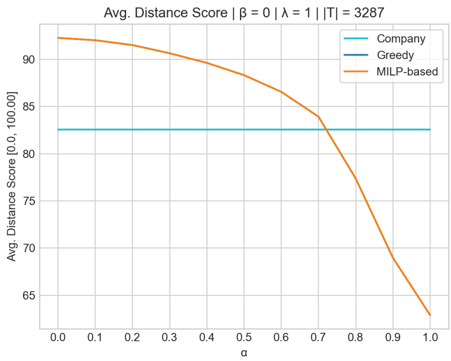
Additional results are reported in Figure 3.6. In Figure 3.6-(a), where we plot the number of contracts over capacity, we observe that the *greedy* configuration obtained significantly better results than the *company* configuration in terms of number of contracts over capacity (with an average reduction of 20.2%). Then, the *MILP-based* configuration obtained a further reduction of 0.3% if compared to the *greedy* configuration. This indicates that both the *greedy* configuration and the *MILP-based* configuration show an interesting potential in managing the available capacity of suppliers, as the two curves almost overlap. In Figure 3.6-(b), where we plot the average quality score values, we see that the results are in line with those found in the previous scenario, thus indicating that the average quality score obtained by the *greedy* configuration and the *MILP-based* configuration remained steady, despite the number of contracts assigned over capacity was significantly decreased. In addition, it is worth noting how the *greedy* configuration performed slightly better than the *MILP-based* configuration



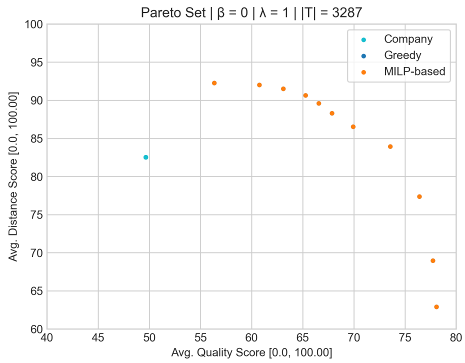

(a) No. of contracts over capacity




(b) Avg. quality score




(c) Avg. distance score



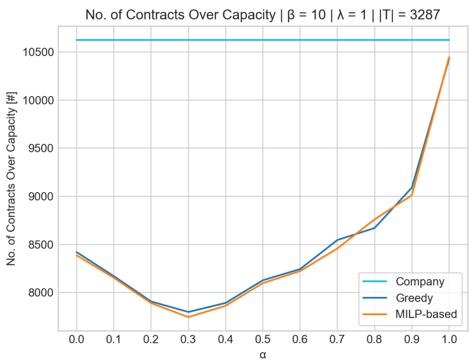

(d) Pareto set

Figure 3.5: Experimental results for $\beta = 0$

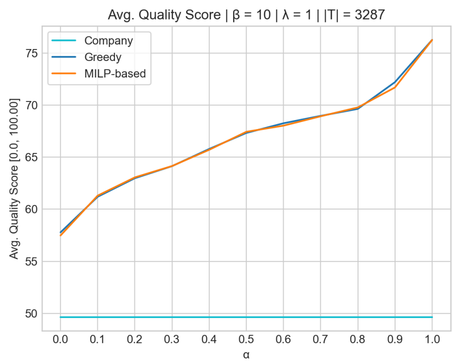
Table 3.5: Objective function values for $\beta = 10$. Best values in **boldface**

| $\alpha$ | $(1-\alpha)$ | *company* | *greedy* | *MILP-based* | %gap$_{company-greedy}$ | %gap$_{greedy-MILP\text{-}based}$ |
|---|---|---|---|---|---|---|
| 0.0 | 1.0 | 918355 | 1052938 | **1055364** | 14.7 | 0.2 |
| 0.1 | 0.9 | 877512 | 1012479 | **1015138** | 15.4 | 0.3 |
| 0.2 | 0.8 | 836670 | 975412 | **978717** | 16.6 | 0.3 |
| 0.3 | 0.7 | 795828 | 940162 | **943141** | 18.1 | 0.3 |
| 0.4 | 0.6 | 754985 | 907508 | **910560** | 20.2 | 0.3 |
| 0.5 | 0.5 | 714143 | 876919 | **880117** | 22.8 | 0.4 |
| 0.6 | 0.4 | 673300 | 850154 | **852057** | 26.3 | 0.2 |
| 0.7 | 0.3 | 632458 | 824887 | **827507** | 30.4 | 0.3 |
| 0.8 | 0.2 | 591616 | 802538 | **804898** | 35.7 | 0.3 |
| 0.9 | 0.1 | 550773 | **803928** | 800594 | 46.0 | -0.4 |
| 1.0 | 0.0 | 509931 | 842442 | **842058** | 65.2 | 0.0 |
| avg | | 714143 | 899033 | **900923** | 28.3 | 0.2 |

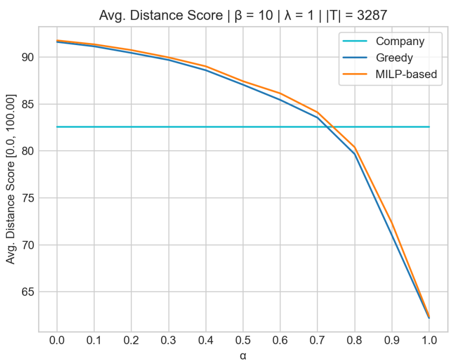
on this indicator. In Figure 3.6-(c) we plot the average distance score values. As before, we see that the *company* configuration obtained a higher distance score for higher values of $\alpha$; in addition, the *MILP-based* configuration performed constantly better than the *greedy* configuration. Finally, the graphical representation of the Pareto sets is reported in Figure 3.6-(d). In this scenario, we see that the Pareto set of the *MILP-based* configuration is slightly above the Pareto set of the *greedy* configuration.



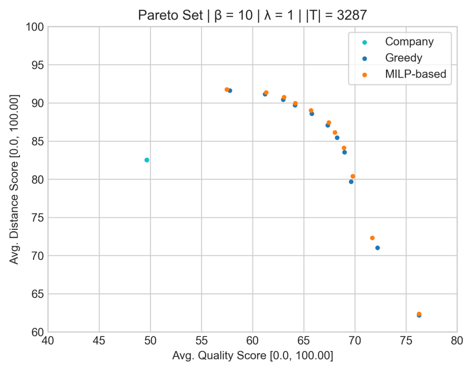

(a) No. of contracts over capacity




(b) Avg. quality score




(c) Avg. distance score




(d) Pareto set

Figure 3.6: Experimental results for $\beta = 10$

### Scenario with a High Penalty on Over Capacity

The results of the experiments performed for $\beta = 100$ are reported in Table 3.6. We observe that the objective function values for the *company* configuration are negative; this is due to the predominant effect of the penalty score. For what concerns the gap between the *greedy* configuration and the *MILP-based* configuration, we see that, on average, the latter outperformed the former by 4.7%, which means that with a high penalty on over capacity the advantage of using the *MILP-based* configuration becomes more evident.

Additional results are reported in Figure 3.7. In Figure 3.7-(a), we observe that the *greedy* configuration significantly improved the results of the previous scenario. Indeed, the average reduction in the number of contracts over capacity increased to 43.1%, if

Table 3.6: Objective function values for $\beta = 100$. Best values in **boldface**

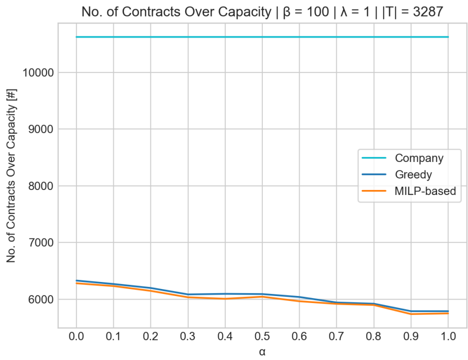
| $\alpha$ | $(1-\alpha)$ | *company* | *greedy* | *MILP-based* | %gap$_{company-greedy}$ | %gap$_{greedy-MILP\text{-}based}$ |
|---|---|---|---|---|---|---|
| 0.0 | 1.0 | -37985 | 387276 | **409646** | n/a | 5.8 |
| 0.1 | 0.9 | -78828 | 362090 | **383584** | n/a | 5.9 |
| 0.2 | 0.8 | -119670 | 341156 | **363213** | n/a | 6.5 |
| 0.3 | 0.7 | -160512 | 328754 | **347393** | n/a | 5.7 |
| 0.4 | 0.6 | -201355 | 304095 | **324492** | n/a | 6.7 |
| 0.5 | 0.5 | -242197 | 283767 | **298414** | n/a | 5.2 |
| 0.6 | 0.4 | -283040 | 270371 | **284656** | n/a | 5.3 |
| 0.7 | 0.3 | -323882 | 261322 | **270801** | n/a | 3.6 |
| 0.8 | 0.2 | -364724 | 248492 | **255195** | n/a | 2.7 |
| 0.9 | 0.1 | -405567 | 255339 | **261031** | n/a | 2.2 |
| 1.0 | 0.0 | -446409 | 260632 | **265015** | n/a | 1.7 |
| avg | | -242197 | 300299 | **314858** | n/a | 4.7 |

compared to the *company* configuration. Again, the *MILP-based* configuration shows an additional average reduction of 0.8%. This confirms the potential in managing the available capacity of suppliers shown both by the *greedy* configuration and the *MILP-based* configuration, especially in those contexts in which the assignment of contracts over capacity is particularly penalizing. In Figure 3.7-(b), we observe that the results on the average quality score values remained steady, if compared to the previous scenario. So, we may conclude that the great reduction in terms of number of contracts over capacity is not accompanied with a significant drop of the average quality score. Again, the *greedy* configuration performed slightly better than the *MILP-based* configuration on this indicator. In Figure 3.7-(c), we observe that the *company* configuration outperformed both the *greedy* configuration and the *MILP-based* configuration in terms of average distance score. This means that, in this scenario, the noteworthy improvement on the number of contracts over capacity and the stable result on the average quality score were obtained at the expense of proximity (i.e., by assigning contracts to suppliers which have a good quality score, but are farther from the facilities of customers). Finally, the graphical representation of the Pareto sets is reported in Figure 3.7-(d). Differently from the previous scenarios, here we may notice that the solutions obtained by the *company* configuration are not always dominated by those obtained by the *greedy* configuration.

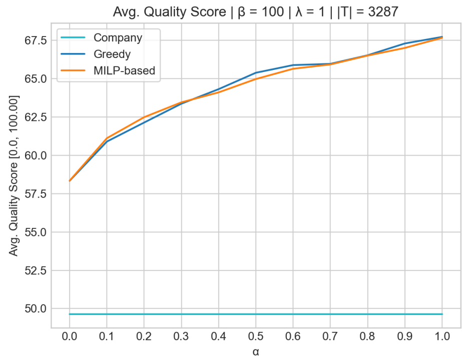
## 3.7 Conclusions

In this paper, we presented a multi-objective supplier selection problem (SSP) arising at *H2H Facility Solutions SpA*, an italian global service provider (GSP) company for which we developed a decision support system (DSS) to aid the decision makers in the process of supplier evaluation and selection. The SSP was formulated as a multi-objective generalized assignment problem, and the evaluation of suppliers was based on a multi-criteria decision analysis (MCDA) performed in partnership with the company.

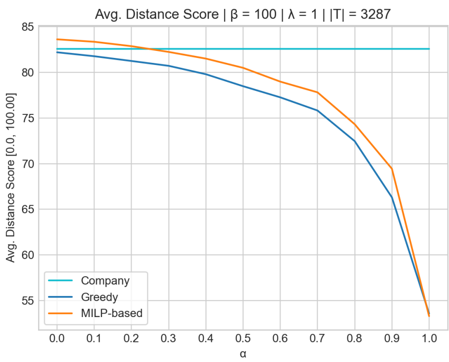
The DSS was implemented using a modular architecture. The first module is a MySQL relational database that stores information on contracts and suppliers. The



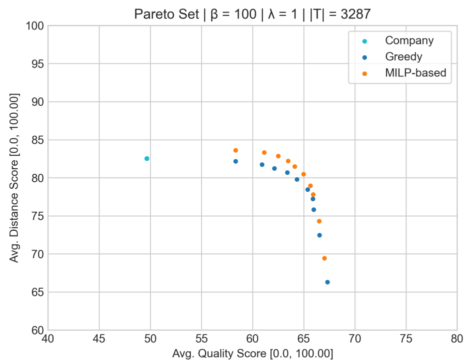

(a) No. of contracts over capacity




(b) Avg. quality score




(c) Avg. distance score




(d) Pareto set

Figure 3.7: Experimental results for $\beta = 100$

second module is responsible for evaluating the quality score of each supplier. The third module simulates the assignment of contracts to suppliers based on a rolling horizon algorithm provided with three alternative configurations: a *company* configuration, a *greedy* configuration and a *MILP-based* configuration. A user-friendly interface, which gives quick access to simple visual tools, was also developed. The effectiveness of the proposed rolling horizon algorithm was tested by means of several computational experiments over a seven-year period of real-data. Given the alternative solutions provided by the discrete Pareto front, a decision maker may then choose the most appropriate one based on his/her experience. The results proved the advantage of using a DSS based on such a rolling horizon algorithm to aid the decision makers in the process of supplier evaluation and selection, especially in those contexts in which we have a considerable number of contracts that must be assigned to a multitude of suppliers (for several categories of service).

In general, we found that the proposed approach is extremely flexible and highly repeatable. Therefore, it may be adapted with some adjustments to other real-world supplier evaluation and selection problems, in different contexts as well. Indeed, in case of adaptation to other companies and industries, the proposed criteria should be slightly reconsidered. However, once redefined the tree of criteria, the methodology may be fully replied. As future work, given the rising importance of GSPs in several sectors and the easy applicability of the proposed methodology, we are interested in implementing analogous DSSs, possibly embedding enhanced heuristics, for other real-world applications. Further future research directions may be represented by the introduction of a weight for each contract, which would make the problem NP-HARD to solve, and the execution of additional experiments to find a good estimation of the minimum number of suppliers that are needed to efficiently cover each geographical area or category of service.

# Chapter 4

# Solution of a Practical Vehicle Routing Problem for Monitoring Water Distribution Networks

Atefi, R., Iori, M., Salari, M., Vezzali, D. (2022). Solution of a Practical Vehicle Routing Problem for Monitoring Water Distribution Networks (submitted to an international journal).

## Abstract

In this work, we introduce a generalization of the well-known Vehicle Routing Problem for a specific application in the monitoring of a Water Distribution Network (WDN). In this problem, multiple technicians must visit a sequence of nodes in the WDN and perform a series of tests to check the quality of water. Some special nodes (i.e., wells) require technicians to first collect a key from a key center. The key must then be returned to the same key center after the test has been performed, thus introducing precedence constraints and multiple visits in the routes. To solve the problem, three mathematical models and an Iterated Local Search have been implemented. The efficiency of the proposed methods is demonstrated by means of extensive computational tests on randomly created instances, as well as on instances derived from a real-world case study.

## 4.1 Introduction

Water contamination is related to the presence of one or more chemical compounds or pathogens to the extent that they become dangerous to the consumer and might lead to diseases [131]. The risk of accidental contamination of drinking water is a well-known issue, and, recently, concerns regarding the deliberate contamination of urban water networks have called for additional safeguards.

In general, any threat to urban water networks directly affects the users in the community [146]. Indeed, according to a report recently released by the *World Health Organization*, contaminated drinking water is estimated to cause 485,000 diarrhoeal deaths each year [187]. The safety of water distribution networks has always been an important issue for the communities. However, many distribution systems in cities around the world face the threat of accidental or intentional contamination during the transportation from treatment plants to consumers due to reverse flows (i.e., the return of contaminated water flows from facilities), old infrastructures, insufficient use of disinfectants, and so forth. Consequently, water contamination in distribution networks is considered as the most diffused cause behind the spread of water-borne diseases [125].

In recent years, several studies have been conducted to identify the main sources of water pollution and improve the quality of water thanks to innovative treatment methods and plants, but still an accidental event, such as a large-scale contamination or a destructive attack to the transmission system, can significantly affect both the economy and the society. In 2014, for example, 300,000 consumers in West Virginia were affected by the accidental contamination of their drinking water distribution system caused by 4-Methylcyclohexanemethanol [155]. During the same year, as reported by [127], a spill of benzene from a chemical plant in China accidentally reached the water distribution network. More recently, 27,000 Norwegian consumers were exposed to water contaminated with Clostridium [185].

Supply, treatment, transmission and distribution of drinking water in urban distribution networks require substantial expenses; therefore, not only water in urban distribution networks is considered an essential resource, but also an economic commodity. The results of a study conducted by the *World Bank* show that nearly 15% of treated water is wasted annually in developed countries. This amount arises to a range of 35-60% for developing countries [192]. Timely control of *Water Distribution Networks* (WDNs) is thus of fundamental importance, both from an economical and public health point of view.

In this paper, a new variant of the well-known *Vehicle Routing Problem* (VRP) in the context of WDNs is proposed. In this problem, a set of technicians must visit a set of nodes, including wells, reservoirs and treatment plants, within a network to evaluate the water quality. When visiting a well, the technicians need a key to open the well and perform the required tests. Since the technicians do not have the key, they have to visit a specified node at which the key is located, called *key center* in the following, to acquire it. As a result, they need to visit this node before reaching the well. After the tests have been performed, they have to take the key back to its original key center before returning to the depot where they started their route. Note that it is not compulsory to visit the key center immediately before and after the well; in other words, the technicians can keep the key with them while visiting other nodes. In addition to that, it is imposed that all nodes are visited and that the duration of any route performed by a technician does not exceed a maximum traveling time. The aim of the problem is to minimize the sum of the traveled times.

The problem originates from a real-world application that we encountered in Mashhad (Iran), where 5 technicians daily inspect a WDN comprising 3,124 households/shops, 293 reservoirs/tanks, 356 wells and 14 treatment plants. Apart from the real-world application, the problem is of broad interest as it models routing problems for the inspection and/or maintenance of equipment where material should be collected from a depot before the execution of the service and then returned to the same depot at the end of the activity. To solve the problem, we propose three *Mixed Integer Linear Programming* (MILP) models, and an *Iterated Local Search* (ILS) algorithm. While the models managed to solve small-size instances with up to 20 nodes, the ILS efficiently tackled cases with up to 200 nodes, allowing us to produce good-quality solutions for randomly created instances, as well as for realistic instances derived from the case study, in short computing times.

The remainder of the paper is organized as follows. In Section 4.2, the relevant literature is revised. The problem is formally described in Section 4.3. Sections 4.4 and 4.5 present the mathematical models and the ILS algorithm, respectively. Computational results are described in Section 4.6, and final conclusions and future research directions are discussed in Section 4.7.

## 4.2 Literature Review

The VRP is an iconic class of problems in operations research, with applications in the fields of transportation, distribution, logistics and services. We refer the interested reader to [168] for an extensive overview, to [126] for a recent survey, and to [169] and [82] for recent collections of benchmark datasets. The problem we face generalizes the VRP by considering precedence constraints and multiple visits. In this section, we only revise routing problems involving these two features, with a particular focus on real-world applications.

In the context of the *Traveling Salesman Problem* (TSP), precedence constraints were first addressed in the seminal work by [21], and, since then, have been widely investigated. In [125], the authors proposed a formulation for the TSP with precedence constraints using a two-commodity network flow model and developed a genetic algorithm based on a topological sorting of customers. In [151], novel formulations for the asymmetric TSP and the precedence constrained asymmetric TSP were proposed. To tighten the formulations, the authors proposed and tested valid inequalities. [165] presented a new model for the time-dependent capacitated profitable tour problem, a generalization of the TSP with time windows and precedence constraints, and developed a tailored labeling algorithm. [150] describe the precedence constrained generalized TSP, in which customers are partitioned into groups and exactly one visit per group must be performed. They presented a novel branching technique and compared several bounding methods.

Precedence constraints have also been widely studied for problems involving multiple vehicles. [146] developed a genetic algorithm based on a topological sorting of customers to solve the VRP with precedence constraints. The algorithm includes a route repair method to generate feasible offspring. A VRP variant with time windows, synchronization and precedence constraints was introduced by [83]. The authors focused on an attended home health care application, and proposed some exact and heuristic solution methods, including a novel MILP formulation, a greedy heuristic, and three metaheuristics.

Precedence constraints naturally arise in the context of *Pickup-and-Delivery Problems* (PDP), where each demand must be first collected at an origin node before being delivered at a destination node. We refer the reader to [24] and [52] for detailed surveys on PDPs for goods transportation and PDPs for people transportation, respectively, and to [108] for a recent survey on simultaneous PDPs. Recently, [20] studied a multi-PDP with time windows. They defined a 2-index formulation, an asymmetric representatives formulation, and a 3-index formulation improved by preprocessing and valid inequalities. The problem was solved exactly using a branch-and-cut algorithm. Dedicated branch-and-cut algorithms were also developed by [87], to solve the single-vehicle two-echelon one-commodity PDP, and by [186], to solve a PDP with split loads and transshipments. The problem addressed in the latter work includes multiple visits to the same node. This is common when split deliveries are allowed, or multiple pickup and delivery operations can be performed at a single node. These generalizations were considered by [34], where non-elementary formulations were proposed for a single-vehicle PDP and then extended to the cases of split deliveries, intermediate drop-offs, and multiple vehicles.

Overall, we may find many routing problems that are inspired by real-world applications and involve precedence constraints and multiple visits. [156] studied an application of a PDP with time windows and precedence constraints arising in the transportation of live animals. In this case, the precedence constraints are given by veterinary rules, imposing that the livestock holdings are visited in a predefined sequence to avoid the spread of potential diseases. The authors proposed a tight formulation of the problem based on a Dantzig-Wolfe decomposition. [144] presented an application in the context of military operations, that was modeled as a generalized VRP with synchronization and precedence constraints. The peculiarity of the problem is due to the nature of the attack, which may require aircraft synchronization, multiple attacks to the same target, and precedence constraints among different targets. The problem was solved by a MILP model.

[74] addressed a particular PDP with time windows originating from the oil industry. The aim of the problem is to determine the routing and scheduling of vessels that collect crude oil from offshore platforms and transport it to terminals on the coast. The authors proposed a MILP model, solving it by means of two different branch-and-cut algorithms. Another valuable example of routing and scheduling in the context of large-scale disaster relief operations was examined by [149]. The authors solved a PDP arising from a case study in the city of Tehran (Iran). They proposed an integrated logistic system to evacuate people from areas affected by natural or man-made disasters. The problem was formulated as a MILP model, and a memetic algorithm was developed to solve large-scale instances.

Recently, a real-world routing application with precedence constraints in the context of healthcare logistics was addressed by [9]. The problem, arisen in the province of Québec and related to the transportation of biomedical samples from specimen collection centers to specific laboratories, was formulated as a MILP model and solved using an ILS.

For what concerns WDNs, the literature mainly contains works on the location of sensors (see, e.g., [145]). The VRP has been applied in many areas, but, to the best of our knowledge, not yet to the inspection of WDNs. In this paper, we fill this lack in the literature and propose exact and heuristic solution methods for a real-world VRP on a WDN.

## 4.3 Problem Description

The WDN is an essential infrastructure that consists of many elements, including reservoirs, wells, pipes and treatment plants.

An effective way to constantly monitor a WDN is by means of water quality sensors, which can be positioned all over the network. In cities where these sensor systems have not been installed, technicians are required to regularly visit nodes of the WDN and perform tests. The nodes to be visited, called for simplicity *demand nodes* in the following, are divided into two types:

1. *Type I*: households, shops, reservoirs, tanks and treatment plants. For this kind of nodes, the technicians can directly go on site and perform the required tests. Reservoirs, tanks and treatment plants are characterized by larger service times than households and shops, due to the larger amount of tests that have to be performed;

2. *Type II*: wells. For these nodes, the technicians need a key to access the well and perform the tests. So, they have to visit first a specified key center, and take the key. Once all tests have been completed at the well, the key needs to be returned to its original key center, thus imposing a second visit.

A simple illustrative example derived from the real-world application we are facing is depicted in Figure 4.1. It comprises three routes starting and ending at the depot. Two of them (top and left part of the figure) visit just reservoirs and treatment plants, so demand nodes of type I. The third (right part of the figure) also visits a well, and is thus forced to pass twice by the corresponding key center.

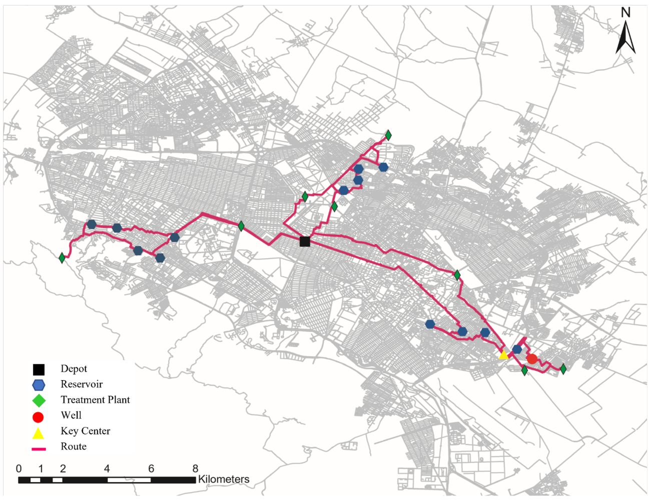



Figure 4.1: An illustrative example of a VRPWDN solution in Mashhad (Iran)

Formally, we are given a directed graph $G = (V, A)$, where the node set is $V = \{0, 1, \ldots, n, n+1\}$ and is partitioned as $V = V_1 \cup V_2 \cup V_3 \cup \{0, n+1\}$. Nodes 0 and $n+1$ represent, respectively, the beginning and end of all routes, and in our application coincide with a unique central depot. Sets $V_1$ and $V_2$ are associated with, respectively, the demand nodes of types I and II. Set $V_3$ comprises nodes associated with all key centers. With each node $i \in V_2$, we associate a predecessor $p_i \in V_3$ and a successor node $d_i \in V_3$. In our application, $p_i$ and $d_i$ correspond to a unique key center, so they have the same geographical location, but the models and algorithms that we propose below can also solve the case in which they correspond to different locations.

Each demand node has to be visited exactly once, while each vehicle visits a particular key center at most once for picking up all the keys, and then another single time for delivering all the keys that were previously collected. This implies that, in case a center has the keys for multiple demand nodes and these nodes are visited by a unique vehicle, then such keys must be collected all together in a unique visit (to $p_i$), and then later delivered all together in another visit (to $d_i$). We recall that it is not compulsory to visit $p_i$ immediately before $i$. In other words, the vehicle can collect the key for $i$ but then visit other nodes before reaching $i$. The same holds for $d_i$, which is not required to be visited immediately after $i$.

The graph is complete, and with each arc $(i, j) \in A$ we associate a traveling time $c_{ij}$. A service time $v_i$ is associated with each node $i \in V$. We suppose that triangle inequality holds for all our instances (i.e., $c_{ij} \leq c_{ik} + v_k + c_{kj}$ for all $i, j, k \in V$). We are also given a set $K$ of homogeneous vehicles based at the central depot. Each vehicle performs a single route. A route starts and ends at the depot. Its duration is given by the sum of the service and traveling times of the nodes and arcs covered by the vehicle, and it should not exceed a maximum duration $L$. Whenever a route visits a node $i$ of type II, then it should also visit $p_i$ and $d_i$.

The aim of the *Vehicle Routing Problem for Water Distribution Networks* (VRPWDN) is to visit all demand nodes, while satisfying all constraints and minimizing the sum of the route durations. The VRPWDN is NP-hard in the strong sense, because it generalizes the well-known VRP. In the next sections, we attempt its solution through mathematical models and heuristic algorithms.

## 4.4 Mathematical Models

In this section, we investigate three mathematical models that describe the VRPWDN and are derived from the literature. The first model is based on a time representation of the problem and is inspired by the formulation proposed by [50] for the VRP with time windows. The second is a flow-based model that builds upon the formulation presented by [99] and later used by, among others, [100], [129], and [8]. The third is a node-based model that we derive from the classical Miller, Tucker and Zemlin formulation (see, e.g., [25]).

### 4.4.1 Time-based Model

Let $y_{ik}$ be a binary variable taking value 1 if node $i$ is visited by vehicle $k$ and 0 otherwise, $x_{ijk}$ be another binary variable taking value 1 if arc $(i, j)$ is covered by vehicle $k$ and 0 otherwise, and $t_{ik}$ be a continuous variable corresponding to the time at which vehicle $k$ arrives at node $i$. The time-based model for the VRPWDN can be formulated as follows:

$$(\text{VRPWDN}_{\text{tb}}) \qquad \min z_{(\text{VRPWDN}_{\text{tb}})} = \sum_{i \in V \setminus \{n+1\}} \sum_{j \in V \setminus \{0\}} \sum_{k \in K} (c_{ij} + v_i) x_{ijk} \tag{4.1}$$

subject to

$$\sum_{j\in V\setminus\{0\}}\sum_{k\in K} x_{ijk} = 1 \qquad i \in V_1 \cup V_2 \tag{4.2}$$

$$\sum_{j\in V\setminus\{0\}} x_{0jk} = 1 \qquad k \in K \tag{4.3}$$

$$y_{ik} = \sum_{j\in V\setminus\{0\}} x_{ijk} = \sum_{j\in V\setminus\{n+1\}} x_{jik} \qquad i \in V, k \in K \tag{4.4}$$

$$\sum_{i\in V\setminus\{n+1\}} x_{i,n+1,k} = 1 \qquad k \in K \tag{4.5}$$

$$\sum_{k\in K} t_{0k} = 0 \tag{4.6}$$

$$0 \leq t_{ik} \leq L y_{ik} \qquad i \in V, k \in K \tag{4.7}$$

$$t_{jk} \geq t_{ik} + v_i + c_{ij} - M_{ij}(1 - x_{ijk}) \qquad i \in V \setminus \{n+1\}, j \in V \setminus \{0\}, k \in K \tag{4.8}$$

$$y_{p_ik} + y_{d_ik} \geq 2y_{ik} \qquad i \in V_2, k \in K \tag{4.9}$$

$$t_{p_ik} + v_{p_i} + c_{p_ii} - M'_i(1 - y_{ik}) \leq t_{ik} \leq t_{d_ik} - (c_{id_i} + v_i)y_{ik} \qquad i \in V_2, k \in K \tag{4.10}$$

$$x_{ijk} \in \{0, 1\} \qquad i, j \in V, k \in K \tag{4.11}$$

$$y_{ik} \in \{0, 1\} \qquad i \in V, k \in K \tag{4.12}$$

Objective function (4.1) is to minimize the total duration of the routes. Constraints (4.2) impose that each node $i \in V_1 \cup V_2$ has exactly one outgoing arc. Each vehicle starts its route from the depot and such condition is imposed by means of constraints (4.3). Constraints (4.4) and (4.5) ensure that each node $i$ has exactly one incoming and one outgoing arc and that each vehicle $k$ end its route at the depot. Constraints (4.6) impose that all routes start at time 0. Constraints (4.7) impose that arrival times are non-negative and limit the duration of each route to be at most $L$. The time at which vehicle $k$ arrives at node $j$ is modeled by means of constraints (4.8), in which we set $M_{ij} = L + v_i + c_{ij} - c_{j,n+1}$. Constraints (4.9) impose that if vehicle $k$ visits node $i$, then it also visits nodes $p_i$ and $d_i$. Since $p_i$ may contain keys not only for $i$ but for other nodes, vehicle $k$ may visit $p_i$ but not $i$, and the same holds for $d_i$. For this reason, the equation cannot be an equality. Constraints (4.10), in which we set $M'_i = L + v_{p_i} + c_{p_i,i} - c_{i,n+1}$, guarantee the respect of precedence constraints, by forcing time dependency between visits to $p_i$, $i$ and $d_i$. Note that if $y_{ik}$ is equal to 0, constraints (4.10) become redundant with respect to constraints (4.7). Constraints (4.11) and (4.12) define the domain of the $x_{ijk}$ and $y_{ik}$ variables.

Furthermore, the aforementioned model can be enhanced with the addition of the following valid inequalities

$$(c_{0p_i} + v_{p_i} + c_{p_ii})y_{ik} \leq t_{ik} \qquad i \in V_2, k \in K \tag{4.13}$$

$$(c_{0p_i} + v_{p_i} + c_{p_ii} + v_i + c_{id_i})y_{ik} \leq t_{d_ik} \qquad i \in V_2, k \in K \tag{4.14}$$

$$t_{jk} \geq (c_{0i} + v_i + c_{ij})x_{ijk} \qquad i \in V \setminus \{n+1\}, j \in V \setminus \{0\}, k \in K \tag{4.15}$$

which strengthen the values taken by the arrival time variables.

### 4.4.2 Flow-based Model

Let $f_{ijk}$ be a variable representing the "load" of vehicle $k$ when traveling along arc $(i,j) \in A$. The load represents the number of nodes visited by vehicle $k$ before it travels along arc $(i,j)$. The flow-based model for the VRPWDN can be formulated as follows:

$$(\text{VRPWDN}_{\text{fb}}) \qquad \min z_{(\text{VRPWDN}_{\text{fb}})} = \sum_{i \in V \setminus \{n+1\}} \sum_{j \in V \setminus \{0\}} \sum_{k \in K} (c_{ij} + v_i) x_{ijk} \tag{4.16}$$

subject to (4.2), (4.3), (4.11) and

$$\sum_{j \in V \setminus \{0\}} x_{ijk} = \sum_{j \in V \setminus \{n+1\}} x_{jik} \qquad i \in V, k \in K \tag{4.17}$$

$$\sum_{i \in V \setminus \{n+1\}} \sum_{j \in V \setminus \{0\}} (c_{ij} + v_i) x_{ijk} \leq L \qquad k \in K \tag{4.18}$$

$$\sum_{j \in V \setminus \{0\}} f_{0jk} = 0 \qquad k \in K \tag{4.19}$$

$$\sum_{i \in V \setminus \{n+1\}} f_{i,n+1,k} = \sum_{i \in V \setminus \{n+1\}} \sum_{j \in V \setminus \{0\}} x_{ijk} - 1 \qquad k \in K \tag{4.20}$$

$$\sum_{j \in V \setminus \{0\}} f_{ijk} \geq \sum_{j \in V \setminus \{n+1\}} (f_{jik} + x_{jik}) \qquad i \in V \setminus \{0, n+1\}, k \in K \tag{4.21}$$

$$\sum_{j \in V \setminus \{0\}} (f_{p_i jk} - f_{ijk} + x_{ijk}) \leq (n-1)(1 - \sum_{j \in V \setminus \{0\}} x_{ijk}) \qquad i \in V_2, k \in K \tag{4.22}$$

$$\sum_{j \in V \setminus \{0\}} f_{p_i jk} \geq \sum_{j \in V \setminus \{0\}} x_{ijk} \qquad i \in V_2, k \in K \tag{4.23}$$

$$\sum_{j \in V \setminus \{0\}} (f_{d_i jk} - f_{ijk}) \geq \sum_{j \in V \setminus \{0\}} x_{ijk} \qquad i \in V_2, k \in K \tag{4.24}$$

$$0 \leq f_{ijk} \leq (n-1) x_{ijk} \qquad i, j \in V, k \in K \tag{4.25}$$

As in the previous model, objective function (4.16) minimizes the total route duration. Constraints (4.17) correspond to the previous constraints (4.4) except for the $y_{ik}$ term. The maximum duration of each route is bounded by means of constraints (4.18). Constraints (4.19) and (4.20) impose the load on vehicle $k$ when leaving 0 and entering $n+1$, respectively. Constraints (4.21) impose the load conservation at node $i$. Constraints (4.22)–(4.24) guarantee the respect of precedence constraints. Constraints (4.25) impose lower and upper bounds on the $f_{ijk}$ variables. The above model can be improved by the addition of the following constraints:

$$\sum_{j \in V \setminus \{n+1\}} x_{jp_i k} + \sum_{j \in V \setminus \{0\}} x_{d_i jk} \geq 2 \sum_{j \in V \setminus \{n+1\}} x_{jik} \qquad i \in V_2, k \in K \tag{4.26}$$

$$\sum_{j \in V \setminus \{0\}} (f_{d_i jk} - f_{p_i jk}) \geq \sum_{l \in V \setminus \{n+1\}: p_i = p_l} \sum_{j \in V \setminus \{0\}} x_{ljk} \qquad i \in V_2, k \in K \tag{4.27}$$

$$\sum_{l \in V \setminus \{n+1\}} (f_{lik} - f_{ljk}) + n x_{ijk} + (n-2) x_{jik} \leq (n-1) \qquad i, j \in V \setminus \{0, n+1\}, k \in K \tag{4.28}$$

Constraints (4.26) are equivalent to (4.9). Constraints (4.27) enforce an additional relation between the flows leaving $p_i$ and $d_i$. Constraints (4.28) are derived from the lifted constraints proposed by [51].

### 4.4.3 Node-based Model

Let $u_{ik}$ be a variable representing the load on vehicle $k$ after leaving node $i$. With respect to the previous model, this implies setting $u_{ik} = \sum_{j \in V} f_{ijk}$. The node-based model for the VRPWDN can be formulated as follows:

$$(\text{VRPWDN}_{\text{nb}}) \qquad \min z_{(\text{VRPWDN}_{\text{nb}})} = \sum_{i \in V \setminus \{n+1\}} \sum_{j \in V \setminus \{0\}} \sum_{k \in K} (c_{ij} + v_i) x_{ijk} \tag{4.29}$$

subject to (4.2), (4.3), (4.11), (4.17), (4.18) and

$$u_{0k} = 0 \qquad k \in K \tag{4.30}$$

$$u_{n+1,k} = \sum_{i \in V \setminus \{n+1\}} \sum_{j \in V \setminus \{0\}} x_{ijk} \qquad k \in K \tag{4.31}$$

$$u_{ik} - u_{jk} + n x_{ijk} \leq (n-1) \qquad i \in V \setminus \{n+1\}, j \in V \setminus \{0\}, k \in K \tag{4.32}$$

$$u_{p_i k} - u_{ik} + \sum_{j \in V \setminus \{0\}} x_{ijk} \leq n(1 - \sum_{j \in V \setminus \{0\}} x_{ijk}) \qquad i \in V_2, k \in K \tag{4.33}$$

$$u_{p_i k} \geq \sum_{j \in \{0\}} x_{ijk} \qquad i \in V_2, k \in K \tag{4.34}$$

$$u_{d_i k} - u_{ik} \geq \sum_{j \in V \setminus \{0\}} x_{ijk} \qquad i \in V_2, k \in K \tag{4.35}$$

$$0 \leq u_{ik} \leq n \sum_{j \in V \setminus \{0\}} x_{ijk} \qquad i \in V, k \in K \tag{4.36}$$

For each vehicle $k$, constraints (4.30) set the load after leaving node 0, while constraints (4.31) define the load when arriving at node $n+1$. Constraints (4.32) impose the load conservation when traveling from node $i$ to node $j$. Constraints (4.33)–(4.35) guarantee the respect of precedence constraints. Constraints (4.36) impose both the non-negativity of the $u_{ik}$ variables and their relation with the $x_{ijk}$variables. The model can be improved by the addition of (4.26) and of

$$u_{d_i k} - u_{p_i k} \geq \sum_{l \in V \setminus \{n+1\}: p_i = p_l} \sum_{j \in V \setminus \{0\}} x_{ljk} \qquad i \in V_2, k \in K \tag{4.37}$$

$$u_{ik} - u_{jk} + n x_{ijk} + (n-2) x_{jik} \leq (n-1) \qquad i, j \in V \setminus \{0, n+1\}, k \in K \tag{4.38}$$

which correspond to the above (4.27) and (4.28), respectively.

## 4.5 Iterated Local Search

We developed an ILS algorithm with the purpose of finding good-quality VRPWDN solutions in short computing times. The choice of this metaheuristic is motivated by its simplicity and effectiveness, in addition to the wide applicability it has found on related VRPs (see, e.g., [173], [157], [83]) as well as on practical applications (see, e.g., [15] and [9]). On the other hand, the need for short computing times is justified by the number of visits usually scheduled in a day in our real-world application, and by the fact that candidate locations might change at the beginning or in the course of a day. Two examples which typically cause a re-scheduling of visits can be a new warning for potential water contamination coming from a household or shop or, when visiting a well, the unfortunate event that the well's door is broken and it is not possible to open it.

Following the general framework proposed by [119], the ILS starts from an initial solution and then improves it by iteratively invoking local search and perturbation procedures. The pseudo-code of the proposed ILS is provided in Algorithm 2. First, we generate an initial solution $x_0$ by means of a heuristic algorithm (line 1), and then we improve it with a local search procedure (line 2). The current solution, $x$, is stored as the incumbent, $x^*$, and inserted in the set of best known solutions obtained during the search, called $BKSet$ (lines 3 and 4). Next, we execute two phases, one after the other.

In the first phase, by applying a perturbation on $x$ followed by a call to the local search (lines 6–8), the algorithm tries to escape from local optima. The perturbation is randomly selected between two tailored procedures. Let $z(x)$ and $l(x)$ be the cost of $x$ and the maximum duration of a route in $x$, respectively. In case $x$ has better cost than $x^*$, or same cost but lower maximum duration, then we use it to update $x^*$. In such a case, we also insert $x$ in $BKSet$. This set contains the $\beta$ different solutions found during the search and having the smallest $z(x)$ costs, breaking ties by smallest $l(x)$ value. If, instead, $x$ does not improve $x^*$, then we set $x \leftarrow x^*$ as starting solution to be shaken at the next iteration. This loop is repeated until no improvement is found for $max_{iter}$ iterations.

With the aim of further improving the solution obtained, at line 16 we enter the second ILS phase, in which a new series of improving attempts is performed. The idea is to intensify the search around the solutions contained in $BKSet$. For each such solution, we perform once more a loop of shaking and local search procedures, which is repeated until the same termination condition used above is met. Should one of these attempts manage to improve the incumbent solution, this time only in terms of costs, then the search restarts from the beginning of the first phase.

In the following, we provide the details of the main elements of the algorithm.

### 4.5.1 Initialization Procedure

Algorithm 3 gives the `Initialization` procedure that is used to generate an initial solution. At the beginning, $|K|$ routes are built in parallel by randomly selecting a first node $i \in V_1 \cup V_2$ per route. In case $i$ belongs to $V_2$, then the predecessor and the successor of $i$ (i.e., $p_i$ and $d_i$) are also inserted into the route. In the next $|V_1 \cup V_2| - |K|$ iterations, a new node is randomly selected and inserted into an existing route. In these iterations, both the node and, in case $i \in V_2$, its predecessor and successor are inserted in the route in the positions that lead to the minimum extra mileage cost. Note that the insertion of node $i$ or tuple $(p_i, i, d_i)$ into an existing route is led by procedure `CheapestInsertion`, which evaluates among the $|K|$ routes the best candidate for the

**Algorithm 2** Iterated Local Search (ILS)

1: $x_0 \leftarrow$ `Initialization()` ▷ Generate an initial solution
2: $x \leftarrow$ `LocalSearch`$(x_0)$
3: $x^* \leftarrow x$
4: $BKSet \leftarrow \{x^*\}$ ▷ $BKSet$: set of best known solutions
5: **repeat** ▷ Phase 1
6:     `Shake()` $\leftarrow$ `Rand`$\{S_1, S_2\}$ ▷ Randomly select a shaking procedures
7:     $x \leftarrow$ `Shake`$(x)$
8:     $x \leftarrow$ `LocalSearch`$(x)$
9:     `Insert`$(x, BKSet)$
10:     **if** $z(x) < z(x^*)$ OR $(z(x) = z(x^*)$ AND $l(x) < l(x^*))$ **then**
11:         $x^* \leftarrow x$
12:     **else**
13:         $x \leftarrow x^*$
14:     **end if**
15: **until** no improvement is found for $max_{iter}$ iterations
16: **for** $j \leftarrow 1, \ldots, |BKSet|$ **do** ▷ Phase 2
17:     $x \leftarrow BKSet_j$ ▷ Select the $j^{th}$ solution $\in BKSet$
18:     **repeat**
19:         `Shake()` $\leftarrow$ `Rand`$\{S_1, S_2\}$
20:         $x \leftarrow$ `Shake`$(x)$
21:         $x \leftarrow$ `LocalSearch`$(x)$
22:         **if** $z(x) < z(x^*)$ **then**
23:             $x^* \leftarrow x$
24:             `Insert`$(x^*, BKSet)$
25:             `Go to line 5`
26:         **end if**
27:     **until** no improvement is found for $max_{iter}$ iterations
28: **end for**
29: **return** $x^*$

expansion. At line 22, the algorithm checks whether the solution is feasible. If not, then the whole procedure is repeated from scratch.

### 4.5.2 Local Search

The `LocalSearch` procedure invokes, one after the other, the following neighborhood searches:

LS1 *Swap intra-route*: swap two sequences with up to three consecutive nodes in the same route. Potential nodes belonging to $V_3$ are extracted from the two sequences and reinserted after the swap following the minimum extra mileage cost and respecting the precedence constraints;

LS2 *Swap inter-route*: swap two sequences with up to three consecutive nodes from different routes, taking care of nodes belonging to $V_3$;

**Algorithm 3** Initialization Procedure

$\mathcal{S}, \mathcal{V} \leftarrow \emptyset$<br>
**for** $k \leftarrow 1, \ldots, |K|$ **do** ▷ Initialization of $|K|$ routes in parallel<br>
    $i \leftarrow$ `Rand`$\{1, ..., |V_1 \cup V_2|\}$<br>
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{i\}$ ▷ Add $i$ to the set of visited nodes<br>
    **if** $i \in V_2$ **then**<br>
        $r_k \leftarrow (0, p_i, i, d_i, n+1)$<br>
        `Insert`$(r_k, \mathcal{S})$<br>
    **else**<br>
        $r_k \leftarrow (0, i, n+1)$<br>
        `Insert`$(r_k, \mathcal{S})$<br>
    **end if**<br>
**end for**<br>
**for** $j \leftarrow 1, \ldots, |V_1 \cup V_2| - |K|$ **do** ▷ Expansion of existing routes<br>
    $i \leftarrow$ `Rand`$\{\{1, ..., |V_1 \cup V_2|\} \setminus \mathcal{V}\}$<br>
    $\mathcal{V} \leftarrow \mathcal{V} \cup \{i\}$<br>
    **if** $i \in V_2$ **then**<br>
        `CheapestInsertion`$((p_i, i, d_i), r_k \in \mathcal{S})$<br>
    **else**<br>
        `CheapestInsertion`$(i, r_k \in \mathcal{S})$<br>
    **end if**<br>
**end for**<br>
**if** `Feasible`$(\mathcal{S}) = 1$ **then**<br>
    `Continue`<br>
**else**<br>
    `Go to line 1`<br>
**end if**<br>
**return** $\mathcal{S}$

LS3 *Relocate intra-route*: remove a sequence with up to three consecutive nodes and reinsert it in a different position within the same route, taking care of nodes belonging to $V_3$;

LS4 *Relocate inter-route*: remove a sequence with up to three consecutive nodes and reinsert it in a different route, taking care of nodes belonging to $V_3$;

LS5 *3-opt*: in a preliminary step, select a route and remove potential nodes belonging to $V_3$. Following this step, apply the standard 3-opt algorithm to the remaining nodes. After each iteration of the 3-opt algorithm update the solution by reinserting the previously extracted nodes belonging to $V_3$.

Procedures from LS1 to LS4 have all complexity $O(n^2)$, whereas LS5 has complexity $O(n^3)$. To limit the computational effort required by LS5, a random logic search is added. In particular, a candidate route $k$ is selected randomly and potential nodes belonging to $V_3$ are removed as follows. For each node $i$ in the route, the saving $s_i$ that could be obtained by removing $i$ and directly connecting the predecessor and successor nodes of $i$ in the route is computed. Then, the probability of removing $i$ is set to

$p_i = s_i/\sum_j s_j$. By means of the roulette wheel mechanism, three non-adjacent nodes are selected for removal, and then the resulting route is optimized by a 3-opt algorithm. A threshold of $\gamma$ iterations is set to limit the number of attempts.

The calls to LS1–LS5 are repeated as long as an improvement is found. Procedure `LocalSearch` hence returns a solution which is a local optimum with respect to all five neighborhoods.

### 4.5.3 Shaking Procedure

To perturb a solution, we randomly select, with same probability, one of the two following procedures.

- S1 *Shaking 1:* randomly select a route $k$ and execute a random iteration of the 3-opt algorithm to update the order of visits. If the cost of the current solution is not worse than $\alpha z(x^*)$, with $\alpha$ being an input parameter, randomly select a second route $k'$ and perform another 3-opt iteration. The procedure is iterated as long as the cost of the perturbed solution is not worse than $\alpha z(x^*)$;

- S2 *Shaking 2:* compute the cost saving obtained by removing any node from the solution, similarly to what is done in LS5. Then use the roulette wheel mechanism to select a node $i \in V \setminus \{0, n+1\}$, and remove $i$ from its route. The removal procedure is iterated until at least $\alpha$ percent of all nodes have been removed. If the selected node belongs to $V_2$, then its saving is computed as the average cost saving obtained by removing $i$, $p_i$, and $d_i$. At the end of this step, the algorithm invokes the `Initialization` procedure to rebuild a feasible solution.

## 4.6 Computational Results

In this section, we present the results of extensive computational tests performed with the aim of assessing the performance of the proposed methods. The mathematical models and the ILS were coded in C++ using Microsoft Visual Studio 2010. The computational tests were executed on a PC equipped with an Intel Core i7 CPU processor @ 2.70 GHz and 6 GB of RAM, using CPLEX 12.3 as MILP solver. In Section 4.6.1, we describe the sets of randomly-created instances that we used for our tests. The comparison among the mathematical models is reported in Section 4.6.2, while the behavior of the ILS is analyzed in Sections 4.6.3 and 4.6.4. In Section 4.6.5, we report the results of additional computational experiments performed on a set of realistic instances derived from the case study.

### 4.6.1 Randomly-created Instances

We created several random instances with the aim of assessing the performance of the algorithms under different situations. In detail, we created two sets of instances, each comprising different subsets having homogeneous values of $|V_1 \cup V_2|$, $(|V_2|, |V_3|)$ and $|K|$, and composed by three random instances per subset. We obtained the following sets:

- *Small-size*: 18 instances with $|V_1 \cup V_2|$=10, $(|V_2|, |V_3|) \in \{(1, 1), (2, 1), (2, 2)\}$, and $|K| \in \{1, 2\}$; 24 instances with $|V_1 \cup V_2|$=15, $(|V_2|, |V_3|) \in \{(3, 2), (3, 3), (4, 2), (4, 3)\}$, and $|K| \in \{2, 3\}$; 24 instances with $|V_1 \cup V_2|$=20, $(|V_2|, |V_3|) \in \{(2, 2), (3, 2), (3, 3), (5, 3)\}$, and $|K| \in \{2, 3\}$;

- *Medium- and large-size*: 24 instances with $|V_1 \cup V_2|$=50, $(|V_2|, |V_3|) \in \{(5, 5), (8, 8), (10, 5), (10, 8)\}$, and $|K| \in \{5, 8\}$; 24 instances with $|V_1 \cup V_2|$=100, $(|V_2|, |V_3|) \in \{(5, 5), (10, 5), (10, 10), (15, 10)\}$, and $|K| \in \{10, 15\}$; 24 instances with $|V_1 \cup V_2|$=200, $(|V_2|, |V_3|) \in \{(10, 10), (20, 10), (20, 20), (30, 20)\}$, and $|K| \in \{15, 20\}$.

For each instance, the coordinates of the nodes are integer values randomly selected between 0 and 100. The distances between the nodes are computed as the Euclidean ones, rounded to the second closest digit. The maximum duration is set to $L = 1.5(\sum_{i \in V_1 \cup V_2} \bar{c}_i + |K| \sum_{i \in V_3 \cup \{0\}} \bar{c}_i)/|K|$, where $\bar{c}_i$ is the average travel time of the arcs leaving $i$, computed as $\bar{c}_i = \sum_{j \in V \setminus \{i\}} c_{ij}/(|V| - 1)$ for each node $i \in V \setminus \{n + 1\}$. The service time $v_i$ for each node $i \in V_1 \cup V_2 \cup V_3$ is set to a random integer value between 20 and 40.

In the following, a subset of instances is identified by the tuple $(|V_1 \cup V_2|, |V_2|, |V_3|, |K|)$, while a single instance is identified by $(|V_1 \cup V_2|, |V_2|, |V_3|, |K|, u)$, where $u$ is a numerical index going from 1 to 3.

To favor future research on the problem, the randomly-created instances have been made publicly available at `https://github.com/DarioVezzali/VRPWDN`.

### 4.6.2 Comparison among the Mathematical Models

In this section, the performance of the three mathematical models from Section 4.4 is investigated. A time limit of 3,600 CPU seconds was imposed on each execution. The aggregated results that we obtained are reported in Table 4.1. Each line reports average/total values for a group of three instances having the same numbers of vertices and vehicles. For each group, columns "$z_{lb}$" and "$z_{ub}$" give the average lower and upper bound values, respectively, column "%gap" gives the average percentage gap and column "t(s)" the average run time. An entry "tlim" indicates that the time limit was reached for all the three instances in the group. Column "opt" gives the total number of instances solved to proven optimality.

From Table 4.1, we can observe that just on a few large-size instances the time-based model and the node-based model find better results in terms of average upper bound. Overall, the flow-based model outperforms the other two models in terms of average lower bound, average percentage gap, average run time, and number of optimal solutions obtained. Consequently, we adopted this model to assess the quality of the solutions obtained by the ILS (see Section 4.6.3).

For all the instances belonging to the medium- and large-size sets, the mathematical models could not obtain proven optimal solutions and the computer frequently ran out of memory because of the large model size. Overall, we can conclude that the results prove the need of a good heuristic for these instances. This need is further motivated by the dimension of the original real-world problem, where the number of visits per day (i.e., around 70) is out of scale if compared to the size of instances solved to optimality within the time limit.

Table 4.1: Comparison of mathematical models (three inst. per line). Best average lower and upper bound values in **boldface**

| | | | | time-based | | | | | flow-based | | | | | node-based | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\lvert V_1 \cup V_2\rvert$ | $\lvert V_2\rvert$ | $\lvert V_3\rvert$ | $\lvert K\rvert$ | $z_{lb}$ | $z_{ub}$ | %gap | t(s) | opt | $z_{lb}$ | $z_{ub}$ | %gap | t(s) | opt | $z_{lb}$ | $z_{ub}$ | %gap | t(s) | opt |
| 10 | 1 | 1 | 1 | **706.39** | **706.39** | 0.00 | 0.87 | 3 | **706.39** | **706.39** | 0.00 | 1.13 | 3 | **706.39** | **706.39** | 0.00 | 1.42 | 3 |
| 10 | 1 | 1 | 2 | **730.91** | **730.91** | 0.00 | 7.45 | 3 | **730.91** | **730.91** | 0.00 | 3.26 | 3 | **730.91** | **730.91** | 0.00 | 12.20 | 3 |
| 10 | 2 | 1 | 1 | **743.25** | **743.25** | 0.00 | 6.54 | 3 | **743.25** | **743.25** | 0.00 | 3.13 | 3 | **743.25** | **743.25** | 0.00 | 0.66 | 3 |
| 10 | 2 | 1 | 2 | **793.75** | **793.75** | 0.00 | 23.12 | 3 | **793.75** | **793.75** | 0.00 | 9.45 | 3 | **793.75** | **793.75** | 0.00 | 10.96 | 3 |
| 10 | 2 | 2 | 1 | **803.96** | **803.96** | 0.00 | 177.59 | 3 | **803.96** | **803.96** | 0.00 | 16.73 | 3 | **803.96** | **803.96** | 0.00 | 44.80 | 3 |
| 10 | 2 | 2 | 2 | **833.78** | **833.78** | 0.00 | 110.68 | 3 | **833.78** | **833.78** | 0.00 | 75.01 | 3 | **833.78** | **833.78** | 0.00 | 436.67 | 3 |
| sum/avg (10) | | | | **768.67** | **768.67** | 0.00 | 54.38 | 18 | **768.67** | **768.67** | 0.00 | 18.12 | 18 | **768.67** | **768.67** | 0.00 | 84.45 | 18 |
| 15 | 3 | 2 | 2 | **1036.61** | **1036.61** | 0.00 | 1811.64 | 3 | **1036.61** | **1036.61** | 0.00 | 509.71 | 3 | **1036.61** | **1036.61** | 0.00 | 823.41 | 3 |
| 15 | 3 | 2 | 3 | 1014.04 | **1049.45** | 2.88 | 2054.35 | 2 | **1049.45** | **1049.45** | 0.00 | 1325.68 | 3 | 1021.59 | **1049.45** | 2.41 | 2419.87 | 1 |
| 15 | 3 | 3 | 2 | 1065.96 | **1114.68** | 4.39 | tlim | 0 | **1114.68** | **1114.68** | 0.00 | 1104.10 | 3 | 1067.19 | **1114.68** | 4.28 | 2973.83 | 1 |
| 15 | 3 | 3 | 3 | 1101.81 | **1160.32** | 5.05 | tlim | 0 | **1153.10** | 1160.74 | 0.65 | 1358.60 | 2 | 1096.70 | **1160.32** | 5.49 | tlim | 0 |
| 15 | 4 | 2 | 2 | **1036.61** | **1036.61** | 0.00 | 1822.00 | 3 | **1036.61** | **1036.61** | 0.00 | 510.81 | 3 | **1036.61** | **1036.61** | 0.00 | 825.53 | 3 |
| 15 | 4 | 2 | 3 | 1081.47 | **1084.12** | 0.28 | 2439.70 | 2 | **1084.12** | **1084.12** | 0.00 | 1108.56 | 3 | **1084.12** | **1084.12** | 0.00 | 1637.68 | 3 |
| 15 | 4 | 3 | 2 | 1073.81 | 1150.81 | 6.81 | tlim | 0 | **1142.24** | **1149.80** | 0.69 | 1420.98 | 2 | 1082.75 | 1150.81 | 6.03 | tlim | 0 |
| 15 | 4 | 3 | 3 | 1111.81 | 1203.22 | 7.72 | tlim | 0 | **1190.64** | **1202.40** | 1.03 | 1430.51 | 2 | 1113.29 | 1204.85 | 7.73 | tlim | 0 |
| sum/avg (15) | | | | 1065.27 | 1104.48 | 3.39 | 2815.96 | 10 | **1100.93** | **1104.30** | 0.30 | 1096.12 | 21 | 1067.36 | 1104.68 | 3.24 | 2435.64 | 11 |
| 20 | 2 | 2 | 2 | 1225.52 | **1275.44** | 3.87 | 3274.98 | 1 | **1243.73** | 1277.82 | 2.61 | 1354.86 | 2 | 1228.48 | **1275.44** | 3.62 | 2266.04 | 2 |
| 20 | 2 | 2 | 3 | 1262.37 | 1330.87 | 5.14 | tlim | 0 | **1294.95** | **1329.12** | 2.56 | 1458.53 | 2 | 1251.59 | 1332.91 | 6.09 | tlim | 0 |
| 20 | 3 | 2 | 2 | 1197.44 | **1269.71** | 5.60 | tlim | 0 | **1243.52** | **1269.71** | 2.02 | 1271.07 | 2 | 1236.21 | **1269.71** | 2.58 | 1531.30 | 2 |
| 20 | 3 | 2 | 3 | 1220.35 | **1301.73** | 6.13 | tlim | 0 | **1301.73** | **1301.73** | 0.00 | 1127.63 | 3 | 1253.86 | **1301.73** | 3.58 | 2510.73 | 2 |
| 20 | 3 | 3 | 2 | 1210.01 | **1289.02** | 5.94 | tlim | 0 | **1260.69** | 1296.61 | 2.60 | 2661.64 | 1 | 1215.65 | 1295.67 | 5.95 | tlim | 0 |
| 20 | 3 | 3 | 3 | 1223.85 | **1321.84** | 7.26 | tlim | 0 | **1281.30** | **1321.84** | 2.97 | tlim | 0 | 1204.51 | 1337.03 | 9.69 | tlim | 0 |
| 20 | 5 | 3 | 2 | 1200.10 | 1280.16 | 6.12 | tlim | 0 | **1241.98** | 1280.43 | 2.91 | 2986.79 | 1 | 1235.59 | **1278.32** | 3.24 | 2739.77 | 1 |
| 20 | 5 | 3 | 3 | 1228.58 | 1324.16 | 7.12 | tlim | 0 | **1270.11** | **1322.70** | 3.84 | 2863.57 | 1 | 1251.23 | 1402.59 | 9.43 | 2891.87 | 1 |
| sum/avg (20) | | | | 1221.03 | **1299.11** | 5.90 | 3559.37 | 1 | **1267.25** | 1300.00 | 2.44 | 2165.51 | 12 | 1234.64 | 1311.67 | 5.52 | 2842.59 | 8 |
| overall sum/avg | | | | 1041.02 | **1083.67** | 3.38 | 2333.13 | 29 | **1070.80** | 1083.93 | 0.99 | 1190.99 | 51 | 1046.73 | 1088.31 | 3.19 | 1942.39 | 37 |

To assess the performance of the proposed valid inequalities, six small-size instances were selected and solved running the three models with and without the addition of the valid inequalities. The results are reported in Table 4.2. We can notice that the inequalities help improve the performance of all models, by reducing the average percentage gap and execution time, and increasing the number of proven optimal solutions.

Table 4.2: Effect of valid inequalities on six small-size instances

| | without valid inequalities | | | | | with valid inequalities | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| mathematical model | $z_{lb}$ | $z_{ub}$ | %gap | t(s) | opt | $z_{lb}$ | $z_{ub}$ | %gap | t(s) | opt |
| time-based | 1036.25 | 1077.49 | 3.24 | 2484.51 | 2 | 1040.67 | 1076.85 | 2.81 | 2288.63 | 3 |
| flow-based | 1062.79 | 1076.85 | 1.00 | 953.69 | 5 | 1064.43 | 1076.85 | 0.88 | 751.28 | 5 |
| node-based | 1033.41 | 1084.99 | 3.86 | 1912.63 | 3 | 1037.93 | 1068.09 | 2.38 | 1745.36 | 4 |

### 4.6.3 ILS Parameter Tuning

The ILS procedure adopts four main parameters (i.e., $\alpha$, $\beta$, $\gamma$ and $max_{iter}$). To set their values, we randomly selected six instances (two with $0 \leq n \leq 20$, two with $50 \leq n \leq 100$, and two with $n = 200$). We then tested the ILS on these instances by attempting all possible combinations of parameter values chosen in the sets $\alpha \in \{0.05.0.10, 0.15, 0.25\}$, $\beta \in \{2, 5, 10, 20\}$, $\gamma \in \{50, 100\}$ and $max_{iter} \in \{200, 500, 1000, 5000\}$. The results are reported in Table 4.3. For each combination of parameters, column "t(s)" gives the average ILS run time on the six instances, and column "%gap" gives the average gap computed as the average over the six instances of $100(z - z^*)/z^*$. Here, $z$ is the value of the solution obtained by the given configuration and $z^*$ is the value of the best solution obtained by all configurations.

The configuration with $\alpha = 0.10$, $\beta = 5$, $\gamma = 50$ and $max_{iter} = 1000$ is the one that obtained the best results (highlighted in bold in the table). It could always achieve the best solution values, at the expense of a limited increase in the computing time with respect to configurations adopting a smaller number of iterations. This configuration was thus adopted for all successive ILS tests.

### 4.6.4 ILS Evaluation

In this section, we investigate the performance of the ILS. In Table 4.4, the results of the ILS are compared with those obtained by the best mathematical model (i.e., the flow-based one) on groups of three instances per line. We recall that column "$z_{ub}$" gives the average upper bound value, column "opt" the number of proven optimal solutions, and column "t(s)" the average run time. The ILS was executed five times on each instance. We report the best, average and worst solution values achieved, as well as their standard deviation, in columns "$z_{best}$", "$z_{avg}$", "$z_{worst}$" and "$\sigma_z$", respectively. More in detail, $z_{best}$ gives the average of the best solution values produced on the three instances, $z_{avg}$ the average of the average values, and $z_{worst}$ the average of the worst values. The average computational time is shown in column "t(s)".

According to the results, for those groups of three instances that were all solved to optimality by the flow-based model, the ILS obtained the same optimal values in a shorter computational time. For all the remaining small-size sets, the ILS achieved

Table 4.3: ILS parameter tuning. Best configuration in **boldface**

| | $(\gamma, max_{iter})$ | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | (50, 200) | | (50, 500) | | (50, 1000) | | (50, 5000) | | (100, 200) | | (100, 500) | | (100, 1000) | | (100, 5000) | |
| $(\alpha, \beta)$ | t(s) | %gap | t(s) | %gap | t(s) | %gap | t(s) | %gap | t(s) | %gap | t(s) | %gap | t(s) | %gap | t(s) | %gap |
| (0.05,2) | 1.53 | 0.91 | 1.79 | 0.87 | 2.13 | 0.84 | 2.79 | 0.82 | 1.56 | 0.83 | 1.80 | 0.79 | 1.89 | 0.78 | 3.28 | 0.77 |
| (0.05,5) | 1.67 | 0.76 | 1.83 | 0.75 | 2.27 | 0.73 | 2.91 | 0.72 | 1.79 | 0.75 | 1.90 | 0.74 | 2.02 | 0.74 | 3.49 | 0.73 |
| (0.05,10) | 1.91 | 0.76 | 2.18 | 0.74 | 2.66 | 0.73 | 3.41 | 0.72 | 2.08 | 0.73 | 2.19 | 0.73 | 2.26 | 0.73 | 3.91 | 0.71 |
| (0.05,20) | 1.73 | 0.76 | 1.93 | 0.74 | 2.34 | 0.73 | 2.86 | 0.72 | 2.20 | 0.73 | 2.35 | 0.72 | 2.43 | 0.69 | 4.67 | 0.69 |
| (0.10,2) | 2.09 | 0.08 | 2.33 | 0.03 | 2.68 | 0.02 | 3.58 | 0.01 | 1.91 | 0.13 | 1.97 | 0.11 | 2.09 | 0.10 | 2.58 | 0.10 |
| (0.10,5) | 2.74 | 0.08 | 3.25 | 0.02 | **4.02** | **0.00** | 5.44 | 0.00 | 2.06 | 0.11 | 2.38 | 0.07 | 2.89 | 0.06 | 3.28 | 0.05 |
| (0.10,10) | 3.13 | 0.08 | 4.06 | 0.02 | 5.04 | 0.00 | 6.47 | 0.00 | 2.49 | 0.07 | 2.61 | 0.07 | 3.05 | 0.06 | 3.39 | 0.05 |
| (0.10,20) | 3.57 | 0.08 | 4.53 | 0.02 | 5.23 | 0.00 | 8.02 | 0.00 | 2.84 | 0.07 | 3.11 | 0.06 | 3.24 | 0.06 | 3.46 | 0.05 |
| (0.15,2) | 1.83 | 0.43 | 2.06 | 0.39 | 2.30 | 0.38 | 3.12 | 0.35 | 2.04 | 0.38 | 2.37 | 0.35 | 2.49 | 0.35 | 3.12 | 0.35 |
| (0.15,5) | 2.49 | 0.40 | 2.86 | 0.38 | 3.13 | 0.35 | 5.08 | 0.35 | 2.33 | 0.36 | 2.59 | 0.35 | 3.20 | 0.34 | 4.85 | 0.33 |
| (0.15,10) | 3.35 | 0.40 | 3.88 | 0.38 | 4.16 | 0.35 | 6.37 | 0.35 | 2.48 | 0.35 | 3.79 | 0.33 | 4.11 | 0.33 | 5.09 | 0.33 |
| (0.15,20) | 4.55 | 0.40 | 5.02 | 0.38 | 5.23 | 0.35 | 8.64 | 0.35 | 2.71 | 0.35 | 4.26 | 0.33 | 4.82 | 0.33 | 5.94 | 0.32 |
| (0.25,2) | 2.25 | 1.34 | 2.64 | 1.07 | 3.30 | 0.94 | 4.56 | 0.92 | 2.21 | 0.88 | 2.27 | 0.86 | 2.84 | 0.86 | 4.19 | 0.86 |
| (0.25,5) | 2.54 | 1.18 | 2.93 | 0.91 | 4.05 | 0.89 | 5.62 | 0.89 | 2.68 | 0.86 | 3.16 | 0.85 | 3.74 | 0.85 | 4.80 | 0.83 |
| (0.25,10) | 3.00 | 1.16 | 3.94 | 0.90 | 4.94 | 0.86 | 7.33 | 0.86 | 3.52 | 0.83 | 4.86 | 0.82 | 6.07 | 0.82 | 7.83 | 0.82 |
| (0.25,20) | 3.72 | 1.16 | 5.12 | 0.90 | 6.31 | 0.86 | 8.89 | 0.86 | 4.67 | 0.83 | 6.13 | 0.82 | 6.63 | 0.82 | 8.46 | 0.82 |

better values than the flow-based model (without proof of their optimality). In addition, the constantly null average standard deviation among the different runs indicates the robustness of the algorithm on these very simple instances. When comparing the average run times, we can notice that the ILS needed an overall average time of just 0.23 seconds against the 1,190.99 seconds of the flow-based model.

In Table 4.5, we report the results of the ILS on medium- and large-size instances. On instances having $|V_1 \cup V_2| = 50$ the average standard deviation is 0.00, on those having $|V_1 \cup V_2| = 100$ it becomes 0.50, while on those having $|V_1 \cup V_2| = 200$ it increases to 0.92, thus resulting in an overall average standard deviation of 0.47. This confirms the robustness of the algorithm. Concerning the run time, the ILS took on average 1.91 seconds to solve instances having $|V_1 \cup V_2| = 50$, 8.60 seconds for those having $|V_1 \cup V_2| = 100$, and 13.02 seconds for those having $|V_1 \cup V_2| = 200$. The overall average run time is 7.84 seconds, proving that the method is suitable for a quick use in practical situations.

Finally, Table 4.6 reports a sensitivity analysis on the average percentage of computational time needed by each ILS component, grouped by set of instances. On the small-size sets, LS2 and LS3 are the most time-consuming local search procedures, while for medium- and large-size sets the largest effort is required by LS1 and LS4.

### 4.6.5 Results on Realistic Instances

The flow-based model and the ILS were also tested on a set of realistic instances generated from the WDN in the city of Mashhad (Iran). Our real case study consists of 3,124 households/shops, 293 reservoirs/tanks, 356 wells and 14 treatment plants. For all of these nodes the exact locations were collected.

Following the same rationale described in Section 4.6.1, we generated 108 realistic instances divided into two sets of *small-size* and *medium- and large-size* instances, each comprising different subsets having homogeneous values of $|V_1 \cup V_2|$, $(|V_2|, |V_3|)$, and $|K|$, and composed by three random instances per subset. The resulting sets are:

Table 4.4: Computational results on small-size instances (three inst. per line)

| | | | | flow-based | | | ILS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\lvert V_1 \cup V_2\rvert$ | $\lvert V_2\rvert$ | $\lvert V_3\rvert$ | $\lvert K\rvert$ | $z_{ub}$ | t(s) | opt | $z_{best}$ | $z_{avg}$ | $z_{worst}$ | $\sigma_z$ | t(s) |
| 10 | 1 | 1 | 1 | 706.39 | 1.13 | 3 | 706.39 | 706.39 | 706.39 | 0.00 | 0.00 |
| 10 | 1 | 1 | 2 | 730.91 | 3.26 | 3 | 730.91 | 730.91 | 730.91 | 0.00 | 0.00 |
| 10 | 2 | 1 | 1 | 743.25 | 3.13 | 3 | 743.25 | 743.25 | 743.25 | 0.00 | 0.00 |
| 10 | 2 | 1 | 2 | 793.75 | 9.45 | 3 | 793.75 | 793.75 | 793.75 | 0.00 | 0.00 |
| 10 | 2 | 2 | 1 | 803.96 | 16.73 | 3 | 803.96 | 803.96 | 803.96 | 0.00 | 0.00 |
| 10 | 2 | 2 | 2 | 833.78 | 75.01 | 3 | 833.78 | 833.78 | 833.78 | 0.00 | 0.00 |
| sum/avg (10) | | | | 768.67 | 18.12 | 18 | 768.67 | 768.67 | 768.67 | 0.00 | 0.00 |
| 15 | 3 | 2 | 2 | 1036.61 | 509.71 | 3 | 1036.61 | 1036.61 | 1036.61 | 0.00 | 0.14 |
| 15 | 3 | 2 | 3 | 1049.45 | 1325.68 | 3 | 1049.45 | 1049.45 | 1049.45 | 0.00 | 0.22 |
| 15 | 3 | 3 | 2 | 1114.68 | 1104.10 | 3 | 1114.68 | 1114.68 | 1114.68 | 0.00 | 0.14 |
| 15 | 3 | 3 | 3 | 1160.74 | 1358.60 | 2 | 1155.96 | 1155.96 | 1155.96 | 0.00 | 0.24 |
| 15 | 4 | 2 | 2 | 1036.61 | 510.81 | 3 | 1036.61 | 1036.61 | 1036.61 | 0.00 | 0.16 |
| 15 | 4 | 2 | 3 | 1084.12 | 1108.56 | 3 | 1084.12 | 1084.12 | 1084.12 | 0.00 | 0.22 |
| 15 | 4 | 3 | 2 | 1149.80 | 1420.98 | 2 | 1146.56 | 1146.56 | 1146.56 | 0.00 | 0.20 |
| 15 | 4 | 3 | 3 | 1202.40 | 1430.51 | 2 | 1202.40 | 1202.40 | 1202.40 | 0.00 | 0.27 |
| sum/avg (15) | | | | 1104.30 | 1096.12 | 21 | 1103.30 | 1103.30 | 1103.30 | 0.00 | 0.20 |
| 20 | 2 | 2 | 2 | 1277.82 | 1354.86 | 2 | 1275.44 | 1275.44 | 1275.44 | 0.00 | 0.26 |
| 20 | 2 | 2 | 3 | 1329.12 | 1458.53 | 2 | 1316.03 | 1316.03 | 1316.03 | 0.00 | 0.33 |
| 20 | 3 | 2 | 2 | 1269.71 | 1271.07 | 2 | 1262.52 | 1262.52 | 1262.52 | 0.00 | 0.35 |
| 20 | 3 | 2 | 3 | 1301.73 | 1127.63 | 3 | 1301.73 | 1301.73 | 1301.73 | 0.00 | 0.41 |
| 20 | 3 | 3 | 2 | 1296.61 | 2661.64 | 1 | 1277.25 | 1277.25 | 1277.25 | 0.00 | 0.35 |
| 20 | 3 | 3 | 3 | 1321.84 | tlim | 0 | 1301.37 | 1301.37 | 1301.37 | 0.00 | 0.61 |
| 20 | 5 | 3 | 2 | 1280.43 | 2986.79 | 1 | 1265.46 | 1265.46 | 1265.46 | 0.00 | 0.33 |
| 20 | 5 | 3 | 3 | 1322.70 | 2863.57 | 1 | 1303.93 | 1303.93 | 1303.93 | 0.00 | 0.73 |
| sum/avg (20) | | | | 1300.00 | 2165.51 | 12 | 1287.97 | 1287.97 | 1287.97 | 0.00 | 0.42 |
| overall sum/avg | | | | 1083.93 | 1190.99 | 51 | 1079.19 | 1079.19 | 1079.19 | 0.00 | 0.23 |

- *Small-size*: 12 instances with $|V_1 \cup V_2|$=10, $(|V_2|, |V_3|) \in \{(1,1),\ (2,1),\ (2,2)\}$, and $|K| \in \{1,2\}$; 12 instances with $|V_1 \cup V_2|$=15, $(|V_2|, |V_3|) \in \{(1,1),\ (2,1),\ (2,2)\}$, and $|K| \in \{1,2,3\}$; 12 instances with $|V_1 \cup V_2|$=20, $(|V_2|, |V_3|) \in \{(1,1),\ (2,1),\ (2,2)\}$, and $|K| \in \{2,3\}$;

- *Medium- and large-size*: 12 instances with $|V_1 \cup V_2|$=40, $(|V_2|, |V_3|) \in \{(4,2),\ (4,3),\ (6,2),\ (6,3)\}$, and $|K| \in \{2,3\}$; 12 instances with $|V_1 \cup V_2|$=50, $(|V_2|, |V_3|) \in \{(4,2),\ (4,3),\ (6,2),\ (6,3)\}$, and $|K| \in \{2,3\}$; 12 instances with $|V_1 \cup V_2|$=60, $(|V_2|, |V_3|) \in \{(4,2),\ (4,3),\ (6,2),\ (6,3)\}$, and $|K| \in \{2,3\}$; 12 instances with $|V_1 \cup V_2|$=100, $(|V_2|, |V_3|) \in \{(8,4),\ (8,5),\ (10,4),\ (10,5)\}$, and $|K| \in \{4,5\}$; 12 instances with $|V_1 \cup V_2|$=150, $(|V_2|, |V_3|) \in \{(8,4),\ (8,5),\ (10,4),\ (10,5)\}$, and $|K| \in \{4,5\}$; 12 instances with $|V_1 \cup V_2|$=200, $(|V_2|, |V_3|) \in \{(8,4),\ (8,5),\ (10,4),\ (10,5)\}$, and $|K| \in \{4,5\}$.

For each instance, the coordinates of the nodes were randomly selected among the given real locations. The flow-based model and the ILS were used to run the experiments. The results are reported in the Tables 4.7 and 4.8. In Table 4.7, the results of the ILS are compared with those obtained by the flow-based model. We recall that columns "$z_{ub}$", "t(s)" and "opt" give the average upper bound value, the average run time and the total number of instances solved to proven optimality by the mathematical model, respectively. Note that an entry "tlim" indicates that the time limit of 3,600 CPU seconds was reached for all the three instances in the group.

Table 4.5: Computational results on medium- and large-size instances (three inst. per line)

| | | | | ILS | | | | |
|---|---|---|---|---|---|---|---|---|
| $\lvert V_1 \cup V_2\rvert$ | $\lvert V_2\rvert$ | $\lvert V_3\rvert$ | $\lvert K\rvert$ | $z_{best}$ | $z_{avg}$ | $z_{worst}$ | $\sigma_z$ | t(s) |
| 50 | 5 | 5 | 5 | 2583.27 | 2583.27 | 2583.27 | 0.00 | 1.17 |
| 50 | 5 | 5 | 8 | 2721.90 | 2721.90 | 2721.90 | 0.00 | 1.59 |
| 50 | 8 | 8 | 5 | 2883.07 | 2883.07 | 2883.07 | 0.00 | 1.67 |
| 50 | 8 | 8 | 8 | 3001.70 | 3001.70 | 3001.70 | 0.00 | 2.10 |
| 50 | 10 | 5 | 5 | 2664.02 | 2664.02 | 2664.02 | 0.00 | 2.09 |
| 50 | 10 | 5 | 8 | 2807.41 | 2807.41 | 2807.41 | 0.00 | 2.19 |
| 50 | 10 | 8 | 5 | 2863.64 | 2863.64 | 2863.64 | 0.00 | 1.91 |
| 50 | 10 | 8 | 8 | 3003.07 | 3003.07 | 3003.07 | 0.00 | 2.52 |
| avg (50) | | | | 2816.01 | 2816.01 | 2816.01 | 0.00 | 1.91 |
| 100 | 5 | 5 | 10 | 4430.65 | 4430.92 | 4431.53 | 0.40 | 7.61 |
| 100 | 5 | 5 | 15 | 4642.24 | 4642.45 | 4643.13 | 0.39 | 8.29 |
| 100 | 10 | 5 | 10 | 4507.07 | 4507.27 | 4508.07 | 0.45 | 7.19 |
| 100 | 10 | 5 | 15 | 4750.73 | 4750.92 | 4751.65 | 0.41 | 8.17 |
| 100 | 10 | 10 | 10 | 4856.94 | 4857.16 | 4857.99 | 0.47 | 9.03 |
| 100 | 10 | 10 | 15 | 5062.41 | 5062.62 | 5063.43 | 0.45 | 9.43 |
| 100 | 15 | 10 | 10 | 4826.28 | 4826.62 | 4827.96 | 0.75 | 9.18 |
| 100 | 15 | 10 | 15 | 5070.19 | 5070.50 | 5071.74 | 0.69 | 9.90 |
| avg (100) | | | | 4768.31 | 4768.56 | 4769.44 | 0.50 | 8.60 |
| 200 | 10 | 10 | 15 | 8244.39 | 8244.97 | 8246.12 | 0.82 | 9.86 |
| 200 | 10 | 10 | 20 | 8636.53 | 8637.11 | 8638.33 | 0.84 | 10.34 |
| 200 | 20 | 10 | 15 | 8550.63 | 8551.32 | 8552.62 | 0.98 | 12.27 |
| 200 | 20 | 10 | 20 | 8814.41 | 8815.00 | 8816.05 | 0.82 | 13.29 |
| 200 | 20 | 20 | 15 | 9128.90 | 9129.63 | 9130.63 | 0.82 | 13.66 |
| 200 | 20 | 20 | 20 | 9305.35 | 9305.98 | 9307.13 | 0.81 | 14.96 |
| 200 | 30 | 20 | 15 | 9372.60 | 9373.86 | 9375.17 | 1.16 | 14.05 |
| 200 | 30 | 20 | 20 | 9497.20 | 9498.20 | 9499.67 | 1.10 | 15.70 |
| avg (200) | | | | 8943.75 | 8944.51 | 8945.72 | 0.92 | 13.02 |
| overall avg | | | | 5509.36 | 5509.69 | 5510.39 | 0.47 | 7.84 |

Conversely, columns "$z_{best}$", "$z_{avg}$", "$z_{worst}$", "$\sigma_z$" and "t(s)" give the best, average and worst solution values, the standard deviation and the computational time of the ILS, respectively. We can notice that on small-size instances, the flow-based model and the ILS obtained the same optimal values on instances having $|V_1 \cup V_2| \in \{10, 15\}$, and on one subset out of four of instances having $|V_1 \cup V_2| = 20$. For the remaining subsets, the ILS achieved better values than the flow-based model (again, without proof of their optimality).

On medium- and large-size instances, the ILS achieved very robust results on instances having $|V_1 \cup V_2| \in \{40, 50, 60\}$. Indeed, the standard deviation is constantly null for all the subgroups, and the run times are very short. The robustness of the ILS slightly decreases for instances having $|V_1 \cup V_2| \in \{100, 150, 200\}$, however remaining acceptable for a practical use. For these instances, the average run times are around 7.32, 9.80 and 11.08 seconds, respectively, thus confirming that the algorithm could

Table 4.6: Percentage of the computational time needed by each ILS component

| Set | LS1 | LS2 | LS3 | LS4 | LS5 | S1 | S2 |
|---|---|---|---|---|---|---|---|
| Small-size | 3.63% | 39.41% | 28.65% | 6.37% | 20.50% | 0.37% | 1.07% |
| Medium-size | 59.50% | 9.02% | 1.30% | 18.36% | 10.69% | 0.23% | 0.89% |
| Large-size | 50.03% | 3.48% | 3.33% | 32.35% | 9.98% | 0.13% | 0.71% |

Table 4.7: Computational results on realistic small-size instances (three inst. per line)

| | | | | flow-based | | | ILS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\|V_1 \cup V_2\|$ | $\|V_2\|$ | $\|V_3\|$ | $\|K\|$ | $z_{ub}$ | t(s) | opt | $z_{best}$ | $z_{avg}$ | $z_{worst}$ | $\sigma_z$ | t(s) |
| 10 | 1 | 1 | 1 | 82475.75 | 12.69 | 3 | 82475.75 | 82475.75 | 82475.75 | 0.00 | 0.00 |
| 10 | 1 | 1 | 2 | 80629.79 | 8.95 | 3 | 80629.79 | 80629.79 | 80629.79 | 0.00 | 0.00 |
| 10 | 2 | 1 | 1 | 121874.44 | 5.53 | 3 | 121874.44 | 121874.44 | 121874.44 | 0.00 | 0.00 |
| 10 | 2 | 2 | 2 | 116649.98 | 135.45 | 3 | 116649.98 | 116649.98 | 116649.98 | 0.00 | 0.00 |
| sum/avg (10) | | | | 100407.49 | 40.66 | 12 | 100407.49 | 100407.49 | 100407.49 | 0.00 | 0.00 |
| 15 | 1 | 1 | 1 | 97874.84 | 1218.80 | 2 | 97874.84 | 97874.84 | 97874.84 | 0.00 | 0.05 |
| 15 | 1 | 1 | 2 | 186943.98 | 46.03 | 3 | 186943.98 | 186943.98 | 186943.98 | 0.00 | 0.08 |
| 15 | 2 | 1 | 2 | 126742.54 | 1204.46 | 2 | 126742.54 | 126742.54 | 126742.54 | 0.00 | 0.12 |
| 15 | 2 | 2 | 3 | 135254.03 | 2437.79 | 1 | 135254.03 | 135254.03 | 135254.03 | 0.00 | 0.12 |
| sum/avg (15) | | | | 136703.85 | 1226.77 | 8 | 136703.85 | 136703.85 | 136703.85 | 0.00 | 0.09 |
| 20 | 1 | 1 | 2 | 158682.88 | 1205.87 | 2 | 158588.73 | 158588.73 | 158588.73 | 0.00 | 0.18 |
| 20 | 1 | 1 | 3 | 175830.70 | 3342.99 | 1 | 175655.43 | 175655.43 | 175655.43 | 0.00 | 0.20 |
| 20 | 2 | 1 | 2 | 170015.90 | 1466.86 | 3 | 170015.90 | 170015.90 | 170015.90 | 0.00 | 0.20 |
| 20 | 2 | 2 | 3 | 188935.98 | tlim | 0 | 187949.09 | 187949.09 | 187949.09 | 0.00 | 0.26 |
| sum/avg (20) | | | | 173366.36 | 2403.93 | 6 | 173052.29 | 173052.29 | 173052.29 | 0.00 | 0.21 |
| overall sum/avg | | | | 136825.90 | 1223.79 | 26 | 136721.21 | 136721.21 | 136721.21 | 0.00 | 0.10 |

efficiently solve realistic instances having a considerable number of nodes in a few seconds.

## 4.7 Conclusions

In this paper, we introduced a generalization of the well-known Vehicle Routing Problem (VRP), called VRP for Water Distribution Networks (VRPWDN), that includes precedence constraints among nodes and multiple visits to some of the nodes. The problem is NP-hard in the strong sense and, to the best of our knowledge, has not yet been applied in the context of distribution networks where regular inspections have to be performed to detect potential sources of contamination. To solve the VRPWDN, three alternative mathematical models (time-based, flow-based and node-based) were proposed, and an Iterated Local Search (ILS) algorithm was developed.

Extensive computational tests on randomly generated small-size instances were performed to compare the performance of the three mathematical models, showing that the flow-based model outperforms the other two in terms of solution quality and speed. On the same instances, the accuracy of the ILS in finding good-quality solutions in a short time was proved. The ILS was also used to perform a series of tests on randomly generated medium- and large-size instances with up to 200 nodes, confirming its efficacy and robustness.

Additional computational tests were executed on small-, medium-, and large-size realistic instances derived from the Mashhad (Iran) distribution network, proving that our methods can be applied with profit even in a practical case.

Interesting future research directions include the application of the developed techniques to other related VRPs with precedence constraints and multiple visits. In addition, we are interested in studying the generalization of the VRPWDN to the case of multiple periods. In this generalization, one should first of all determine in which day

Table 4.8: Comp. results on realistic medium- and large-size instances (three inst. per line)

| | | | | ILS | | | | |
|---|---|---|---|---|---|---|---|---|
| $\lvert V_1 \cup V_2 \rvert$ | $\lvert V_2 \rvert$ | $\lvert V_3 \rvert$ | $\lvert K \rvert$ | $z_{best}$ | $z_{avg}$ | $z_{worst}$ | $\sigma_z$ | t(s) |
| 40 | 4 | 2 | 2 | 180613.80 | 180613.80 | 180613.80 | 0.00 | 0.78 |
| 40 | 4 | 3 | 3 | 219227.54 | 219227.54 | 219227.54 | 0.00 | 0.79 |
| 40 | 6 | 2 | 2 | 195113.71 | 195113.71 | 195113.71 | 0.00 | 0.85 |
| 40 | 6 | 3 | 3 | 201338.25 | 201338.25 | 201338.25 | 0.00 | 0.84 |
| avg (40) | | | | 199073.33 | 199073.33 | 199073.33 | 0.00 | 0.82 |
| 50 | 4 | 2 | 2 | 250479.23 | 250479.23 | 250479.23 | 0.00 | 0.97 |
| 50 | 4 | 3 | 3 | 278476.01 | 278476.01 | 278476.01 | 0.00 | 0.97 |
| 50 | 6 | 2 | 2 | 263769.77 | 263769.77 | 263769.77 | 0.00 | 1.18 |
| 50 | 6 | 3 | 3 | 293179.73 | 293179.73 | 293179.73 | 0.00 | 1.28 |
| avg (50) | | | | 271476.19 | 271476.19 | 271476.19 | 0.00 | 1.10 |
| 60 | 4 | 2 | 2 | 263976.92 | 263976.92 | 263976.92 | 0.00 | 1.67 |
| 60 | 4 | 3 | 3 | 264029.28 | 264029.28 | 264029.28 | 0.00 | 1.73 |
| 60 | 6 | 2 | 2 | 222473.25 | 222473.25 | 222473.25 | 0.00 | 2.07 |
| 60 | 6 | 3 | 3 | 291979.18 | 291979.18 | 291979.18 | 0.00 | 2.35 |
| avg (60) | | | | 260614.66 | 260614.66 | 260614.66 | 0.00 | 1.96 |
| 100 | 8 | 4 | 4 | 399442.47 | 399442.63 | 399443.14 | 0.30 | 6.41 |
| 100 | 8 | 5 | 5 | 438923.85 | 438923.90 | 438924.05 | 0.09 | 7.38 |
| 100 | 10 | 4 | 4 | 344216.84 | 344217.11 | 344217.64 | 0.38 | 7.72 |
| 100 | 10 | 5 | 5 | 376473.23 | 376473.42 | 376473.86 | 0.28 | 7.78 |
| avg (100) | | | | 389764.10 | 389764.27 | 389764.67 | 0.26 | 7.32 |
| 150 | 8 | 4 | 4 | 438156.93 | 438157.42 | 438158.64 | 0.72 | 9.64 |
| 150 | 8 | 5 | 5 | 440416.07 | 440416.44 | 440417.48 | 0.61 | 8.85 |
| 150 | 10 | 4 | 4 | 466864.57 | 466864.99 | 466866.19 | 0.70 | 10.54 |
| 150 | 10 | 5 | 5 | 569988.53 | 569988.92 | 569990.11 | 0.68 | 10.17 |
| avg (150) | | | | 478856.53 | 478856.94 | 478858.11 | 0.68 | 9.80 |
| 200 | 8 | 4 | 4 | 491986.12 | 491986.81 | 491988.10 | 0.86 | 10.76 |
| 200 | 8 | 5 | 5 | 495553.12 | 495553.59 | 495555.14 | 0.88 | 9.82 |
| 200 | 10 | 4 | 4 | 646227.25 | 646227.71 | 646228.81 | 0.71 | 11.93 |
| 200 | 10 | 5 | 5 | 696373.63 | 696374.24 | 696375.29 | 0.73 | 11.80 |
| avg (200) | | | | 582535.03 | 582535.58 | 582536.83 | 0.79 | 11.08 |
| overall avg | | | | 363719.97 | 363720.16 | 363720.63 | 0.29 | 5.35 |

inspecting the given nodes, and then creating the routes for each day.

# Chapter 5

# Smart-Meter Installation Scheduling Project in the Context of Water Distribution

Baschieri, D., Iori, M., Magni, C.A., Marchioni, A., Vezzali, D. (2021) Smart-Meter Installation Scheduling in the Context of Water Distribution. *31st European Conference on Operational Research (EURO 2021)*, Athens, Greece, July 11-14.

## Abstract

In this work, we propose a mixed integer linear programming (MILP) formulation to model a Smart-Meter Installation Scheduling Project (SMISP) in the context of water distribution. The model is intended to solve a real case study from *IRETI*, a multi-utility company operating in the Italian water distribution sector. Specifically, in compliance with the European and the Italian regulations on metering, a distribution company is required to periodically control meters and substitute them in case they have reached their lifespan. In the examined case study, *IRETI* has opted for a massive substitution plan to install innovative "walk-by" smart meters in place of traditional mechanical meters. The MILP formulation aims at integrating both the operational and the financial perspective of the SMISP. In particular, the objective function has been carefully defined to maximize the net present value (NPV) of the massive substitution plan, including the annual conditional cost savings obtained by the introduction of "walk-by" smart meters, some additional revenues established by ARERA, the Italian Authority that regulates the water distribution sector, the capital expenditures for the installation of "walk-by" smart meters, and the depreciation charges. The final goal of the proposed formulation is to define the optimal schedule for the massive substitution plan, such that both the financial and the operational constraints are satisfied and the NPV is maximized.

## 5.1 Introduction

Smart metering systems allow multi-utility companies to perform readings and manage electrical, gas and water meters from remote. As such, these systems produce a number of advantages. Among these, we mention (i) the reduced operational costs in the process of collecting readings, (ii) the increased frequency of readings, (iii) the increased availability of data for analytics, (iv) the increased capability of multi-utility companies in monitoring distribution networks and detecting potential leakages, (v) the increased awareness of customers for what concerns their energy consumption.

Scheduling projects are very diffused in service industries, as they deal with the allocation of tasks to resources over given time periods with the goal of optimizing one or more objectives [138]. In this work, we focus on the smart-meter installation scheduling project (SMISP) in the context of water distribution, which consists in determining the optimal schedule of smart meter installations to maximize the net present value (NPV) of the project. Such an objective is obtained by combining both operational and financial parameters. Note that an installation is to replace a traditional mechanical meter with a smart electronic device.

In the following, we present a mixed integer linear programming (MILP) formulation which was defined to solve the particular SMISP arising in the province of Reggio Emilia. A representation of the project at hand is provided in Figure 5.1, where the smart meters to install were approximately 46,000. These data were provided by *IRETI*, a multi-utility company operating in the Italian water distribution sector. We also highlight that not all the municipalities were involved in the project, as we were only interested in the installation of "walk-by" smart meters. Indeed, the remaining municipalities required an alternative smart metering technology, which was out of the scope of this work.

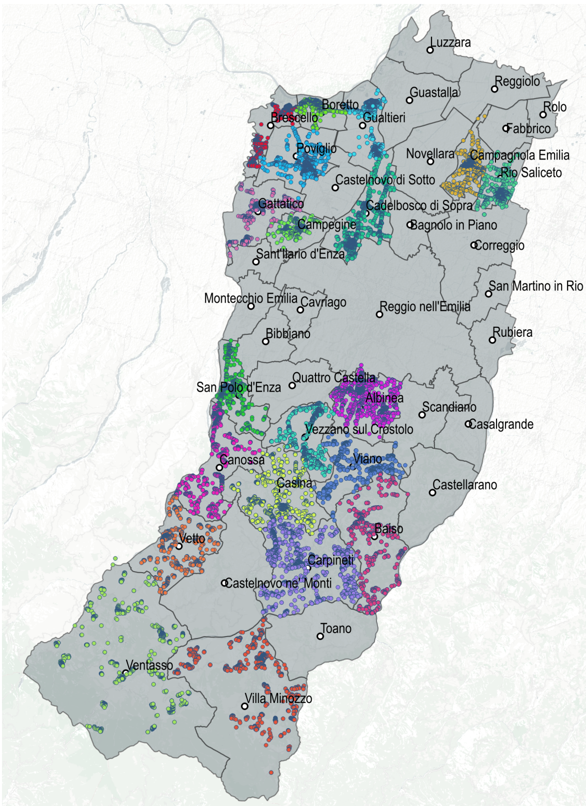



Figure 5.1: Smart meters to install in the province of Reggio Emilia

The research was motivated by a regulation on measurement instruments issued by the Italian Ministry of Economic Development, which establishes a periodic check of mechanical water meters (i.e., every 10 years) and imposes their substitution in case they have reached their lifespan.

## 5.2 Problem Definition

The SMISP is formally defined in the following.

### 5.2.1 Operational Parameters

We are given a set $J$ of meter groups with $N_j$ traditional meters to substitute; note that the reading of meters being part of the same group is performed in the same period and, as established by ARERA, the Italian Authority that regulates the market, each meter must be read two or three times per year. Then, we are given a set $T$ of time intervals, and a set $K$ of substitution squads, which can install a maximum of $Q$ smart meters per time interval. Additional parameters are represented by $b_{jt}$, which takes value 1 if substitutions can occur in meter group $j$ during time interval $t$ and 0 otherwise, and $\sigma$, which corresponds to the maximum number of meters groups each squad $k$ can work in a time interval. The first parameter has been imposed by the multi-utility company to avoid the concurrence of readings and substitutions, while the latter allows a substitution squad to work in more than a single meter group during the same time interval. Finally, we define $S_{jt}$ as the conditional cost savings obtained for meter group $j$ during time interval $t$ if readings are collected by means of smart meters instead of traditional meters.

### 5.2.2 Accounting and Financial Parameters

We define $S_j$ as the annual conditional cost savings obtained for meter group $j$ once all the substitutions have been completed, and as the investment cost incurred by *IRETI* to buy and install a single smart meter. In addition, $P$ represents the number of periods (i.e., years) during which the project occurs, while $DH$ corresponds to the number of years as regards the depreciation horizon. Note that we use a straight-line method to depreciate fixed assets. Further parameters are represented by $r$, as the annual cost of capital, and $\gamma$, as the annual tax rate. The project duration $P$ is estimated using the following formula:

$$P = SP + (DH + 1)$$

where $SP$ limits the *operational horizon* during which the substitutions can be scheduled. $SP$ is fixed to 3 years and it is an input for the problem. Consequently, $T_{SP}$ indicates the set of time intervals within $SP$.

### 5.2.3 Variables

Let $x_{jkt}$ be an integer variable corresponding to the number of smart meters installed in meter group $j$ by substitution squad $k$ during time interval $t$, $y_{jkt}$ be a binary variable taking value 1 if meter group $j$ is worked by squad $k$ during time interval $t$ and 0 otherwise, $\overline{y}_{jt}$ be another binary variable taking value 1 if installations in meter group $j$ are completed during time interval $t$ and 0 otherwise, and $z_{jt}$ be a binary variable taking value 1 if meter group $j$ is already smart during time interval $t$ and 0 otherwise.

We also define $S_p$ as the conditional cost savings, $R_p$ as the additional revenues defined by ARERA [12], $X_p$ as the capital expenditures, and $D_p$ as the depreciation charges. All these support variables are defined for each period $p$.

### 5.2.4 Mathematical Formulation

We define:

$$\text{NPV} = \sum_{p=0}^{P} \frac{F_p}{(1+r)^p}$$

where

$$\begin{aligned} F_p &= (1-\gamma)(S_p + R_p - D_p) - (X_p - D_p) \\ &= (1-\gamma)(S_p + R_p) - X_p + \gamma D_p \end{aligned}$$

is the estimated cash flow of a single period $p$, comprising the impact of income taxes. Therefore, our SMISP can be formulated as follows:

$$\text{(SMISP)} \qquad \max \ \text{NPV} = \sum_{p=0}^{P} \frac{1}{(1+r)^p}[(1-\gamma)(S_p + R_p) - X_p + \gamma D_p] \tag{5.1}$$

subject to

$$\sum_{t \in T_{SP}} \overline{y}_{jt} = 1 \qquad j \in J \tag{5.2}$$

$$z_{jt} \leq \sum_{\tau=1}^{t-1} \overline{y}_{j\tau} \qquad j \in J, t \in T_{SP} \tag{5.3}$$

$$N_j \overline{y}_{jt} \leq \sum_{k \in K} \sum_{\tau=1}^{t} x_{jk\tau} \qquad j \in J, t \in T_{SP} \tag{5.4}$$

$$\sum_{j \in J} x_{jkt} \leq Q \qquad k \in K, t \in T_{SP} \tag{5.5}$$

$$\sum_{j \in J} y_{jkt} \leq \sigma \qquad k \in K, t \in T_{SP} \tag{5.6}$$

$$x_{jkt} \leq \min\{N_j, Q\} y_{jkt} \qquad j \in J, k \in K, t \in T_{SP} \tag{5.7}$$

$$\sum_{k \in K} y_{jkt} \leq b_{jt}(1 - \sum_{\tau=1}^{t-1} \overline{y}_{j\tau}) \qquad j \in J, t \in T_{SP} \tag{5.8}$$

$$S_p = \sum_{j \in J} \sum_{t \in T_{SP}} S_{jt} z_{jt} \qquad p = 0, \ldots, P : p < SP \tag{5.9}$$

$$S_p = \sum_{j \in J} S_j \qquad p = 0, \ldots, P : SP \leq p \leq P - 2 \tag{5.10}$$

$$S_{P-1} = S_P = 0 \tag{5.11}$$

$$X_p = C \sum_{j \in J} \sum_{k \in K} \sum_{t \in T_{SP}} x_{jkt} \qquad p = 0, \ldots, P : p < SP \tag{5.12}$$

$$X_p = 0 \qquad p = 0, \ldots, P : p \geq SP \tag{5.13}$$

$$D_0 = 0 \tag{5.14}$$

$$D_p = \frac{1}{DH} \sum_{\varphi = \max\{0, p-DH\}}^{p-1} X_\varphi \qquad p = 0, \ldots, P : p > 0 \tag{5.15}$$

$$R_0 = R_1 = 0 \tag{5.16}$$

$$R_2 = rX_0 \tag{5.17}$$

$$R_p = D_{p-2} + r \sum_{\varphi=0}^{p-2} (X_\varphi - D_\varphi) \qquad p = 3, \ldots, P \tag{5.18}$$

$$\overline{y}_{jt} \in \{0, 1\} \qquad j \in J, t \in T_{SP} \tag{5.19}$$

$$z_{jt} \in \{0, 1\} \qquad j \in J, t \in T_{SP} \tag{5.20}$$

$$y_{jkt} \in \{0, 1\} \qquad j \in j, k \in K, t \in T_{SP} \tag{5.21}$$

$$x_{jkt} \geq 0, \text{integer} \qquad j \in j, k \in K, t \in T_{SP} \tag{5.22}$$

The objective function (5.1) maximizes the NPV, based on the conditional cost savings obtained by the introduction of "walk-by" smart meters, the additional revenues established by ARERA, the total capital expenditures for the installation of "walk-by" smart meters, and the depreciation charges. The cash flows are discounted considering the annual cost of capital $r$. Constraints (5.2) impose that all the substitutions have to be completed within $|T_{SP}|$ years. On the other hand, constraints (5.3) define the condition such that a meter group is considered "smart", while constraints (5.4) express the condition to complete the installations for each meter group $j$. According to constraints (5.5) and constraints (5.6), respectively, the capacity per time interval of each squad $k$ is limited by $Q$ and a single squad $k$ can work in a maximum of $\sigma$ meter groups during the same time interval $t$. Constraints (5.7) establish the connection between variables $x_{jkt}$ and variables $y_{jkt}$, while constraints (5.8) impose that substitutions can occur in meter group $j$ during time interval $t$ only if readings are not performed. Additional constraints (5.9)-(5.18) define the support variables $S_p$, $R_p$, $X_p$, and $D_p$, that are necessary to model the financial machinery. Finally, constraints (5.19)-(5.22) define the domain of the operational variables.

## 5.3 Solution Approach

To solve the SMISP, we developed a simple heuristic approach in which we generate an initial feasible solution using the constructive heuristic algorithm described in Section 5.3.1. The solution generated by the constructive heuristic algorithm is then given as an input to the MILP solver to fix some variables and speed up the search for an optimal solution.

### 5.3.1 Constructive Heuristic Algorithm

In this section, we describe the tailored constructive heuristic algorithm that we developed for the SMISP. Such an algorithm solves the operational problem of finding an initial feasible schedule of installations by minimizing the total completion time. The algorithm is inspired by the shortest remaining processing time (SRPT) rule for parallel machine models [138]. In our case study, the processing time for each meter group $j$ is set to $p_j = N_j/Q$. We assume that all the substitution squads are identical.

The main idea of the algorithm is to create a sorted candidate list of meter groups based on the SRPT rule during an initial ordering phase (line 4). Then, the ordering phase is followed by an assignment phase during which the substitution squad schedules are updated. In particular, each meter group in the candidate list is assigned to a substitution squad (line 8), according to the order given by the candidate list and as long as there are available substitution squads. Such an order-assign mechanism is repeated for each time interval $t$ within the operational horizon.

In case two or more meter groups have the same remaining processing time at the beginning of a particular time interval $t$, the additional information provided by the meter reading calendar is considered and those meter groups having the same remaining processing time are re-ordered based on the farthest reading (i.e., the meter group with the farthest reading first, and so forth). In addition, note that if a meter group cannot be worked during time interval $t$ due to the concurrent collection of readings, neither can it enter in the sorted candidate list.

Finally, the algorithm returns the initial schedule. $\hat{y}_{jkt}$ heuristic variables taking value 1 may be provided as an input to the MILP formulation to fix the initial value of the corresponding $y_{jkt}$ variables.

In the following, we report the pseudo-code of the algorithm.

**Algorithm 4** Constructive Heuristic Algorithm

1: $\hat{y}_{jkt} \leftarrow 0 \ \forall\, j \in J, k \in K, t \in T_{SP}$
2: $Schedule \leftarrow \emptyset$
3: **while** $t \leq |T_{SP}|$ **do**
4:     $CandidateList \leftarrow \texttt{SRPT}(J)$ ▷ Build a sorted candidate list of meter groups
5:     **for** $j \leftarrow 1, \ldots, |J|$ such that $j \in CandidateList$ **do**
6:         **for** $k \leftarrow 1, \ldots, |K|$ **do**
7:             **if** $k$ is idle during $t$ **then**
8:                 $\hat{y}_{jkt} \leftarrow 1$ ▷ Assign meter group $j$ to squad $k$, during time interval $t$
9:                 $Schedule \leftarrow Schedule \cup \{\hat{y}_{jkt}\}$ ▷ Update the schedule
10:             **end if**
11:         **end for**
12:     **end for**
13: **end while**
14: **return** $Schedule$

## 5.4 Computational Results

In this section, we present the results of some computational tests that were performed on a computer server equipped with two Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz processors and 64 GB of RAM, using FICO Xpress as MILP solver. The mathematical formulation and the constructive heuristic algorithm were coded in Mosel language and a time limit of 3,600 CPU seconds was imposed on each execution.

To test the proposed solution approach, we created 135 realistic base instances adapted from the original real-world application. Each instance comprises 5 meter groups with a fixed number of traditional meters to substitute. Let $s$ be the estimated unitary cost of a single smart reading, from which the conditional cost savings are derived, and $C$ the unitary expenditure for installing a single smart meter. We chose $s \in \{0.07, 0.14, 0.21\}$, $C \in \{10, 50\}$, and, for what concerns the other parameters, we selected $r \in \{0.04, 0.06, 0.08\}$ and $Q \in \{10, 50\}$. Several combinations of parameters $r$, $s$, $Q$, and $C$ were considered, while the number of substitution squads was varied between 1 and 5 for each combination of parameters.

We performed three computational experiments. In the first experiment, we solved the problem over weekly time intervals via branch-and-bound. The computational results are reported in Table 5.1. In particular, from column "$|J|$" to column "$C$" we report the different parameters that were used for each subgroup of instances, while column "# of instances" indicates the number of instances per subgroup. Columns "$\text{NPV}_{lb}$" and "$\text{NPV}_{ub}$" give the average lower bound and the average upper bound, respectively, while columns "$\%\text{gap}_{\text{min}}$" and "$\%\text{gap}_{\text{max}}$" give the minimum gap and the maximum gap, respectively. Column "time(s)" gives the computing time, and the entry "tlim" indicates that the time limit was reached for all the five instances in the group. Column "opt" gives the total number of instances solved to proven optimality. Here, we observe that only 33 instances were solved to optimality. For the remaining instances, the solver reached the time limit with an average minimum gap of 17% and an average maximum gap of 21%.

In the second experiment, we solved the problem over monthly time intervals via branch-and bound. The computational results are reported in Table 5.2. Here, all the instances were solved to optimality. However, from a practical perspective, the solutions provided over monthly time intervals are less detailed and may require further scheduling activities. Note that the slight difference among the $\text{NPV}_{lb}$ and $\text{NPV}_{ub}$ values is given by the MILP solver tolerance.

In the third experiment, we built an initial feasible schedule with the aforementioned constructive heuristic algorithm. According to this schedule, we fixed the initial value of $y_{jkt}$ variables, before solving the problem over weekly time intervals via branch-and-bound. The computational results are reported in Table 5.3. Column "$\text{NPV}_{heur}$" gives the average solution value obtained for each subgroup of instances, while column "$\%\text{gap}_{\text{NPV}_{\text{heur}}-\text{NPV}_{\text{lb}}}$" gives the gap with the average lower bound found by solving the problem via branch-and-bound and without variable fixing. Here, we observe that the computing times were significantly reduced thanks to the variable fixing procedure, while the gap with the average lower bound found by solving the problem via branch-and-bound and without variable fixing is limited. Nonetheless, the average NPV value is lower if compared to the previous tests.

Table 5.1: Computational results over weekly time intervals without variable fixing

| $\|J\|$ | $\sum_{j\in J} N_j$ | $r$ | $s$ | $\|K\|$ | $Q$ | $C$ | # of instances | $\text{NPV}_{lb}$ | $\text{NPV}_{ub}$ | %$\text{gap}_{\min}$ | %$\text{gap}_{\max}$ | time(s) | opt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 1150 | 0.04 | 0.07 | {1,2,3,4,5} | 10 | 10 | 5 | 20284.5 | 20355.0 | 0.00 | 0.02 | 881.6 | 4 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 15716.2 | 17903.0 | 0.10 | 0.15 | tlim | 0 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 17025.9 | 18832.2 | 0.07 | 0.11 | tlim | 0 |
| | | | 0.14 | {1,2,3,4,5} | 10 | 10 | 5 | 19206.6 | 19277.5 | 0.00 | 0.02 | 898.1 | 4 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 14676.0 | 16916.8 | 0.10 | 0.16 | tlim | 0 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 15873.4 | 18202.1 | 0.10 | 0.15 | tlim | 0 |
| | | | 0.21 | {1,2,3,4,5} | 10 | 10 | 5 | 18128.7 | 18196.1 | 0.00 | 0.02 | 957.5 | 4 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 13576.8 | 15786.0 | 0.12 | 0.18 | tlim | 0 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 14740.9 | 16849.3 | 0.10 | 0.14 | tlim | 0 |
| 5 | 1150 | 0.06 | 0.07 | {1,2,3,4,5} | 10 | 10 | 5 | 17609.7 | 17708.3 | 0.00 | 0.03 | 1217.1 | 4 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 11283.4 | 15127.5 | 0.24 | 0.30 | tlim | 0 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 12523.5 | 16230.3 | 0.20 | 0.27 | tlim | 0 |
| | | | 0.14 | {1,2,3,4,5} | 10 | 10 | 5 | 16644.1 | 16742.4 | 0.00 | 0.03 | 1271.5 | 4 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 10316.8 | 14450.9 | 0.25 | 0.31 | tlim | 0 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 11494.2 | 15396.8 | 0.22 | 0.28 | tlim | 0 |
| | | | 0.21 | {1,2,3,4,5} | 10 | 10 | 5 | 15677.5 | 15798.2 | 0.00 | 0.04 | 1365.2 | 4 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 9320.8 | 13332.3 | 0.28 | 0.32 | tlim | 0 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 10463.9 | 14397.4 | 0.25 | 0.31 | tlim | 0 |
| 5 | 1150 | 0.08 | 0.07 | {1,2,3,4,5} | 10 | 10 | 5 | 15337.1 | 15514.7 | 0.00 | 0.05 | 1638.7 | 3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 7463.7 | 12605.4 | 0.38 | 0.43 | tlim | 0 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 8620.8 | 13577.9 | 0.34 | 0.40 | tlim | 0 |
| | | | 0.14 | {1,2,3,4,5} | 10 | 10 | 5 | 14466.6 | 14610.3 | 0.00 | 0.05 | 1639.3 | 3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 6574.6 | 12022.5 | 0.43 | 0.48 | tlim | 0 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 7725.6 | 13052.3 | 0.39 | 0.44 | tlim | 0 |
| | | | 0.21 | {1,2,3,4,5} | 10 | 10 | 5 | 13586.0 | 13819.6 | 0.00 | 0.06 | 1707.9 | 3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 5704.0 | 11675.9 | 0.50 | 0.52 | tlim | 0 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 6790.7 | 12196.7 | 0.39 | 0.48 | tlim | 0 |
| | | sum/avg | | | | | 135 | 12993.8 | 15576.9 | 0.17 | 0.21 | 2829.1 | 33 |

Table 5.2: Computational results over monthly time intervals without variable fixing

| $\|J\|$ | $\sum_{j\in J} N_j$ | $r$ | $s$ | $\|K\|$ | $Q$ | $C$ | # of instances | NPV$_{lb}$ | NPV$_{ub}$ | %gap$_{\min}$ | %gap$_{\max}$ | time(s) | opt |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 1150 | 0.04 | 0.07 | {1,2,3,4,5} | 10 | 10 | 5 | 20155.1 | 20155.4 | 0.00 | 0.00 | 4.1 | 5 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 15657.0 | 15657.2 | 0.00 | 0.00 | 44.4 | 5 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 16937.4 | 16937.4 | 0.00 | 0.00 | 16.6 | 5 |
| | | | 0.14 | {1,2,3,4,5} | 10 | 10 | 5 | 19083.9 | 19083.9 | 0.00 | 0.00 | 4.1 | 5 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 14585.8 | 14585.8 | 0.00 | 0.00 | 53.6 | 5 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 15799.4 | 15799.4 | 0.00 | 0.00 | 22.9 | 5 |
| | | | 0.21 | {1,2,3,4,5} | 10 | 10 | 5 | 18012.8 | 18012.8 | 0.00 | 0.00 | 4.0 | 5 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 13514.6 | 13514.7 | 0.00 | 0.00 | 64.3 | 5 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 14661.6 | 14661.6 | 0.00 | 0.00 | 32.1 | 5 |
| 5 | 1150 | 0.06 | 0.07 | {1,2,3,4,5} | 10 | 10 | 5 | 17484.5 | 17484.5 | 0.00 | 0.00 | 5.0 | 5 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 11213.9 | 11214.0 | 0.00 | 0.00 | 160.9 | 5 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 12440.7 | 12440.7 | 0.00 | 0.00 | 80.7 | 5 |
| | | | 0.14 | {1,2,3,4,5} | 10 | 10 | 5 | 16525.4 | 16525.4 | 0.00 | 0.00 | 5.1 | 5 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 10254.9 | 10255.0 | 0.00 | 0.00 | 226.4 | 5 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 11415.2 | 11415.4 | 0.00 | 0.00 | 84.4 | 5 |
| | | | 0.21 | {1,2,3,4,5} | 10 | 10 | 5 | 15566.4 | 15566.4 | 0.00 | 0.00 | 5.1 | 5 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 9295.9 | 9296.0 | 0.00 | 0.00 | 302.0 | 5 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 10389.8 | 10389.8 | 0.00 | 0.00 | 123.3 | 5 |
| 5 | 1150 | 0.08 | 0.07 | {1,2,3,4,5} | 10 | 10 | 5 | 15215.3 | 15215.3 | 0.00 | 0.00 | 5.5 | 5 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 7409.3 | 7409.3 | 0.00 | 0.00 | 478.6 | 5 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 8571.9 | 8571.9 | 0.00 | 0.00 | 287.2 | 5 |
| | | | 0.14 | {1,2,3,4,5} | 10 | 10 | 5 | 14351.1 | 14351.1 | 0.00 | 0.00 | 5.8 | 5 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 6545.1 | 6545.2 | 0.00 | 0.00 | 634.1 | 5 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 7641.6 | 7641.6 | 0.00 | 0.00 | 365.8 | 5 |
| | | | 0.21 | {1,2,3,4,5} | 10 | 10 | 5 | 13487.0 | 13487.0 | 0.00 | 0.00 | 6.1 | 5 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 5681.0 | 5681.2 | 0.00 | 0.00 | 826.5 | 5 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 6711.4 | 6711.4 | 0.00 | 0.00 | 562.5 | 5 |
| | | | sum/avg | | | | 135 | 12911.4 | 12911.5 | 0.00 | 0.00 | 163.4 | 135 |

Table 5.3: Computational results over weekly time intervals with variable fixing

| $\|J\|$ | $\sum_{j\in J} N_j$ | $r$ | $s$ | $\|K\|$ | $Q$ | $C$ | # of instances | $\text{NPV}_{heur}$ | $\%\text{gap}_{\text{NPV}_{\text{heur}}-\text{NPV}_{\text{lb}}}$ | time(s) |
|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 1150 | 0.04 | 0.07 | {1,2,3,4,5} | 10 | 10 | 5 | 20025.8 | 0.01 | 0.3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 15523.9 | 0.01 | 0.3 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 16967.7 | 0.00 | 0.6 |
| | | | 0.14 | {1,2,3,4,5} | 10 | 10 | 5 | 18961.0 | 0.01 | 0.3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 14459.1 | 0.02 | 0.3 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 15828.2 | 0.00 | 0.8 |
| | | | 0.21 | {1,2,3,4,5} | 10 | 10 | 5 | 17896.2 | 0.01 | 0.3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 13394.2 | 0.01 | 0.3 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 14688.6 | 0.00 | 0.8 |
| 5 | 1150 | 0.06 | 0.07 | {1,2,3,4,5} | 10 | 10 | 5 | 17358.1 | 0.01 | 0.3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 11079.8 | 0.02 | 0.3 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 12471.1 | 0.00 | 0.5 |
| | | | 0.14 | {1,2,3,4,5} | 10 | 10 | 5 | 16405.2 | 0.01 | 0.3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 10126.9 | 0.02 | 0.3 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 11444.0 | 0.00 | 0.6 |
| | | | 0.21 | {1,2,3,4,5} | 10 | 10 | 5 | 15452.3 | 0.01 | 0.3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 9174.0 | 0.02 | 0.3 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 10416.8 | 0.00 | 0.5 |
| 5 | 1150 | 0.08 | 0.07 | {1,2,3,4,5} | 10 | 10 | 5 | 15091.4 | 0.02 | 0.3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 7272.9 | 0.03 | 0.3 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 8602.3 | 0.00 | 0.5 |
| | | | 0.14 | {1,2,3,4,5} | 10 | 10 | 5 | 14233.3 | 0.02 | 0.3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 6414.8 | 0.02 | 0.3 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 7670.3 | 0.01 | 0.6 |
| | | | 0.21 | {1,2,3,4,5} | 10 | 10 | 5 | 13375.1 | 0.02 | 0.3 |
| | | | | {1,2,3,4,5} | 10 | 50 | 5 | 5556.6 | 0.03 | 0.3 |
| | | | | {1,2,3,4,5} | 50 | 50 | 5 | 6738.4 | 0.01 | 0.5 |
| | | | sum/avg | | | | 135 | 12838.1 | 0.01 | 0.4 |

## 5.5 Conclusions and Future Developments

In this work, we described a real-world smart-meter installation scheduling project (SMISP) arising at *IRETI*, a multi-utility company operating in the Italian water distribution sector. The SMISP was accurately defined through a MILP formulation, in which the objective function is to maximize a net present value while satisfying both financial and operational constraints.

A number of computational experiments were performed by solving the problem via branch-and-bound over alternative time intervals (i.e., weekly and monthly), as well as using a variable fixing approach based on a tailored constructive heuristic algorithm for the SMISP. Although the variable fixing approach turned out to find good quality solutions in short computing times, the computational results that we obtained on 135 realistic base instances proved the need for developing a more sophisticated solution approach, like a metaheuristic algorithm or a decomposition method. As a possible future development, we intend to use such an approach to solve the real-world application encountered in the province of Reggio Emilia.

# Bibliography


[1] J. Aczél and T. L. Saaty, "Procedures for synthesizing ratio judgements," *Journal of Mathematical Psychology*, vol. 27, no. 1, pp. 93–102, 1983.

[2] N. Agatz, A. M. Campbell, M. Fleischmann, and M. Savelsbergh, "Challenges and Opportunities in Attended Home Delivery," in *The Vehicle Routing Problem: Latest Advances and New Challenges*, B. L. Golden, S. Raghavan, and E. Wasil, Eds., Boston: Springer, 2008, pp. 379–396.

[3] N. Agatz, A. M. Campbell, M. Fleischmann, and M. Savelsbergh, "Time Slot Management in Attended Home Delivery," *Transportation Science*, vol. 45, no. 3, pp. 435–449, 2011.

[4] N. Agatz, A. M. Campbell, M. Fleischmann, J. van Nunen, and M. Savelsbergh, "Revenue management opportunities for internet retailers," *Journal of Revenue and Pricing Management*, vol. 12, no. 2, pp. 128–138, 2013.

[5] N. Agatz, Y. Fan, and D. Stam, "The Impact of Green Labels on Time Slot Choice and Operational Sustainability," *Production and Operations Management*, vol. 30, no. 7, pp. 2285–2303, 2021.

[6] N. Agatz, M. Fleischmann, and J. van Nunen, "E-fulfillment and multi-channel distribution – A review," *European Journal of Operational Research*, vol. 187, no. 2, pp. 339–356, 2008.

[7] O. Ali, J.-F. Côté, and L. C. Coelho, "Models and algorithms for the delivery and installation routing problem," *European Journal of Operational Research*, vol. 291, no. 1, pp. 162–177, 2021.

[8] S. Allahyari, M. Salari, and D. Vigo, "A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem," *European Journal of Operational Research*, vol. 242, no. 3, pp. 756–768, 2015.

[9] A. M. Anaya-Arenas, C. Prodhon, J. Renaud, and A. Ruiz, "An iterated local search for the biomedical sample transportation problem with multiple and interdependent pickups," *Journal of the Operational Research Society*, vol. 72, no. 2, pp. 367–382, 2021.

[10] C. Archetti and L. Bertazzi, "Recent challenges in Routing and Inventory Routing: E-commerce and last-mile delivery," *Networks: An International Journal*, vol. 77, no. 2, pp. 255–268, 2021.

[11] ARERA, *Integrated text of the authority regulations for accounting unbundling for companies operating in the same gas, water or electricity business market*, Available at `https://www.arera.it/allegati/docs/16/137-16all_ti.pdf`, [Online; Accessed 18 November 2022], 2016.

[12] ARERA, *Integrated text of the authority regulations for water service compensations*, Available at `https://www.arera.it/allegati/docs/17/665-17all.pdf`, [Online; Accessed 18 November 2022], 2017.

[13] ARERA, *Single text of the instructions of quality regulation and tariffs of gas distribution and measurement services for the 2020-2025 regulatory period*, Available at `https://www.arera.it/allegati/docs/19/569-19rqdg.pdf`, [Online; Accessed 18 November 2022], 2019.

[14] K. Asdemir, V. S. Jacob, and R. Krishnan, "Dynamic pricing of multiple home delivery options," *European Journal of Operational Research*, vol. 196, no. 1, pp. 246–257, 2009.

[15] R. Atefi, M. Salari, L. C. Coelho, and J. Renaud, "The open vehicle routing problem with decoupling points," *European Journal of Operational Research*, vol. 265, no. 1, pp. 316–327, 2018.

[16] N. Azi, M. Gendreau, and J.-Y. Potvin, "An exact algorithm for a single-vehicle routing problem with time windows and multiple routes," *European Journal of Operational Research*, vol. 178, no. 3, pp. 755–766, 2007.

[17] N. Azi, M. Gendreau, and J.-Y. Potvin, "An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles," *European Journal of Operational Research*, vol. 202, no. 3, pp. 756–763, 2010.

[18] N. Azi, M. Gendreau, and J.-Y. Potvin, "A dynamic vehicle routing problem with multiple delivery routes," *Annals of Operations Research*, vol. 199, no. 1, pp. 103–112, 2012.

[19] N. Azi, M. Gendreau, and J.-Y. Potvin, "An adaptive large neighborhood search for a vehicle routing problem with multiple routes," *Computers & Operations Research*, vol. 41, pp. 167–173, 2014.

[20] I. Aziez, J.-F. Côté, and L. C. Coelho, "Exact algorithms for the multi-pickup and delivery problem with time windows," *European Journal of Operational Research*, vol. 284, no. 3, pp. 906–919, 2020.

[21] E. Balas, M. Fischetti, and W. R. Pulleyblank, "The precedence-constrained asymmetric traveling salesman polytope," *Mathematical Programming*, vol. 68, pp. 241–265, 1995.

[22] R. Baldacci, A. Mingozzi, and R. Roberti, "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints," *European Journal of Operational Research*, vol. 218, no. 1, pp. 1–6, 2012.

[23] S. Bannerjee, J. Bone, Y. Finger, and C. Haley, *European digital city index methodology report*, Available at `https://www.nesta.org.uk/blog/edci-2016-updating-the-european-digital-city-index/`, [Online; Accessed 18 November 2022], 2016.

[24] M. Battarra, J.-F. Cordeau, and M. Iori, "Chapter 6: Pickup-and-Delivery Problems for Goods Transportation," in *Vehicle Routing: Problems, Methods, and Applications*, P. Toth and D. Vigo, Eds., Philadelphia: SIAM, 2014, pp. 161–191.

[25] T. Bektaş and L. Gouveia, "*Requiem* for the Miller–Tucker–Zemlin subtour elimination constraints?" *European Journal of Operational Research*, vol. 236, no. 3, pp. 820–832, 2014.

[26] R. W. Bent and P. Van Hentenryck, "Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers," *Operations Research*, vol. 52, no. 6, pp. 977–987, 2004.

[27] Bloomberg, *The Dark Side of 15-Minute Grocery Delivery*, Available at `https://www.bloomberg.com/news/articles/2021-12-07/what-instant-delivery-services-could-do-to-cities`, [Online; Accessed 18 November 2022], 2021.

[28] N. Boysen, S. Fedtke, and S. Schwerdfeger, "Last-mile delivery concepts: A survey from an operational research perspective," *OR Spectrum*, vol. 43, no. 1, pp. 1–58, 2021.

[29] O. Bräysy and M. Gendreau, "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms," *Transportation Science*, vol. 39, no. 1, pp. 104–118, 2005.

[30] O. Bräysy and M. Gendreau, "Vehicle Routing Problem with Time Windows, Part II: Metaheuristics," *Transportation Science*, vol. 39, no. 1, pp. 119–139, 2005.

[31] D. Bredström and M. Rönnqvist, "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints," *European Journal of Operational Research*, vol. 191, no. 1, pp. 19–31, 2008.

[32] B. P. Bruck, F. Castegini, J.-F. Cordeau, M. Iori, T. Poncemi, and D. Vezzali, "A Decision Support System for Attended Home Services," *INFORMS Journal on Applied Analytics*, vol. 50, no. 2, pp. 137–152, 2020.

[33] B. P. Bruck, J.-F. Cordeau, and M. Iori, "A practical time slot management and routing problem for attended home services," *Omega*, vol. 81, pp. 208–219, 2018.

[34] B. P. Bruck and M. Iori, "Non-Elementary Formulations for Single Vehicle Routing Problems with Pickups and Deliveries," *Operations Research*, vol. 65, no. 6, pp. 1597–1614, 2017.

[35] B. P. Bruck, D. Vezzali, M. Iori, C. A. Magni, and D. Pretolani, "A Decision Support System to Evaluate Suppliers in the Context of Global Service Providers," in *Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 1: ICEIS*, 2021, pp. 420–430.

[36] D. Bühler, R. Klein, and M. Neugebauer, "Model-based delivery cost approximation in attended home services," *Computers & Industrial Engineering*, vol. 98, pp. 78–90, 2016.

[37] A. M. Campbell and M. Savelsbergh, "Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems," *Transportation Science*, vol. 38, no. 3, pp. 369–378, 2004.

[38] A. M. Campbell and M. Savelsbergh, "Decision Support for Consumer Direct Grocery Initiatives," *Transportation Science*, vol. 39, no. 3, pp. 313–327, 2005.

[39] A. M. Campbell and M. Savelsbergh, "Incentive Schemes for Attended Home Delivery Services," *Transportation Science*, vol. 40, no. 3, pp. 327–341, 2006.

[40] P. Cappanera and M. G. Scutellà, "Joint Assignment, Scheduling, and Routing Models to Home Care Optimization: A Pattern-Based Approach," *Transportation Science*, vol. 49, no. 4, pp. 830–852, 2015.

[41] P. Cappanera, M. G. Scutellà, F. Nervi, and L. Galli, "Demand uncertainty in robust Home Care optimization," *Omega*, vol. 80, pp. 95–110, 2018.

[42] D. Cattaruzza, N. Absi, D. Feillet, and J. González-Feliu, "Vehicle routing problems for city logistics," *EURO Journal on Transportation and Logistics*, vol. 6, no. 1, pp. 51–79, 2017.

[43] F. Çebi and D. Bayraktar, "An integrated approach for supplier selection," *Logistics Information Management*, vol. 16, no. 6, pp. 395–400, 2003.

[44] J. Chai, J. N. Liu, and E. W. Ngai, "Application of decision-making techniques in supplier selection: A systematic review of literature," *Expert Systems with Applications*, vol. 40, no. 10, pp. 3872–3885, 2013.

[45] C. Cleophas and J. F. Ehmke, "When Are Deliveries Profitable?" *Business & Information Systems Engineering*, vol. 6, no. 3, pp. 153–163, 2014.

[46] J.-F. Cordeau, M. Gendreau, and G. Laporte, "A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems," *Networks: An International Journal*, vol. 30, no. 2, pp. 105–119, 1997.

[47] J.-F. Cordeau, G. Laporte, and A. Mercier, "A unified tabu search heuristic for vehicle routing problems with time windows," *Journal of the Operational Research Society*, vol. 52, no. 8, pp. 928–936, 2001.

[48] C. F. Daganzo, "Modeling Distribution Problems with Time Windows: Part I," *Transportation Science*, vol. 21, no. 3, pp. 171–179, 1987.

[49] G. Desaulniers, F. Errico, S. Irnich, and M. Schneider, "Exact Algorithms for Electric Vehicle-Routing Problems with Time Windows," *Operations Research*, vol. 64, no. 6, pp. 1388–1405, 2016.

[50] G. Desaulniers, O. B. Madsen, and S. Ropke, "Chapter 5: The Vehicle Routing Problem with Time Windows," in *Vehicle Routing: Problems, Methods, and Applications*, P. Toth and D. Vigo, Eds., Philadelphia: SIAM, 2014, pp. 119–159.

[51] M. Desrochers and G. Laporte, "Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints," *Operations Research Letters*, vol. 10, no. 1, pp. 27–36, 1991.

[52] K. F. Doerner and J.-J. Salazar-González, "Chapter 7: Pickup-and-Delivery Problems for People Transportation," in *Vehicle Routing: Problems, Methods, and Applications*, P. Toth and D. Vigo, Eds., Philadelphia: SIAM, 2014, pp. 193–212.

[53] E. N. Duman, D. Taş, and B. Çatay, "Branch-and-price-and-cut methods for the electric vehicle routing problem with time windows," *International Journal of Production Research*, vol. 60, no. 17, pp. 5332–5353, 2022.

[54] P. M. Duque, M. Castro, K. Sörensen, and P. Goos, "Home care service planning. The case of Landelijke Thuiszorg," *European Journal of Operational Research*, vol. 243, no. 1, pp. 292–301, 2015.

[55] F. Dweiri, S. Kumar, S. A. Khan, and V. Jain, "Designing an integrated AHP based decision support system for supplier selection in automotive industry," *Expert Systems with Applications*, vol. 62, pp. 273–283, 2016.

[56] J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics* (International Series in Operations Research & Management Science), 1st. New York: Springer, 2012, vol. 177.

[57] J. F. Ehmke and A. M. Campbell, "Customer acceptance mechanisms for home deliveries in metropolitan areas," *European Journal of Operational Research*, vol. 233, no. 1, pp. 193–207, 2014.

[58] J. F. Ehmke and D. C. Mattfeld, "Vehicle routing for attended home delivery in city logistics," *Procedia-Social and Behavioral Sciences*, vol. 39, pp. 622–632, 2012.

[59] J. F. Ehmke, S. Meisel, and D. C. Mattfeld, "Floating car based travel times for city logistics," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 338–352, 2012.

[60] J. F. Ehmke, A. Steinert, and D. C. Mattfeld, "Advanced routing for city logistics service providers based on time-dependent travel times," *Journal of Computational Science*, vol. 3, no. 4, pp. 193–205, 2012.

[61] F. Errico, G. Desaulniers, M. Gendreau, W. Rei, and L.-M. Rousseau, "The vehicle routing problem with hard time windows and stochastic service times," *EURO Journal on Transportation and Logistics*, vol. 7, no. 3, pp. 223–251, 2018.

[62] J. Euchi, M. Masmoudi, and P. Siarry, "Home health care routing and scheduling problems: A literature review," *4OR*, vol. 20, no. 3, pp. 351–389, 2022.

[63] European Commission, *Competence Centre on Composite Indicators and Scoreboards (COIN)*, Available at `https://knowledge4policy.ec.europa.eu/composite-indicators_en`, [Online; Accessed 18 November 2022], 2002.

[64] European Parliament, *Directive 2009/73/EC of the European parliament and of the council*, Available at `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:211:0094:0136:en:PDF`, [Online; Accessed 18 November 2022], 2009.

[65] P. Eveborn, P. Flisberg, and M. Rönnqvist, "LAPS CARE–an operational system for staff planning of home care," *European Journal of Operational Research*, vol. 171, no. 3, pp. 962–976, 2006.

[66] E. Fadda, L. Gobbato, G. Perboli, M. Rosano, and R. Tadei, "Waste Collection in Urban Areas: A Case Study," *Interfaces*, vol. 48, no. 4, pp. 307–322, 2018.

[67] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, "An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to Some Vehicle Routing Problems," *Networks: An International Journal*, vol. 44, no. 3, pp. 216–229, 2004.

[68] J. Figueira and B. Roy, "Determining the weights of criteria in the ELECTRE type methods with a revised Simos' procedure," *European Journal of Operational Research*, vol. 139, no. 2, pp. 317–326, 2002.

[69] C. Fikar and P. Hirsch, "Home health care routing and scheduling: A review," *Computers & Operations Research*, vol. 77, pp. 86–95, 2017.

[70] M. Fischetti, C. Polo, and M. Scantamburlo, "A Local Branching Heuristic for Mixed-Integer Programs with 2-Level Variables, with an Application to a Telecommunication Network Design Problem," *Networks: An International Journal*, vol. 44, no. 2, pp. 61–72, 2004.

[71] M. L. Fisher and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing," *Networks: An International Journal*, vol. 11, no. 2, pp. 109–124, 1981.

[72] D. Fleckenstein, R. Klein, and C. Steinhardt, "Recent advances in integrating demand management and vehicle routing: A methodological review," *European Journal of Operational Research*, 2022, [Article In Press].

[73] A. M. Florio, D. Feillet, and R. F. Hartl, "The delivery problem: Optimizing hit rates in e-commerce deliveries," *Transportation Research Part B: Methodological*, vol. 117, pp. 455–472, 2018.

[74] M. G. S. Furtado, P. Munari, and R. Morabito, "The pickup and delivery problem with time windows in the oil industry: model and Branch-and-cut methods," *Gestão & Produção*, vol. 24, no. 3, pp. 501–513, 2017.

[75] G. Gallego, R. Ratliff, and S. Shebalov, "A General Attraction Model and Sales-Based Linear Program for Network Revenue Management Under Customer Choice," *Operations Research*, vol. 63, no. 1, pp. 212–232, 2015.

[76] M. S. Garcia-Cascales and M. T. Lamata, "On rank reversal and TOPSIS method," *Mathematical and Computer Modelling*, vol. 56, no. 5–6, pp. 123–132, 2012.

[77] S. H. Ghodsypour and C. O'Brien, "A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming," *International Journal of Production Economics*, vol. 56-57, pp. 199–212, 1998.

[78] S. H. Ghodsypour and C. O'Brien, "The total cost of logistics in supplier selection, under conditions of multiple sourcing, multiple criteria and capacity constraint," *International Journal of Production Economics*, vol. 73, no. 1, pp. 15–27, 2001.

[79] K. Goffin, M. Szwejczewski, and C. New, "Managing suppliers: when fewer can mean more," *International Journal of Physical Distribution & Logistics Management*, vol. 27, no. 7, pp. 422–436, 1997.

[80] K. Govindan, S. Rajendran, J. Sarkis, and P. Murugesan, "Multi criteria decision making approaches for green supplier evaluation and selection: a literature review," *Journal of Cleaner Production*, vol. 98, pp. 66–83, 2015.

[81] S. Greco, M. Ehrgott, and J. R. Figueira, *Multiple Criteria Decision Analysis: State of the Art Surveys* (International Series in Operations Research & Management Science), 2nd. New York: Springer, 2016, vol. 233.

[82] A. Gunawan, G. Kendall, B. McCollum, H.-V. Seow, and L. S. Lee, "Vehicle routing: Review of benchmark datasets," *Journal of the Operational Research Society*, vol. 72, no. 8, pp. 1794–1807, 2021.

[83] S. R. A. Haddadene, N. Labadie, and C. Prodhon, "A GRASP × ILS for the vehicle routing problem with time windows, synchronization and precedence constraints," *Expert Systems with Applications*, vol. 66, pp. 274–294, 2016.

[84] S. Han, L. Zhao, K. Chen, Z.-w. Luo, and D. Mishra, "Appointment scheduling and routing optimization of attended home delivery system with random customer behavior," *European Journal of Operational Research*, vol. 262, no. 3, pp. 966–980, 2017.

[85] T. Hanne, T. Melo, and S. Nickel, "Bringing Robustness to Patient Flow Management Through Optimized Patient Transports in Hospitals," *Interfaces*, vol. 39, no. 3, pp. 241–255, 2009.

[86] F. Hernandez, M. Gendreau, and J.-Y. Potvin, "Heuristics for tactical time slot management: A periodic vehicle routing problem view," *International Transactions in Operational Research*, vol. 24, no. 6, pp. 1233–1252, 2017.

[87] H. Hernández-Pérez, M. Landete, and I. Rodriguez-Martin, "The single-vehicle two-echelon one-commodity pickup and delivery problem," *Computers & Operations Research*, vol. 127, p. 105 152, 2021.

[88] A. Hertz and N. Lahrichi, "A patient assignment algorithm for home care services," *Journal of the Operational Research Society*, vol. 60, no. 4, pp. 481–495, 2009.

[89] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl, "The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations," *European Journal of Operational Research*, vol. 252, no. 3, pp. 995–1018, 2016.

[90] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 11th. New York: McGraw-Hill Education, 2021.

[91] W. Ho, "Integrated analytic hierarchy process and its applications–A literature review," *European Journal of Operational Research*, vol. 186, no. 1, pp. 211–228, 2008.

[92] W. Ho and X. Ma, "The state-of-the-art integrations and applications of the analytic hierarchy process," *European Journal of Operational Research*, vol. 267, no. 2, pp. 399–414, 2018.

[93] W. Ho, X. Xu, and P. K. Dey, "Multi-criteria decision making approaches for supplier evaluation and selection: A literature review," *European Journal of Operational Research*, vol. 202, no. 1, pp. 16–24, 2010.

[94] International Center for Sustainable Development of Energy, Water and Environment Systems, *SDEWES index*, Available at `https://www.sdewes.org/sdewes_index.php`, [Online; Accessed 18 November 2022], 2016.

[95] A. Ishizaka and A. Labib, "Review of the main developments in the analytic hierarchy process," *Expert Systems with Applications*, vol. 38, no. 11, pp. 14 336–14 345, 2011.

[96] A. Ishizaka and P. Nemery, *Multi-Criteria Decision Analysis: Methods and Software*, 1st. Chichester: John Wiley & Sons, Ltd., 2013.

[97] O. Jabali, R. Leus, T. Van Woensel, and T. de Kok, "Self-imposed time windows in vehicle routing problems," *OR Spectrum*, vol. 37, no. 2, pp. 331–352, 2015.

[98] B. Kallehauge, "Formulations and exact algorithms for the vehicle routing problem with time windows," *Computers & Operations Research*, vol. 35, no. 7, pp. 2307–2330, 2008.

[99] I. Kara, "Arc Based Integer Programming Formulations for the Distance Constrained Vehicle Routing Problem," in *3rd IEEE International Symposium on Logistics and Industrial Informatics*, 2011, pp. 33–38.

[100] I. Karaoglan, F. Altiparmak, I. Kara, and B. Dengiz, "The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach," *Omega*, vol. 40, no. 4, pp. 465–477, 2012.

[101] M. Keskin and B. Çatay, "Partial recharge strategies for the electric vehicle routing problem with time windows," *Transportation Research Part C: Emerging Technologies*, vol. 65, pp. 111–127, 2016.

[102] M. Keskin and B. Çatay, "A matheuristic method for the electric vehicle routing problem with time windows and fast chargers," *Computers & Operations Research*, vol. 100, pp. 172–188, 2018.

[103] M. Keskin, B. Çatay, and G. Laporte, "A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations," *Computers & Operations Research*, vol. 125, p. 105 060, 2021.

[104] M. Keskin, G. Laporte, and B. Çatay, "Electric vehicle routing problem with time-dependent waiting times at recharging stations," *Computers & Operations Research*, vol. 107, pp. 77–94, 2019.

[105] R. Klein, S. Koch, C. Steinhardt, and A. K. Strauss, "A review of revenue management: Recent generalizations and advances in industry applications," *European Journal of Operational Research*, vol. 284, no. 2, pp. 397–412, 2020.

[106] R. Klein, J. Mackert, M. Neugebauer, and C. Steinhardt, "A model-based approximation of opportunity cost for dynamic pricing in attended home delivery," *OR Spectrum*, vol. 40, no. 4, pp. 969–996, 2018.

[107] R. Klein, M. Neugebauer, D. Ratkovitch, and C. Steinhardt, "Differentiated Time Slot Pricing Under Routing Considerations in Attended Home Delivery," *Transportation Science*, vol. 53, no. 1, pp. 236–255, 2019.

[108] Ç. Koç, G. Laporte, and İ. Tükenmez, "A review of vehicle routing with simultaneous pickup and delivery," *Computers & Operations Research*, vol. 122, p. 104 987, 2020.

[109] S. Koch and R. Klein, "Route-based approximate dynamic programming for dynamic pricing in attended home delivery," *European Journal of Operational Research*, vol. 287, no. 2, pp. 633–652, 2020.

[110] C. Köhler, J. F. Ehmke, and A. M. Campbell, "Flexible time window management for attended home deliveries," *Omega*, vol. 91, p. 102 023, 2020.

[111] K. J. Kostuk and K. A. Willoughby, "A Decision Support System for Scheduling the Canadian Football League," *Interfaces*, vol. 42, no. 3, pp. 286–295, 2012.

[112] T. J. Kull and S. Talluri, "A supply risk reduction model using integrated multicriteria decision making," *IEEE Transactions on Engineering Management*, vol. 55, no. 3, pp. 409–419, 2008.

[113] E. Lam, G. Desaulniers, and P. J. Stuckey, "Branch-and-cut-and-price for the Electric Vehicle Routing Problem with Time Windows, Piecewise-Linear Recharging and Capacitated Recharging Stations," *Computers & Operations Research*, vol. 145, p. 105 870, 2022.

[114] M. A. Lang, C. Cleophas, and J. F. Ehmke, "Anticipative Dynamic Slotting for Attended Home Deliveries," *Operations Research Forum*, vol. 2, no. 4, p. 70, 2021.

[115] M. A. Lang, C. Cleophas, and J. F. Ehmke, "Multi-criteria decision making in dynamic slotting for attended home deliveries," *Omega*, vol. 102, p. 102 305, 2021.

[116] E. Lanzarone and A. Matta, "Robust nurse-to-patient assignment in home care services to minimize overtimes under continuity of care," *Operations Research for Health Care*, vol. 3, no. 2, pp. 48–58, 2014.

[117] D. Lebedev, P. Goulart, and K. Margellos, "A dynamic programming framework for optimal delivery time slot pricing," *European Journal of Operational Research*, vol. 292, no. 2, pp. 456–468, 2021.

[118] I. I. Lin and H. S. Mahmassani, "Can Online Grocers Deliver?: Some Logistics Considerations," *Transportation Research Record*, vol. 1817, no. 1, pp. 17–24, 2002.

[119] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated Local Search: Framework and Applications," in *Handbook of Metaheuristics*, J.-Y. Potvin and M. Gendreau, Eds., 3rd, Cham: Springer, 2019, pp. 129–168.

[120] J. Mackert, "Choice-based dynamic time slot management in attended home delivery," *Computers & Industrial Engineering*, vol. 129, pp. 333–345, 2019.

[121] O. B. Madsen, K. Tosti, and J. Vælds, "A heuristic method for dispatching repair men," *Annals of Operations Research*, vol. 61, no. 1, pp. 213–226, 1995.

[122] F. Mafakheri, M. Breton, and A. Ghoniem, "Supplier selection-order allocation: A two-stage multiple criteria dynamic programming approach," *International Journal of Production Economics*, vol. 132, no. 1, pp. 52–57, 2011.

[123] C. A. Magni, *Investment Decisions and the Logic of Valuation*, 1st. Cham: Springer, 2020.

[124] S. Mancini and M. Gansterer, "Bundle generation for last-mile delivery with occasional drivers," *Omega*, vol. 108, p. 102 582, 2022.

[125] C. Moon, J. Kim, G. Choi, and Y. Seo, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *European Journal of Operational Research*, vol. 140, no. 3, pp. 606–617, 2002.

[126] A. Mor and M. G. Speranza, "Vehicle routing problems over time: A survey," *Annals of Operations Research*, vol. 314, pp. 255–275, 2022.

[127] R. Mukherjee, U. M. Diwekar, and A. Vaseashta, "Optimal sensor placement with mitigation strategy for water network systems under uncertainty," *Computers & Chemical Engineering*, vol. 103, pp. 91–102, 2017.

[128] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: An overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[129] Z. Naji-Azimi and M. Salari, "The time constrained maximal covering salesman problem," *Applied Mathematical Modelling*, vol. 38, no. 15-16, pp. 3945–3957, 2014.

[130] M. Nardo, M. Saisana, A. Saltelli, S. Tarantola, A. Hoffmann, and E. Giovannini, *Handbook on Constructing Composite Indicators: Methodology and User Guide*, Available at `https://knowledge4policy.ec.europa.eu/publication/handbook-constructing-composite-indicators-methodology-user-guide-0_en`, [Online; Accessed 18 November 2022], 2020.

[131] S. S. Naserizade, M. R. Nikoo, and H. Montaseri, "A risk-based multi-objective model for optimal placement of sensors in water distribution system," *Journal of Hydrology*, vol. 557, pp. 147–159, 2018.

[132] D. H. Nguyen, S. de Leeuw, and W. E. Dullaert, "Consumer Behaviour and Order Fulfilment in Online Retailing: A Systematic Review," *International Journal of Management Reviews*, vol. 20, no. 2, pp. 255–276, 2018.

[133] Organisation for Economic Co-operation and Development, *E-commerce in the times of COVID-19*, Available at `https://www.oecd.org/coronavirus/policy-responses/e-commerce-in-the-time-of-covid-19-3a2b78e8/`, [Online; Accessed 18 November 2022], 2020.

[134] S. S. Özarık, L. P. Veelenturf, T. Van Woensel, and G. Laporte, "Optimizing e-commerce last-mile vehicle routing and scheduling under uncertain customer presence," *Transportation Research Part E: Logistics and Transportation Review*, vol. 148, p. 102 263, 2021.

[135] B. Pan, Z. Zhang, and A. Lim, "Multi-trip time-dependent vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 291, no. 1, pp. 218–231, 2021.

[136] S. Pan, V. Giannikas, Y. Han, E. Grover-Silva, and B. Qiao, "Using customer-related data to enhance e-grocery home delivery," *Industrial Management & Data Systems*, vol. 117, no. 9, pp. 1917–1933, 2017.

[137] K. S. Pasupathy and A. Medina-Borja, "Integrating Excel, Access, and Visual Basic to Deploy Performance Measurement and Evaluation at the American Red Cross," *Interfaces*, vol. 38, no. 4, pp. 324–337, 2008.

[138] M. L. Pinedo, *Scheduling*, 5th. Cham: Springer, 2016.

[139] D. Pisinger and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, vol. 34, no. 8, pp. 2403–2435, 2007.

[140] M. Polnik, A. Riccardi, and K. Akartunalı, "A multistage optimisation algorithm for the large vehicle routing problem with time windows and synchronised visits," *Journal of the Operational Research Society*, vol. 72, no. 11, pp. 2396–2411, 2021.

[141] J.-C. Pomerol and S. Barba-Romero, *Multicriterion Decision in Management: Principles and Practice* (International Series in Operations Research & Management Science), 1st. New York: Springer, 2000, vol. 25.

[142] D. Pretolani, "Multiple criteria decision analysis theory and tools for the sdewes index," *Journal of Sustainable Development of Energy, Water and Environment Systems*, vol. 8, no. 4, pp. 654–677, 2020.

[143] M. Punakivi and J. Saranen, "Identifying the success factors in e-grocery home delivery," *International Journal of Retail & Distribution Management*, vol. 29, no. 4, pp. 156–163, 2001.

[144] N.-H. Quttineh, T. Larsson, K. Lundberg, and K. Holmberg, "Military aircraft mission planning: A generalized vehicle routing model with synchronization and precedence," *EURO Journal on Transportation and Logistics*, vol. 2, no. 1-2, pp. 109–127, 2013.

[145] S. Rathi and R. Gupta, "Sensor Placement Methods for Contamination Detection in Water Distribution Networks: A Review," *Procedia Engineering*, vol. 89, pp. 181–188, 2014.

[146] N. M. Razali, "An Efficient Genetic Algorithm for Large Scale Vehicle Routing Problem Subject to Precedence Constraints," *Procedia - Social and Behavioral Sciences*, vol. 195, pp. 1922–1931, 2015.

[147] M. I. Restrepo, F. Semet, and T. Pocreau, "Integrated Shift Scheduling and Load Assignment Optimization for Attended Home Delivery," *Transportation Science*, vol. 53, no. 4, pp. 1150–1174, 2019.

[148] T. L. Saaty, "How to make a decision: The analytic hierarchy process," *European Journal of Operational Research*, vol. 48, no. 1, pp. 9–26, 1990.

[149] F. Sabouhi, A. Bozorgi-Amiri, M. Moshref-Javadi, and M. Heydari, "An integrated routing and scheduling model for evacuation and commodity distribution in large-scale disaster relief operations: A case study," *Annals of Operations Research*, vol. 283, pp. 643–677, 2019.

[150] R. Salman, F. Ekstedt, and P. Damaschke, "Branch-and-bound for the Precedence Constrained Generalized Traveling Salesman Problem," *Operations Research Letters*, vol. 48, no. 2, pp. 163–166, 2020.

[151] S. C. Sarin, H. D. Sherali, and A. Bhootra, "New tighter polynomial length formulations for the asymmetric traveling salesman problem with and without precedence constraints," *Operations Research Letters*, vol. 33, no. 1, pp. 62–70, 2005.

[152] M. W. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operations Research*, vol. 4, no. 1, pp. 285–305, 1985.

[153] V. Schmid and K. F. Doerner, "Examination and Operating Room Scheduling Including Optimization of Intrahospital Routing," *Transportation Science*, vol. 48, no. 1, pp. 59–77, 2014.

[154] M. Schneider, A. Stenger, and D. Goeke, "The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations," *Transportation Science*, vol. 48, no. 4, pp. 500–520, 2014.

[155] H. Seok Jeong and D. M. Abraham, "Operational Response Model for Physically Attacked Water Networks Using NSGA-II," *Journal of Computing in Civil Engineering*, vol. 20, no. 5, pp. 328–338, 2006.

[156] M. Sigurd, D. Pisinger, and M. Sig, "Scheduling Transportation of Live Animals to Avoid the Spread of Diseases," *Transportation Science*, vol. 38, no. 2, pp. 197–209, 2004.

[157] M. M. Silva, A. Subramanian, and L. S. Ochi, "An iterated local search heuristic for the split delivery vehicle routing problem," *Computers & Operations Research*, vol. 53, pp. 234–249, 2015.

[158] A. Snoeck, D. Merchán, and M. Winkenbach, "Revenue management in last-mile delivery: State-of-the-art and future research directions," *Transportation Research Procedia*, vol. 46, pp. 109–116, 2020.

[159] M. M. Solomon, "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.

[160] R. Spliet and G. Desaulniers, "The discrete time window assignment vehicle routing problem," *European Journal of Operational Research*, vol. 244, no. 2, pp. 379–391, 2015.

[161] R. Spliet and A. F. Gabor, "The Time Window Assignment Vehicle Routing Problem," *Transportation Science*, vol. 49, no. 4, pp. 721–731, 2015.

[162] A. Strauss, N. Gülpınar, and Y. Zheng, "Dynamic pricing of flexible time slots for attended home delivery," *European Journal of Operational Research*, vol. 294, no. 3, pp. 1022–1041, 2021.

[163] A. K. Strauss, R. Klein, and C. Steinhardt, "A review of choice-based revenue management: Theory and methods," *European Journal of Operational Research*, vol. 271, no. 2, pp. 375–387, 2018.

[164] A. Subramanyam, A. Wang, and C. E. Gounaris, "A scenario decomposition algorithm for strategic time window assignment vehicle routing problems," *Transportation Research Part B: Methodological*, vol. 117, pp. 296–317, 2018.

[165] P. Sun, L. P. Veelenturf, S. Dabia, and T. Van Woensel, "The time-dependent capacitated profitable tour problem with time windows and precedence constraints," *European Journal of Operational Research*, vol. 264, no. 3, pp. 1058–1073, 2018.

[166] K. T. Talluri and G. J. Van Ryzin, *The Theory and Practice of Revenue Management* (International Series in Operations Research & Management Science), 1st. New York: Springer, 2004, vol. 68.

[167] The Guardian, *How our home delivery habit reshaped the world*, Available at `https://www.theguardian.com/technology/2019/nov/21/how-our-home-delivery-habit-reshaped-the-world`, [Online; Accessed 18 November 2022], 2019.

[168] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*, 2nd. Philadelphia: SIAM, 2014.

[169] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian, "New benchmark instances for the Capacitated Vehicle Routing Problem," *European Journal of Operational Research*, vol. 257, no. 3, pp. 845–858, 2017.

[170] M. W. Ulmer, J. C. Goodson, D. C. Mattfeld, and B. W. Thomas, "On modeling stochastic dynamic vehicle routing problems," *EURO Journal on Transportation and Logistics*, vol. 9, no. 2, p. 100 008, 2020.

[171] M. W. Ulmer, B. W. Thomas, A. M. Campbell, and N. Woyak, "The Restaurant Meal Delivery Problem: Dynamic Pickup and Delivery with Deadlines and Random Ready Times," *Transportation Science*, vol. 55, no. 1, pp. 75–100, 2021.

[172] United Nations Environment Programme, *Application of the Sustainability Assessment of Technologies Methodology: Guidance Manual*, Available at `https://www.unep.org/resources/report/application-sustainability-assessment-technologies-methodology-guidance-manual-nov`, [Online; Accessed 18 November 2022], 2012.

[173] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. Van Oudheusden, "Iterated local search for the team orienteering problem with time windows," *Computers & Operations Research*, vol. 36, no. 12, pp. 3281–3290, 2009.

[174] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows," *Computers & Operations Research*, vol. 40, no. 1, pp. 475–489, 2013.

[175] T. Vidal, G. Laporte, and P. Matl, "A concise guide to existing and emerging vehicle routing problem variants," *European Journal of Operational Research*, vol. 286, no. 2, pp. 401–416, 2020.

[176] A. Vinsensius, Y. Wang, E. P. Chew, and L. H. Lee, "Dynamic Incentive Mechanism for Delivery Slot Management in E-Commerce Attended Home Delivery," *Transportation Science*, vol. 54, no. 3, pp. 567–587, 2020.

[177] S. A. Voccia, A. M. Campbell, and B. W. Thomas, "The Same-Day Delivery Problem for Online Purchases," *Transportation Science*, vol. 53, no. 1, pp. 167–184, 2019.

[178] G. Wang, S. H. Huang, and J. P. Dismukes, "Product-driven supply chain selection using integrated multi-criteria decision-making methodology," *International Journal of Production Economics*, vol. 91, no. 1, pp. 1–15, 2004.

[179] X. C. Wang, W. Kim, J. Holguín-Veras, and J. Schmid, "Adoption of delivery services in light of the COVID pandemic: Who and how long?" *Transportation Research Part A: Policy and Practice*, vol. 154, pp. 270–286, 2021.

[180] X. Wang and E. Wasil, "On the road to better routes: Five decades of published research on the vehicle routing problem," *Networks: An International Journal*, vol. 77, no. 1, pp. 66–87, 2021.

[181] Y.-M. Wang and Y. Luo, "On rank reversal in decision analysis," *Mathematical and Computer Modelling*, vol. 49, no. 5–6, pp. 1221–1229, 2009.

[182] N. Ware, S. Sing, and D. Banwet, "Supplier selection problem: A state-of-the-art review," *Management Science Letters*, vol. 2, no. 5, pp. 1465–1490, 2012.

[183] K. Waßmuth, C. Köhler, N. Agatz, and M. Fleischmann, "Demand Management for Attended Home Delivery – A Literature Review," *ERIM Report Series Reference Forthcoming*, 2022.

[184] C. A. Weber, J. R. Current, and W. Benton, "Vendor selection criteria and methods," *European Journal of Operational Research*, vol. 50, no. 1, pp. 2–18, 1991.

[185] C. de Winter, V. R. Palleti, D. Worm, and R. Kooij, "Optimal placement of imperfect water quality sensors in water distribution networks," *Computers & Chemical Engineering*, vol. 121, pp. 200–211, 2019.

[186] D. Wolfinger and J.-J. Salazar-González, "The Pickup and Delivery Problem with Split Loads and Transshipments: A Branch-and-Cut Solution Approach," *European Journal of Operational Research*, vol. 289, no. 2, pp. 470–484, 2021.

[187] World Health Organization, *Drinking-water*, Available at `https://www.who.int/news-room/fact-sheets/detail/drinking-water`, [Online; Accessed 18 November 2022], 2019.

[188] X. Yang and A. K. Strauss, "An approximate dynamic programming approach to attended home delivery management," *European Journal of Operational Research*, vol. 263, no. 3, pp. 935–945, 2017.

[189] X. Yang, A. K. Strauss, C. S. Currie, and R. Eglese, "Choice-Based Demand Management and Vehicle Routing in E-Fulfillment," *Transportation Science*, vol. 50, no. 2, pp. 473–488, 2016.

[190] B. Yildiz and M. Savelsbergh, "Provably High-Quality Solutions for the Meal Delivery Routing Problem," *Transportation Science*, vol. 53, no. 5, pp. 1372–1388, 2019.

[191] B. Yildiz and M. Savelsbergh, "Pricing for delivery time flexibility," *Transportation Research Part B: Methodological*, vol. 133, pp. 230–256, 2020.

[192] Q. Zhang, Z. Y. Wu, M. Zhao, J. Qi, Y. Huang, and H. Zhao, "Leakage Zone Identification in Large-Scale Water Distribution Systems Using Multiclass Support Vector Machines," *Journal of Water Resources Planning and Management*, vol. 142, no. 11, p. 04 016 042, 2016.