UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

CICLO
XXXVI

Rich Vehicle Routing Problems: models, algorithms and applications

Dottorando:
Valentina BONOMI

Relatore:
Prof.ssa Renata MANSINI

Settore scientifico disciplinare:
MAT/09 - Ricerca Operativa

UNIVERSITÀ DEGLI STUDI DI BRESCIA

Abstract

Dipartimento di Ingegneria dell'Informazione
MAT/09 - Ricerca Operativa

Dottorato di ricerca in Ingegneria dell'Informazione

Rich Vehicle Routing Problems: models, algorithms and applications

by Valentina BONOMI

The Vehicle Routing Problem (VRP) is one of the most central transportation problems in the field of Operations Research. Introduced by Dantzig and Ramser, 1959, the problem aims at finding the optimal routes for a fleet of vehicles in order to serve a set of costumers. The traditional version of the VRP and its variants have been extensively studied in the academic literature. However, recent years have witnessed a surge in the application of optimization models by businesses and organizations. This shift in focus aims to address real-world complexities by introducing novel features and constraints. The family of these extended problems is called Rich Vehicle Routing Problems (RVRPs). RVRPs extend the traditional academic formulations of VRPs by incorporating problem-specific constraints that closely mirror decisions made at both tactical and operational levels in practical settings. In this thesis, we delve into the study of RVRPs in the field of healthcare and logistics providing efficient mathematical model formulations, exact and heuristic resolution approaches and a comprehensive analysis of the computational results.

In the healthcare domain, our research focuses on addressing critical issues within the Nurse Routing Problems (NRPs). Our goal is to enhance logistic outcomes for healthcare organizations while simultaneously improving the working conditions of healthcare providers and the quality of care delivered to patients. To achieve this, we introduce the concept of *fairness* into NRPs, along with quality-enhancing constraints such as nurse-patient consistency and time window specifications. Our analysis begins by examining several fairness metrics, considering both patients and nurses, within a Single-Objective Single-Period NRP framework. We provide a set of objective functions that can be interchanged to assess the interaction between different metrics and their cost implications. Next, we extend our investigation on fairness inserting new measures and providing a Multi-Objective formulation of the previous NRP. Employing a lexicographic approach, we simultaneously consider multiple objective functions, selecting triplets of functions to represent the interests of each stakeholder (hospital, nurses, and patients). Furthermore, we present a Dynamic Multi-Period NRP with Consistency Constraints in which temporal distribution of patients requests is unknown. Objective of the problem is to decide which nurse visit wich patients over several days based on new requests daily revealed. We propose two approaches: a pure myopic dynamic method, which lacks future event information, and a scenario-based optimization method that leverages historical data to forecast future developments.

In the logistics domain, we propose two key problem formulations: the Last Mile Logistic Delivery Problem with Parcel Lockers (LMDP-LS) and the Attended Home Delivery Problem with Recovery Options (AHDP-RO). In the LMDP-LS we evaluate the environmental impact of parcel lockers when the ecological footprint of consumers is taken into account. The problem has the objective of deriving meaningful insights on the environmental impact of both the company and the consumers in the switch from a door-to-door delivery service to a locker-based one. Additionally, in the AHDP-RO, we model and solve the traditional attended home delivery service, which mandates the customer's presence at home to avoid delivery failures. Specifically, we model the probability of finding the customer at home through Availability Profiles (APs) plotting the probability of successful deliveries during the working day. We introduce the possibility for couriers to take recovery actions when customers are unavailable, with associated penalties included in the objective function. The overarching objective is to plan daily courier routes to minimize both routing and penalty costs. Throughout our research, we provide the results of small-size instances solved to optimality and we employ the Adaptive Large Neighborhood Search (ALNS) meta-heuristic to obtain good quality solutions for large-size instances. Our heuristic results are compared with the one obtained by the commercial solver Gurobi.

Versione italiana: I Vehicle Routing Problems (VRPs) sono una branca di problemi centrali nella Ricerca Operativa. Introdotto da Dantzig et al (1959), il problema mira a trovare le rotte ottimali per una flotta di veicoli per servire i clienti. Negli ultimi anni si è assistito a un incremento nell'applicazione dei modelli di ottimizzazione da parte di aziende e organizzazioni. Questo cambiamento di focus mira ad affrontare le complessità del mondo reale introducendo caratteristiche e vincoli innovativi. La famiglia di questi problemi estesi è chiamata Rich VRP (RVRPs). I RVRPs estendono le formulazioni tradizionali dei VRP incorporando vincoli specifici del problema che riflettono decisioni prese sia a livello tattico che operativo in contesti pratici. Questa tesi approfondisce lo studio dei RVRPs nel settore sanitario e logistico. Forniamo formulazioni di modelli matematici efficienti, approcci di risoluzione esatti ed euristici, e un'analisi comprensiva dei risultati computazionali. La nostra ricerca affronta questioni critiche all'interno dei Nurse Routing Problems (NRPs) nel settore sanitario. Il nostro obiettivo è migliorare i risultati logistici per le organizzazioni sanitarie migliorando contemporaneamente le condizioni di lavoro dei fornitori di assistenza e la qualità dell'assistenza fornita, Per raggiungere questo scopo, introduciamo il concetto di equità negli NRPs, insieme a vincoli che migliorano la qualità come la coerenza infermiere-paziente e le specifiche delle finestre temporali. La nostra analisi inizia esaminando diverse metriche di equità, fornendo un insieme di funzioni obiettivo che possono essere scambiate per valutare l'interazione tra diverse metriche e le loro implicazioni sui costi. Successivamente, estendiamo la nostra indagine sull'equità inserendo nuove misure e fornendo una formulazione Multi-Obiettivo del precedente NRP in cui selezioniamo triplette di funzioni per rappresentare gli interessi di ogni stakeholder. Inoltre, presentiamo un NRP Dinamico Multi-Periodo con Consistenza in cui la distribuzione temporale delle richieste dei pazienti è sconosciuta. L'obiettivo del problema è decidere le assegnazioni infermiere-paziente per diversi giorni basandosi su richieste rivelate giornalmente. Proponiamo due approcci: un metodo dinamico puramente miope, che manca di informazioni sugli eventi futuri, e un metodo di ottimizzazione basato su scenari che sfrutta i dati storici per prevedere sviluppi futuri. Nel dominio logistico proponiamo il Last Mile Delivery Problem with Locker Selection (LMDP-LS) e l'Attended Home Delivery Problem with Recovery Options (AHDP-RO). Nel LMDP-LS, valutiamo l'impatto ambientale dei locker per pacchi tenendo conto dell'impronta ecologica dei consumatori. Nell'AHDP-RO, modelliamo e risolviamo anche il servizio tradizionale di consegna a domicilio assistita, modellando la probabilità di trovare il cliente a casa attraverso profili di disponibilità e tracciando la probabilità di consegne riuscite durante la giornata lavorativa. Introduciamo la possibilità per i corrieri di intraprendere azioni di recupero quando i clienti sono indisponibili, con penalità associate incluse nella funzione obiettivo. L'obiettivo generale è pianificare le rotte giornaliere dei corrieri per minimizzare i costi di routing e di penalità. Nel corso della nostra ricerca, forniamo i risultati di istanze di piccole dimensioni risolte all'ottimalità e impieghiamo la meta-euristica di Adaptive Large Neighborhood Search (ALNS) per ottenere soluzioni di buona qualità per quelle di grandi dimensioni.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The global transportation industry, as a pivotal component of the world's economy, is projected to experience significant growth, with a Compounded Average Growth Rate (CAGR) of 3.6% through 2030, culminating in a value of approximately $8.9 trillion by 2030 (ReportLinker, 2023). This growth trajectory underscores the vital role of an efficient transportation and logistics system in driving economic success and sustainability, especially in a world increasingly shaped by technological advancements and escalating demands. The challenges emerging from this growth span various transportation industry sectors, including logistics, healthcare, rail, trucking, infrastructure, and passenger vehicles. This expansion has been further accelerated by the change in consumer habits over recent years, a direct consequence of the Covid-19 pandemic. The 2022 European E-Commerce Report (Lone and Weltevreden, 2022) highlights a significant shift in this trend, with e-commerce in Europe experiencing a 6% increase in B2C sales compared to 2019. It is estimated that around 75% of internet users purchased products or services online during 2022, exerting a profound impact on the transportation dynamics within the delivery sector. Moreover, the transportation needs within the healthcare sector have been influenced mainly by the pandemic, which has reshaped how transportation facilitates access to healthcare (Chen et al., 2021). This has led to the emergence of new trends in home healthcare businesses. However, this switch in habits has also highlighted environmental concerns. According to the European Environment Agency, transportation is responsible for about a quarter of the EU's total greenhouse gas emissions, with road transport being one of Europe's leading contributors to environmental noise pollution. This sector is Europe's only central economic area where greenhouse gas emissions have increased since 1990. In the face of these evolving demands and environmental implications, traditional Vehicle Routing Problem (VRP) models are increasingly inadequate in capturing the multifaceted nature of these challenges. The necessity for Rich Vehicle Routing Problems (RVRP) has emerged, offering more sophisticated and adaptable solutions to accommodate these new trends and effectively manage the complexities of the contemporary transportation sector.

These advanced RVRP models are essential to optimize transportation operations, balancing efficiency and sustainability in an environment of rapid change and growth. This thesis aims to investigate the models and the algorithms of RVRPs in two main domains: the home healthcare sector and the last-mile delivery process. We first delve into the healthcare part by describing three different applications of a Nurse Routing Problem (NRP) studied from multiple aspects. We will investigate the role of fairness, intended to create an equal and impartial environment for all the stakeholders involved through a single and multi-objective formulation. Then, we propose a stochastic and dynamic Consistent VRP, and we solve it using two different problem-tailored heuristics. The second part addresses the Last-Mile Delivery Problems (LMDPs) in door-to-door and locker-based systems. Many practical constraints are considered, such as the client's presence at home, managing failed deliveries, or the computation of the ecological impact of both clients and couriers.

Contributions

In what follows, we summarise the main contributions we achieve in this thesis.

- We have described, formulated, and efficiently solved, exactly and heuristically, a multitude of Vehicle Routing Problems (VRPs), incorporating a range of practical constraints. This comprehensive approach has provided the necessary tools to adapt VRPs to various domains, including but not limited to home care and home delivery services. This has allowed us to tailor VRP solutions that are theoretically sound and practically viable, ensuring that they can address the specific logistical and operational demands of the transportation sector.

- We have characterized the fairness needs of stakeholders involved in a nurse routing problem through meticulous mathematical modeling. This detailed modeling has enabled us to capture the complex interplay of objectives within and across different stakeholders. We have developed both single-objective and multi-objective approaches to provide a comprehensive overview of the intricate relationships among these often conflicting objectives. The analysis conducted in this study extends beyond the examination of objectives specific to individual stakeholders. Instead, it adopts a broader perspective, exploring how different stakeholders' objectives interact.

- We have developed a novel meta-heuristic approach (`ParallelALNS`) capable of efficiently handling multi-actor and multi-objective problems. The results of our study demonstrate how this heuristic outperforms commercial solvers in various scenarios.

- We have conducted an in-depth analysis of last-mile delivery problems from various aspects. Our focus has been optimizing the logistical side by developing an operational model to minimize costs and maximize the delivery hit rate. This model considers the likelihood of customers being at home, ensuring a more efficient delivery process. Additionally, we have explored the environmental side of these problems. Our findings demonstrate that a locker-based solution is efficient only when customers are environmentally conscious. Combining logistical efficiency with environmental considerations, this dual approach highlights the complexity of last-mile delivery challenges and provides a comprehensive framework for addressing them effectively.

Structure of the thesis

In this section, we summarise the content of the thesis in more detail. The thesis comprises eight chapters, introduction and conclusions excluded, divided into two main parts: Part I refers to healthcare applications, and Part II refers to the logistics. Table (1.1) reports the primary information about each chapter's content. Each row corresponds to a chapter. The columns specify the primary sector, the model name, and if the problem has been solved through an exact or a heuristic solutions method.

TABLE 1.1: content of the chapters

	Sector	Model Name	Exact	Heuristic
Chapter (2)		Literature Review		
Chapter (3)		Solution Approaches		
Chapter (4)	Healthcare	NRP	yes	-
Chapter (5)		NRP	yes	ALNS meta-heuristic
Chapter (6)		SMHHP-C	yes	ALNS meta-heuristic
Chapter (7)		Literature Review		
Chapter (8)	Logistic	LR-LMDP	yes	-
Chapter (9)		AHDP-RO	yes	-

In what follows, we describe the content of each chapter in more detail.

- In Chapter (2) and Chapter (7), we review the main literature applications of RVRPs in healthcare and logistics, respectively.

- Chapter (3) presents the literature and the general framework of the main solution methods that will be implemented in Chapter (5) and Chapter (6). We present the meta-heuristic Adaptive Large Neighborhood Search (ALNS) and two dynamic approaches, the Myopic Dynamic Heuristic (MDH) and the Multi-Scenario.Based Progressive Fixing (MSB-PF).

- Chapters from (4) to (6) include the healthcare applications. Precisely, in Chapter (4) and Chapter (5), we study the insertion of fairness measures in the NRP. Then, Chapter (6) presents the Stochastic and Dynamic Home Healthcare Problem with consistency constraints. Chapter (4) is published as a conference paper on IFAC-PapersOnLine. Chapter (5) is published on the *International Journal of Production Research*.

- Chapters (8) and Chapter (9) are dedicated to the logistic applications. In Chapter (8), we study the environmental impact of a Location-Routing Problem in last-mile delivery, including the presence of parcel lockers to collect packages. In Chapter (9) we study an Attended Home Delivery Problem in the operational efficiency is reached by minimizing both the traveling and the failed delivery costs. The latter depends on the customers' presence at home and the recovery option the courier has to perform in case of failed delivery.

- Finally, Chapter (10) presents the main conclusions.

Part I

Rich Vehicle Routing Problems in Healthcare Applications

This part explores the application of Rich Vehicle Routing Problems (RVRPs) in the Healthcare sector. Specifically, our research addresses critical challenges in the Nurse Routing Problems (NRPs) - a variant of the Vehicle Routing Problems that involves nurses or healthcare workers as vehicles and patients as nodes to visit. Our objective is to identify and explore the challenges that arise from real-life scenarios by introducing realistic settings and constraints. As a result, our focus is on optimizing a service provider's costs while considering the working conditions of the nurses and the quality of care provided to patients. After an extensive review of the literature in Chapter (2), we introduce the concept of fairness in the NRP formulation. Chapter (4) explores this idea further with a Single-Objective NRP that addresses a multi-actor implementation of fairness. Moreover, we extend our exploration of fairness by introducing novel metrics and formulating a Multi-Objective variant of the preceding problem in Chapter (5). In addition, we present a Dynamic Multi-Period NRP with Consistency Constraints in Chapter (6). This problem deals with the temporal distribution of patient requests, which remains unknown. The goal is to make informed decisions regarding nurse-patient assignments across several days, with new requests unveiled daily. For each work, we provide a general description of the problem and the specific setting, followed by the mathematical model and a comprehensive analysis of the computational outcomes on both exact and heuristic approaches.

Chapter 2

Literature Review

Contents

In this chapter we present the relevant literature in the domain of RVRPs in the Home Healthcare (HCC) domain. In particular, our review mainly focuses on the articles dealing with Nurse Routing Problems (NRPs) including features in common with the works introduced in this chapter.

The study is divided as follows. In Section (2.1) we present a comprehensive review of the variations of NRPs in literature with particular attention on works including fairness and multi-actor formulations; we highlight how it is missing a contribution treating fairness in a multi-objective and multi-actor framework. Then, Section (2.2) delves into healthcare applications of hierarchical Multi-Objective approaches. Finally Section (2.3) reviews the most relevant works dealing with Vehicle Routing Problems including the service-level consistency constraint.

2.1 Nurse Routing Problem Variants Including Fairness

The home healthcare sector has various issues that need to be addressed for an efficient and functional optimization of the overall system, from staff rostering to visit scheduling and service allocation. The reader is referred to Grieco et al., 2020 for an extensive survey about home healthcare optimization. To our knowledge, there are only a few works that include fairness and most of them approach it as a secondary objective in addition to cost minimization. Frequently, fairness is associated with providing low-cost, timely, and high-quality services to patients by maximizing their satisfaction and/or minimizing their waiting time. Among works addressing fairness, the majority presents a single-objective formulation including the presence of stakeholders other than the hospital only as additional constraints. Table (2.1) provides a detailed overview of various NRP variants as reported in known literature. This table captures the essential features of each variant, including their solution approaches. The 'Obj' column specifies the objective function aimed for optimization, which varies from minimizing total time (TT) or cost (TC), total tardiness (TTA), number of unattended clients (NC), and time window violations (TWV) for patients, to maximizing workload balance (WB), consistency (CON), profit per visit (PP), nurse skill alignment (NS) with visits, task management flexibility (TF), nurse preferences (NP), and patient satisfaction (PS) using certain problem parameters. The 'Multi-obj' column is marked with a checkmark to indicate multi-objective problem formulation. Resolution methods are shown in brackets: epsilon-constraint method (ϵ-const), and lexicographic method (lex). The 'Multi-actor' column notes if multiple stakeholders are involved, while 'Problem Type' categorizes each as deterministic (Det), stochastic (Stoch), or dynamic (Dyn). The 'Approach' column distinguishes between exact (Exact) and heuristic (Heur) solution methods. Table (2.2) compiles key constraints from these studies. The 'Fairness' column underscores equitable considerations for patients (P), nurses (N), or the hospital/TOC (H/T). The 'Horizon' column distinguishes between single-period and multi-period problems. The 'Serv Freq' column indicates if patient requests are for single or multiple appointments over the planning period. The 'Serv Type' column classifies services as either heterogeneous (heterog) or homogeneous (homog). The presence of consistency or coordinated visits by multiple nurses is marked by a checkmark in the 'Cons' and 'Synchro' columns. Finally, the 'TW' and 'Skills' columns address the type of time windows (hard or soft) for patient visits and the integration of nurse skill considerations.

TABLE 2.1: NRP variants: problem type and solution methodology.

Article	Obj	Multi-Obj	Multi-Actor	Problem Type	Approach
Liu et al., 2016	TT,NC	–	–	Det	Exact
Carello et al., 2018	TT,NP,CON	✓ (ϵ-const)	✓	Det/Stoch/Dyn	Exact
Decerle et al., 2019	TT,LST,PP,WB	–	✓	Det	Heur
Mosquera et al., 2019	TF	✓ (lex)	✓	Det	Heur
Gobbi et al., 2019	PP	–	–	Det	Heur
Khodabandeh et al., 2021	TT,NS	✓ (ϵ-const)	✓	Det	Exact
Bhattarai et al., 2022	TC, LST, WB	✓ (ϵ-const)	✓	Det	Exact
Bonomi et al., 2022	TT,CON,TWV,NS,PS	–	✓	Det	Exact
Jiang et al., 2023	CON,WB	✓	✓	Det	Heur
Belhor et al., 2023	TT,TWV	✓ (lex)	–	Det	Exact/Heur
Gobbi et al., 2022	PP	–	–	Det	Heur
Current work	TT,TTA,TWV,CON,WB,NS	✓ (lex)	✓	Det	Exact/Heur

In the realm of nursing research, recent literature over the past five years has revealed a notable gap in studies that simultaneously consider multiple stakeholders' perspectives and a variety of fairness measures. This overview

TABLE 2.2: NRP variants: constraint types.

			Patients				Nurses		
Authors	Fairness	Horizon	Serv Freq	Serv Type	Cons	TW	Syncro	Skills	
Liu et al., 2016	–	single	single	homog	–	hard	–	–	
Carello et al., 2018	✓ (H,N,P)	multi	multi	heterog	✓	–	–	✓	
Decerle et al., 2019	✓ (N,P)	multi	multi	heterog	–	soft	✓	✓	
Mosquera et al., 2019	–	multi	multi	heterog	–	hard	–	✓	
Gobbi et al., 2019	–	single	multi	heterog	–	hard	–	–	–
Khodabandeh et al., 2021	✓ (N)	single	single	heterog	–	hard	–	✓	
Bhattarai et al., 2022	✓ (H,N,P)	multi	multi	homog	–	hard	✓	✓	
Bonomi et al., 2022	✓ (N,P)	single	multi	heterog	✓	soft	–	✓	
Jiang et al., 2023	✓ (N)	multi	multi	homog	✓	–	–	–	
Belhor et al., 2023	✓ (P)	single	single	homog	–	soft	–	–	
Gobbi et al., 2022	–	single	multi	heterog	–	soft	–	–	
Current work	✓ (T,N,P)	single	multi	heterog	–	–	–	–	

reorders the cited articles while maintaining their thematic relevance. The literature shows a varied approach regarding the stakeholder perspectives and fairness measures. For instance, Carello et al., 2018 stands out for its inclusive approach towards stakeholders' requirements. The study proposes mathematical models to explore the effects of prioritizing different stakeholders, focusing on quality of service for patients, workload balance among operators, and cost minimization for service providers. Bhattarai et al., 2022, while addressing the role of multiple actors, limits its scope to a single measure for each stakeholder. This contrasts with the broader set of measures used in the current article to analyze stakeholder perspectives. Several studies have focused on specific stakeholder needs in the context of fairness. Belhor et al., 2023 emphasizes patients' needs, aiming to minimize total tardiness relative to visiting time preferences and overall service time. Conversely, Khodabandeh et al., 2021 and Jiang et al., 2023 center on nurses' requirements. The former integrates the minimization of the gap between actual and potential skills of nurses with the minimization of total travel time, while the latter employs a heuristic approach to maximize workload balance and continuity of care. The work of Decerle et al., 2019 proposes a memetic algorithm for an NRP, balancing caregivers' workload to minimize total working time and enhance service quality. Mosquera et al., 2019 introduces a novel perspective on task flexibility, encompassing flexible scheduling, duration, and caregivers' workload. From a resolution approach standpoint, Liu et al., 2016 contributes a Branch-and-Cut algorithm for a worker scheduling and routing problem, including lunch break considerations. Building on the foundational work of Manerba and Mansini, 2016 in the Nurse Routing Problem with Workload Constraints and Incompatible Services, Gobbi et al., 2019 extends the problem to allow for multiple services required by patients in a single day and integrates minimum demand requirements. Their research aims to maximize visit numbers and optimize patient rewards, employing a hybrid matheuristic approach enhanced by Kernel Search for improved solution space exploration.

2.2 Multi-Objective Approaches

Multi-objective problems (MOPs) involve the challenging task of optimizing several, often conflicting, objectives concurrently. Various methodologies have been developed to address MOPs, which are well-documented in the literature. The $\epsilon-$constraint method, a prominent approach where one objective is optimized while others are restricted by pre-set values ϵ is one of the most applied. This technique is instrumental in generating the Pareto frontier, as exemplified in works by Carello et al. 2018, Khodabandeh et al. 2021, and Bhattarai et al. 2022. The scalarization or weighted sum method, which consolidates multiple objectives into a singular goal through weighted summation. The hierarchical or lexicographic method, prioritizes objectives based on their perceived importance or decision-maker preference. Additionally, various alternative strategies are employed for solving MOPs, such as goal programming Tirkolaee et al., 2020, evolutionary algorithms Ghannadpour and Zarrabi, 2019, and non-dominated sorting methods Long et al., 2021. For an extensive review, Gunantara, 2018 thoroughly analyzes Multi-Objective Optimization. Hierarchical multi-objective optimization (H-MOO) particularly addresses MOPs by prioritizing objectives based on their relative importance. This approach seeks optimal solutions at each hierarchical level, starting from the highest priority objective. H-MOO is beneficial for its ability to simplify multi-objective problems into a series of single-objective ones, and its flexibility in reordering objectives. The healthcare sector frequently utilizes MOO, often adopting a hierarchical approach. Examples include Mosquera et al., 2019, which applies hierarchical optimization for home care visit scheduling to enhance task flexibility; Malagodi et al., 2021, employing a lexicographic method for home health care (HHC) scheduling with an emphasis on stakeholder preferences; and Belhor et al., 2023, which combines lexicographic and evolutionary algorithm approaches. Our research departs from these models by not fixing the hierarchy of objectives. Instead, we explore various combinations of objectives related to different stakeholders to understand their inter-impact. We introduce a novel parallel ALNS designed for multi-objective, multi-actor problems. Beyond healthcare, hierarchical approaches have been successfully applied in other domains, such as surgery scheduling (Al Hasan et al., 2018), pick-up and delivery services (Al Chami et al., 2021), and workforce allocation in scheduling perishable products (Bolsi et al., 2022).

2.3 The Consistent Vehicle Routing Problem

Rich VRP variants introduce real-world constraints, leading to more complex formulations like the Consistent VRP (ConVRP) and Capacitated VRP

(CVRP). ConVRP, particularly relevant when customer satisfaction is essential, strives to create cost-efficient routes while adhering to specific customer requirements. In the context of ConVRP, Groër et al., 2009 emphasizes consistency in both visit timing and the driver serving a patient. Kovacs et al., 2015a extends this concept by treating consistency as soft constraints, where any violations are penalized in the objective function. For an overview of ConVRP resolution methods, Goeke et al., 2019 provides a comprehensive analysis. A novel approach is seen in Mancini et al., 2021, where consistency is integrated into a CVRP framework, allowing for vehicle collaboration and customer sharing to enhance profit. Our study focuses on a HHC-oriented ConVRP, which considers nurse-patient consistency constraints, ensuring each customer is visited by no more than one driver. The significance of HCC problems has surged recently, especially with the planning and optimizing medical services at home becoming a competitive edge for hospitals and enhancing patient satisfaction. The Covid-19 pandemic has further underscored the importance of efficient HHC management to maintain social distancing and cater to vulnerable patient groups. In this context, Allen et al., 2020 presents a model simulating a worst-case Covid-19 scenario's impact on an outpatient dialysis network. The authors apply a Monte-Carlo VRP model to test the transportation plan's resilience, aiming to transport more patients simultaneously. Pacheco and Laguna, 2020 models a VRP for the urgent delivery of face shields in Burgos, Spain, incorporating pick-up and delivery dynamics. Service and time consistency in HHC problems are explored in Grenouilleau et al., 2019, focusing on service-level improvement by maximizing the number of patients served. Similarly, Demirbilek et al., 2018 aims to maximize daily nurse visits, implementing strict time window constraints. To our knowledge, the works that most resemble the application of consistency of our problem are the ones from Cappanera and Scutellà, 2021 and Demirbilek et al., 2019. In Cappanera and Scutellà, 2021 the authors address consistency and demand uncertainty in Home Care planning by proposing a mathematical model and alternative policies to develop a pattern-based algorithmic framework. However, in the resolution method, they address uncertainty through a robust approach. Demirbilek et al., 2019 proposes a new heuristic approach for routing and scheduling multiple nurses in home healthcare, considering real-life aspects such as clustered service areas and skill requirements. The authors propose a new heuristic based on generating several scenarios, including current schedules of nurses, new requests, and randomly generated future requests, to solve three decision problems: patient acceptance, nurse assignment, and assignment of visit days and times. The approach is compared with a greedy heuristic from the literature and empirically demonstrates higher average daily visits and shorter travel times.

Chapter 3

Solution Methods

Contents

The inherent complexity of Rich Vehicle Routing Problems often renders exact methods impractical or overly time-consuming when solving large instances. Heuristic approaches can give a trade-off between solution quality and computational efficiency, operating on the principle of taking efficient problem-tailored paths to reach good quality, if nonoptimal, solutions. This chapter aims to outline the heuristic resolution techniques utilized in this thesis. While the chapter offers a comprehensive overview of the fundamental methods, detailed adaptations tailored to specific problems will be elaborated upon in the following chapters, alongside their respective problems. The chapter is divided as follows. First, Section (3.1) introduces the fundamentals of Local Search Algorithms, highlighting the principal characteristics. Then, Section (3.2) describes the Adaptive Large Neighborhood Search (ALNS) framework. Section (3.3) shifts the focus to dynamic approaches that, in contrast with the so-called Offline approach, address problems with uncertainty in specific data sets or problems where data are not completely know but become available as time goes on. Within this section, two distinct dynamic approaches are presented. Section (3.3.1) describes the Myopic Dynamic Heuristic (MDH), which operates without foresight in future events. Meanwhile, Section (3.3.2) explores a dynamic strategy incorporating anticipated future events into the solution process by using scenarios.

3.1 Large Neighborhood Search (LNS)

The Large Neighborhood Search (LNS) is a meta-heuristic presented by Shaw, 1998 and belongs to the class of Large-Scale Neighbourhood Search (VLSN) heuristics. LNS is built around efficiently examining the *neighborhood* of a solution by iteratively deconstructing and rebuilding it, aiming to find a better solution or exit from local optima. To have a formal definition of neighborhood, let $P = (\{, S)$ be an optimization problem, S the set of all the feasible solutions for P and $\{: S \leftarrow \mathcal{R}$ its objective function. The Neighborhood function $N : S \leftarrow 2^{|}S|$ defines for each solution $i \in S$ the neighborhood $N(i) \subseteq S$ set of all the solutions close to i. $N(i)$ comprises solutions derived by applying a specific move to a solution $s \in S$. In the LNS, the way the neighborhood search is carried out is defined by a set of *destroy* and *repair* functions. A destroy function $d(s)$ removes part of the elements present in the current solution, while a repair method $r(d(s))$ reconstructs it by reinserting them following problem-specific criteria. As shown in Algorithm (1), the method inputs an initial feasible solution s, and, at each iteration, destruction, and repair operators are applied to obtain a new solution s'. This solution is then submitted to an *acceptance criteria*(s', s) that allows the method to accept non-improving solutions to widen the search in the neighborhood and exit from possible local optima. If accepted, the new solution becomes the current solution s', and the method is repeated until the termination criteria are met, returning the best solution s^*. The best solution is evaluated through the cost function $c(s)$ mapping for each solution in the solution space and its cost. We present the case of a minimization problem where a solution s^* is better than s if $c(s^*) < c(s)$. Regarding routing problems, we will delve into the most commonly used destroy and repair operators in Section (3.1.1) and Section (3.1.2), respectively. A detailed examination of the acceptance criteria is given in Section (3.1.4)

Algorithm 1 Large Neighborhood Search

Require: a feasible solution s

$s^* \leftarrow s$
while stopping criteria not met **do**
    $s' = r(d(s))$
    if accept(s', s) **then**
        $s \leftarrow s'$
        if $(c(s) < c(s^*))$ **then**
            $s^* \leftarrow s$
        end if
    end if
end while
return s^*.

3.1.1 Destroy Operators

Destroy functions deconstruct part of the initial solution. One key parameter for an effective search is the *degree of destruction* q, representing the number of removed elements from the current solution. If this degree is low, the heuristic explores only a small part of the space and might have problems in finding improving solutions. If the parameter is too high, there is the risk of a total re-optimization, losing the link with the initial solution. Since every destroy heuristic works on any degree of destruction, the setting and performance of the operator according to the value of q may vary depending on problem-specific characteristics. In Shaw, 1998, the value is gradually increased at each iteration between two values q_{min} and q_{max}, while Ropke and Pisinger, 2006 choose it randomly inside a predefined range. Some examples of general destruction heuristics are now presented. Then, Chapter (5) and Chapter (6) contain a more detailed adaptation to our specific problems.

Random Removal

This heuristic randomly selects q elements (e.g., routes, nodes, or requests) without considering their role inside the solution. It is primarily practical when diversification is needed and when it is paired with other removal heuristics, as is the case of the framework shown in Section (3.2).

Worst Removal

The idea behind implementing this heuristic is to remove the q elements that most adversely affect the solution—specifically, those that most degrade the objective function's value. An element's cost is typically determined by comparing the solution's total cost with and without that particular element. Taking a solution s with objective $f(s)$ and an element r, the cost of the element is computed as $C(r,s) = f(s) - f(s^{-r})$, where $f(s^{-r})$ is the objective of the solution without the element r. Then, the elements are sorted in descending order according to their cost; the first is removed, and the costs are updated. The process is iterated until q elements are removed.

Shaw or Related Removal

This heuristic is initially presented in Shaw, 1998. A number q of elements is removed based on their similarity. Removing similar elements increases the chances for the insertion heuristic to reinsert them in shuffled positions with a little computational cost. How the similarity is computed is highly problem-tailored. In Shaw, 1998 the presented problem is a Vehicle Routing Problem where the relatedness function is:

$$R(i,j) = \frac{1}{c_{ij} + V_{ij}} \tag{3.1}$$

where c_{ij} is the traveling distance to go from node i to node j normalized in the range [0,1], and V_{ij} is a parameter taking value 1 if the two nodes are served by the same vehicle, 0 otherwise. In this case, relatedness depends on the proximity of the visits and whether they are inserted in the same route or not.

3.1.2 Repair Operators

In Shaw, 1998, the authors propose a Branch-and-Bound heuristic to reinsert removed nodes. At the same time, Ropke and Pisinger, 2006 implements less precise but less time-consuming heuristics presented in this section.

Greedy Insertion

This method calculates the cheapest insertion position for each element that has been previously removed. The insertion cost is computed as the difference between the solution cost with and without the inserted part. The elements are then ordered in non decreasing cost and inserted starting from the first one, computing the new insertion cost at each iteration. It provides a rapid way to construct or repair solutions, ensuring each step adheres to a locally optimal criterion. However, like all greedy methods, it might suffer from shortsightedness, potentially getting trapped in local optima. In particular, it postpones the insertion of high-cost elements to the end of the algorithm, where fewer inserting possibilities are available. Taking a routing problem with N nodes as an example, let us define as $\Delta_{f_i^k}$ as the cost of inserting node i into its $k-th$ best route. A greedy insertion would insert the node $j \in N$ that produces the least increase of costs, i.e.

$$j = arg\ min_{i \in N} \Delta f_i^1 \tag{3.2}$$

Regret Heuristic

To overcome the greedy nature of the previous heuristic, the regret heuristic incorporates the regret value of not inserting an element in its best position compared to its second-best, third-best, and so on. A Regret-`n` heuristic generally calculates the element with the most significant cost difference, hence the regret, between the cheapest and the $n-1$ position. The highest regret element is the one with the highest difference, thus the one for which a later insertion might cause the most significant additional cost. Differently from the greedy heuristic, in a routing problem the regret considers the difference in terms of costs between inserting a node into its best route or in its k-th best. Hence, a Regret-`k` heuristic inserts node j as:

$$j = arg\ max_{i \in N} (\sum_{h=2}^{k} \Delta f_i^h - \Delta f_i^1) \tag{3.3}$$

3.1.3 Initial Solution

The initial solution given as input to the algorithm is strictly problem-related. It can be obtained through a mathematical model or a heuristic method like a greedy constructive algorithm.

3.1.4 Acceptance Criterion

In LNS, the acceptance criterion is a critical factor that strongly influences the quality of the solutions obtained. It is instrumental in enabling the algorithm to accept non-improving solutions, which can be vital in escaping from local optima and exploring the search space more effectively. Several different acceptance criteria can be used in LNS, and the choice of criterion can significantly impact the algorithm's performance. One of the most famous ones is the acceptance rule used by *Simulated Annealing*, first presented by Laarhoven and Aarts, 1987. The method mimics the man-made process of metal annealing in which the metal is melted and then slowly cooled down with a steady decrease in temperature. Based on this, the heuristic idea is to accept a worsening solution with a probability related to a current temperature T, that starting from an initial value T_0 is decreased at each iteration. This means that at the initial steps, the probability of accepting non-improving solutions is higher and then reduces. This allows the algorithm to refine the search at the end, focusing primarily on improving solutions. More precisely, given a current solution s, the probability of accepting an obtained solution s' is computed as

$$e^{-\frac{f(s')-f(s)}{T_n}}. \tag{3.4}$$

In Equation (3.4), $f(s') - f(s)$ represents the objective function degradation, and T_n is the time-dependent global variable representing the temperature at iteration n that can be computed starting from T_0 as

$$T_n = cT_{n-1} = c^n T_0 \tag{3.5}$$

where $c \in [0, 1]$ is the *cooling rate* parameter determining how fast the temperature decreases. Alternatively, other less efficient criteria can be the *Random Walk*, where every new solution s' is accepted, or the *Greedy Acceptance*, which agrees with every improving solution. Choosing the proper acceptance criterion is crucial for optimizing the performance of the LNS algorithm. Some criteria may be more effective for specific problems or solution spaces, while others may be more robust or easier to implement. In practice, a combination of acceptance criteria may balance exploration and exploitation of the search space.

3.2 Adaptive Large Neighborhood Search (ALNS)

The Adaptive Large Neighborhood Search (ALNS) is an advanced metaheuristic technique first introduced in Ropke and Pisinger, 2006 in the context of a Pickup and Delivery Problem. It builds upon and extends the Large Neighborhood Search (LNS) by integrating adaptive mechanisms. These mechanisms empower ALNS to efficiently explore and exploit the solution space, making it especially adept at tackling intricate combinatorial optimization challenges. Contrary to the conventional LNS, where a single destruction and repair function is employed, ALNS harnesses the power of multiple heuristics to address a given problem collaboratively. Central to this approach is a weight-based selection system. This system monitors and evaluates the performance of each heuristic combination in real-time. Simply put, heuristic pairs that yield better solutions are given higher probabilities of being chosen in subsequent iterations. Weights are not static but are instead periodically adjusted throughout the solution process. These adjustments occur at specified intervals called "segments," representing user-defined iterations. The weight adjustment process is detailed in Section (3.2.1). Refer to Algorithm (2) for a detailed view of the ALNS framework. This structure expands upon the foundation laid out in Algorithm (1), adding elements of adaptability. The ALNS algorithm takes an initial solution as input and works with two sets of operators: destroy operators (Ω^-) and repair operators (Ω^+). At each iteration, it selects a pair of these operators based on their respective weights, w^- and w^+, and dynamically recalibrates them based on the solution's quality.

Algorithm 2 Adaptive Large Neighborhood Search

Require: a feasible solution s, sets of destroy Ω^- and repair Ω^+ operators.
$s^* \leftarrow s, w^- \leftarrow (1, \ldots, 1), w^+ \leftarrow (1, \ldots, 1)$
while stopping criteria not met **do**
  select destroy $d \in \Omega^-$ and repair $r \in \Omega^+$ operators using weights w^- and w^+
  $s' \leftarrow r(d(s))$
  if accept(s, s') **then**
    $s \leftarrow s'$
  end if
  if $c(s') < c(s^*)$ **then**
    $s^* \leftarrow s$
  end if
  update w^- and w^+
end while
return s^*.

3.2.1 Weight Adjustment

To improve the performance of a heuristic in resolving a problem, it is essential to adjust the weights of the heuristics used dynamically. One example of this is the Roulette wheel selection principle presented in Ropke and Pisinger, 2006, where a set of h heuristics with weight $w_k, k = \{1, \ldots, h\}$ is used to select a function i in the set, based on the probability of its weight relative to the sum of all the weights. This means that the higher the operator's weight, the higher the chances of it being selected.

During the search process, the weights of the heuristics are automatically adjusted based on the scores of the heuristics recorded during iterations. The scores measure how well the heuristics perform, with a higher score indicating a more successful heuristic. The search is divided into segments, which represent the number of iterations, where the scores are all set to zero at the beginning of a segment, and the weights are adjusted using the recorded scores at the end of the segment.

The score of a pair is increased based on the quality of the solution found at a specific iteration, with three possible increments:

- σ_1: the remove-insert pair resulted in a new global solution;
- σ_2: the remove-insert pair resulted in a solution that was not accepted before but had an improving cost relative to the current solution.
- σ_3: the remove-insert pair resulted in a solution that was not accepted before, has a non-improving cost, but is still accepted.

Cases σ_1 and σ_2 are considered successful, as the pair improved the current solution by finding a never-visited solution, thereby promoting exploitation in the search space. The third case rewards the pair of heuristics that found a non-improving but never-visited solution, promoting diversification in the search space.

Once the scores are recorded during a segment j, the weights that will be used in segment $j+1$ are computed based on equation (3.6):

$$w_{i,j+1} = w_{ij}(1-\rho) + \rho\frac{\pi_i}{\Theta_i} \tag{3.6}$$

where π_i is the score of heuristic i in segment j, and Θ_i is the number of times the heuristic i has been used. The reaction factor ρ adjusts the weight variation's sensitivity based on the heuristic's effectiveness. When $\rho = 0$, the weights are never updated among segments, while at the increase of the factor, the weights tend to be equal to the score of the last segment.

3.3 Dynamic Approaches

In this section, we explore the two predominant methodologies utilized alongside ALNS in our research to address complex problems, especially those plagued with uncertainty. Uncertainty in real-world situations often means not all information is available or guaranteed. These methodologies help simulate such scenarios to ascertain the effect of uncertainty on problem-solving and the ultimate solutions derived.

Firstly, we address dynamic approaches. These methods' core is the recognition that real-world problems are constantly evolving. Traditional static solutions might be inadequate because they don't adjust or adapt to changing inputs or environments. By employing dynamic approaches, we ensure that solutions are flexible and can respond to variations. The work of Psaraftis, 1995, provides a comprehensive overview of the dynamic aspects that can be incorporated in dynamic vehicle routing problems. The author categorizes the characteristics of the problem that can be deterministic, such as the number of nodes, vehicles or the distances matrix, from the others that are subjected to uncertainty and become dynamically known during the routing process, such as travel times or nodes' demand. The latter category can be either forecasted through historical data, drawn by probabilistic distributions or revealed on real-time during the process.

Within the realm of dynamic approaches, an effective heuristic is the Myopic Dynamic Heuristic (MDH), discussed in detail in Section (3.3.1). The MDH exemplifies how a purely dynamic structure can be integrated into problem-solving. It operates on a short-sighted basis, making decisions that seem best at the present moment without extensive foresight into the future. This can be particularly valuable when immediate response is paramount and waiting for complete data is not feasible.

On the other hand, we also explore Multi-Scenario Approaches. These are essential when dealing with uncertainties as they offer a broader perspective by analyzing multiple potential scenarios arising from uncertain data or conditions. Instead of basing decisions on a single expected outcome, multi-scenario methods assess various outcomes, providing a comprehensive view of potential risks and rewards. Specifically, in Section (3.3.2), we focus on the Multi-Scenario-Based Progressive-Fixing (MSB-PF), a novel method highlighted in this thesis. This approach aims to find solutions that perform well across various plausible scenarios rather than optimizing for just one. In our exploration, both small instances for economic analysis and larger ones using ALNS have been solved, demonstrating the versatility and applicability of these methods. Both methods are applied to the problem *"A Dynamic Multi-Period Home Healthcare Problem with Consistency Constraints"* presented in Section (6).

3.3.1 Myopic Dynamic Heuristic (MDH)

The Myopic Dynamic Heuristic (MDH) is presented by Hvattum et al., 2006 to solve a Dynamic and Stochastic Vehicle Routing Problem. The term *myopic* refers to the fact that the approach considers the problem as purely dynamic without using any knowledge about future events or any stochastic information. In practice, it considers a series of static and deterministic subproblems, obtained by dividing the main problem into decision epochs, moments in which replanning is allowed and new decisions are taken, updating information considering only the known information. The solution obtained at each decision epoch is used as input to the following, adding available information and fixing the already taken decisions. In our thesis, we use the MDH to solve the Multi-Period Dynamic Nurse Routing Problem with Consistency Constraints (MPDNRP-CC) presented in Chapter (6) where the objective is to create a set of routes to serve patients with requests that become known during the time horizon. We set each decision epoch at the end of the working day, and the known information is the patients' requests already arrived at the decision maker. Routes of the previous days can't be changed, and new decisions involve only the routes of the following day. Each subproblem is solved using the ALNS. In Algorithm (3), the decision process of MDH is detailed.

Algorithm 3 Myopic Dynamic Heuristic (MDH)

```
for each decision epoch do
    Formulate a static and deterministic sub-problem with known patients and fix the variables corresponding to decisions that have already been made.
    Solve the problem using the ALNS of Algorithm (2).
    Use the solution as the new plan for the overall problem.
end for
```

The MDH can be used to obtain a fast benchmark for the overall problem, and it represents the application most similar to the real-world standard practice. However, due to its myopic nature, it tends to create good-quality solutions, overlooking the long-term consequences of initial decisions.

3.3.2 Multi-Scenario-Based Progressive Fixing (MSB-PF)

In our research, we have employed the Multi-Scenario-Based Progressive Fixing (MSB-PF), a heuristic method based on the traditional Multi-Scenario Approach (MSA) initially proposed by Bent and Van Hentenryck, 2004 and the Branch and Regret Heuristic (BRH) by Hvattum et al., 2007. The MSA framework is specifically tailored to address the intricacies inherent in resolving partially dynamic vehicle routing problems involving stochastic customer behavior. The primary objective of employing MSA is to leverage the wealth of stochastic information available when dealing with routing problems characterized by dynamic and stochastic attributes. At the core of the MSA technique is the concept of crafting multiple sample scenarios. Each scenario encapsulates all the known information and accommodates potential future data by drawing from relevant probability distributions. At each decision point, these scenarios are tackled as if they are static and deterministic. This means they are treated as problems with fixed, known outcomes. After solutions for all scenarios are obtained, the challenge is to pick the most indicative of the future. This chosen solution becomes the input solution for the decisions made in the next phase. In the original MSA method, this "most indicative" scenario is determined using a consensus function, which measures which scenario aligns most closely with the rest.

The BRH, instead, maintains the use of sample scenarios from the MSA but improves the evaluation of stochastic information. Rather than choosing one scenario as representative of future events, the BRH branches on all the possible decisions that can be made in a decision epoch and then selects one of the branches according to a regret criterion. For better understanding, let us give an overview of the specific application used by Hvattum et al., 2007 for a Stochastic and Dynamic Vehicle Routing Problem in which customers may place orders at any time during the day, and the objective is to maximize the served customers while minimizing the number of vehicles and the traveled distance. The working day is divided into decision intervals, and at each interval, the BRH works on two different levels of decisions. The first level objective is to decide which customers to visit during the current interval and which to postpone. Then, the most frequent customer-vehicle pair is fixed for the customers that have to be served in the interval. A decision is evaluated by solving the sample scenarios with the current decision and with all the alternatives. For each alternative, the resulting solutions are evaluated on the problem objective function and averaged over the set of sample scenarios. The alternative with the best average value is selected. In Algorithm (4), the decision process is detailed.

Algorithm 4 Branch and Regret Heuristic (BRH)

```
for each decision epoch do
    Lock all previous decisions.
    create the set of sample scenarios S
    for each scenario s ∈ S do
        solve s
    end for
    while there are decisions to take do
        select a decision d.
        create the set Φ(d) of alternatives of d.
        for each alternative φ ∈ Φ(d) do
            for each scenario s ∈ S do
                solve scenario s^φ with decision φ locked.
            end for
            Compute the regret cost of φ.
        end for
        lock the least regret action φ.
    end while
end for
```

Our MSB-FP heuristic includes the generation of multiple scenarios as in the MSA approach but disregards the decision process based on consensus functions using the branching technique present in the BRH. However, due to the computational burden of considering all the possible alternatives for all the possible sample scenarios, we modify the decision using a *progressive fixing* of information. Precisely, as in the BRH, the Algorithm starts with solving the sample scenarios to find the most common decision. Then, instead of re-evaluating all the scenarios on all the alternatives, in the MSB-PF, this decision is fixed, and the resolution moves either to the next decision that has to be taken or to the next decision epoch. Each sub-problem is solved using the ALNS presented in Section (3.2). Algorithm (5) shows the general decision process.

Algorithm 5 Multi-Scenario-Based Progressive Fixing (MSB-PF)

```
for each decision epoch do
    Create the set of sample scenarios S
    while there are decisions to make do
        Fix all previous decisions.
        for each scenario s ∈ S do
            solve s
        end for
        select the decision d with more consensus among scenarios
        add decision d to locked decisions
    end while
end for
```

The specific application of the MSB-PF is detailed in the problem of Chapter (6) in the Section (6.4.3).

Chapter 4

Optimizing Fairness in Home Healthcare

Contents

The content of this chapter was presented to the 10^{th} *IFAC Conference on Manufacturing Modelling, Management and Control (IFAC MIM 2022). This chapter corresponds to the conference paper "V.Bonomi, R.Mansini, R.Zanotti, Fairness in Home Healthcare: can patient-centered and nurse-centered measures concur to the same goals?" appeared in IFAC-PapersOnLine, 2022, 55, 3136-3141. (DOI: 10.1016/j.ifacol.2022.10.211)*

4.1 Introduction

In the context of home healthcare (HHC) systems, the notion of fairness is multifaceted and varies in significance based on the stakeholders involved. Fairness within the workplace boosts the morale and productivity of employees such as nurses and medical professionals, fostering greater job engagement. Conversely, patient safety and satisfaction, widely recognized as key metrics in HHC, reflect the standard of care provided. A fair environment for patients and care workers is critically essential for HHC organizations. According to public data, the HHC industry is expected to see considerable expansion shortly. The World Health Organization projects that by 2050, the global population aged 60 and over will double, representing 22% of the total population. This demographic shift, along with progress in medical technologies and broader healthcare accessibility, is anticipated to escalate the need for HHC services. In Italy, for instance, about 2.7% of the population aged 65 and older are currently utilizing HHC services, a statistic mirrored in several other European countries. These statistics highlight the need of an evolving system open to the incorporation of different perspectives. This includes looking beyond just the minimization of costs, adopting an inclusive approach that considers the point of view of the external stakeholders such as patients' needs and nurses' satisfaction. By doing so, home healthcare systems can evolve to meet the increasing demands of the sector while maintaining a balance between cost-effectiveness and fair, quality care for all involved. In this chapter, we aim at analyzing the impact that switching the perspective from the costs to fairness achievement could have on the operational activities of a Territorial Operations Center (TOC), a district-level authority tasked with the management of patient care through the deployment of a fleet of professional caregivers. The study models a single-period Nurse Routing Problem (NRP) to introduce the concepts of fairness. We examine various fairness metrics pertinent to the two primary entities involved other than the TOC, and model these via min/max objective functions. The first entity, the nurse, is central to our analysis. When prioritizing nurses, our objectives are:

- Workload Fairness: this aspect emphasizes the desire of nurses for a balanced distribution of tasks. Achieving this involves minimizing the maximum total service time or the maximum time spent on daily tasks outside the hospital by all nurses.

- Fairness in Preferred Working Zone: nurses generally prefer to work in areas that are more accessible based on their transportation means. This involves allocating zones in such a way that the maximum ranking level of zones assigned to nurses is minimized, based on their preferences.

- Fairness in Qualification Level: it is crucial for nurses to perform tasks that match their qualification levels. A fair distribution in this regard means minimizing the maximum discrepancy from the ideal qualification level for each service assigned to nurses, ensuring that they are not relegated to tasks below their skill level.

The patients represent the other relevant actor. Their care, encompassing rehabilitation or post-discharge follow-up plans, necessitates a daily optimization of visits. This optimization incorporates various aspects of fairness, with a focus on patient-centric goals:

- Fairness in visiting time windows: the objective here is to align the timing of services provided to a patient as closely as possible with a pre-established time window. The fairness goal minimizes the maximum sum of early and late deviation from the scheduled visiting time window computed for all patients;
- Fairness in waiting time: when a patient receives more than one service and does not want to wait too much between the first and the last service's execution. The fairness goal minimizes the maximum idle time endured by all patients;
- Fairness in nurse assignment consistency: patients often prefer continuity in their care, desiring the same nurse for all services. To satisfy this preference, the aim is to minimize the number of nurses attending to patients needing multiple services.
- Fairness in adequacy level: patients expect to receive care from nurses whose qualifications match the required level of service. The goal is to minimize the gap between the lowest average adequacy level assigned to each patient and the highest possible qualification level.

This chapter is organized as follows. Section (4.2) describes the studied problem and Section (4.3) gives the mathematical formulation, whereas Section (4.4) introduces the different fairness measures related to nurses and patients and their mathematical formulations. Section (4.5) analyzes the results obtained when using the different fairness measures and draws some interesting managerial insights. Finally, Section (4.6) provides conclusions and future developments.

4.2 Problem Description

In our study, we examine the daily operational activities of routing and scheduling of a TOC. A non-trivial task is to meet different stakeholders' often conflicting needs and expectations while minimizing operational costs. For instance, the needs of patients might not align with the one of the hospitals providing the service or with the fleet of nurses performing visits. Specifically, our research delves into the daily organization of patient visits conducted by the TOC fleet of nurses at their homes. The process includes scheduling and routing nurses to various patient locations, taking into account factors such as the nurses' skills, travel time, and the specific healthcare needs of each patient. The aim is to optimize these visits to maximize patient care while being mindful of the constraints and resources of the healthcare system, particularly the hospital and its nursing staff. We perform a double analysis in which, starting from the traditional problem of minimizing costs we try to incorporate different perspectives through a switch in the objective function. The needs of the other stakeholders other than the hospital are modeled through the concept of *fairness*. Patients require a fair and impartial care delivery while nurses seek an equitable working environment. The main indicators to measure fairness for each stakeholder and their mathematical formulation are detailed in Section (4.4). The standard model can be formulated as a NRP characterized by a set of nurses tasked with visiting a set of patients over the territory. The activities are planned daily and, during the day, patients might require more than one visit. Visits have a predefined service time and do not need to be performed by the same vehicles. Nurses are characterized by different levels of specialization, yet they can perform any visit. The objective of the problem is to find a set of routes that minimizes the total traveling time without exceeding the working time limit.

The problem can be defined over a complete graph $G = (V, A)$, where $V = \{0\} \cup \{1, ..., n\}$ is the node set, and $A = \{(i, j) : i, j \in V; i \neq j\}$ is the arc set. $N = \{1, ..., n\}$ represents the set of services requested by patients, whereas 0 is the hospital (starting and ending point for the routes of the nurses). Let K be the set of nurses and P the set of patients. Node set N is partitioned into $|P|$ disjoint and non-empty subsets $N_p, p \in P$, i.e. $N = \cup_{p \in P} N_p$ and $N_p \cap N_{p'} = \emptyset$ for $p \neq p'$. Nodes belonging to the subset N_p represent the set of services required by patient $p \in P$ and have the same location. A positive service time s_i is associated with each patient's service $i \in N$. We define as t_{ij} the non-negative time needed to travel from node i to node j, where $t_{ij} = 0$ for all $i, j \in N_p, p \in P$. Each nurse has a working time threshold equal to T_{max}. We assume that travel times satisfy the triangle inequality. The NRP assigns nurses to patients by scheduling their visits in a set of routes (starting and ending at node 0) such that each service is satisfied by precisely one nurse and the global traveling time is minimized while complying with nurses' working time limit.

4.3 Mathematical Formulation

In this section, we first present the decision variables and the formulation of the standard NRP. Then, in Section (4.4), we detail how the various fairness measures are formulated and inserted in the standard model. We define three distinct sets of variables. For each arc $(i, j) \in A$ and each vehicle $k \in K$ we define the binary variable as follows:

- $x_{ij}^k = \begin{cases} 1, & \text{if arc } (i, j) \text{ is traversed by vehicle } k \\ 0, & \text{otherwise.} \end{cases}$

The second set contains, for each node $i \in N$ and each vehicle $k \in K$ the binary variable:

- $y_i^k = \begin{cases} 1, & \text{if nurse } k \text{ performs a visit on node } i \\ 0, & \text{otherwise.} \end{cases}$

The third set contains the continuous variables z_{ij} for each arc (i, j) defining the arrival time at node j when arriving from node i. For each set $S \subset N$, let $\delta^+(S) = \{(i, j) \in A : i \in S, j \notin S\}$ and $\delta^-(S) = \{(i, j) \in A : i \notin S, j \in S\}$ be the set of arcs leaving and entering set S, respectively, with a special case where $|S| = 1$ indicated as $\delta^+(i) = \delta^+(\{i\})$ and $\delta^-(i) = \delta^-(\{i\})$.

The model can be formulated as follows:

$$(NRP) \quad \min \sum_{(i,j)\in A} t_{ij} x_{ij}^k \tag{4.1}$$

Subject to:

$$\sum_{(i,j)\in\delta^+(i)} x_{ij}^k = \sum_{(j,i)\in\delta^-(i)} x_{ji}^k = y_i^k \quad i \in N, k \in K \tag{4.2}$$

$$\sum_{k\in K} y_i^k = 1 \quad i \in N \tag{4.3}$$

$$\sum_{(0,j)\in\delta^+(0)} x_{0j}^k = \sum_{(j,0)\in\delta^-(0)} x_{j0}^k \leq 1 \quad k \in K \tag{4.4}$$

$$\sum_{(i,j)\in\delta^+(i)} z_{ij} - \sum_{(j,i)\in\delta^-(i)} z_{ji} = \sum_{k\in K} \sum_{(i,j)\in\delta^+(i)} (s_i + t_{ij}) x_{ij}^k \quad i \in N \tag{4.5}$$

$$z_{0j} \geq t_{0j} \sum_{k\in K} x_{0j}^k \quad j \in N \tag{4.6}$$

$$(t_{0i} + s_i + t_{ij}) \sum_{k\in K} x_{ij}^k \leq z_{ij} \leq (T_{max} - t_{j0} - s_j) \sum_{k\in K} x_{ij}^k \quad (i, j) \in A \tag{4.7}$$

$$z_{ij} \geq 0 \quad (i, j) \in A \tag{4.8}$$

$$x_{ij}^k \in \{0, 1\} \quad (i, j) \in A, k \in K \tag{4.9}$$

$$y_i^k \in \{0, 1\} \quad i \in N, k \in K \tag{4.10}$$

The objective function (4.1) minimizes the operational costs of the hospital computed as the Total Travelling Time (TTT). Constraints (4.2) regulate the arc flow in visit nodes, imposing that if a vehicle k is assigned to node i, exactly one arc can enter and leave from it. Constraints (4.3) guarantee that each node (each service) is visited (is executed) exactly once by one nurse. Constraints (4.4) impose that at maximum $|K|$ nurses vehicles start from the starting node 0 and come back. Constraints (4.5) ensure that, if a nurse k serves node j immediately after node i (i.e. $x_{ij}^k = 1$), then the time elapsed between the arrival times in the two nodes is equal to the execution time t_i required to serve node i plus the travel time t_{ij} to move from node i to node j. Constraints (4.6) set a bound on the minimum time required to reach the starting node after the depot, whereas constraints (4.7) define lower and upper bounds on the arrival time and duration of each route. Finally, constraints (4.8)– (4.10) impose nonnegative and binary conditions on variables.

The creation of subtours is prevented by constraints (4.5)–(4.7). Moreover, we strengthened the formulation by adding the following connectivity constraints from Hanafi et al., 2020:

$$\sum_{(i,j)\in\delta^+(S)} x_{ij}^k \geq y_s^k \quad S \subseteq N, |S| \geq 2, k \in K, s \in S. \tag{4.11}$$

Finally, it is worth noticing that, to correctly model some of the fairness objectives discussed in the next section, we need to identify which nurse provides which service for each patient. For this reason, we use a three-index formulation. Alternatively, one can avoid this by duplicating the depot a number of times equal to the number of nurses, thus separating the starting and ending depot for each nurse in a fictitious way as referenced in Luo et al., 2015.

4.4 Fairness Measures

In this section, we introduce several objective functions expressed through min/max formulations classified according to the main stakeholder they refer to.

For the sake of space, we do not provide the entire model formulation associated with each function. Instead, we introduce the objective function and additional constraints to extend the model in (4.2)–(4.10).

4.4.1 Nurse-centered measures

From the viewpoint of nurses, implementing fairness measures is essential for ensuring that the distribution of workload is balanced and that service assignments are equitable. These measures must reflect the nurses' skills and experience. By optimizing these aspects, we not only foster fair treatment of the nursing staff but also enhance patient satisfaction and the quality of care provided.

Four primary fairness measures are considered in our approach: the balance of workload, evaluated in terms of Total Time (TTW) and Service Time (STW), the equitable allocation of Working Zones (PWZ) based on the preferences of nurses, and the assignment of tasks that align with the nurses' level of qualification (NQL).

Total Time Workload (TTW) and Service Time Workload (STW)

In the workload distribution, fairness is achieved by minimizing the maximum workload w_{max} assigned to nurses. The workload for each nurse is quantified in two ways: firstly, as the service time workload, which encompasses the total duration a nurse spends in attending to their assigned patients, and secondly, as the total working time, defined by the time of return to the TOC. In both cases, we formulate a min-max problem as follows:

$$\min w_{max} \tag{4.12}$$

$$w_k \leq w_{max} \qquad k \in K \tag{4.13}$$

$$w_k \geq 0 \qquad k \in K \tag{4.14}$$

where w_k can be defined either as the service time workload of nurse k using constraints (4.15) or her/his total working time expressed as traveling time plus service time, using constraints (4.16):

$$\sum_{i \in N} s_i y_i^k = w_k \quad k \in K \tag{4.15}$$

$$\sum_{(i,j) \in A} t_{ij} x_{ij}^k + \sum_{i \in N} s_i y_i^k = w_k \quad k \in K \tag{4.16}$$

Fairness on Preferred Working Zone (PWZ)

Nurses often have varying preferences for different zones within the same district, as these zones may differ in accessibility. For example, a nurse who doesn't own a car might prefer not to be assigned to areas with sparse public transport options. To model these preferences, we assign a set of positive integer numbers γ_{ik} for each nurse k and each service i, indicating the preference level for the zone associated with the service. A rank value equal to 1 corresponds to the best possible level. If two services i and j are located in the same zone, then $\gamma_{ik} = \gamma_{jk}$, for each nurse $k \in K$. The formulation for this fairness goal is:

$$\min zon_{max} \tag{4.17}$$

$$zon_{max} \geq zon_k \qquad k \in K \tag{4.18}$$

$$zon_k \geq \gamma_{ik} y_i^k \qquad k \in K, i \in N \tag{4.19}$$

$$zon_k \geq 0 \qquad k \in K \tag{4.20}$$

$$zon_{max} \geq 0 \tag{4.21}$$

Variable zon_k will take the maximum γ_{ik} value (worst ranking position) among all services $i \in N$ directly assigned to nurse k (i.e., $\max\{\gamma_{ik} | y_i^k = 1, i \in N\}$).

Fairness on Nurse Qualification Level (NQL)

Regarding the skill levels of nurses, it's noted that they possess varied qualifications, with certain nurses being more adept and comfortable in performing specific services compared to others. We indicate as ρ_{ik} the *qualification level* of nurse $k \in K$ in executing service $i \in N$. This coefficient takes an integer value between 1 (lowest level) and ρ (the highest level, equal to 5 in our case). The present fairness measure aims at equitably assigning nurses to services according to their qualification:

$$\min q_{max} \tag{4.22}$$

$$q_{max} \geq q_k \qquad k \in K \tag{4.23}$$

$$q_k \geq \rho - \rho_{ik} y_i^k \qquad k \in K, i \in N \tag{4.24}$$

$$q_k \geq 0 \qquad k \in K \tag{4.25}$$

$$q_{max} \geq 0 \tag{4.26}$$

where variable q_k represents the maximum difference between the best value ρ and each value ρ_{ik} among those related to services assigned to nurse k (i.e., all $i \in N$ for which $y_i^k = 1$). Variable q_{max} takes, according to the constraints (4.23) and the objective function, the maximum of q_k out of all nurses $k \in K$.

4.4.2 Patient-centered measures

Ensuring fairness in patient care involves providing services equally, and avoiding any form of prioritization. To this end, we have developed three distinct equity measures that facilitate equitable service delivery in various aspects, ranging from consistency in nursing to scheduling of visits. The key measures identified are: equity in Visiting Time Windows (VTW) expressed as earliness or lateness in respect to patients' time windows; equity in Patient Waiting Time (PWT) as a measure of fairness balancing the maximum idle time between visits of a patient; Nurse Adequacy Level (NAL) as a measure of fairness in assigning nurses based on their qualifications compared to patients' needs, and Nurse Assignment Consistency (NAC) guarantees a limited number of nurses for each patient.

Equity in Visiting Time Windows (VTW)

This fairness measure helps model in a fair way the time in which the patients receive their visits. Let $[a_p, b_p]$ be the time window in which patient $p \in P$ would like to receive his/her visits. We model as e_i and l_i two continuous variables, for each service $i \in N$, that measure the earliness and lateness in executing the service concerning the assigned time window, respectively. The fairness goal is to balance the total lateness and earliness among all patients. We introduce continuous variable t_p which measures the total lateness and earliness for each patient $p \in P$ and variable t_{max}, representing the maximum t_p value among all the different patients $p \in P$. The formulation is as follows:

$$\min t_{max} \tag{4.27}$$

$$t_p \leq t_{max} \tag{4.28}$$

$$t_p \geq \sum_{i \in N_p} (e_i + l_i) \qquad p \in P \tag{4.29}$$

$$a_p - e_j \leq \sum_{(i,j) \in \delta^-(j)} z_{ij} \leq b_p + l_j \qquad p \in P, j \in N_p \tag{4.30}$$

$$e_i \geq 0, \quad l_i \geq 0 \qquad i \in N \tag{4.31}$$

$$t_p \geq 0 \qquad p \in P \tag{4.32}$$

$$t_{max} \geq 0 \tag{4.33}$$

Constraints (4.30) model the time window assigned to each service $j \in N_p$, for each patient $p \in P$. Notice that, if the arrival time at node j, measured by $\sum_{(i,j) \in \delta^-(j)} z_{ij}$, is included in the time window $[a_p, b_p]$, the values of e_i and l_i are forced to zero by the objective function.

Equity in Patient Waiting Time (PWT)

A crucial aspect of time-related fairness in patient care involves reducing the waiting period between consecutive visits to the same patient. Especially in cases where a patient needs multiple visits, it's preferable to conduct these visits with minimal delay.

To achieve this, our focus is on promoting fairness by aiming to minimize the maximum idle time represented by the non-negative variable $idle_{max}$. This variable represents the interval between the start of the first service and the end of the last service to which it is substracted the time to serve the patient as in constraints (4.36). By doing so, we ensure that patients are not left waiting unnecessarily long between visits, thereby enhancing the efficiency and responsiveness of the care provided.

$$\min idle_{max} \tag{4.34}$$
$$idle_p \le idle_{max} \qquad p \in P \tag{4.35}$$
$$idle_p \ge high_p - low_p - \sum_{i \in N_p} s_i \qquad p \in P \tag{4.36}$$
$$low_p \le \sum_{j \in \delta^-(i)} z_{ji} \qquad i \in N_p \tag{4.37}$$
$$high_p \ge \sum_{j \in \delta^-(i)} z_{ji} + s_i \qquad i \in N_p \tag{4.38}$$
$$idle_p \ge 0, \quad low_p \ge 0, \quad high_p \ge 0 \qquad p \in P \tag{4.39}$$
$$idle_{max} \ge 0 \tag{4.40}$$

where variables low_p and $high_p$ are the earliest starting time and the latest concluding time for all services required by patient $p \in P$, and their difference net of the service times (nonnegative variable $idle_p$) measures the idle time for the same patient.

Equity in Nurse Assignment Consistency (NAC)

In practical scenarios, there's generally no limit to the number of nurses that can attend to a single patient. This function aims to prevent situations where some patients receive care from a higher number of nurses than others who require a similar level of service. To create a fair environment, we minimize the maximum number of nurses assigned to patients needing multiple visits. We define complete consistency as the possibility that the same nurse provides all the services for a given patient. Such consistency becomes particularly critical during health crises like the COVID-19 pandemic, as it significantly reduces the risk of contamination. In our model, a solution is considered fair if it is close to guarantee complete consistency. By maintaining this consistency, we aim to elevate the level of service provided and simultaneously minimize the risk to the patient.

Following Kovacs et al., 2015b, we minimize a variable k_{max} representing the maximum number of nurses assigned to patients as follows:

$$\min k_{max} \tag{4.41}$$
$$k_{max} \ge \sum_{k \in K} r_{pk} \qquad p \in P \tag{4.42}$$
$$r_{pk} \ge y_i^k \qquad p \in P, i \in N_p, k \in K \tag{4.43}$$
$$r_{pk} \in \{0, 1\} \qquad p \in P, k \in K \tag{4.44}$$
$$k_{max} \ge 0 \tag{4.45}$$

where binary variable r_{pk} takes value 1 if nurse k visits patient p. Notice that, according to (4.43), r_{pk} is forced to 1 if at least one of the binary variables y_i^k takes value 1, indicating that nurse k has performed at least one of the services $i \in N_p$.

Equity in Nurse Adequacy Level (NAL)

Patients have a strong interest in receiving highly specialized services. The more qualified a nurse is for a particular service, the higher the patient's perceived adequacy level of the service. As already explained, ρ_{ik} is the qualification level of nurse k with respect to service i. We indicate as adq_{max} the continuous variable representing the worst adequacy measured as the difference between the maximum possible qualification level ρ and the average level of qualification assigned to each patient. The higher such difference, the lower the adequacy of the service. The goal is to minimize such value:

$$\min adq_{max} \tag{4.46}$$

$$adq_{max} \geq \rho - \frac{1}{|N_p|} \sum_{k \in K} \sum_{i \in N_p} \rho_{ik} y_i^k \qquad p \in P \tag{4.47}$$

$$adq_{max} \geq 0 \tag{4.48}$$

Constraints (4.47) impose that adq_{max} has to be greater than or equal to the difference between ρ and the average qualification level assigned to each patient $p \in P$ and computed out of all ρ_{ik} values of nurses performing all the services $i \in N_p$.

4.5 Computational Results and Managerial Insights

This section details the results achieved by employing various fairness measures as objective functions in a standard NRP. We assess their impact on the overall structure of the solutions and the total costs, quantified here as the total traveled time (TTT) expended by nurses in completing their tasks. Section (4.5.1) presents the structure of the benchmark instances used in the study, while Section (4.5.2) gives an extensive analysis of the correlation that can be found among different measures. In particular, we draw conclusions on the effect that each fairness measure has on the others when optimized and we gives some insight on the correlation that can exist between measures of different stakeholders. The experiments were conducted on a Ubuntu 20.04.2 system, powered by an AMD Ryzen 9 3950x CPU, featuring 16 cores, 32 threads, and equipped with 32 GB of RAM. The optimization tasks utilized Gurobi 9.1.2 for mixed integer linear programming (MILP). All test cases were resolved to their optimal solutions within one hour. The only exceptions were instances with $|N| = 30$ and $|N| = 40$ when applying fairness measures TTW and VTW as objectives.

4.5.1 Instances Generation

We performed the experiments on a total of 30 benchmark instances that differ from the number of nodes $|N|$, the number of vehicles $|K|$, and the maximum number of services that can be required by a patient $|F|$. Table (4.1) reports the structure and the size of the instances. The column $Pat.$ indicates the average number of patients among the instances, and the column $\#Inst.$ indicates the number of instances generated per each tuple $(|N|, |K|, |F|)$. Patients' locations have been randomly generated within a geometrical square in such a way that the maximum travel time between any two nodes is equal to 120 minutes. The geographical area is divided into 16 zones for which the nurses display their preferences. The duration of a service denoted as s_i for a visit i within the set N varies based on the patient's required service type. We categorize these services into three types: the short service, lasting between 5 to 15 minutes, the medium service, with a duration ranging from 16 to 30 minutes, and the long service, taking between 31 to 45 minutes. The maximum working shift of the nurses T_{max} equals 8 hours.

TABLE 4.1: Instances: structure and size.

$\|N\|$	$\|K\|$	$\|F\|$	$Pat.$	$\#Inst$
20	2	2	13	10
30	3	2	20	10
40	3	3	21	10

In Table (4.2), we report the average time required to reach optimality. It is worth noticing how STW and VTW have a much higher average time than the other objectives, reaching the time limit for every instance for $|N| = 40$.

$\|N\|$	TTT	TTW	STW	NQL	PWZ	NAC	VTW	PWT	NAL
20	6	1	351	12	7	2	706	4	100
30	114	5	3241	35	24	14	3342	30	16
40	403	85	3600	122	92	56	3600	283	191

TABLE 4.2: Computational Times (s)

4.5.2 Fairness Measures correlation

To explore the correlation between different fairness measures, we approach each instance by solving it with every possible fairness objective. Once we obtain a solution for a specific objective function, we are interested in the value that the non-optimized measures assume. Let us define as z^{obj}_{fair} the value that the measure $fair$ assume when function obj is being optimized. Table (4.3) reports all the obtained values of z^{obj}_{fair}. Row caption (from now on obj) indicates the fairness measure used as objective function, the remaining measures ($fair$) are listed in the columns.

$\|N\|$		TTT	TTW	STW	NQL	PWZ	NAC	VTW	PWT	NAL
20	**TTT**	297.9	415.2	22.1	3.1	11.0	1.0	660.0	0.0	3.1
	TTW	310.1	359.9	21.8	3.1	12.0	1.3	587.3	47.4	3.1
	STW	501.4	462.8	19.4	3.1	11.6	1.9	496.8	224.0	3.1
	NQL	479.7	470.6	23.0	1.4	11.2	2.0	575.2	264.9	1.4
	PWZ	426.5	463.4	22.9	3.1	6.7	1.3	528.2	114.4	2.9
	NAC	489.6	466.2	21.3	3.3	11.4	1.0	566.8	215.6	3.2
	VTW	513.7	473.6	20.5	3.3	12.0	1.8	63.9	43.2	3.2
	PWT	456.2	448.6	20.8	3.1	11.6	1.8	657.1	0.0	3.1
	NAL	465.7	468.6	22.7	9.5	11.4	2.0	555.3	219.8	1.0
30	**TTT**	345.6	469.1	28.5	3.9	9.5	1.0	534.4	0.0	3.9
	TTW	438.5	339.8	21.4	3.9	12.0	1.7	598.0	91.9	3.9
	STW	775.9	464.6	18.8	3.9	12.2	2.0	571.4	291.4	3.9
	NQL	747.2	468.8	24.5	1.1	13.1	2.0	540.8	268.8	1.1
	PWZ	655.4	468.2	23.9	3.9	5.2	2.0	614.7	239.1	3.9
	NAC	718.5	465.0	21.5	3.9	11.6	1.0	631.9	200.9	3.9
	VTW	806.9	472.9	21.2	3.9	12.2	2.0	82.6	86.8	3.9
	PWT	738.9	463.4	20.7	3.9	12.5	2.0	664.9	0.0	3.9
	NAL	730.0	474.7	24.1	1.5	12.9	2.0	538.9	285.8	1.1
40	**TTT**	312.6	471.1	38.0	5.6	17.6	1.0	817.2	3.3	3.2
	TTW	397.6	354.7	28.5	3.5	13.2	1.9	772.6	111.9	3.2
	STW	727.9	473.6	24.8	3.5	14.0	2.6	718.0	327.3	3.4
	NQL	717.7	474.6	28.3	1.3	13.9	2.5	700.9	330.2	1.3
	PWZ	638.9	471.7	30.1	3.5	5.9	2.0	695.7	300.3	3.4
	NAC	665.2	469.3	29.7	3.5	13.2	1.0	818.3	253.2	3.3
	VTW	730.7	474.1	27.9	3.5	13.7	2.3	128.1	144.2	3.4
	PWT	676.3	469.8	29.3	3.5	14.0	2.1	775.4	0.0	3.4
	NAL	726.0	474.9	27.4	1.9	13.7	2.2	779.3	258.3	1.1

TABLE 4.3: Value of different measures at the variation of the optimized function, divided per $|N|$

For example, taking $|N| = 30$ nodes, when PWZ is optimized the function STW has a value of 23.9 that corresponds to $z^{PWZ}_{STW} = 23.9$. The highlighted entrances on the diagonals correspond to z^{obj}_{obj}, more precisely, the value of a fairness measures when it is optimized. It can be noticed how these correspond to the lowest value in each column, meaning that every function has its best fairness level when it is optimized as an objective. Analyzing the table column by column offers insights into the varying outcomes a function can achieve, whether it is the main focus of optimization or not. For example, in the column dedicated to PWT, which tracks patient waiting time, the number drops to 0 when PWT is the primary objective. However, this measure can escalate to as high as 330.2 as in instances with $|N| = 40$ and PWZ as main objective.

However, the pure values of z^{obj}_{fair} do not give any immediate insight into the impact that a switch in the objective function has on the fairness results. For this reason, we measure the improvement that each function has on a specific obj compared to its worse case. Let us define, as obj^W the measure obj that provides the worst outcome for a specific measure $fair$. We computed the percentage improvement of z^{obj}_{fair} for various obj as :

$$\Delta_{\%}(fair, obj) = \frac{z^{obj}_{fair} - z^{obj^W}_{fair}}{z^{obj^W}_{fair}} * 100 \tag{4.49}$$

Table (4.4) reports the results of these improvements for all the combinations of obj and $fair$.

$\|N\|$		TTT	TTW	STW	NQL	PWZ	NAC	VTW	PWT	NAL
	TTT	42.0	12.3	3.9	67.4	8.3	50.0	0.0	100.0	1.6
	TTW	39.6	24.0	5.2	67.4	0.0	35.0	11.0	82.1	1.6
	STW	2.4	2.3	15.7	67.4	3.3	5.0	24.7	15.5	1.6
	NQL	6.6	0.6	0.0	85.3	6.7	0.0	12.8	0.0	57.1
20	**PWZ**	17.0	2.2	0.4	67.4	44.2	35.0	20.0	56.8	7.9
	NAC	4.7	1.6	7.4	65.3	5.0	50.0	14.1	18.6	0.0
	VTW	0.0	0.0	10.9	65.3	0.0	10.0	90.3	83.7	0.0
	PWT	11.2	5.3	9.6	67.4	3.3	10.0	0.4	100.0	1.6
	NAL	9.3	1.1	1.3	0.0	5.0	0.0	15.9	17.0	69.8
	TTT	57.2	1.2	0.0	0.0	27.5	50.0	19.6	100.0	0.0
	TTW	45.7	28.4	24.9	0.0	8.4	15.0	10.1	68.5	1.3
	STW	3.8	2.1	34.0	0.0	6.9	0.0	14.1	0.0	1.3
	NQL	7.4	1.2	14.0	71.8	0.0	0.0	18.7	7.8	71.8
30	**PWZ**	18.8	1.4	16.1	0.0	60.3	0.0	7.5	17.9	1.3
	NAC	11.0	2.1	24.6	0.0	11.5	50.0	5.0	31.1	0.0
	VTW	0.0	0.4	25.6	0.0	6.9	0.0	87.6	70.2	1.3
	PWT	8.4	2.4	27.4	0.0	4.6	0.0	0.0	100.0	0.0
	NAL	9.5	0.0	15.4	61.5	1.5	0.0	18.9	1.9	73.1
	TTT	57.2	0.8	0.0	0.0	0.0	61.5	0.1	99.0	5.9
	TTW	45.6	25.3	25.0	37.5	25.0	26.9	5.6	66.1	5.9
	STW	0.4	0.3	34.7	37.5	20.5	0.0	12.3	0.9	0.0
	NQL	1.8	0.1	25.5	76.8	21.0	3.8	14.3	0.0	62.7
40	**PWZ**	12.6	0.7	20.8	37.5	66.5	23.1	15.0	9.1	0.0
	NAC	9.0	1.2	21.8	37.5	25.0	61.5	0.0	23.3	3.9
	VTW	0.0	0.2	26.6	37.5	22.2	11.5	84.3	56.3	1.5
	PWT	7.5	1.1	22.9	37.5	20.5	19.2	5.2	100.0	1.5
	NAL	0.7	0.0	27.9	66.1	22.2	15.4	4.8	21.8	67.2

TABLE 4.4: Percentage improvements comparison.

In the table, each $(obj - fair)$ entry represents the percentage improvement of z_{fair}^{obj} with respect to the worst result measure $fair$ has got among all the objs. For example, entry $\Delta_{\%}(PWT, TTW)$ indicates the improvement that the function PWT has when TTW is optimized in respect to its worst outcome. Considering the 20 nodes case, $\Delta_{\%}(PWT, TTW) = 82.1\%$ meaning that the result of PWT improves of the 82.1% in respect of the worst value that is, in this case, when NQL is optimized. The worst case of each $fair$ are identified by the entry 0. Having the highest improvements on the diagonal proves how each function's best fairness level is reached when it is used as an objective. Considering the table by column, it becomes easier to highlight which objectives yield to the most significant improvements for each fairness measure. Let us take, for example, the goal of optimizing the total traveled time (TTT). The best measures are TTT (as expected), TTW, PWT but also PWZ, and when the number of patients increases, also NAC. These last two measures indicate that attributing the preferred zones to nurses and the optimization of nurse consistency positively impact on the total traveling time since all services of the same patient are generally assigned to the same nurse. Table (4.3) and Table (4.4) help identify the behavior of specific measures when the objective function varies.

The second scope of our computational analysis is to examine the correlations existing among different fairness goals. We conducted the analysis by examining functions in pairs and deriving two distinct types of correlations: a pair-external correlation (`Type 1`) and a pair-internal correlation (`Type 2`). For each pair of objective function (obj_1, obj_2), `Type 1` correlation is computed as the complement to 1 of the average gap between $z_{fair}^{obj_1}$ and $z_{fair}^{obj_2}$ for all the values of $fair$. The gap for each $fair$ is computed as in equation (4.50).

$$\Delta_{fair} = \frac{|z_{fair}^{obj_1} - z_{fair}^{obj_2}|}{max\{z_{fair}^{obj_1}, z_{fair}^{obj_2}\}} \tag{4.50}$$

`Type 1` correlation indicates the similarity among two objectives obj_1 and obj_2 in respect of the measures external of the pair. The highest the correlation value the highest the pair's similarity, meaning that the other measures' results are similar in the solutions obtained optimizing obj_1 and obj_2.

On the other hand, `Type 2` correlation measure the similarity within the pair. `Type 2` computes the complement to one of the average percentage gaps between $z_{obj_2}^{obj_1}$ and the optimal value of obj_1 $z_{obj_1}^{obj_1}$ over $z_{obj_2}^{obj_1}$ and the same is done using $z_{obj_1}^{obj_2}$ and obj_2 and average the resulting values. The highest the value the more directly correlated are the two measures, meaning that optimizing one generally implies optimizing the other. Figure (4.1) visually represents each pair's correlations.

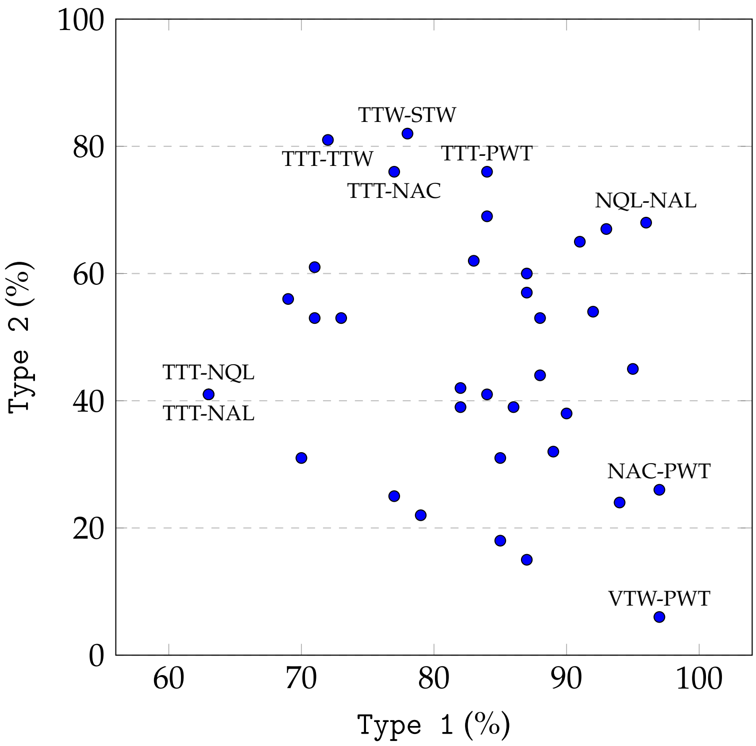


FIGURE 4.1: Correlation between fairness measures.

The graph delineates three principal regions where pivotal pairs are highlighted.

- In the leftmost area are included pairs with low `Type 1` values, meaning a low correlation regarding the external measures. An example are $TTT - NQL$ and $TTT - NAL$ with a `Type 1` correlation equal to 65%. The `Type 2` correlation for these pairs is even lower, aligning with the expectations of a low in-pair similarity regarding the solutions. Function TTT prioritizes the reduction of travel times while both $NAL$and NQL deal with the assignment of patients to nurses.

- The upper part of the graph contains pairs with high value of `Type 2` correlation, hence a higher similarity between the in-pair solution. The involved pairs, such as $TTT - TTW$, $TTT - NAC$, $TTW - STW$, and $TTT - PWT$, all pertain to travel time optimization. Notably, TTT also correlates significantly with PWT and NAC. This correlation is due to these fairness measures often obtaining solutions where the visits from the same patients are close to each other, reducing total costs.

- In the rightmost area pairs are characterized by high `Type 1` correlations, showing solutions similar in the results regarding the other measures. In $VTW - PWT$ and $NAC - PWT$ the solutions are incredibly similar. In fact, all these functions schedule visits to patients close to each other to achieve their fairness objective. A low level of `Type 2` correlation, however, suggests that considering the pair measures the values are completely different. In fact, taking as example, $NAC - PWT$ we can see solutions in which visits are closer but that drastically differ in the number of nurses assigned to a patient. NAC will reach the closeness by sending only one nurse to each patient, the opposite will happen in PWT where, to achieve closeness, multiple vehicles are sent to a node simultaneously.

In conclusion, the pair with robust correlation in both `Type 1` and `Type 2` is $NQL - NAL$. This pairing unites two measures focused on nurses' qualifications, striving to create the most beneficial nurse-patient assignments from either the patient's (NAL) or the nurse's (NQL) perspective.

4.6 Conclusions

In this chapter, we study the daily activities of a TOC tasked with the routing and scheduling of a fleet of nurses to visit patients at their homes. We consider different fairness measures that summarize the needs of both patients and nurses, proposing a set of alternative objective functions beyond the basic problem formulation of minimizing total traveled times. Doing so, we aim to encapsulate in the logistical setting of a NRP the multifaceted nature of different stakeholders. To facilitate our analysis, we've employed a range of alternative models, each one a derivative of distinct fairness measures. These models have been rigorously solved to obtain optimal solutions utilizing advanced Mixed Integer Linear Programming (MILP) solvers. The computational experiments show how specific fairness measures can concur to similar results despite belonging to different stakeholders. These similarities might be helpful from an optimization perspective in reducing the list of fairness measures considered in the development of a more realistic multi-objective problem.

Chapter 5

A Multi-Objective Multi-Actor Nurse Routing Problem including Fairness

Contents

The content of this chapter was presented to the following workshops: 7th AIROYoung Workshop 2023, 3rd EUROYoung Workshop 2023. This chapter corresponds to the paper "V.Bonomi, R.Mansini, R.Zanotti, Mediating governance goals with patients and nurses satisfaction: a multi-actor multi-objective problem including *fairness" published on the International Journal of Production Research on August 2023. (10.1080/00207543.2023.2248522)*

5.1 Introduction

This chapter aims to deepen the study of the problem addressed in Chapter (4) by facing the challenges arising from the optimization of a Home Healthcare (HHC) System involving multiple actors with conflicting goals through a multi-objective formulation. Chapter (4) tackles the diverse goals of multiple stakeholders through a single-objective optimization, a framework that doesn't suit well with the complexity of the relationships that arise from having multiple decision-makers, each with their often conflicting unique priorities. This multifaceted scenario necessitates a more advanced approach to integrate the varying perspectives and needs of each actor involved.

In this context, we keep the problem description of a routing and scheduling problem in home healthcare in which we insert the concept of fairness, recognizing that equitable resource allocation and service provision are crucial for the well-being of both nurses and patients. We keep the majority of fairness measures related to nurses and patients already defined in Chapter (4), like workload balance, equitable patient-nurse matching, and the minimization of disparities in care quality while formulating new governance measures for the TOC. Along with the pure minimization of traveled times, we insert a more patient-oriented policy of minimizing lateness and a nurse-oriented one with the minimization of the time of the last visit. We introduce a variety of multi-objective formulations for the problem, each the broader multi-objective context is not lost while focusing on one primary goal at a time involves a hierarchical prioritization of these objectives, leading to a sequential problem-solving method. In this approach, we address a series of single-objective problems, where each problem's solution becomes a constraint in the subsequent ones. This method ensures that while focusing on one primary goal at a time, the broader multi-objective context is not lost. For every set of three objectives—one for each stakeholder—we conduct a thorough analysis of all six permutations to derive the most advantageous outcome for each stakeholder. This systematic evaluation is crucial for understanding the trade-offs involved allowing the identification of the most effective ordering of objectives. This hierarchy-based analysis is particularly valuable in scenarios where prioritization can significantly impact the overall system's efficiency and fairness. Computationally, addressing hierarchical multi-objective problems, particularly their single-objective formulations in medium to large instances, presents a non-trivial challenge. Advanced Mixed Integer Programming (MIP) solvers, such as Gurobi, often struggle to find feasible solutions within a reasonable time due to the complexity of these problems. We introduce a cutting-edge method to overcome this obstacle: the Parallel Adaptive Large Neighborhood Search (`ParallelALNS`). This approach innovates key components of the traditional ALNS to better handle our problem's multi-objective nature. We develop unique destroy and repair operators that are sensitive to multiple objectives. These operators are specifically designed to effectively deconstruct and reconstruct solutions, taking into account the various goals in our hierarchical framework. Additionally, we create several acceptance operators that leverage the hierarchical structure, guiding the search process more effectively toward optimal solutions. The versatility of these methods makes them suitable for a wide range of hierarchical multi-objective problems. To validate the efficacy of our `ParallelALNS` method, we conduct a comparative analysis with solutions provided by the Gurobi MIP solver, which runs for one hour. To enhance the performance of Gurobi's Branch-and-Cut algorithm, we implement a two-phase matheuristic approach, dubbed `MathALNS`. The first phase utilizes a variant of `ParallelALNS` to generate a strong initial solution. In the second phase, we re-engage Gurobi with this initial solution, effectively incorporating an ALNS as a primal heuristic within the branch-and-bound search tree. This approach aims to refine the upper and lower bounds achieved by Gurobi. Moreover, we position the solutions obtained through `ParallelALNS` on a Pareto frontier, which is established by multiple runs of a `ParetoALNS` strategy. This strategy is dedicated to iteratively producing non-dominated solutions for the multi-objective formulation. By doing so, we can assess the performance of our solutions in the context of a broader solution space, ensuring that our approach not only finds feasible solutions but also contributes to the overall optimality of the problem-solving process. This comprehensive approach underscores the innovative potential of `ParallelALNS` in tackling complex hierarchical multi-objective problems in the realm of home healthcare and beyond.

The Chapter is organized as follows. Section (5.2) reports, for clarity, the problem notation and the mathematical formulation already detailed in Chapter (4), Section (4.2) and (4.3), respectively. Section (5.3) presents the new policies introduced for the stakeholder and Section (5.4) details the applied multi-objective approach implemented and the method used to evaluate the quality of a triplet as objective function. The description of the `ParallelALNS` framework is given in Section (5.5), with a focus on the novel characteristics inserted to handle multi-objective multi-actors problems. Section (5.6) focuses on the computational analysis of the results obtained by solving multiple small-size instances using Gurobi and their comparison using the scoring method. The performance of ParallelALNS is also analyzed on large-size instances through the implementation of a method generating the Pareto frontier, and by devising a matheuristic (`MathALNS`) that allows Gurobi to obtain better upper and lower bounds. Finally, in Section (5.7) we draw conclusions on the presented problem and its results.

5.2 Problem Description and Mathematical Formulation

In this section, the model notation and the mathematical formulation presented in Chapter (4) are detailed again, to give the reader a better understanding of the chapter. The described problem is a Nurse Routing and Scheduling Problem, which involves managing different actors' presence and influence. To address this challenge, we adopt a multi-objective formulation that combines the goals of all three stakeholders (TOC, nurses, and patients) involved.

Let $P = \{1, \cdots, p_{max}\}$ be a set of patients geographically dispersed over a district manned by a TOC, and let $K = \{1, \cdots, m\}$ be a set of nurses in charge of performing services at patients' homes. We indicate as N_p the set of services demanded by patient $p \in P$ and as $N = \{1, \cdots, n\}$ the set of services needed by all the patients, where $N = \cup_{p \in P} N_p$ and $N_p \cap N_{p'} = \emptyset$ for $p \neq p'$. Each service $i \in N$ is associated with a positive service time s_i and must be satisfied by precisely one nurse. Every morning, each nurse starts his/her tour from the TOC, visits a subset of patients fulfilling some or all of their service requests, and then returns to the TOC within the shift duration T (working time). The sequence of patients a nurse visits depends on the optimized objective function.

The problem can be formalized on a directed graph $G = (V, A)$, where $V = \{0\} \cup N$ is the node set, with node 0 representing the TOC, and $A = \{(i, j) : i, j \in V; i \neq j\}$ is the arc set. For each arc $(i, j) \in A$, we denote as t_{ij} the non-negative time required to travel from node i to node j, where $t_{ij} = 0$ for all $i, j \in N_p, p \in P$. Travel times satisfy the triangle inequality.

To define the model, we introduce the following three sets of variables:

- Binary variables $x^k_{ij}, (i, j) \in A, k \in K$. Each variable x^k_{ij} takes value 1 if nurse k traverses arc (i, j), and 0 otherwise;
- Binary variables $y^k_i, i \in N, k \in K$. Each variable y^k_i takes value 1 if nurse k performs service i, and 0 otherwise;
- Continuous variables z_{ij}, $(i, j) \in A$. Each variable z_{ij} indicates the arrival time at node j when arriving from node i.

Regardless of the objective function used, all feasible solutions must satisfy the following set of basic constraints:

$$\text{(5.1)} \qquad \sum_{(i,j)\in\delta^+(i)} x_{ij}^k = \sum_{(j,i)\in\delta^-(i)} x_{ji}^k = y_i^k \quad i \in N, k \in K$$

$$\text{(5.2)} \qquad \sum_{k\in K} y_i^k = 1 \quad i \in N$$

$$\text{(5.3)} \qquad \sum_{(0,j)\in\delta^+(0)} x_{0j}^k = \sum_{(j,0)\in\delta^-(0)} x_{j0}^k \leq 1 \quad k \in K$$

$$\text{(5.4)} \qquad \sum_{(i,j)\in\delta^+(i)} z_{ij} - \sum_{(j,i)\in\delta^-(i)} z_{ji} = \sum_{k\in K} \sum_{(i,j)\in\delta^+(i)} (s_i + t_{ij}) x_{ij}^k \quad i \in N$$

$$\text{(5.5)} \qquad z_{0j} \geq t_{0j} \sum_{k\in K} x_{0j}^k \quad j \in N$$

$$\text{(5.6)} \qquad (t_{0i} + s_i + t_{ij}) \sum_{k\in K} x_{ij}^k \leq z_{ij} \leq (T - t_{j0} - s_j) \sum_{k\in K} x_{ij}^k \quad (i,j) \in A$$

$$\text{(5.7)} \qquad z_{ij} \geq 0 \quad (i,j) \in A$$

$$\text{(5.8)} \qquad x_{ij}^k \in \{0,1\} \quad (i,j) \in A, k \in K$$

$$\text{(5.9)} \qquad y_i^k \in \{0,1\} \quad i \in N, k \in K$$

For each subset $S \subset N$ of nodes, let $\delta^+(S) = \{(i,j) \in A : i \in S, j \notin S\}$ and $\delta^-(S) = \{(i,j) \in A : i \notin S, j \in S\}$ be the sets of arcs leaving and entering set S, respectively. To simplify notation, the special case with $|S| = 1$ is indicated as $\delta^+(i)$ ($\delta^-(i)$) instead of $\delta^+(\{i\})$ ($\delta^-(\{i\})$). Constraints (5.2) impose that if nurse k visits node i (i.e., $y_i^k = 1$), precisely one arc entering and one arc leaving the node are selected. Constraints (5.2) guarantee that each node (each service) is visited (is executed) exactly once by one nurse. Constraints (5.3) restrict the number of nurses who can depart from and return to the TOC (node 0) to be no more than $|K|$. Constraints (5.4) ensure that if nurse k visits node j immediately after node i (i.e., $x_{ij}^k = 1$), then the time elapsed between the arrival times in the two nodes is equal to the service time s_i at node i plus the travel time t_{ij} to move from node i to node j. Constraints (5.5) set a lower bound on the time required to reach the first visited node after leaving the TOC, whereas constraints (5.6) define lower and upper bounds on the arrival time and duration of each route. More precisely, they ensure that if a nurse arrives at node j from node i, it is early enough to complete the service and return to the TOC before the shift duration T expires, and late enough to account for the travel time from the TOC to node i, from i to j, and for the service time at node i. Finally, constraints (5.7)–(5.9) enforce non-negative and binary conditions on variables.

Notice that constraints (5.4)–(5.6) prevent the construction of subtours. Nevertheless, we have further strengthened this formulation by adding the generalized connectivity constraints (see Hanafi et al., 2020):

$$\sum_{(i,j)\in\delta^+(S)} x_{ij}^k \geq y_s^k \quad S \subseteq N, |S| \geq 2, k \in K, s \in S. \tag{5.10}$$

Finally, it is worth noticing that, to correctly model some of the fairness objectives discussed in the next section, we need to identify which nurse provides which service for each patient. For this reason, we use a three-index formulation. Alternatively, one can avoid this by duplicating the depot a number of times equal to the number of nurses, thus separating the starting and ending depot for each nurse in a fictitious way (see Luo et al., 2015).

5.3 Stakeholders' Goals

IIn Section (5.3.1), we detail the new governance goals for the Territorial Operations Center (TOC), complete with their descriptions and mathematical formulations. Following this, Sections (5.3.2) and (5.3.3) provide an overview of the objectives for nurses and patients, as initially introduced in Section (4.4). Here, we highlight the key changes and continuities from the previous chapter. For the nurses' objectives, we retain the four functions STW,TTW,PWZ and NQL, without modification. On the patient side, however, we have made notable adjustments. Specifically, we have eliminated the PWT goal due to its high optimality gaps in smaller-sized instances. Moreover, we have restructured the *Equity in Visiting Time Windows* (VTW) measure. Rather than quantifying earliness and lateness concerning a patient-given time window, we have introduced the *Equity in Late Service Time* (LST) metric. This new measure focuses exclusively on the lateness aspect, calculated from a specific point in time, thus providing a more targeted assessment of service timeliness from the patients' perspective.

5.3.1 TOC-centered measures

In contrast with the previous chapter, we introduce two new governance measures for the TOC, one aimed at improving the operational costs of the staff and the other for guaranteeing a good quality of service for the patients. In total we study three TOC-related measures: the minimization of Total Traveled Time (TTT), the Total Tardiness (TTA) and the Time of Last Visit (TLV).

Total Traveled Time (TTT)

Already presented as objective function of the overall model in (4.1), TTT aims at reduce the total time spent by nurses in travelling to visit patients. It can be formulated as:

$$\min \sum_{k\in K} \sum_{(i,j)\in A} t_{ij}x_{ij}^k \tag{5.11}$$

This is the most frequently used measure in the computation of costs for healthcare operations. However, other than a pure optimization of operations, it doesn't account for the patients or the nurses needs.

Total Tardiness (TTA)

This governance goal aims at minimizing the overall tardiness in respect to patients given deadline of visit. To each patient $p \in P$ it is associated with a deadline a_p, which represents the time by which the patient expects to complete all the required services. Whether the patient requires one or more visits, the deadline represents the moment after which the patient would like not to receive any more nurse visits. Although this is not a hard constraint in terms of TOC operations, the patients perceive a lower level of tardiness as a higher level of service quality. Let $t_i, i \in N_p$, be a non-negative variable measuring the time overrun (tardiness) of deadline a_p for patient p on his service i. TTA can be formulated as follows:

$$\min \sum_{i\in N} t_i \tag{5.12}$$

$$\sum_{(i,j)\in\delta^-(j)} z_{ij} \leq a_p + t_j \quad p \in P, j \in N_p \tag{5.13}$$

$$t_i \geq 0 \quad i \in N \tag{5.14}$$

Constraints (5.13)–(5.14), along with objective function (5.12), define the value of tardiness at node j of customer p as the positive difference between the arrival time at node $j \in N_p$ and the deadline a_p, i.e., $\max\{0, \sum_{(i,j)\in\delta^-(j)} z_{ij} - a_p\}$. Notice that optimizing global tardiness may not be equitable, as it could result in an unfair distribution of tardiness values among patients.

Time of the Last Visit (TLV)

To prioritize the nurses perspective while minimizing operational costs, the TOC might be interested in minimizing the completion time of all the services (which is equivalent to the time of completion of the last visit). In this way, the overall working shift of nurses would also be reduced. Let t_{last} denote this completion time. TLV can be formulated as follows:

$$\min t_{last} \tag{5.15}$$

$$t_{last} \geq \sum_{(i,j)\in\delta^-(j)} z_{ij} + s_j \quad j \in N \tag{5.16}$$

$$t_{last} \geq 0 \tag{5.17}$$

Given objective function (5.15), constraints (5.16) set t_{last} exactly equal to the maximum value of the leaving time for each node $j \in N$, computed as the arrival time in j ($\sum_{(i,j)\in\delta^-(j)} z_{ij}$) plus the service time in the node.

5.3.2 Nurse-centered measures

Given that the included nurses measures have not been modified since Section (4.4), we report here only the denomination and a brief description. For each one are referenced the objective function and constraints indexed from Section (4.4). The modeled nurses' perspectives are:

- Total Time Workload (TTW), modeled in Constraints (4.12)–(4.14) and Constraints(4.16), aimed at achieving a fair distribution of workload computed as traveled time plus service times;
- Service Time Workload (STW), modeled in Constraints (4.12)–(4.14) and Constraints(4.15)), aimed at achieving a fair distribution of workload computed only as service times at patients;
- Preferred Working Zone (PWT), modeled in Constraints (4.17)–(4.21), achieving a fair assignment of patients to nurses according to their preferred working zone;
- Nurse Qualification Level (NQL), modeled in Constraints (4.22)–(4.26), equitably assigning visits to nurses according to their qualification level.

5.3.3 Patient-centered measures

We have modeled a total of three different equity measures for the patients. In particular, we re-implemented the measures

- Nurse Assignment Consistency (NAC), modeled in Constraints (4.41)–(4.48), minimizing the maximum number of nurses assigned to any patient needing multiple visits;
- Nurse Adequacy Level (NAL), modeled in Constraints (4.46)–(4.48), making fair nurse-patients assignments according to nurses qualifications, the higher the specialization level of a nurse in a specific service, the higher the adequacy perceived by the patient;

from Section (4.4) and added the Equity of Late Service Time (LST).

Equity in Late Service Time (LST)

This equity measure aims to establish a fair time for patients to receive their visits in relation to the deadline a_p set by each patient $p \in P$, which is the time by which they would like to receive all of their visits. For each service $j \in N_p$, we introduce the continuous variable l_j that measures the lateness (the discard) in starting service j with respect to a_p (i.e., $l_j = \max\{0, \sum_{(i,j)\in\delta^-(j)} z_{ij} - a_p\}$). The goal is to balance the worst lateness among all patients. The continuous variable t_p measures the total lateness for each patient $p \in P$ computed as the sum of all positive discards from deadline a_p for all the services belonging to patient p (i.e., $t_p = \sum_{i\in N_p} l_i$). Finally, variable t_{max} represents the maximum t_p value among all the different patients $p \in P$. The formulation is as follows:

$$\min t_{max} \tag{5.18}$$

$$t_{max} \geq t_p \qquad p \in P \tag{5.19}$$

$$t_p \geq \sum_{i\in N_p} l_i \qquad p \in P \tag{5.20}$$

$$l_j \geq \sum_{(i,j)\in\delta^-(j)} z_{ij} - a_p \qquad p \in P, j \in N_p \tag{5.21}$$

$$l_i \geq 0 \qquad i \in N \tag{5.22}$$

$$t_p \geq 0 \qquad p \in P \tag{5.23}$$

$$t_{max} \geq 0 \tag{5.24}$$

Constraints (5.21) model the lateness for each service $j \in N_p$, for each patient $p \in P$. Notice that, if the arrival time at node j, measured by $\sum_{(i,j)\in\delta^-(j)} z_{ij}$, is lower than the deadline a_p, the value of l_i will be greater than or equal to zero due to (5.22) and thus forced to zero by the objective function.

5.4 Multi-Objective Optimization Approach

In this section, we first present the multi-objective optimization approach selected to tackle the multi-objective and multi-actor nature of our problem in Section (5.4.1). Then, in Section (5.4.2) we introduce the scoring method used to determine the best triplet of objective functions (one for each actor).

5.4.1 The Lexicographic Approach

MOO problems are characterized by the presence of multiple, possibly conflicting, objectives that need to be optimized simultaneously. In general, an

MOO problem consists of an unordered sequence of objective functions defined over a feasible set X generated by common constraints:

$$\min_{x \in X} \{f_i(x), i = 1, ..., r\}$$

In this article, we propose several alternative formulations of our problem. Each one is characterized by a defined triplet of objective functions in a specific order $[f_1, f_2, f_3]$, where each $f_i, i = 1, ..., 3$ is associated with one of the three actors involved. To solve this model, we employ a hierarchical (lexicographic) method, which establishes a predefined order (usually determined by the decision maker) among objective functions and then solves a series of single-objective optimization problems.

Let us assume that we have an ordered sequence $f_1, \ldots, f_r$ of goals, all of which need to be minimized. For each level i of optimization, the associated single objective problem will be:

$$\min \quad f_i(x) \tag{5.25}$$

$$\text{s.t.}$$

$$f_j(x) = f_j(x_j^*) \quad j \in \{1, \ldots, i-1\}, i \geq 2 \tag{5.26}$$

$$\mathbf{x} \in X \tag{5.27}$$

where $f_j(x_j^*)$ represents the optimal solution value for the higher-priority objective j computed in the previous $j = 1, ..., i-1$ problem resolutions.

This approach presents a distinct advantage in facilitating explicit considerations of trade-offs among diverse objectives. It empowers decision-makers to balance prioritizing critical goals while retaining a degree of flexibility for secondary objectives. In applying the lexicographic framework to our case, we concentrate on two fundamental elements: multiple goals, including those representing fairness measures, and the diversity of stakeholders involved. This allows for a double-level optimization, prioritizing both goals and actors. To this aim, we applied an approach working on two different scopes. The first scope is to highlight the relationships among various goals, with particular attention to those including equity considerations. To this end, we generate all possible combinations of three goals, each representing a different stakeholder. Subsequently, the second objective aims to examine these goal triads individually, adjusting each priority to catch the effect of favoring one stakeholder over others on the overall solution. Through a scoring mechanism, our approach can identify the triplet that represents the optimal goal combination. This combination is characterized by its minimal trade-off in terms of stakeholder dissatisfaction. The specifics of this method are elaborated in the subsequent section

5.4.2 The Scoring Method

Scoring methods are useful tools that help solve complex problems, especially when we need to pick the best options from a set. These methods make it easier to weigh different choices against each other. We examined different methods from literature to find the most suitable for our problem. One popular method is the Analytic Hierarchy Process (AHP) cited by Saaty, 1990 and its different versions like the Analytic Network Process (ANP) and AHPSort of Saaty and Vargas, 2013; Ishizaka et al., 2012. AHP simplifies complex decision-making by decomposing it into more manageable components and employing pairwise comparisons to establish priorities. ANP extends this approach to scenarios characterized by interdependent elements, functioning within a networked structure. Another notable method is the Elimination by Aspects (EbA) presented in Tversky, 1972, which systematically excludes options that fail to meet essential criteria, proving invaluable when dealing with indispensable requirements. Furthermore, collaborative decision-making models, such as those discussed by Sirikijpanichkul et al., 2017, incorporate collective preferences and priorities in the scoring process. However, our research aims to formulate a distinct scoring methodology for VRPs in healthcare and logistics. This novel approach diverges from traditional methods like AHP, which rely on predetermined priorities, or EbA, which operates on inflexible criteria. Our proposed method aims to organically identify the most pertinent objectives, treating all objectives with equal initial significance. This approach offers enhanced adaptability and is more adept at capturing the holistic aspirations of all stakeholders involved in the routing problem.

In the defined problem, the amalgamation of distinct objectives from various stakeholders results in 36 non-ordered triplets. For any given triplet, each objective can assume one of three hierarchical priority levels, leading to six possible ordered triplets due to the permutations of these three objectives. From a stakeholder perspective, each Total Operations Cost (TOC) goal appears in 12 non-ordered triplets, formed by combining three patient goals with four nurse goals. Correspondingly, each patient objective is present in 12 non-ordered triplets, while each nurse objective features in nine. Our scoring methodology determines the optimal goal for each stakeholder by internally ranking their objectives. This ranking is derived from the performance of each objective when optimized as the primary function within all priority levels of the ordered triplets. For instance, a TOC goal undergoes optimization 24 times $(2 \cdot 12)$ at priority level 1, with analogous occurrences at other levels. When an objective is optimized at a specific priority level i, it facilitates the assessment of other objectives for the same stakeholder. We denote the value of the optimized objective at priority level i as $z^*_{obj,i}$, and the value of an alternative objective $o\bar{b}j$ evaluated under the solution derived from optimizing *obj* at priority level i as $z_{\bar{obj}}^{obj,i}$. For example, $z^*_{TLV,2}$ signifies the optimization value of the Time Lost in Traffic (TLV) goal at priority level 2, while $z_{TTA}^{TLV,2}$ denotes the evaluated value of the Total Time of Arrival (TTA) objective using the solution optimized for TLV at priority level 2. Each TOC goal has $24 \cdot 3$ values from optimizing it across all priority levels and an equal number of recomputed values for alternate goals. This optimization is replicated for each goal, leading to a reevaluation of values for other objectives. The objectives are ranked accordingly after calculating the average values for each goal and priority level. The detailed methodology is encapsulated in the pseudocode presented in Algorithm (6).

The ranking method (`StakeholderObjSelector` function) takes as input the set of instances $\mathcal{I}$, the set of stakeholders S, and the collection of all possible goals for all stakeholders $\mathcal{G}$. The method first generates all possible ordered tuples of goals (one for each stakeholder) and stores them in $\mathcal{F}$ (procedure GENERATETRIPLETS, Line (1)). In our case, as there are three stakeholders, GENERATETRIPLETS will produce $36 \cdot 6 = 216$ ordered triplets. Next, for each instance $I \in \mathcal{I}$ and each ordered triplet $[f_1, f_2, f_3]$, the method solves a hierarchical model to obtain a solution *sol* and the optimized values of the objective functions $z^*_{f_i,i}$, $i = 1, ..., 3$ for each stakeholder's goal (Line (4)). Based on such solution, the method computes the values $z_f^{f_i,i}$ for $f \neq f_i, i = 1, ..., 3$ of the other goals for each stakeholder (Lines (7)-(9)).

For each stakeholder, the method then proceeds to assign a score to every goal of the stakeholder itself. This two-step process is repeated for each priority level and stakeholder goal. In the first step, given the results obtained (included in $\mathbf{z}$), goals f and g, and a priority level i, Procedure COMPUTEAVERAGE (Line (16)) returns the average value of g when function $f \in G_s$ is optimized with priority level i. Notice that, when $f = g$ and $i = 1$, this is exactly the value of $z^*_{f,i}$. In the second step, based on the relative position of these average values, a positive score is assigned to each function (procedure COMPUTESCORE at Line (18)). Since all objective functions are minimization ones, the lower the average value, the higher the score of a function. We set the highest score equal to 1. Finally, the global score associated with a function is obtained by summing up its average scores computed with respect to all the stakeholders and all priority levels. Sorting the global scores allows us to identify the best function for each stakeholder (Line (27)). These best functions $\mathbf{f}^*$ are provided as output.

5.5 A Parallel ALNS

The core aim of our research was to develop a methodology capable of tackling realistic-sized scenarios of the problem at hand, as existing state-of-the-art solvers often struggle to generate high-quality solutions within a feasible

Algorithm 6 `StakeholderObjSelector`

Require: Instance set $\mathcal{I}$, Stakeholder set S, collection $\mathcal{G} = \{G_s | s \in S\}$ with G_s set of goals for stakeholder $s \in S$.

$\mathcal{F} \leftarrow$ GENERATETRIPLETS$(\mathcal{G})$
for all $I \in \mathcal{I}$ **do**
    for all $[f_1, f_2, f_3] \in \mathcal{F}$ **do**
        $(sol; z^*_{f_1,1}, z^*_{f_2,2}, z^*_{f_3,3}), \leftarrow$ OPTIMIZE$(I, [f_1, f_2, f_3])$
        for $f_i \in [f_1, f_2, f_3]$ **do**
            $s \leftarrow$ STAKEHOLDER(f_i)
            for all $f \in G_s, f \neq f_i$ **do**
                $z_f^{f_i,i} \leftarrow f(sol)$
            end for
        end for
    end for
    for $s \in S$ **do**
        for $i = 1 \ldots 3$ **do**
            for all $g \in G_s$ **do**
                for all $f \in G_s$ **do**
                    $avg_f \leftarrow$ COMPUTEAVERAGE$(\mathbf{z}, g, f, i)$
                end for
                $score \leftarrow$ COMPUTESCORE(avg, G_s)
                for all $f \in G_s$ **do**
                    $globalScore(f) \leftarrow globalScore(f) + score(f)$
                end for
            end for
        end for
    end for
end for
for $s \in S$ **do**
    $f(s)^* \leftarrow$ SORTANDPICKBEST$(globalScore, G_s)$
end for
return $\mathbf{f}^*$

time frame. In response, we formulated a novel metaheuristic framework with the following characteristics:

- it addresses a multi-objective, multi-actor scenario without necessitating a predefined hierarchy among the input objective functions.
- it demonstrates adaptability in its exploration of the solution space, leveraging the hierarchical and multi-dimensional nature of the problem.
- it employs a parallel approach for handling large-scale instances efficiently, wherein a central *manager* coordinates the flow of crucial information among several *workers*.

Our innovation includes a new parallel adaptation of Adaptive Large Neighborhood Search (ALNS),a meta-heuristic approach originally proposed in Ropke and Pisinger, 2006 presented in Chapter (3), Section (3.2). In our specific context, a feasible solution *sol* for our problem, characterized by objective function values $obj_{sol} = [obj^1_{sol}, obj^2_{sol}, obj^3_{sol}]$, entails creating up to $|K|$

routes (one per nurse), with each route comprising a sequence of services for various patients. The *destroy operators* selectively remove services from routes based on certain criteria, resulting in a temporary infeasible solution. Subsequently, *repair operators* attempt to reinsert services using different criteria to restore feasibility. Our unique implementation, termed `ParallelALNS`, distinguishes itself by identifying optimal solutions for every possible combination of the input objective functions, rather than optimizing a singular predetermined order. This is achieved through parallelization: multiple ALNS instances (*workers*) operate concurrently, exchanging information via a shared *manager*. Each worker handles an ordered set of objectives, with the possibility of several workers addressing the same set, depending on the chosen level of parallelism. This approach is predicated on the understanding that solutions suboptimal for one worker due to different prioritization of objectives might be valuable to another.

Algorithm (7) outlines the general structure of our `ParallelALNS` framework, which is executed by each worker. It requires as input an ordered triplet of objective functions $[f_1, f_2, f_3]$, as well as a feasible solution *sol* and its value obj_{sol}. In addition, it takes a set of destroy and repair operators, Θ^- and Θ^+, a set of acceptance operators Γ, and a time limit τ. The algorithm starts by initializing both the current solution (sol_C) and the best incumbent solution (sol_I) to the input solution *sol* (Lines (1)–(2)). It selects an acceptance operator (Line (3)) and initializes the degree of destruction q to 1 (Line (4)). The algorithm then enters a loop (Line (5)) that continues until the termination criterion (i.e., time limit τ) is met. First, a destroy (θ^-) and a repair operator (θ^+) are selected (Line (6)) and applied to the current solution (Lines (7)-(8)). Then, depending on the acceptance operator γ, one of two things happens: either the newly generated solution (sol_E) is accepted (Lines (9)-(15)), q is set to 1, and the incumbent solution is updated, if necessary (Lines (12)-(14)), or, if $noImpIter$ iterations have passed since the last accepted solution, q is increased by $qStep$. When a solution sol_E is accepted, it is shared with other workers through the $ALNSManager$.PUSH routine (Line (11)). The probabilities of the selected destroy and repair operators are then updated according to their performance (Line (19)). At Lines (20)-(32), the algorithm performs an epoch reset, a standard feature of ALNS implementations. The operators' probabilities are reset (Line (21)), and the current solution is set to the best incumbent (Line (22)). Differently from standard ALNS implementations, if the incumbent has not been improved for $imprEpoch$ epochs, the acceptance operator is changed (Line (24)). The current solution s_C is then set to the best available solution shared by all workers, provided by the ALNS manager through $ALNSManager$.GETBEST (Line (25)). Additionally, to leverage the information collected during the algorithm's execution and try to improve the best incumbent, an improvement phase is called (MIPIMPROVEMENT).

Algorithm 7 `ALNS-Worker`

Require: ordered set of goals $[f_1, f_2, f_3]$; feasible solution (sol, obj_{sol}); sets of destroy and repair operators Θ^-, Θ^+; set of acceptance operators Γ; time limit τ.

1: $sol_C \leftarrow sol$ ▷ current solution
2: $sol_I \leftarrow sol$ ▷ best incumbent solution
3: select $\gamma \in \Gamma$
4: $q \leftarrow 1$
5: **repeat**
6: select $\theta^- \in \Theta^-$ and $\theta^+ \in \Theta^+$
7: $[\overline{sol}_C, N^-] \leftarrow \theta^-(sol_C, q)$
8: $sol_E \leftarrow \theta^+(\overline{sol}_C, N^-)$
9: **if** $\gamma(sol_E, sol_C)$ **then**
10: $sol_C \leftarrow sol_E$
11: $ALNSManager$.PUSH(sol_C)
12: **if** ISBETTER(sol_E, sol_I) **then**
13: $sol_I \leftarrow sol_E$
14: **end if**
15: $q \leftarrow 1$
16: **else if** no solution accepted for $noImpIter$ **then**
17: $q \leftarrow min(q + qStep, q_{max})$
18: **end if**
19: update probabilities of θ^- and θ^+
20: **if** *epochIter* is reached **then**
21: reset probabilities of operators in Θ^- and Θ^+
22: $sol_C \leftarrow sol_I$
23: **if** sol_I not improved for $imprEpoch$ **then**
24: randomly select $\gamma \in \Gamma$
25: $sol_C \leftarrow ALNSManager$.GETBEST($[f_1, f_2, f_3]$)
26: $\bar{A} \leftarrow ALNSManager$.GETRESTRICTEDNETWORK($[f_1, f_2, f_3]$)
27: $sol_C \leftarrow$ MIPIMPROVEMENT($sol_C, \bar{A}$)
28: **if** ISBETTER(sol_C, sol_I) **then**
29: $sol_I \leftarrow sol_C$
30: **end if**
31: **end if**
32: **end if**
33: **until** time limit τ is reached
34: return sol_I

During the improvement phase, the current worker receives a list of promising arcs (restricted network) from the manager. This list corresponds to the arcs present in the best solutions found so far by all workers and made available to the manager through the $ALNSManager$.GETRESTRICTEDNETWORK procedure (Line (26)). Notably, the restricted network is updated continuously with new solutions pushed by the workers. Thus, two consecutive calls to $ALNSManager$.GETRESTRICTEDNETWORK may provide different lists of arcs $\bar{A}$. The MIPIMPROVEMENT procedure uses the arcs contained in the current solution sol_C and the promising set of arcs $\bar{A}$ to construct a restricted problem and solves it through a MIP solver with a very short time limit. The procedure then updates the best incumbent if necessary (Lines (28)-(30)).

In the following sections, we describe the main components of our `ParallelALNS` method, including the initial feasible solution construction, the set of destroy and repair operators, the acceptance operators, and the role played by the manager in sharing information and coordinating the workers.

5.5.1 Initial Feasible Solution

In our approach, the initial step involves identifying a starting feasible solution, denoted as *sol*. We employ a constructive heuristic, beginning with $|K|$ empty routes. A node $i \in N$ is selected at random, adhering to specific selection criteria (to be detailed), and added to a route via the cheapest insertion rule. This rule seeks to minimize the total length of all routes by determining the optimal position for node insertion. The process continues until all nodes are assigned or an insertion failure occurs, which is when a node cannot be feasibly added to any route under the given constraints. In such cases, the current partial solution is discarded, and the process restarts. Upon finding a feasible solution, it is evaluated against the incumbent solution based on criteria such as total distance, cost, and time, and is retained only if it offers an improvement. This heuristic is executed iteratively within a brief time limit, τ_{init}, typically just a few seconds. To fully leverage the computational capacity, particularly in a scenario with *nCore* logical cores available, we run *nCore* instances of the constructive heuristics in parallel. The optimal solution from these parallel runs is then selected. This approach, while efficient, assumes [insert assumptions], and its effectiveness within the limited time frame warrants further statistical analysis for reliability assessment.

5.5.2 Destroy and Repair Operators

In the architecture of our `ParallelALNS` system, each unit is designed to tackle optimization tasks that involve multiple objectives, arranged in a lexicographic order. This is facilitated by the development of specialized destroy and repair mechanisms. These mechanisms are versatile, capable of focusing on a singular objective, a pair of objectives, or all three objectives simultaneously. This flexibility allows them to selectively target services for removal and reinsertion based on one, two, or all three objectives. The advantage of this method lies in its efficiency: by concentrating on fewer objectives at a time, it streamlines neighborhood exploration and minimizes the need for computing multiple objectives for every potential solution. Additionally, our approach achieves a higher degree of diversification by not taking into account one or two of the less significant objective functions.

In the Ω^- set, each destroy operator removes service requests based on the *worst removal* criterion. For its application for single-objective problems, the reader is referred to Section (3.1.1). With multiple objectives, we assess the removal impact either hierarchically, prioritizing higher-ranked objectives, or through a weighted scheme using predefined weights (γ_1, γ_2, γ_3). This results in either prioritizing major improvements in top objectives or balancing improvements across objectives. Given a set of three objectives, we explore all possible combinations (15 in total), which, when accounting for the hierarchical and weighted evaluations (12 out of 15 combinations), leads to two distinct sorting methods for each combination. Additionally, we incorporate a random sorting approach, culminating in 28 distinct sorting possibilities. Once the service requests are sorted, different ways of drawing them produce different destroy operators. We implement two different removal rules; the first rule involves removing the top-ranked request with a set probability, continuing until q requests are removed or restarting the cycle if necessary. The second rule adapts the Shaw removal strategy (Section (3.1.1)), starting not randomly but with the first request in the sorted list. This leads to a total of 56 destroy operators (28 sorting methods times 2 removal rules).

Similarly, the repair operators in the Ω^+ set follow analogous principles. Removed services are re-sorted for reinsertion using various criteria, including random and objective function-based methods, either hierarchically or weighted. Each repair operator then calculates the cheapest insertion cost for each request, arranging them accordingly for reinsertion. Consequently, with three objective functions, we have 28 distinct repair operators.

5.5.3 Acceptance Operators

In contrast to the conventional ALNS, our approach integrates multiple acceptance operators, rather than just one, to effectively navigate the multi-objective dimensions of our problem. These acceptance operators, akin to the destroy and repair operators in traditional models, consider multiple objectives when evaluating whether to accept a solution. Specifically, given a candidate solution sol_E and the current selected solution sol_C, we compute the probability of sol_E to be selected as:

$$e^{\frac{min(0,\sum_{i=1}^{i_{max}} w_i(obj^i_{sol_C} - obj^i_{sol_E})}{T}}$$

where i_{max} is the number of objective functions (possibly lower than 3) taken into account by the acceptance rule, w_i is the weight associated with the i-th objective function, and T is the temperature for the iteration in which the acceptance has to be decided.

Similar to the simulated annealing method, the initial temperature T is set to a default value T_0 and is subsequently reduced after every T_{iter} iterations. This reduction follows a geometric progression with a cooling rate equal to $0 < \alpha < 1$. This approach aims to prevent the algorithm from becoming trapped in certain regions of the solution space, thereby enhancing the likelihood of discovering more favorable solutions, particularly for the secondary and tertiary objectives.

5.5.4 The ALNS Manager

In the proposed parallel infrastructure, the ALNS is executed by multiple workers operating in parallel, under the supervision of a centralized ALNS manager. This manager performs two crucial functions:

- **Best solution record**: The manager is tasked with keeping track of the optimal solutions for each hierarchical arrangement of the objective functions. As workers find acceptable solutions *solC* (referenced in Algorithm (7) through the function $ALNSManager.\textsc{Push}(sol_C)$, these are communicated to the manager. The manager then assesses whether these solutions surpass existing records in any of the objective function hierarchies. Upon a worker's request for the current best solution for a specific order of objectives $[f_1, f_2, f_3]$ (as per Algorithm (7), $ALNSManager.\textsc{GetBest}([f_1, f_2, f_3])$, the manager responds with the leading solution from the relevant list.

- **Escape local minima via mathematical programming**: Drawing on prior research that demonstrates the efficacy of combining ALNS with mathematical programming techniques (Mansini and Zanotti 2019), the manager also engages in facilitating escape from local minima. Specifically, the manager provides a subset of highly promising arcs $\bar{A}$ to assist workers in formulating a constrained version of the problem (as described in Algorithm (7), $ALNSManager.\textsc{GetRestrictedNetwork}$). When a worker reaches a point where the incumbent best solution remains unimproved over several epochs, it resorts to this restricted problem, leveraging the insights accumulated by the manager, including the best solution available (Algorithm (7),$\textsc{MIPImprovement}(sol_C, \bar{A})$). The restricted problem involves only a selected subset of arcs $\bar{A}$ from the complete set A, enabling a swift resolution to optimality. The manager's role here is to discern the most pertinent arcs for a given triplet of objectives, using the recorded best solutions. Solutions are ranked by their performance on the objectives, and each arc in the top χ solutions is scored, with arcs in the r-th ranked solution receiving an incremental score of $\chi - r$. After evaluating all solutions, arcs are ordered based on their cumulative score, and the top ψarcs, determined by a predefined percentage of the total arc count, are provided to the requesting worker.

5.6 Computational Results and Managerial insights

In this section, we delineate the outcomes of our computational experiments on the impact of varying goal combinations, each with distinct priorities, on the overall solution and stakeholder satisfaction. Initially, we introduce the benchmark scenarios that underpin our analysis. This is followed by a discussion offering managerial perspectives, particularly focusing on the implications for small to medium-sized scenarios. Concluding this section, we evaluate the efficacy of our `ParallelALNS` approach in the context of scenarios that mirror real-world complexities. All tests were run on a machine running Ubuntu 20.04.2, equipped with an AMD Ryzen 9 3950x CPU, 16 cores, 32 threads, and 32 GB of RAM. We used Gurobi 10.0.1 as mixed integer programming solver. All methods have been implemented in Java 17.

5.6.1 Instances Generation

We have evaluated the different problem formulations on a large number of benchmark instances. In particular, we use a set of *small-size* instances (Set 1) solved to optimality using Gurobi, on which we conduct the managerial analysis, providing valuable insights for decision-makers and two sets of *large-size* instances (Set 2 and Set 3) to evaluate the performance of `ParallelALNS` in terms of computing time and quality when the main instance parameters (number of patients, maximum number of services per patient, and number of nurses) are varied. For all sets of instances, the locations of patients (and thus, services) are randomly generated within a geographical square such that the maximum travel time between any two nodes is equal to 120 minutes.

More precisely:

- **Set 1** consists of 20 small-size instances characterized by the same number of services (nodes) $|N|$, nurses $|K|$, and the maximum number of services n_{max} per patient (the number of services for each patient is generated uniformly random between 1 and n_{max}). Such parameters are set to 20, 2, and 2, respectively. Each instance differs for the number of patients (generated uniformly random between 10 and 20) and their locations.

- **Set 2** consists of 20 large-size instances all with ($|N|$,$|K|$,n_{max})=(75,5,3). Similarly to Set 1, each instance differs for the number of patients (generated uniformly random between 25 and 75) and their locations.

- **Set 3** consists of 120 instances in which the number of patients is fixed at 30, while the number of nodes $|N|$ is variable and determined by the maximum number of services per patient $n_{max} \in \{1, 3, 5\}$, and the number of nurses varies across $|K| \in \{5, 7\}$. For each combination of parameter values $(|K|, n_{max})$, we generate 20 random instances that differ for the locations of the patients.

All three sets of instances are available for download at https://or-dii.unibs.it/index.php?page=nurse-fairness.

Table (5.1) shows the dimensions and configuration of the first two sets of instances. In particular, it reports the minimum, average, and maximum number of patients per instance (*#Patients*). Table (5.2) displays the features of instances in Set 3, which were introduced to duly analyze the scalability of `ParallelALNS` when increasing both the number of nurses and services.

Set	$\|N\|$	$\|K\|$	n_{max}	*#Patients*			*#Inst*
				min	avg	max	
1	20	2	2	10	13.3	20	20
2	75	5	3	25	37.3	75	20

TABLE 5.1: Instances structure of Set 1 and Set 2.

$\|P\|$	$\|K\|$	n_{max}	$\|N\|$			*#Inst*
			min	avg	max	
30	5	1	30	30	30	20
		3	30	64.35	90	20
		5	30	87.3	150	20
	7	1	30	30	30	20
		3	30	59.3	90	20
		5	30	89.05	150	20

TABLE 5.2: Instances structure of Set 3.

Each small-size instance in Set 1 has been solved for all possible ordered triplets of objective functions, which amounted to 216 in total. Set 2 and Set 3 large-size instances have been tested only on the three objectives finally selected by procedure `StakeholderObjSelector`, representing the best goal for each stakeholder. These correspond to the minimization of the Time of the Last Visit (TLV) for the TOC, of the worst Total Time Workload (TTW) for the nurses, and of Late Service Time (LST) for patients. Notice that both TTW and LST are equity measures, whereas TLV is not.

5.6.2 Managerial Insights on Small-Size Instances

In this section, we detail the outcomes derived from the application of our scoring method, alongside pivotal managerial insights gleaned from examining the solution sets of instances in Set 1. The core objective of these insights is to elucidate for stakeholders the interplay between their individual objectives and the collective impact that emerges when these objectives intersect with those of other stakeholders.

Scoring Methods Results

Table (5.3) provides a summary of the average scores obtained for each instance (listed in columns), grouped by stakeholder (three blocks listed in rows), as computed by the STAKEHOLDEROBJSELECTOR function. We recall that the lower the value the better the ranking position. Within each column, the lowest score for each stakeholder is highlighted in bold font. For example, in instance 12, both NAC and LST in the patient's goals have the same best ranking value of 1.8. The final average score (across all instances) used to identify the triplet to optimize in Set 2 and Set 3 is shown in the *Avg* column. Thus, the triplet finally chosen is `TLV-LST-TTW` with average scores of 1.7, 1.7, and 1.9, respectively. Only the first decimal digit is shown for clarity. TTW has been chosen over STW because its score is slightly better at 1.929 compared to 1.933 for STW.

TABLE 5.3: Average scores for the 20 instances of Set 1.

Obj	Instance																				Avg
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
TTT	2.0	2.3	2.2	2.3	2.3	2.1	2.3	2.2	2.2	2.2	2.1	2.0	2.2	2.1	2.1	2.3	2.2	2.3	2.3	2.2	2.2
TTA	2.2	2.0	2.1	2.0	2.0	2.1	2.0	2.1	2.1	2.1	2.2	2.2	2.1	2.1	2.1	2.0	2.1	2.0	2.0	2.1	2.1
TLV	**1.8**	**1.7**	**1.7**	**1.7**	**1.7**	**1.8**	**1.7**	**1.7**	**1.7**	**1.7**	**1.7**	**1.8**	**1.7**	**1.8**	**1.8**	**1.7**	**1.7**	**1.7**	**1.7**	**1.7**	1.7
NAC	1.9	**1.8**	1.9	**1.7**	**1.8**	1.9	2.0	**1.8**	2.0	1.9	1.8	**1.8**	2.0	**1.7**	1.9	1.4	**1.6**	1.9	1.4	**1.8**	1.8
NAL	1.9	1.9	1.8	2.0	1.9	**1.8**	**1.8**	1.9	**1.8**	1.9	2.0	1.9	**1.7**	2.0	1.9	1.9	2.0	1.8	1.9	1.9	1.9
LST	**1.8**	**1.8**	**1.7**	**1.7**	1.9	1.9	**1.8**	1.9	**1.8**	**1.7**	**1.7**	**1.8**	1.8	1.8	**1.7**	**1.3**	**1.6**	**1.7**	**1.3**	**1.8**	1.7
NQL	2.3	2.7	3.0	2.5	2.7	2.3	2.6	2.7	2.1	2.3	2.5	2.4	2.3	2.6	2.8	2.6	2.8	2.5	2.3	3.1	2.5
STW	2.2	**1.8**	2.2	**1.6**	**1.9**	**1.8**	1.9	**2.1**	**1.8**	2.3	**1.8**	**1.8**	**1.9**	2.0	**2.0**	2.1	2.1	**1.8**	**1.8**	2.0	1.9
TTW	**1.8**	2.3	**1.8**	1.9	2.2	2.1	**1.8**	**2.1**	2.3	**1.8**	**1.8**	**1.8**	2.0	**1.7**	2.2	**1.9**	**1.8**	**1.8**	**1.8**	**1.8**	1.9
PWZ	2.6	2.2	2.3	2.4	2.3	2.8	3.0	2.5	3.0	2.7	2.4	2.6	2.9	2.5	2.4	2.4	2.3	2.2	2.5	2.3	2.5

Figure (5.1) provides a visual representation of how results are distributed for each goal of each stakeholder. The disparity in the TOC scores with TLV as the most favorable measure (average score of 1.7) compared to TTA and TLV highlights a critical insight. It suggests that, from a hospital's perspective, efficiency is better achieved by the minimizing the time of the last service rather than the overall travel time or the total lateness. This insight can be beneficial to the TOC in deploying a different strategy than the minimization of costs knowing that the trade-off in prioritizing nurses working shifts would be low. From Figure (5.1b), the uniformity in scores across various fairness measures indicates that, from the patient's standpoint, no single fairness function significantly outperforms the others in balancing trade-offs. This could mean that patients perceive a relatively equal level of service regardless of the specific fairness measure applied. It's an important finding, suggesting a degree of flexibility in choosing fairness measures without significantly impacting patient perception. On a nurses perspective, Figure (5.1c) reports the time-related functions STW and TTW with an average score equal to 1.9 diverging from the average equal to 2.4 of the two assignment-related functions NQL and PWZ . Lower scores in STW and TTW might reflect nurses' preference or better satisfaction for measures that focus on their workload scheduling and travel aspects. In contrast, higher scores for NQL

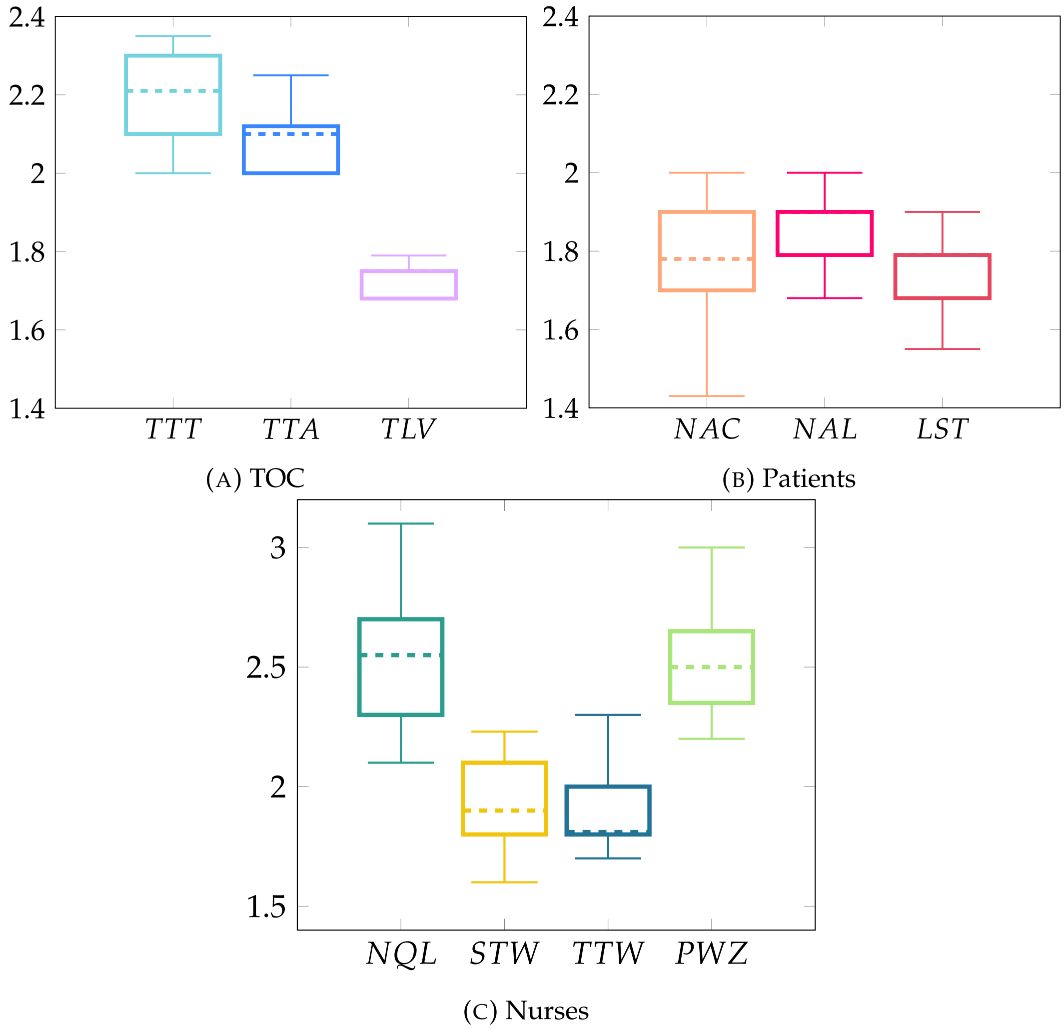
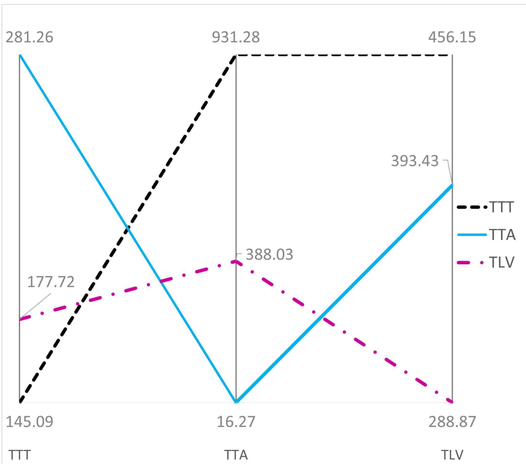


FIGURE 5.1: Distribution of the scores for each objective.

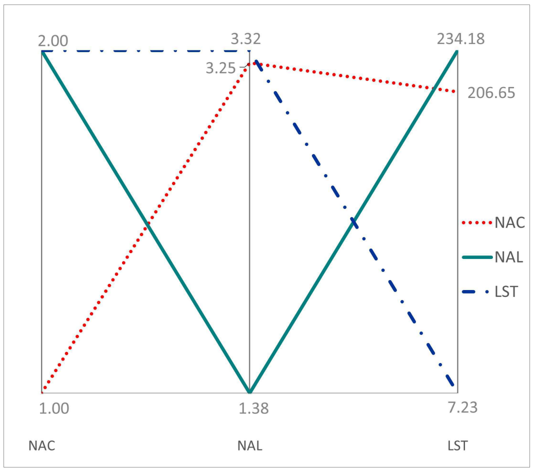
and PWZ indicate less favorable views when the workload is assessed from the perspectives of overall quality of life and patient waiting times.

Intra-Actor Results

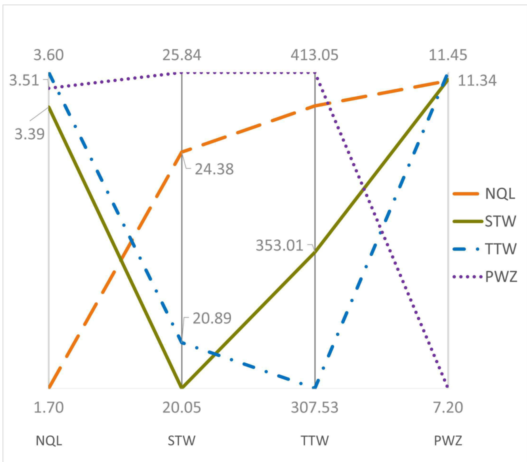
In this section, we delve into the correlation between measures by analyzing the impact of using a specific goal as the first objective function in a triplet solved through hierarchical optimization. We investigate how this affects the values of the remaining objectives, as shown in Figure (5.2). Additionally, we examine how changing the priority order of goals in the triplet impacts the solution values, as illustrated in Figure (5.5).



(A) TOC



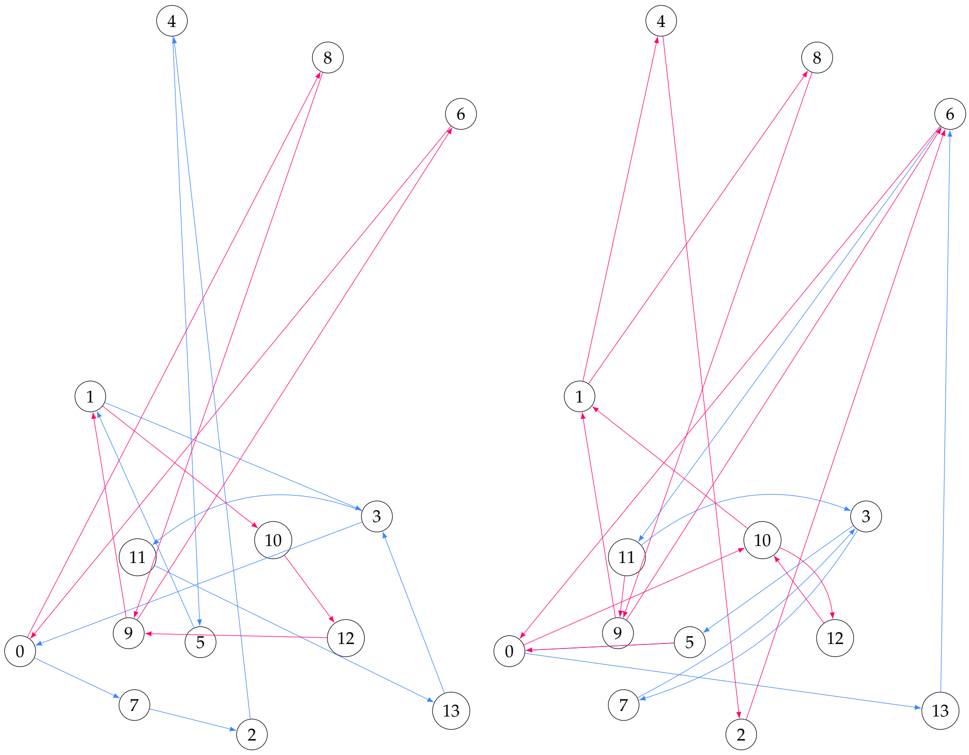
(B) Patients



(C) Nurses

FIGURE 5.2: Correlation between objective functions.

Figure (5.2) presents a comparative analysis of various objectives linked to the same stakeholder, employing a parallel coordinates graph for illustration. This graph is an effective tool for visualizing and comparing multivariate data. Each vertical axis on this graph represents a specific goal, allowing for an effortless examination of the interconnections between these goals and identifying discernible patterns or trends. In this graph, each color-coded piece-wise line is associated with a distinct objective function pertaining to a stakeholder, considered as the primary objective in a triplet of hierarchical multi-objective optimization. The number of points each line connects varies with the stakeholder: three points for some (refer to Figures (5.2a) and (5.2b)) and four for others (as shown in Figure (5.2c)). For example, in Figure (5.2a), a black dotted line represents the goal TTT. This line intersects the TTT axis at its lowest point value of 145.09, indicative of the optimized average objective function value when TTT is prioritized in all tests. Subsequently, it crosses the axes of other goals for the same stakeholder, recording average values of 931.28 for TTA and 456.15 for TLV. These values are the averages achieved across all instances for TTA and TLV when TTT is optimized and the resulting solution is used to evaluate the other two goals of the TOC. By analyzing the values displayed on a given vertical axis, we can effectively assess the performance of a stakeholder's specific objective under different optimization scenarios. This allows us to observe how well the goal performs when it's the primary focus of optimization (resulting in its minimum value) and how it fares when the optimal solution for other objectives is considered, yielding an average value. For instance, take a look at Figure (5.2a). Concentrating on the vertical axis labeled TLV, we notice that its optimal value stands at 288.87 when prioritized in the optimization process. However, when we calculate TLV's average performance using the optimal solution for TTT, its performance worsens significantly, reaching a maximum value of 456.15. Conversely, when TTA takes precedence in optimization, TLV's average value improves to 393.43. These visual representations offer valuable insights into the relationships between different objectives and how their values fluctuate when the top-priority objective changes. This deeper understanding helps decision-makers comprehend the interplay among various measures. For example, Figure (5.2c) reveals a clear connection between the behaviors of functions STW and TTW. Optimizing one of these functions leads to the other achieving its second-best value. Interestingly, both functions perform poorly when the optimization focus shifts to NQL and PWZ. Although certain functions may show positive correlations, Figures (5.3) and (5.4) emphasize significant disparities in the optimal routes between these solutions. These figures provide a detailed view of the routes generated for Instance 13 within Set 1, which involves 20 nodes, 13 patients, and 2 vehicles. In Figure (5.3), we can observe the solutions derived from optimizing NQL and NAL, while Figure (5.4) displays the results of optimizing TTW and STW. It's important to note that, for clarity in visualization, the nodes in these figures represent patients rather than individual services. Consequently, you may notice subtours in the graphs where two services for the same patient are not executed consecutively, leading to multiple visits to the same patient. These subtours are a result of the representation choice and provide insights



(A) NQL (B) NAL

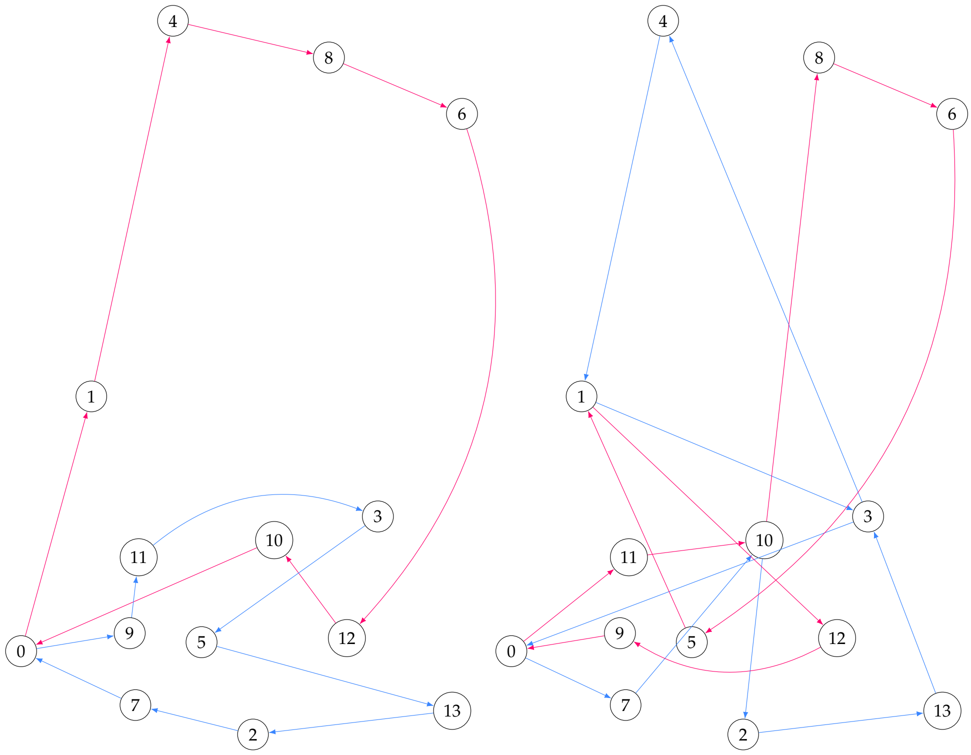
FIGURE 5.3: Optimal routes for functions NQL and NAL for Instance 13 of Set 1.

into the complexities of optimizing routes for healthcare services.

Figure (5.3) compares optimal solutions when optimizing functions related to nurses' skills. Despite the strong correlation between these measures, there are notable discrepancies in nurse-patient assignments. For instance, in Figure (5.3a), Patient 6 is attended to by just one nurse, whereas in Figure (5.3b), the patient's services are distributed among two nurses. This demonstrates that the specific assignments and routes can vary significantly even with similar optimization goals.

Similarly, Figure (5.4) illustrates that, while function STW is encompassed within function TTW, the routes obtained by optimizing these objective functions differ notably. This highlights that the inclusion of one function within another does not necessarily result in identical or even highly similar routes, emphasizing the intricacies of route optimization in healthcare contexts.

In Figure (5.5), the box plots visually represent the percentage improvement in various measures when their priority changes during the lexicographic optimization process. These improvements are quantified as "gaps," denoted as $Gap(i \to j)$, indicating the percentage of value enhancement that occurs when a function is shifted from priority level i to priority level j with $i > j$. Specifically, we focus on three graphs: $Gap(3 \to 2)$ and $Gap(3 \to 1)$ measure the enhancement when moving from the third priority level to the second and first priority levels, respectively and $Gap(2 \to 1)$ represents the



(A) TTW (B) STW

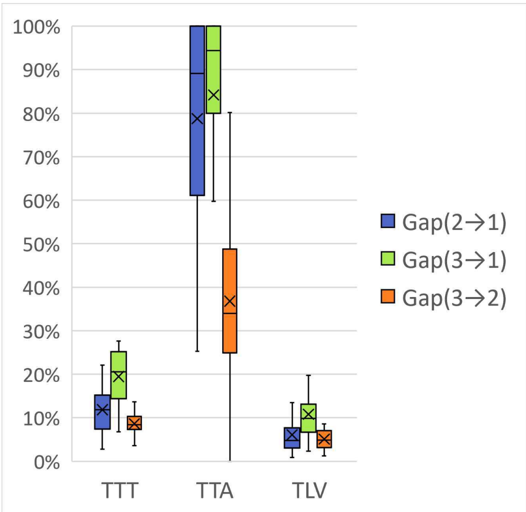
FIGURE 5.4: Optimal routes for functions TTW and STW for Instance 13 of Set 1.

improvement when transitioning from the second to the first priority level. The analysis helps us conclude that the improvement of four measures (two related to the TOC and two to nurses) remains consistently below 30%, while for half of the functions the highest improvement falls short of 40%. For instance, take the measure NAC, which exhibits $Gap(2 \rightarrow 1)$ values ranging from 0% to 17.5%, with an average improvement of 10%. The whisker associated with the maximum value reaches 29%.

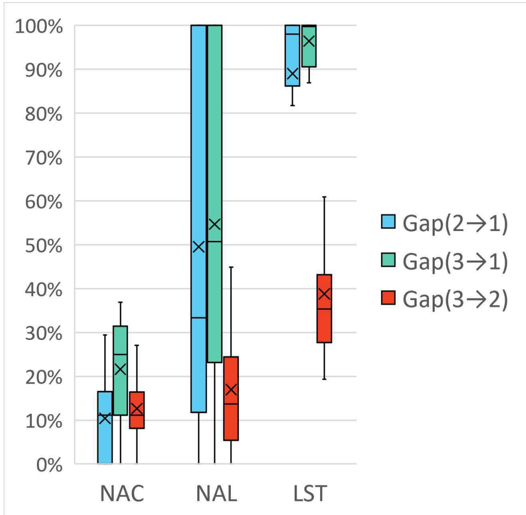
However, for measures like TTA, NAL, LST, and NQL, the percentage improvement can reach as high as 100%. Among these, LST experiences the most significant improvement, with an average enhancement of 98% when transitioning from the third to the first priority level. These findings shed light on how different measures respond to changes in priority levels during optimization.

Inter-Actor Results

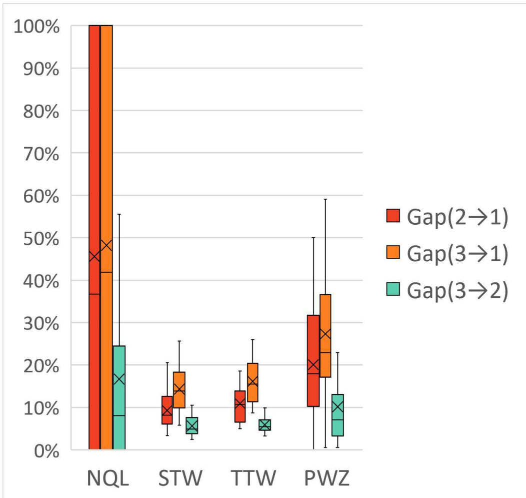
In this section, we delve into the intricate relationships between stakeholders and their objective functions. Specifically, we explore how prioritizing an objective function associated with one stakeholder can impact the optimization of objectives related to non-prioritized stakeholders. Figure (5.6)



(A) TOC



(B) Patients



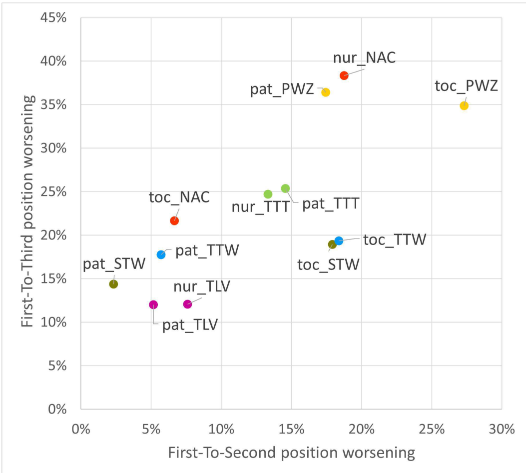
(C) Nurses

FIGURE 5.5: Improvement of the value of the objective functions when changing priority.

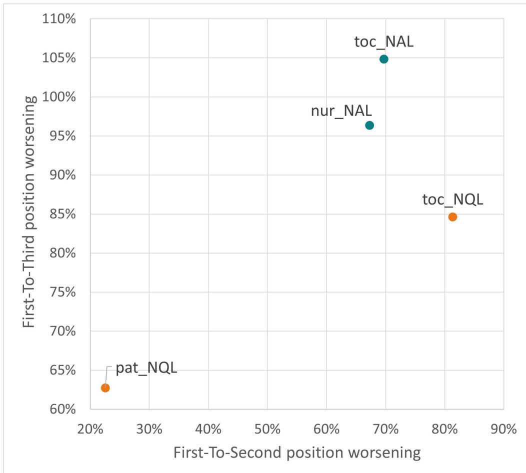
visually represents the percentage deterioration observed in various measures based on the stakeholder with the highest priority. For each measure, two distinct points are plotted to evaluate its deterioration under different scenarios: when each of the other two stakeholders is assumed to have the highest priority level.

Here's how to interpret the graph: the x-axis represents the percentage worsening of a measure when it is optimized as the secondary objective, compared to its value when it is the primary objective. The y-axis displays the percentage decrease in performance when the measure is used as the tertiary (third) objective in the hierarchical order. For example, in Subfigure (5.6a), the point pat_{STW} indicates that when patients' objectives are prioritized as the first to be optimized, measure STW experiences an average decline of 3% when subsequently optimized as the second objective. This decline increases to 14% when STW becomes the third objective, suggesting a high sensitivity to its prioritization. Conversely, functions that are positioned near the bisector line at the center of the figure, like toc_{TTW}, exhibit no significant difference in their percentage deterioration when moved from the second to the last priority level. This implies that their performance remains relatively consistent regardless of their priority. This visual representation serves a dual purpose. Firstly, it clearly illustrates how one stakeholder's objectives can impact the deterioration of another stakeholder's objectives. For example, comparing pat_{PWZ} with toc_{PWZ} allows us to observe that the deterioration of PWZ when shifted from the first to the third priority level is independent of the first-priority stakeholder chosen. However, moving it from the first to the second priority level performs better when patients' objectives are prioritized over TOC's. Secondly, it helps understand the difference in behavior between a shift to the secondary priority level and a shift to the tertiary priority level, providing valuable insights into the dynamics of prioritization in this context.

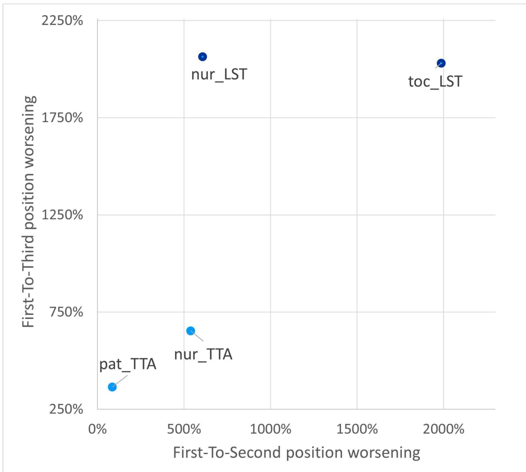
Certainly, the analysis reveals interesting patterns in how the optimization of nurses' measures (STW, TTW, NQL, and PWZ) as the second objective is influenced by the prioritization of the TOC as the main objective. For instance, when nurses' measures are secondary objectives, they exhibit significantly higher declines in performance when the TOC is the primary focus. Take toc_{NQL}, for example, which experiences an average decline of 82% when the TOC is prioritized, compared to a 23% decline when patients' objectives take precedence. However, this sensitivity to the primary actor diminishes when the measure is in the third position, with a difference of only 22% between pat_{NQL} and toc_{NQL}. The plot is divided into three subfigures based on the range of values to enhance the visualization of these results. Subfigure ((5.6a)) includes values ranging from 0% to 45% for the y-axis. Subfigure ((5.6b)) covers values between 60% and 110%, and Subfigure ((5.6c)) represents the highest section of the plane, with a maximum value of 2250%. Consistent with the previous analysis of percentage variations in Figure (5.5), TTA and LST exhibit the widest ranges of value decline. For example, the coordinates for toc_{LST} show a decline of 1986% and 2031%. However, it's crucial to note that your ranking system identifies LST as the best measure for patients. Therefore, despite the significant decline in LST values, the computed values for the other functions remain, on average, better than those obtained when optimizing other patients' goals. In other words, even with the substantial decrease in LST values, the overall solutions still outperform alternative options when it comes to meeting patients' objectives, as determined by your ranking system.



(A) First Range



(B) Second Range



(C) Third Range

FIGURE 5.6: Worsening of objective functions with varying first optimized actor.

5.6.3 Sensitivity Analysis on the Number of Nurses and Services

In this section, we delve into the results obtained from solving the large-size instances within Set 3, consisting of 120 instances. These instances were solved for all possible combinations of the triplet TLV, TTW, LST using the `ParallelALNS` metaheuristic. For each instance and ordered triplet of objective functions, the algorithm was executed three times, with a time limit of 600 seconds per run. In total, we obtained 2160 solutions for the 120 instances, considering six different ordered triplets and three runs per instance. We selected the best run for each instance and each ordered triplet to facilitate the analysis. This approach allows us to focus on the most promising and optimal solutions among the multiple runs conducted for each instance and triplet combination.

TABLE 5.4: Results for instances in Set 3.

			$\|K\| = 5$									$\|K\| = 7$								
$f1$	$f2$	$f3$	$n_{max} = 1$			$n_{max} = 3$			$n_{max} = 5$			$n_{max} = 1$			$n_{max} = 3$			$n_{max} = 5$		
LST	TTW	TLV	**11.32**	**185.25**	**175.11**	162%	93%	98%	715%	147%	153%	0%	-25%	-27%	87%	29%	30%	258%	99%	104%
TLV	TTW	LST	**156.25**	**191.39**	**82.48**	72%	59%	402%	133%	108%	857%	-27%	-23%	-39%	25%	20%	156%	78%	63%	599%
TTW	TLV	LST	**178.51**	**168.32**	**65.91**	62%	67%	435%	114%	123%	1038%	-23%	-25%	-34%	21%	23%	198%	66%	72%	652%
TLV	LST	TTW	**156.25**	**77.34**	**191.49**	72%	426%	59%	133%	919%	108%	-27%	-40%	-23%	25%	156%	20%	78%	646%	63%
LST	TLV	TTW	**11.32**	**166.93**	**194.18**	162%	105%	87%	715%	165%	136%	0%	-30%	-23%	87%	33%	27%	258%	112%	93%
TTW	LST	TLV	**178.51**	**43.89**	**170.70**	62%	542%	65%	114%	1344%	120%	-23%	-44%	-25%	21%	229%	21%	66%	998%	70%

Table (5.4) provides a detailed summary of the results, specifically focusing on ordered triplets of objective functions, categorized by the number of nurses ($|K| = 5$ and $|K| = 7$) and the maximum number of services requested ($n_{max} = 1, 3, 5$). Analyzing the percentage variations, we observe a consistent trend where increasing the number of services leads to higher objective function values, as expected. Regarding LST, it is evident that the objective function is highly sensitive to parameter n_{max} changes. When LST is the first objective, its value increases significantly, by 162% and 715% for $n_{max} = 3$ and 5, respectively. This impact of n_{max} becomes more pronounced when LST occupies the second and third positions. Notably, the difference in the average increase between the second and third positions is relatively small, averaging 484% vs. 418% for $n_{max} = 5$. This highlights the sensitivity of LST to variations in n_{max}. On the other hand, the other two objective functions, TLV and TTW, also show noticeable changes but to a lesser extent. For instance, when TLV is in the first position and n_{max} is increased from 1 to 3, its value increases by 72%. Similarly, when n_{max} is increased from 1 to 5, TLV exhibits a 133% increase. Similar behavior is observed when TLV is in the second and third positions. The behavior of TTW closely resembles that of TLV. In summary, the results in Table (5.4) confirm the expected impact of changing the number of services on objective function values and highlight the sensitivity of LST to variations in n_{max}, while also showing notable but less pronounced effects on TLV and TTW. It's evident from the analysis

TABLE 5.5: Tuning of the `ParallelALNS` parameters.

Parameter	Meaning	Value
$qStep$	Incremental quantity of the degree of destruction q	1
q_{max}	Maximum value of q	$\lfloor 0.9\|N\| \rfloor$
$noImpIter$	Number of iterations without improvement before increasing q	100
$epochIter$	Number of iterations per epoch	10000
$imprEpoch$	Number of epochs without improv. before solving the restricted problem	12,6
τ_{init}	Time limit for the initial solution heuristic	3s
χ	Number of top solutions kept by the ALNS manager	10000
$\bar{A}$	Number of arcs in the restricted network	$\lfloor 0.1\|A\| \rfloor$
τ	Time limit for `ParallelALNS`	120s,600s
$\gamma_1, \gamma_2, \gamma_3$	Weight values in weighted operators	0.9, 0.07, 0.03
T_0	Initial temperature	10
α	Cooling rate	0.9
T_{iter}	Cooling schedule	100
ALNS Workers	Number of ALNS workers running in parallel	30

that when the number of services is kept constant, an increase in the number of nurses generally leads to improved overall performance. However, once again, we observe that LST exhibits the highest volatility among the three measures. For example, when LST is in the third position, and $n_{max} = 5$, the average increase in its value compared to the baseline results decreases from 947.5% to 625.5% when the number of nurses changes from 5 to 7. This sensitivity of LST to changes in the number of nurses is consistent across different positions and values of n_{max}. An interesting exception arises when LST is in the first position, $n_{max} = 1$, and $|K| = 7$. In this particular case, the value of LST remains unchanged compared to the baseline results. This indicates that 5 nurses were sufficient to provide the services when $n_{max} = 1$, and adding more nurses did not lead to any further improvements in this specific scenario. These findings highlight the complex interplay between the number of nurses, the maximum number of services, and the position of the objective function in the optimization process, with LST showing particularly notable sensitivity to these factors.

5.6.4 Evaluating the `ParallelALNS` Framework

In this section, we present the results obtained by our `ParallelALNS` metaheuristic. We conducted preliminary tests using a limited set of instances with varying numbers of service requests and nurses. Based on these tests, we fixed nearly all the ALNS parameters, except for $imprEpoch$, τ, q_{max}, and $\bar{A}$, which depend on the size of the solved instance. All parameter values are listed in Table (5.5). Regarding $imprEpoch$ and τ, we utilized $imprEpoch = 12$ and a time limit of 120 seconds for Set 1 instances, and $imprEpoch = 6$ and $\tau = 600$ for Set 2 instances.

Table (5.6) provides a comprehensive comparison of the results achieved by the `ParallelALNS` metaheuristic on Set 1 instances in comparison to the optimal solutions obtained by Gurobi. The algorithm was run three times with a time limit of 120 seconds for each run.

Each row in the table represents the performance of the algorithm for a specific hierarchical order of the three objective functions. The first three columns indicate the objectives in hierarchical order, and the last two columns display the number of times (out of 20 instances) the best (B) and the worst (W) solution found by `ParallelALNS` matches the optimal solution discovered by Gurobi. The results demonstrate that `ParallelALNS` is generally successful in finding the optimal solution, except when the objective function LST is prioritized as the first one to be optimized. In all other cases, even the worst run of `ParallelALNS` is capable of matching the optimal solution. For example, when considering the best solution, in the triplet $[LST, TTW, TLV]$, `ParallelALNS` obtains 18 optimal solutions out of 20, and for the worst solution, it achieves 16 out of 20. Overall, the performance of `ParallelALNS` on these small instances is relatively stable, with little difference between the best and worst runs. In the cases where `ParallelALNS` cannot find the optimal solution, it's worth noting that the value of the first objective function is consistently equal to the optimal value. In one case, the value of the second objective deviates slightly from the optimal value, with a negligible gap of 0.25%. In the remaining two cases, the average gap between the value obtained for the third objective and the optimal one is 1.24%. These gaps are expressed as percentages and represent the difference between the solution value found by Gurobi for the considered objective function and the solution value obtained by `ParallelALNS`, divided by Gurobi's value.

Considering the worst solution instead, four times the non-optimal objective is the second one (average gap 0.46%) and twice the third one (0.72%).

TABLE 5.6: Comparison between `ParallelALNS` and Gurobi on Set 1 instances.

Objectives			#Opt	
f_1	f_2	f_3	B	W
TLV	LST	TTW	20	20
TLV	TTW	LST	20	20
LST	TTW	TLV	18	16
LST	TLV	TTW	19	18
TTW	TLV	LST	20	20
TTW	LST	TLV	20	20

Table (5.7) presents a comprehensive comparison between `ParallelALNS` (with three runs) and Gurobi (with time limits of 600s and 3600s) on the 20 instances from Set 2. The table's structure is similar to the previous one.

Let's analyze the key observations from this comparison:

- `ParallelALNS` outperforms Gurobi. It's evident that `ParallelALNS` consistently achieves better solutions than Gurobi. In fact, for all hierarchical orders of the three objectives, even the worst solution obtained by `ParallelALNS` surpasses the one found by Gurobi.

- Gurobi faces challenges in finding feasible solutions for certain hierarchical orders, particularly when LST is the first optimized function. In these cases, Gurobi often reaches the time limit without finding the optimal solution but provides a feasible one.

- When LST is the first objective function and Gurobi does find a solution, `ParallelALNS` achieves a significant improvement, with an average gap reduction of 66.5% compared to Gurobi's solutions.

- The difference between the best and worst solutions obtained by `ParallelALNS` for each hierarchical order is limited, indicating the method's consistency in generating competitive solutions.

- When LST is the second objective function, there is a noticeable tradeoff between the value of the first and second objectives. On average, the solutions identified by `ParallelALNS` are 9.35% better than those found by Gurobi for the first objective, but they are 74.7% worse when considering LST (in the second position). This highlights the challenge of optimizing LST when it is not the primary objective.

Overall, the results showcase the effectiveness of `ParallelALNS` in providing competitive solutions across different hierarchical orders, while also highlighting the difficulties Gurobi faces in finding optimal solutions for specific cases.

TABLE 5.7: Comparison between `ParallelALNS` and Gurobi on Set 2 instances.

Objectives			#Improv.			Best			Worst		
f_1	f_2	f_3	B	W	*#Feas.*	Gap f_1	Gap f_2	Gap f_3	Gap f_1	Gap f_2	Gap f_3
TLV	LST	TTW	20	20	15	-13.8	93.9	-7.1	-13.4	91.3	-6.8
TLV	TTW	LST	20	20	15	-8.5	-0.4	116.0	-8.1	-0.1	100.1
LST	TTW	TLV	20	20	13	-66.5	0.9	1.8	-65.0	3.0	3.8
LST	TLV	TTW	20	20	13	-66.5	3.8	6.8	-65.0	6.4	7.9
TTW	TLV	LST	20	20	20	-4.8	-2.9	33.0	-4.5	-2.5	31.0
TTW	LST	TLV	20	20	20	-4.9	55.5	-4.1	-4.5	61.6	-3.8

Table (5.8) provides insights into the effectiveness of the MIPIMPROVEMENT procedure, which is designed to solve the mathematical model of the problem on a restricted network. The table reveals significant variation in the number of times the MIPIMPROVEMENT procedure improves the best solution identified for any of the hierarchical orders across different instances. Some instances, like Instance 3, show a substantial impact of the procedure, with an average of 18.3 improvements. However, there are instances, such as Instance 18, where the procedure does not lead to any improvements. This

TABLE 5.8: Contribution of function MIPIMPROVEMENT when solving Set 2 instances.

Instance	#Impr	Instance	#Impr
1	13.0	11	2.3
2	5.0	12	15.0
3	18.3	13	3.3
4	2.3	14	17.7
5	3.0	15	4.3
6	4.0	16	6.3
7	13.7	17	9.7
8	7.0	18	0.0
9	6.0	19	2.7
10	12.0	20	12.7

variability suggests that the effectiveness of the MIPIMPROVEMENT procedure depends on the specific characteristics of the instances and the initial solutions obtained. Additionally, to address the challenges in accurately evaluating the quality of solutions for certain triplet combinations in Table (5.7) we introduced an algorithm called `MathALNS`, which consists of two phases:

- Phase 1: A modified version of `ParallelALNS` is used to concentrate solely on one hierarchical order, resulting in six times more parallel ALNSs focusing on specific orders. This phase takes the best incumbent integer solution found by `ParallelALNS` in the original three runs as input and produces a potentially improved solution.
- Phase 2: Gurobi receives the solution obtained in Phase 1 as a mild start and runs for 3600 seconds to further enhance it. Based on a non-parallel version of ALNS, our primal heuristic is embedded to strengthen Gurobi's branch-and-cut algorithm. Whenever Gurobi finds a new improved solution, our primal heuristic is executed for 20 seconds to refine it further.

Table (5.9) provides a summary of the improvements achieved by `MathALNS` over `ParallelALNS` on Set 2 instances for the two selected triplet orders. The table showcases the percentage improvement obtained at the end of both phases concerning the best solution obtained by `ParallelALNS` over three runs. It can be observed that, the two triplet orders, TLV-TTW-LST and LST-TLV-TTW, exhibit significantly different solution improvements. For the triplet TLV-TTW-LST both Phase 1 and Phase 2 fail to improve the solutions obtained by `ParallelALNS` by more than half a percentage point for all objectives. This suggests that the solutions achieved by `ParallelALNS` were likely already of very high quality. In contrast, for the LST-TLV-TTW triplet order, Phase 1 demonstrates substantial improvements, particularly in reducing the LST value by almost 5%. Phase 2 further refines the average LST value by a 5.3% reduction, with noticeable improvements for the second and third objectives, TLV and TTW. notably, for the LST-TLV-TTW triplet order, MathALNS determines the optimal objective function value for LST in eight out of 20 cases, matching the one found by ParallelALNS. It's important to consider that these improvements come at the cost of significantly increased execution time. MathALNS utilizes the best solution out of three runs of ParallelALNS (30 minutes in total), then runs a modified ParallelALNS (10 minutes), and finally runs Gurobi (60 minutes), resulting in a total execution time of 100 minutes, which is ten times longer than ParallelALNS. Overall, MathALNS demonstrates its capability to further enhance solutions obtained by ParallelALNS, especially for specific triplet orders where significant improvements are achievable. However, this improvement comes at the expense of increased computational time, highlighting the trade-off between solution quality and execution time.

TABLE 5.9: Improvement of MathALNS over ParallelALNS.

Objectives			Phase 1			Phase 2		
f_1	f_2	f_3	Gap f_1	Gap f_2	Gap f_3	Gap f_1	Gap f_2	Gap f_3
TLV	TTW	LST	−0.3	−0.1	0.7	−0.4	−0.4	−0.4
LST	TLV	TTW	−4.7	−1.4	−1.0	−5.3	−5.2	−6.4

To enhance our assessment of the ParallelALNS method, we've developed a new technique named ParetoALNS. This method is designed to closely approximate the Pareto-efficient frontier, which represents the set of best possible solutions in terms of multiple objectives. It achieves this by keeping track of all non-dominated points – points where no other solution is better in all objectives – that it finds during its operation. By comparing the results from ParallelALNS to this frontier, we can better understand how effective ParallelALNS really is. ParetoALNS is tasked with finding the best possible solutions across all six possible hierarchical orders of a given triplet of objectives. To do this efficiently, we assign one ALNS worker to each hierarchical order, while the rest of the workers focus on solving the problem using objective functions weighted differently. These weights are randomly selected from a range between 0 and 1, ensuring a diverse set of solutions. Our findings, illustrated in Figures (5.7) and (5.8), showcase the Pareto frontier and all the solutions generated by ParetoALNS during multiple runs on a specific instance. For instance, when run 1200 times under the same conditions as ParallelALNS, ParetoALNS produced around 90,000 points. Out of these, 137 were identified as constituting the approximated Pareto frontier, as shown in Figure (5.7). In these figures, we've highlighted in red the six points that represent the best solutions found by ParallelALNS for the six orders of a specific triplet. Interestingly, only four of these points are visible as two of them overlap with points 1 and 4. It's notable that out of the six ParallelALNS solutions, half were not dominated by any other solution throughout the 1200 runs. The other half were only marginally dominated, with minimal improvements noted in the first objective (0.09%) and slightly higher in the second and third objectives (0.14%). However, it's important to mention the computational cost involved in creating a robust approximation of the Pareto frontier for each instance. In this case, `ParetoALNS` managed to generate a large number of points through extensive runs, but only a small fraction contributed to the frontier approximation. The entire process required a substantial amount of time – around 200 hours in total.

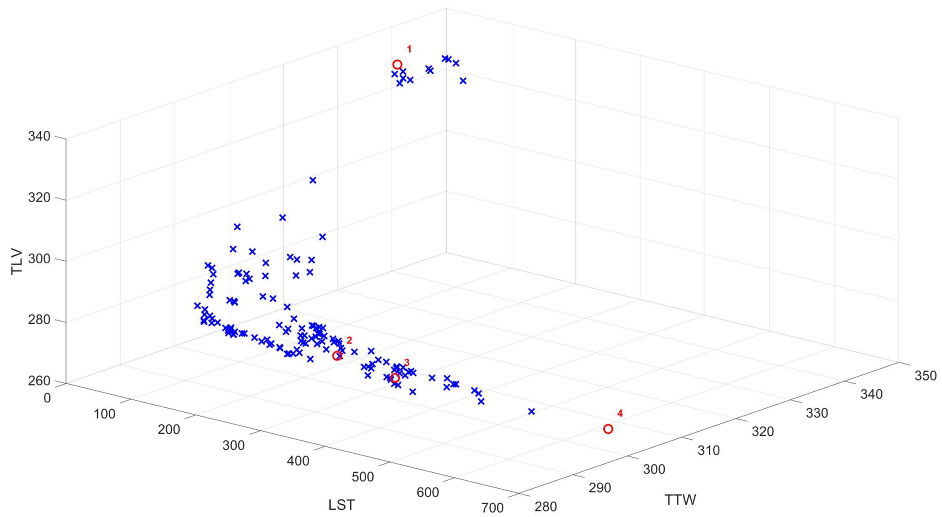


FIGURE 5.7: Approximation of the Pareto frontier: Instance 1 of Set 2.

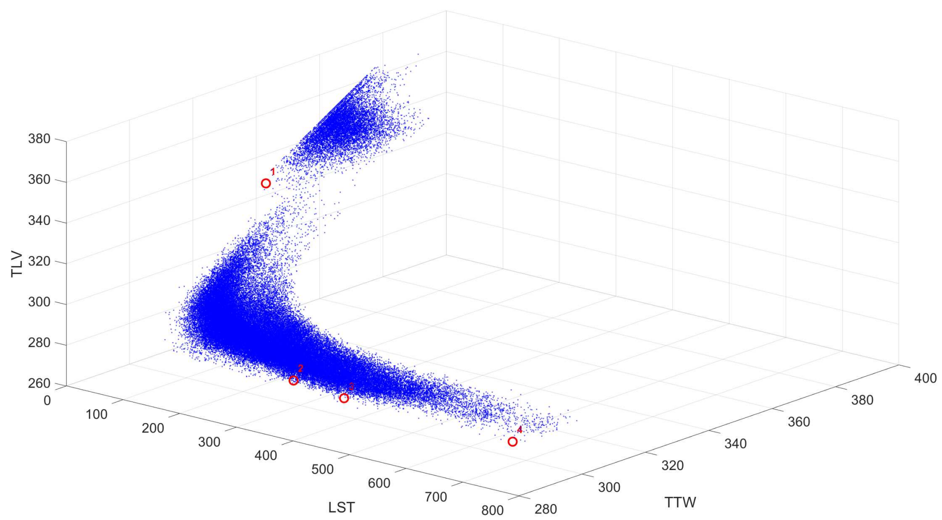


FIGURE 5.8: 90000 solutions generated by `ParetoALNS`: Instance 1 of Set 2.

5.7 Conclusions

Addressing multi-objective, multi-actor problems presents significant challenges, especially in fields requiring intricate balancing of diverse preferences and goals, such as in nurse routing and scheduling. In this study, we have explored an innovative approach to such problems, particularly emphasizing the varying priorities of nurses, patients, and healthcare service providers. Our method stands out by not requiring predetermined objective orderings from different stakeholders. Instead, each stakeholder proposes a range of goals, and our approach independently identifies the most effective goal for each actor, irrespective of its hierarchical position or its interplay with others' objectives. The result is a singular, streamlined problem formulation that focuses solely on the optimal goal for each stakeholder. We have developed a unique adaptation of the Adaptive Large Neighborhood Search (ALNS) algorithm, tailored to address the complexities of multiple objective functions. This version of ALNS innovatively integrates destroy and repair operators alongside acceptance criteria tailored to multi-objective scenarios. To enhance computational efficiency for large-scale instances, we employed a parallel processing strategy. In this setup, multiple ALNS workers, each dedicated to a specific sequence of goals, operate simultaneously. A manager-worker framework orchestrates this parallelism, where the manager acts as a central hub, circulating essential data among the workers. This data includes the most successful solution identified for each goal triplet and a dynamically updated list of promising network arcs. Each worker utilizes this shared knowledge in a specialized matheuristic phase, formulating and resolving the problem on a restricted network using a Mixed Integer Programming (MIP) solver. Our computational experiments offer valuable insights into various equity measures pertinent to nurses and patients. These findings elucidate the relationship between these measures and the objectives set by the healthcare organization responsible for home care services. The efficiency and effectiveness of our developed method are evident in both small and large-scale instances, with its performance being notably superior when compared to leading MIP solvers. We introduced two additional variants of our method for a more thorough evaluation. One variant collaborates with a MIP solver to refine the best solutions identified, while the other focuses on approximating the Pareto-efficient frontier. Both these enhancements further demonstrate the robustness and adaptability of our approach in handling complex, multi-stakeholder optimization problems in healthcare logistics. The success of these methods not only marks a significant advancement in solving such intricate problems but also opens the door for future research and applications in similar multi-objective, multi-actor scenarios.

Chapter 6

A Dynamic Multi-Period Home Healthcare with Consistency

Contents

This chapter corresponds to the paper "V.Bonomi, R.Mansini, R.Zanotti, Dynamically dealing with requests in a Stochastic Multi-Period Home Healthcare Problem with Consistency Constraints", to be submitted

6.1 Introduction

In this section, we examine the Multi-Period Dynamic Vehicle Routing Problem within the context of the healthcare industry, focusing specifically on home care services. This challenge involves dealing with unpredictable requests from patients who require visits from nurses at their homes, where the exact time and location of the needed service are not known in advance. The agency responsible for the nurses must decide daily whether to accept new patients, thus generating revenue, or to outsource these requests to other providers. To improve patients' satisfaction, visits must be scheduled respecting the *consistency* constraint, meaning that it is necessary for a consistent nurse-to-patient assignment in all patient visits. The goal is to plan assignments efficiently, ensuring all accepted patient requests are met while maximizing profits by balancing the revenue against travel costs. We propose several solution approaches to tackle this problem, relying on the dynamic nature of the problem. At first, we consider patient selection and nurse routing day by day in a myopic approach, and then we integrate historical data through scenario generation to obtain better predictions of future demands. All our proposed solutions employ an Adaptive Large Neighborhood Search algorithm to manage the routing challenges, tested on various scenarios to ensure their effectiveness. In Section (6.2) we detailed the problem description and present the mathematical formulation in Section (6.3). Section (6.4) details the main heuristic applied already introduced in Chapter (3). The main computational results are presented in Section (6.5). At first a sensitivity analysis on consistency is made on small size instances.Then, the heuristics performances are studied in more detail.

6.2 Problem Description

We consider the operational model of a private nurse agency tasked with the planning of visits to patients in different geographical areas. When patients need care, they contact the agency and provide details on the number of visits required and the specific days they need service within a given time frame. Requests must be submitted at least one day before the first visit. If a request is submitted more than one day before the first service, the decision to accept or deny the patient will be deferred until the day before the requested first visit. This approach allows the agency to exploit, at maximum, all the new information that may arise between the patient's call and the start of service. If no nurses' routes can accommodate all the patient's visits, the agency declines the patient, referring it to an alternate local provider. The agency does not know in advance how many patients will require service during the time horizon, their locations, the number of associated interventions, and their requests' temporal profile. Every day, the agency plans the routes for its nurses based on the patients already known. After a patient is accepted, it is assigned to a nurse who will be responsible for all the scheduled visits. This commitment to nurse-patient consistency is motivated by the fact that a stable caregiver relationship is crucial to improving the level of care.

Given that patient requests are received continuously, the problem falls under the category of a multi-period stochastic problem with the added complexity of maintaining consistent nurse assignments in a dynamic setting.

The problem is defined over a discrete timeline represented by the set $D = \{1, ..., d_{max}\}$ where the time horizon spans d_{max} days. We indicate as $K =$

$\{1, ..., m\}$ the set of nurses available on each day of the time horizon, and as $P = \{1, \ldots, \bar{p}\}$ the set of patients requiring services over the d_{max} days. The maximum working shift of the nurses is defined as t_{max}. We refer to $D_p \subseteq D$ as the set of days on which patient p requires the service. For each day $d \in D$, we define as $P_d \subseteq P$ the patients that require a visit on day d. Every patient p contributes an amount denoted as r_p (representing the agency's revenue) in exchange for the completion of the service. During the period D, on any given day d, patients are classified in two distinct categories:

- Booked patients: patients who have already contacted the agency specifying their locations and service requests. We denote this set as $P^B \subseteq P$. Among these patients, we can further divide them into those who have been accepted (set P^A), those who have been rejected (set P^R), and those still awaiting a decision (set P^W), forming the partition $P^B = P^A \cup P^R \cup P^W$.

- Potential patients: This group includes individuals whose service requests and locations have not yet been determined. For these patients, the agency must use information from past service requests to anticipate and plan for future needs.

At the end of each day d, decisions about the assignments and the routing of day $d + 1$ are taken. In particular, patients in P^W with the first service in $d + 1$ are either assigned to a nurse or rejected. Once all the assignments are defined, each nurse route is built with the patients in P^A requiring a visit in $d + 1$.

The hospital's costs are calculated as the difference between the revenues of accepted patients and the total travel expenses. Travel costs are proportional to the traveled distances that are Euclidean.

The objective of the problem is to identify which patients to serve to maximize the difference between the revenues and the travel expenses. This has to be achieved while ensuring to comply with consistency constraints. The problem is the Stochastic Multi-Period Home Healthcare Problem with Consistency Constraints (SMHHP-C).

We call the particular version of the problem where it is assumed to know all the patients requiring a visit to the *Offline Problem*. In this case, there is no uncertainty about future patients, and all the needed visits are known.

To better understand the requests management we give here a numerical example over a time horizon of $|D| = 6$ days with $|P| = 4$ patients.

In Table (6.1), for each patient, are shown the day in which the request became known (Release Date) and the days in which the patient requires visits (Service Dates), while a visual representation is given in Figure (6.1).

In Figure (6.1), each patient is represented by a circle on its release date. The arrows exiting the circles indicate in which days the patient needs a visit.

TABLE 6.1: Requests Example

Patient ID	Rel. Date	Serv. Dates
1	0	$\{1,2\}$
2	1	$\{3,4\}$
3	3	$\{4,5\}$
4	4	5

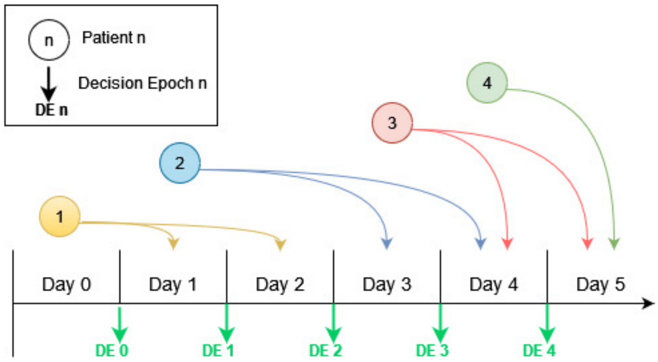


FIGURE 6.1: Visual representation of patients requests

The time horizon is divided in days and the decision epoch is set at the end of the day. Let us assume to be on day $d = 1$. In a real-life situation the subset of booked patients P^B would include Patient 1 and Patient 2; the patients with a release day less or equal than $d = 1$. Patient 3 and Patient 4 are still unknown to the hospital. To this day, decisions about Patient 1 have already been taken on $DE = 0$ while Patient 2, although already known, is will not be planned until $DE = 2$, the day before its first visit that will take place on day 3. Hence, on $DE = 1$ no decision has to be made, and the routes for the subsequent days are not updated. On the other hand, if we consider the Offline Problem, all the patients are already known at $DE = 0$. How different solution methods handle information is presented in Section (6.4).

To provide a compact formulation of the problem, we create a set of $N_s = \{|P|+1,\ldots,|P|+|K|\}$ of $|K|$ dummy nodes representing the starting points for each one of the m vehicles. The same process is applied to the ending nodes, with the set $N_e = \{|P|+|K|+1,\ldots,|P|+2|K|\}$ of $|K|$ dummy nodes. Consequently, for each vehicle $k \in K$ the starting and ending depots are represented by nodes $(\bar{p}+k)$ and $\bar{p}+m+k$, respectively. The duplication of depots avoid us to implement the k index on the arcs variable and was initially presented by Luo et al., 2015. The Offline Problem can be defined over a complete and directed graph $G = (V, A)$ where $V = P \cup N_s \cup N_e$ is the set of nodes including all the patients and depots over the time horizon, and A is the arc set. For each set $S \subset V$, let $\delta^+(S) = \{(i,j) \in A : i \in S, j \notin S\}$ and $\delta^-(S) = \{(i,j) \in A : i \notin S, j \in S\}$ be the set of arcs leaving from and entering set S, respectively. For each $d \in D$, we can define the sub-graph

$G_d = (V_d, A_d)$ where $V_d = P_d \cup N_s \cup N_e$ and $A_d \subseteq A$ are the arcs connecting the nodes of day d. We indicate as c_{ij} and t_{ij} the traveling time associated with arc $(i, j) \in A$ and as s_i the service time at node $i \in P$.

6.3 Mathematical Formulation

In this section, we provide the mathematical formulation of the problem when all information is assumed to be known (Offline problem) and we present a different formulation to include various patients management policies.

To this aim, we define the following sets of variables. For each day $d \in D$, and each arc $(i, j) \in A_d$ connecting node $i \in V_d$ and node $j \in V_d$, we define the binary variable x_{ij}^d as follows:

- $x_{ij}^d = \begin{cases} 1, & \text{if arc } (i,j) \text{ is traversed on day } d \\ 0, & \text{otherwise.} \end{cases}$

The second set of binary variables defines nurse/patient assignments. For each patient $p \in P$ and for each nurse $k \in K$ we introduce the variable y_{pk} as follows:

- $y_{pk} = \begin{cases} 1, & \text{if patient } p \text{ is assigned to nurse } k \\ 0, & \text{otherwise.} \end{cases}$

It's essential to observe that the variable y_{pk} remains independent of $d \in D$, as its value remains constant throughout the time horizon due to the consistency constraint. Once a patient is allocated to a nurse, that patient will consistently receive care from the same nurse across all days $d \in D$.

The last set consists of continuous variables $z_{ij}^d \in A_d, d \in D$ that define the arrival time at node j when arriving from node i in day d. The offline mathematical formulation is as follows.

(6.1) $$\max \sum_{k\in K}\sum_{p\in P} r_p y_{pk} - \sum_{k\in K}\sum_{d\in D}\sum_{(i,j)\in\delta^-(\bar{p}+m+k)} c_{ij} z_{ij}^d$$

Subject to:

(6.2) $$\sum_{(i,j)\in\delta^+(\bar{p}+k)} x_{ij}^d = \sum_{(i,j)\in\delta^-(\bar{p}+m+k)} x_{ij}^d \leq 1 \qquad k\in K, d\in D$$

(6.3) $$\sum_{k\in K} y_{ik} \leq 1 \qquad i\in P$$

(6.4) $$\sum_{(i,j)\in\delta^+(i)} x_{ij}^d = \sum_{k\in K} y_{ik} \qquad d\in D, i\in P_d$$

(6.5) $$\sum_{(j,i)\in\delta^-(i)} x_{ji}^d = \sum_{k\in K} y_{ik} \qquad d\in D, i\in P_d$$

(6.6) $$\sum_{(i,j)\in\delta^+(i)} z_{ij}^d - \sum_{(j,i)\in\delta^-(i)} z_{ji}^d = \sum_{(i,j)\in\delta^+(i)} (t_{ij}+s_i) x_{ij}^d \qquad d\in D, i\in P_d$$

(6.7) $$z_{\bar{p}+k,i}^d \leq t_{\bar{p}+k,i} x_{\bar{p}+k,i}^d \qquad d\in D, i\in P_d, k\in K$$

(6.8) $$z_{ij}^d \geq (t_{\bar{p}+1,i} + t_{ij} + s_i) x_{ij}^d \qquad d\in D, (i,j)\in A_d$$

(6.9) $$z_{ij}^d \leq (t_{max} - t_{j,\bar{p}+m+1} - s_j) x_{ij}^d \qquad d\in D, (i,j)\in A_d$$

(6.10) $$y_{jk} \geq y_{ik} + x_{ij}^d - 1 \qquad d\in D, k\in K, (i,j)\in A_d$$

(6.11) $$y_{\bar{p}+k,k} = y_{\bar{p}+m+k,k} = 1 \qquad k\in K$$

(6.12) $$x_{ij}^d \in \{0,1\} \qquad d\in D, (i,j)\in A_d$$

(6.13) $$z_{ij}^d \geq 0 \qquad d\in D, (i,j)\in A_d$$

(6.14) $$y_{ik} \in \{0,1\} \qquad k\in K, i\in P\cup\{\bar{p}+k, \bar{p}+m+k\}$$

As outlined by objective function (6.1), the goal is to maximize the difference between the patients revenues and the traversing costs, computed as the total costs for the time of arrival to the final depot for all the nurses. Constraints (6.2) regulate the activation of each vehicle k on any given day, imposing that at maximum one arc can leave from node $(\bar{p}+k)$ and entering in node $(\bar{p}+m+k)$. Constraints (6.3) are the nurse-patient consistency constraints, stating that each patient $i \in P$ is served by at most one nurse. Constraints (6.4)–(6.5) manage the arc flow for each patient (node), ensuring that, if accepted, exactly one arc enters and leaves node i on a day d. The elimination of sub-tours is managed through Constraints (6.6) ensuring that if a vehicle visits node j immediately after node i on a day d, then the time elapsed between the arrival times in the two nodes is equal to the execution time s_i required to serve node i plus the travel time t_{ij} to move from node i to node j. Constraints (6.7) set a bound on the minimum time required to reach the starting node after the depot, whereas constraints (6.8)–(6.9) define lower and upper bounds on the arrival time and duration of each route. We can avoid the duplication of Constraints (6.8)–(6.9) for each vehicle ending and starting nodes from the moment that they all have the same location and consequently, selected a specific arc (i, j), for each $k \in K$ $t_{\bar{p}+1} = t_{\bar{p}+k}$ and $t_{j,\bar{p}+m+1} = t_{j,\bar{p}+m+k}$. Constraints (6.10) avoid that an arc (i, j) is traversed by a vehicle $k \in K$ when the node i is assigned to k ($y_{ik} = 1$) but the node j is not ($y_{jk} = 0$). The respective starting and ending depots for each vehicle $k \in K$ are assigned through Constraints (6.11). Finally, Constraints (6.12)–(6.14) are variables domain.

6.3.1 Alternative Policies Mathematical Formulation

The problem presented in this section can be modified to simulate alternative policies that the agency can adopt to provide a different service regarding consistency and temporal distribution of visits. In our main model, consistency is inserted as a hard constraint, meaning that the initial assignment to a nurse can not be modified among visits to the same patient. Also, services are provided precisely on the days requested by the patient. As alternative, the agency might choose to change one or both of these two aspects, removing consistency or providing a more flexible service in terms of days. To this end, we insert the possibility of removing consistency from the main model and we introduce two different versions of visits flexibility: *partial flexibility* where the patient can be served either the day before or after the one he requested and *total flexibility*, stating that the patient can be visited at any time during the time horizon as long as the totality of the provided visits reaches the number of requested services. In total we model six different policies summarized in Table (6.2).

	Consistency	Flexibility
`cons_no`	yes	no
`cons_part`	yes	partial
`cons_tot`	yes	total
`noCons_no`	no	no
`noCons_part`	no	partial
`noCons_tot`	no	total

TABLE 6.2: Alternative policies configuration

The main model is represented by `cons_no` policy, indicating the presence of consistency and no flexibility in visits.

To model these new versions we introduced two new variables:

- $w_p = \begin{cases} 1, & \text{if patient } p \text{ is served by the hospital} \\ 0, & \text{otherwise.} \end{cases}$

- $y^d_{pk} = \begin{cases} 1, & \text{if patient } p \text{ is served by nurse } k \text{ on day } d \\ 0, & \text{otherwise.} \end{cases}$

Moreover, the parameter k_{max} indicates the maximum number of vehicles that can be assigned to patients.

We present here the updated model with highlighted changes to accommodate the new policies.

$$\max \sum_{p\in P} r_p w_p - \sum_{k\in K}\sum_{d\in D}\sum_{(i,j)\in\delta^-(\bar{p}+m+k)} z^d_{ij} \tag{6.15}$$

Subject to:

$$\sum_{(i,j)\in\delta^+(i)} x^d_{ij} = \sum_{k\in K} y^d_{ik} \qquad d\in D, i\in P_d \tag{6.16}$$

$$\sum_{(j,i)\in\delta^-(i)} x^d_{ji} = \sum_{k\in K} y^d_{ik} \qquad d\in D, i\in P_d \tag{6.17}$$

$$\sum_{k\in K} y^d_{ik} \leq 1 \qquad d\in D, i\in P_d \tag{6.18}$$

$$y^d_{ik} \leq y_{ik} \qquad d\in D, i\in P_d, k\in K \tag{6.19}$$

$$\sum_{k\in K} y_{ik} \leq k_{cons} \qquad i\in P \tag{6.20}$$

$$\sum_{d\in D_i}\sum_{k\in K} y^d_{ik} = |D_i| w_i \qquad i\in P \tag{6.21}$$

The objective function (6.15) is changed to accommodate the relaxation of consistency. The variable y_{pk} must to be replaced by the binary variable w_p since multiple nurses could be assigned to the same patient, making the sum of the y variables not equal to one. Constraints (6.16) and (6.17) lose the part of hard consistency stating that on a specific day d only one vehicle k must visit a node i without imposing that it is the same vehicle in all the time horizon. The maximum number of vehicles assigned to a patient is regulated by constraints (6.20). Constraints (6.18) impose that in each day $d \in D$ at maximum one vehicle is assigned to a node $i \in P_d$ while constraints (6.19) activates the y_{ik} variables only if at least one y^d_{ik} is active. Flexibility is handled by constraints (6.21). The total number of visits for each node $i \in P$, when it is served by the hospital ($w_i = 1$) must be equal to the cardinality of the original set D_i of days requested by the patient, regardless of the actual days of visits.

6.4 Solution Approaches

6.4.1 Adaptive Large Neighborhood Search

In this section we present the algorithm used to solve the problem in its Offline version, assuming all the information known. Although we acknowledge the fact that this represent a scenario significantly far from the reality, it is beneficial for us to examine a benchmark of the value derived from possessing perfect information. In this case, we have only one decision epoch (DE) in which all the decisions about accepting patients and creating routes are taken. Figure (6.2) adapts the example given in Figure (6.1), where only DE_0 is needed and all the patients are supposed to be known at day $d = 0$.

Table (6.3) encapsulates the key decisions made at each decision epoch. Given the presence of a singular decision epoch, every request falls within the P^B subset, which comprises booked patients. For each of these patients, a prior determination has been made regarding their acceptance (P^A) or rejection (P^R), leading to the subsequent updating of routes to include these patient visits.

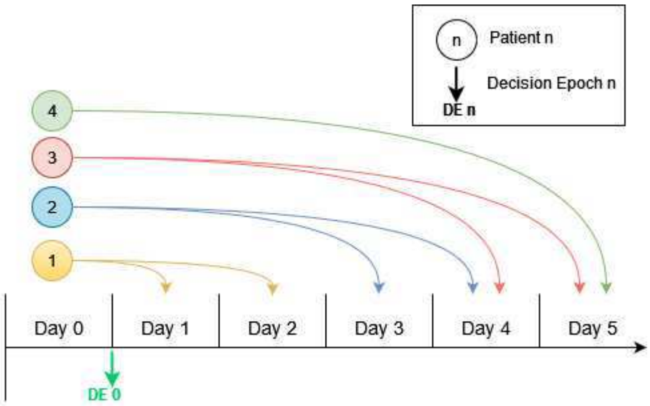


FIGURE 6.2: Requests management in the Offline Approach

TABLE 6.3: Routes creation in the Offline Approach

	Decision Epoch DE_0
P^B	1-2-3-4
$P^A \| P^R$	1-2-3-4
P^W	-
Routes update	yes

The Offline problem is solved using the Adaptive Large Neighborhood Search presented in Chapter (3) which is detailed in respect of our problem characteristics in the current section. The underlying algorithm remains Algorithm (2) presented in Section (3.2). We now give a more specific formulation of the insertion and removal operators as well as a description of how an initial solution is obtained.

Initial Solution

The ALNS algorithm requires as input an initial feasible solution. To this end, we develop a simple constructive heuristic that starts from a set of $|K|$

empty routes. A patient $p \in P$ is then randomly selected and the heuristic tries to insert all its visits, i.e. its nodes, according to the least-cost principle; that is, each node is possibly inserted in a route and position that minimize the total routes length. If it is impossible to accommodate all the visits, the patient is marked as externally assigned. The selection is repeated until there are no more patient left to be inserted. The creation of an initial solution is repeated multiple times in a time limit τ_{init}. Since the choice of the first patient to insert is random, we can obtain a different objective function value each time we create an initial solution. For this reason, we keep track of the found solutions and their objectives saving as incumbent only the one with the best solution value. When a feasible solution is found, it is compared to the incumbent solution and kept only if it represents an improvement.

Destroy Operators

All the destroy operators take as input a feasible solution *sol* and the degree of destruction q and create a set of removed patients P^-. This set is created by sorting in a specific order the patients in *sol*. We implement a set Ω^- of five different destroy operators associated to five sorting procedures:

- Remove Random d_1: randomly selects q patients and remove all their nodes from the solution.
- Remove Shaw d_2: removes from the solution q patients and their visits based on their *Relatedeness measure* $R(i, j)$. Taking a patient i and a patient j their $R(i, j)$ is computed as

  $$R(i, j) = \frac{1}{t_{ij} + 0.1 * |s_i - s_j| + \Lambda_{ij}}$$

  where Λ_{ij} is a parameter equal to 1 if the two nodes belong to the same route, 0 otherwise.
- Remove High Savings d_3: selects and removes the q patients that, if removed, have the biggest impact on traveling time.
- Remove High Service Dates d_4: selects and removes the q patients with the highest number of service dates.
- Remove High Service Dates d_5: selects and removes the q patients that, considering all the days of service, have the highest total service time.

Repair Operators

The repair operators aim at reconstructing the current partial solution by trying to insert the patients removed by the destroy operator. The procedure sorts in a specific order the patients to be inserted. We implemented a set Ω^+ of four different sorting criteria and, thus, repair operators.

- Random Insertion r_1: random ordering of the patients.
- Low Travel Time r_2: Patients are sorted in ascending order according to the incremental travel time incurred by their incorporation into a route.
- Low Service Times r_3: Patients are sorted in ascending order according to their total service time.
- Low service Dates r_4: Patients are sorted in ascending order according to their total number of visits.

Once the patients are ordered, we re-insert them in a sequential order or using the Regret procedure showed in Section (3.2).

6.4.2 Myopic Dynamic Heuristic (MDH)

A pure dynamic approach, with no knowledge about future events, allows us to simulate at best the real-life routing management of the nurse agency. In fact, at each decision epoch, the agency has knowledge only of the patients whom already submitted a request. Additionally, route adjustments are only made on a given day if there is a patient not yet scheduled who requires a visit the following day. Figure (6.3) demonstrates the dynamic approach to the scenario depicted in Figure (6.1), with a focus on day $d = 3$. In this illustration, patients already known to the agency are marked with colored circles, while those unknown, such as Patient 4 whose request is scheduled for day $d = 4$, are indicated in grey. The data and decisions at each decision epoch are concisely detailed in Table (6.4).

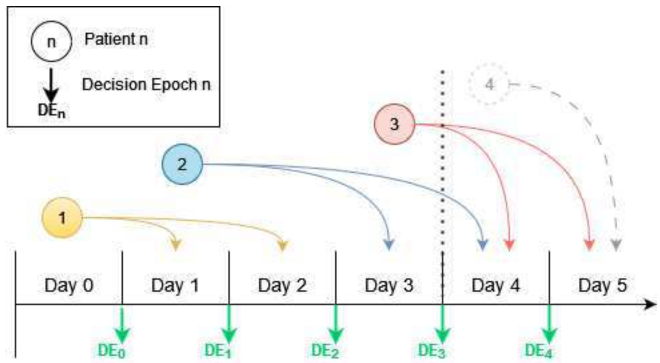


FIGURE 6.3: Requests management in the dynamic approach

We can see that Patient 2 is placed in the P^W subset, which is for waiting requests, from day 2 to day 3. This happens because Patient 2 is known from day $d = 1$ but doesn't need service until day $d = 3$. Therefore, the decision about this patient is delayed until the second decision epoch, DE_2. Similarly,

TABLE 6.4: Routes creation in the Dynamic Approach

	Decision Epoch				
	DE_0	DE_1	DE_2	DE_3	DE_4
P^B	1	1-2	1-2	1-2-3	1-2-3-4
$P^A \vert P^R$	1	-	2	3	4
P^W	-	2	-	-	-
Routes update	yes	no	yes	yes	yes

the routes for day $d = 1$ are not changed, as no new requests need to be added to the routes for day $d = 2$.

The algorithm employed for addressing the dynamic version is the Myopic Dynamic Heuristic (MDH) presented in Section (3.3.1). Algorithm (8) gives a problem specific formulation of the high level Algorithm (3).

Algorithm 8 MyopicDynamicApproach($\mathcal{I}_0, sol_0, P_0^A, P_0^R, P_0^W$)

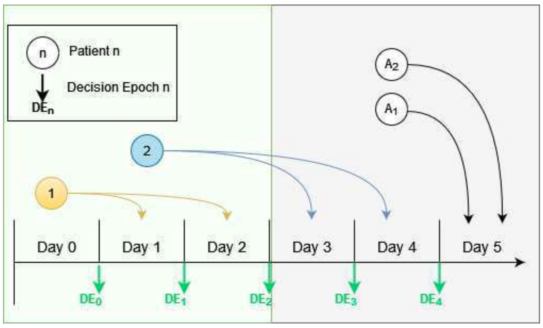
1: $d \leftarrow 1$
2: $(P^A, P^R, P^W) \leftarrow (P_0^A, P_0^R, P_0^W)$
3: $sol \leftarrow sol_0$
4: $\mathcal{I} \leftarrow \mathcal{I}_0$
5: **while** $d < d_{max}$ **do**
6: $\quad sol \leftarrow \texttt{LockAssignments}(sol, P^A)$
7: $\quad \mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{I}_d$
8: $\quad P^B \leftarrow \texttt{GetNewPatients}(\mathcal{I}_d) \cup P^W$
9: $\quad (sol, P_d^A, P_d^R) \leftarrow \texttt{OptimizeRoutes}(\mathcal{I}, sol, P^B)$
10: $\quad P^W \leftarrow P^B \setminus (P_d^A \cup P_d^R)$
11: $\quad P^A \leftarrow P^A \cup P_d^A$
12: $\quad P^R \leftarrow P^R \cup P_d^R$
13: $\quad d \leftarrow d + 1$
14: **end while**
15: **return** (sol, P^A, P^R)

The algorithm requires as input an initial feasible solution sol_0, an instance (i.e. a partial part) of the problem $\mathcal{I}_0$ containing all the available information at the start of the algorithm and the division of patients in the three subset P_0^A, P_0^R, P_0^W according to the solution sol_0. The initialization part from Step (1) to Step (4), initializes the current day d to the first day of the time horizon and sets as current division of patients (P^A, P^R, P^W) the initial division (P_0^A, P_0^R, P_0^W). Moreover, it initializes the current solution sol to sol_0 and the current instance $\mathcal{I}$ to $\mathcal{I}_0$. Then, the Loop from Step (5) to Step (14) iterates for each day $d \in D$. The function `LockAssignments`(sol, P^A) on Step (6) fixes the nurse-patients assignments contained in sol for the request in P^A, meaning that all the patients that have already been assigned to a specific nurse cannot be switched during the iteration. The on Step (7) it adds to the instance $\mathcal{I}$ the instance $\mathcal{I}_d$ containing the aspects of the problem that emerged during day d. More specifically, it adds the patients that have submitted a request during the current day and were unknown in $\mathcal{I}$. On Step (8) the function `GetNewPatients`$(\mathcal{I}_d)$ updates the set of booked patients P^B adding to the pending patients P^W the new patients in $\mathcal{I}_d$. The Algorithm then optimizes routes for the following days on Step (9) using the ALNS from Algorithm (2). The function `OptimizeRoutes`$(\mathcal{I}, sol, P^B)$ takes as input (i) the instance $\mathcal{I}$, (ii) the solution sol containing all the routes for the days previous to d with the fixed assignments for patients contained in P^A and (iii) the set of known patients P^B. Among these patients an accept/reject decision is taken only for the ones with the first service day in $d+1$, the others will be inserted in P^W. The ouptut of the `OptimizeRoutes`$(\mathcal{I}, sol, P^B)$ is an updated sol with the routes for the next day and the set of accepted patients of the day P_d^A and the set of rejected ones P_d^R. From Step (10) to (12) the set of patients are updated. The patients in P^B without a decision are added to P^W and the accepted and rejected patients of the day P_d^A and P_d^R are inserted in P^A and P^R, respectively. The algorithm moved to the next day on Step (13). At the end of the while loop the best solution is returned.

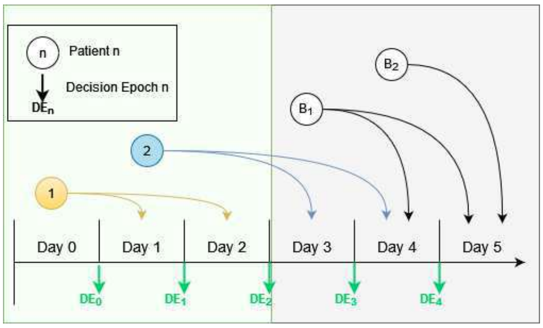
6.4.3 Multi-Scenario-Based Progressive Fixing (MSB-PF)

In this section we detailed the MSB-PF already presented in Chapter (3), Section (3.3.2). MSB-PF tackles the problem generating multiple scenarios to simulate future events. Meaning that, at each decision epoch, known information are merged with possible future events coming from scenarios creating a set of deterministic sub-problems. All the sub-problems are then solved as Offline problems. The term *Progressive Fixing* refers to the fact that, during a decision epoch, rather than making simultaneous decisions for patients in P^W, the heuristic progressively selects one decision to fix for the subsequent iterations. Specifically, this decision involves fixing the most prevalent nurse-patient assignment observed across all sub-problems. The selected patient is removed from P^W and each sub-problem is then resolved including this new decisions. The procedure moves to the next decision epoch only when all the patients in P^W are accepted or rejected. To give a better understanding of scenarios generation, in Figure (6.4) is shown information that the nurse agency has at Day 2 or DE_2, considering the example in Figure (6.1) as representative of reality. The part highlighted in green represents the events belonging to the past, hence known, while the grey part symbolizes a possible realization of future events according to two different scenarios, Scenario A in Figure (6.4a) and Scenario B in Figure (6.4b).

On Day 2, Patient 1 and Patient 2 are already known since their release date is previous. On DE_2, Patient 2 is included in P^W since it requires a visit on Day 3. However, in taking this decision the agency has no knowledge



(A) Scenario A



(B) Scenario B

FIGURE 6.4: Requests management in a Scenario-Based approach

about future events but can take advantage of the information contained in two fictitious scenarios. According to Scenario A, for example, two new requests are expected on Day 4 with a service on Day 5, this wouldn't affect the scheduling of Patient 2. On the other hand, Scenario B foresees the arrival of Patient B_1 on Day 3 with two visits, one of them on Day 4 as Patient 2. Hence, the agency has to take into account that, according to this scenario the choice on Patient 2 it might be affected by future requests. Algorithm (9) shows the exact structure of the procedure.

The algorithm requires an initial feasible solution sol_0, the instance $\mathcal{I}_0$ containing all information available at the beginning of the time horizon, the three subsets P_0^A, P_0^R, P_0^W and the set of scenarios S. In Steps (1) to (4), the current day is set to the beginning of the time horizon, the current solution sol is initialized to the initial solution sol_0 as well as the current sets P^A, P^R, P^W and $\mathcal{I}$. The main loop (Steps (5)-(32)) is repeated for all the days of the time horizon. On Step (6), the function $\texttt{LockAssignments}(sol, P^A)$ locks the decision already taken in the current solution sol about the accepted patients P^A. It then updates the instance $\mathcal{I}$ merging it with the partial instance $\mathcal{I}_d$ composed of the patients become known on day d on Step (7). The set P^B of booked patients is updated merging the new patients from $\mathcal{I}_d$ to the set of waiting patients P^W. On Step (9) is created the empty set of scenarios solutions $\mathcal{F}$. The Loop from Step (10) to Step (14) is repeated for each scenario

Algorithm 9 ProgressiveFixing$(\mathcal{I}_0, sol_0, P_0^A, P_0^R, P_0^W, S)$

$d \leftarrow 1$
$(P^A, P^R, P^W) \leftarrow (P_0^A, P_0^R, P_0^W)$
$sol \leftarrow sol_0$
$\mathcal{I} \leftarrow \mathcal{I}_0$
while $d < d_{max}$ **do**
    $sol \leftarrow$ `LockAssignments`(sol, P^A)
    $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{I}_d$
    $P^B \leftarrow$ `GetNewPatients`$(\mathcal{I}_d) \cup P^W$
    $\mathcal{F} \leftarrow \emptyset$
    for all $s \in S$ **do**
        $P_s^B \leftarrow$ `GetNewPatients`$(s, d) \cup P^B$
        $(sol_s, P_{ds}^A, P_{ds}^R) \leftarrow$ `OptimizeRoutes`$(\mathcal{I}, sol, s, P_s^S)$
        $\mathcal{F} \leftarrow \mathcal{F} \cup (sol_s, P_{ds}^A, P_{ds}^R)$
    end for
    $(sol, P_d^A, P_d^R, P_d^W, \bar{P}_d^W) \leftarrow$ `FixCommonAssignments`$(sol, \mathcal{F}, P^B)$
    while $\bar{P}_d^W \neq \emptyset$ **do**
        $\bar{P}_d^W \leftarrow$ `SortByConsensus`$(\bar{P}_d^W, \mathcal{F})$
        $(sol, P_d^A, P_d^R) \leftarrow$ `FixFirstAssignment`$(sol, \mathcal{F}, P_d^A, P_d^R, \bar{P}_d^W)$
        $\bar{P}_d^W \leftarrow$ `RemoveFirst`$(\bar{P}_d^W)$
        $\mathcal{F} \leftarrow \emptyset$
        for all $s \in S$ **do**
            $P_s^S \leftarrow$ `GetNewPatients`$(s, d) \cup \bar{P}_d^W$
            $(sol_s, P_{ds}^A, P_{ds}^R) \leftarrow$ `OptimizeRoutes`$(\mathcal{I}, sol, s, \bar{P}_d^W)$
            $\mathcal{F} \leftarrow \mathcal{F} \cup (sol_s, P_{ds}^A, P_{ds}^R)$
        end for
        $(sol, P_d^A, P_d^R, P_d^W, \bar{P}_d^W) \leftarrow$ `FixCommonAssignments`$(sol, \mathcal{F}, P_d^A, P_d^R, \bar{P}_d^W)$
    end while
    $P^W \leftarrow P^B \setminus (P_d^A \cup P_d^R)$
    $P^A \leftarrow P^A \cup P_d^A$
    $P^R \leftarrow P^R \cup P_d^R$
    $d \leftarrow d + 1$
end while
return (sol, P^A, P^R)

$s \in S$. At first, on Step (11), the set of scenario's patients P_s^B is created emerging the already known patients in P^B and the new patients of day d coming from scenario s. Then, function `OptimizeRoutes`$(\mathcal{I}, sol, s, P_s^B)$ optimizes the scenario using the ALNS algorithm. The optimization is made considering the decisions already locked in the current solution *sol* and new information coming from the scenario s. In this step, decision for all the patients in P_s^B is made, whether they are real patients coming from instance $\mathcal{I}$ or fictitious one coming from scenario s. However, only decisions about real ones will be fixed. The solution of the scenario is then added to the set $\mathcal{F}$. Once all the solutions are collected, on Step (14), function `FixCommonAssignments` fixes the decisions that are common among all the scenarios. After the fixing, the set of remaining decisions $\bar{P}_d^W$ is created. The while loop from Step (16) to Step (27) continues until the set in $P\bar{P}_d^W$ is not empty, meaning that all the pending patients have a decision. To assign these decisions, the set $P\bar{P}_d^W$ is sorted by the function `SortByConsensus`$(\bar{P}_d^W, \mathcal{F})$. The function collects data on assignments from solutions across various scenarios and arranges the set of patients $\bar{P}_d^W$ based on the frequency of decision-making occurrences among these scenarios. This is achieved by counting the times a specific decision for a patient has been made across scenarios and subsequently organizing the patients in a descending order of decision frequency. The decision about the first patient of the ordered list is fixed in the solution *sol* by the function `FixFirstAssignment`$(sol, \mathcal{F}, P_d^A, P_d^R, \bar{P}_d^W)$ that also updates the sets P_d^A, P_d^R according to the decision. The patient is then removed from the set $\bar{P}_d^W$ and the set of solutions $\mathcal{F}$ is emptied. In loop from step (21) to Step (25) the scenarios are solved again with the new solution fixed. When the set $\bar{P}_d^W$ is empty the while loops ends and the subset of patients are updated. In Step (28) the set P^W of pending patients is created removing from P^B the patients either accepted (P_d^A) or rejected (P_d^R) during the day. The sets P_d^A and P_d^R are merged with the current set P^A and P^R on steps (29) and (30), respectively. The algorithm then moves to the next day and all the process is repeated from step (6). When d_{max} is reached, the current solution *sol* is returned with the sets of accepted and rejected patients.

6.5 Computational Results and Managerial Insights

In this section, we present the results obtained from solving the model with different methods to draw interesting analyses for the nurse agency to deal with dynamic patients management. We first describe the benchmark instances used in our study. Then, we propose a valuable sensitivity analysis on small-size instances valid to draw managerial insights. Lastly, we present the performances of the methods shown in Section (6.4) applied on large-size instances. All tests were run on a Ubuntu 20.04.2 machine with an AMD Ryzen 9 3950x CPU, 16 cores, 32 threads, and 32 GB RAM. We used Gurobi

10.0.1 as mixed integer programming solver. All methods have been implemented in Java 17.

6.5.1 Instance Generation

We have evaluated the solutions methods on a large number of benchmark instances. Specifically, we employed a set of *small-size* instances (`Set 1`) which were optimally solved using Gurobi. This set is used to conduct a sensitivity analysis concerning key parameters on the model of Section (6.3) and for deriving managerial insights about the policies the agency might adopt presented in Section (6.3.1). Together with Set 1 instances, a set of *large-size* instances (Set 2) is employed to evaluate the performance of the solutions method of Section (6.4) in terms of computing time and quality of solutions. Among all the instances, the locations of patients are randomly generated in a geographical area such that the maximum travel time between any two nodes is equal to 120 minutes. The sets are differentiated by the number of patients $|P|$, the number of nurses $|K|$, and the time horizon $|D|$. Two key parameters vary as follows: the parameter λ, which indicates the percentage of patients having a release date of $d = 1$, and the parameter ρ, utilized in calculating patients profits, as in Equation (6.22), to alter the profit-to-travel cost ratio. The profit of a patient $p \in P$ is computed as:

$$r_p = |D_p|(s_p + 2t_{0p}\rho) * \omega \tag{6.22}$$

where ω represents a random part of noise included in the interval [0.8,1.2]. We can notice how the impact of the travel times, in particular of the round trip from the depot to the patient $2t_{0p}$, is regulated by the parameter ρ; the higher the value, the higher the impact of the revenue compared to the travelling costs. The distribution of visits over the time horizon is governed by a sequence of events. An *event*, in this context, refers to an occurrence that triggers a series of requests for visits. Each event is characterized by its distinct number and frequency of visits. For example, we defined an event to simulate a Covid-19 outbreak that, if selected, generates a set of patients geographically close to each other with a high number of visits (modeling the Covid-19 illness duration) but with a low frequency in time. Conversely, an event simulating for example a day aimed at home vaccine visits will assign to patients a profile including only one visit over the time horizon. In our study we tested instances with values of ρ equal to $\{0.25, 0.5, 1, 2\}$ for the small size instances and $\rho = \{0.5, 1, 2\}$ for the large size ones. The percentage of patients known on the starting day varies as $\lambda = \{0\%, 20\%, 50\%, 100\%\}$; where the $\lambda = 100\%$ case corresponds to the Offline version of the problem, with all the patients known, while the $\lambda = 0\%$ indicates that the problem starts with no knowledge about future events. The sensitivity analysis on small-size instances is made on the Offline problem, hence setting $\lambda = 100\%$.

For each benchmark instance we generated a set of $|S| = 100$ sample scenarios starting from the realistic scenario s^*. For each scenario $s \in S$, the total number of patients P^s is drawn from a normal distribution having mean $|P|$ and variance **, where $|P|$ is the number of patients belonging s^*. Of these $|P^s|$ patients the first $\lambda * |P|$ come from the real scenarios being the ones with release date equal to $d = 0$ and known at the beginning of the time horizon while the others are randomly generated.

Table (6.5) and Table (6.6) show the dimension and configurations of the two sets of instances. #*Inst* represents the number of instances generated per combination.

Set	Values
$\|P\|$	25
$\|D\|$	7
$\|K\|$	2
$\lambda(\%)$	100
ρ	$\{0.25, 0.5, 1, 2\}$
$\#Inst$	20
Total	80

TABLE 6.5: Instance structure of Set 1

Set	Values
$\|P\|$	$\{50, 100, 200\}$
$\|D\|$	15
$\|K\|$	$\{3, 4\}$
$\lambda(\%)$	$\{0, 20, 50, 100\}$
ρ	$\{0.5, 1\}$
$\#Inst$	3
Total	144

TABLE 6.6: Instance structure of Set 2

In our tests, the total number of instances is determined by the product of the cardinalities of each parameter set. Precisely, for Set 1, the total number of instances is calculated by considering the single values of $|P|$, $|D|$, $|K|$, and λ, along with the four distinct values in the set ρ, multiplied for the number of instances per combination $\#Inst$. Therefore, the total number of Set 1 instances is 80. For Set 2, the total number of instances is calculated by considering three distinct values of $|P|$, a single value of $|D|$, two distinct values of $|K|$, four distinct percentage values of λ, and three distinct values in the set ρ, multiplied by the number of instances per combination $\#Inst$. Therefore, the total number of Set 2 instances is 144.

6.5.2 Managerial Insights on small size instances

In this section, we present the results on parameters sensitivity analysis and the key managerial insights obtained by analyzing the instances in Set 1. At first, we give an overview of how the parameter ρ affects the overall results. This can give to the agency more understanding when it comes to patients revenue configuration. Then, we present some insights on results variations among the model versions of Section (6.3.1). Table (6.7) provides a summary of the average results obtained solving to optimality Set 1 instances for each of the different policies. For each policy, on the rows, are listed the value of the objective function *Obj*, the revenue component PR, the travel time part TTT and the average number of accepted patients $|P^A|$. The corresponding value of the ρ parameter is indicated on the columns.

		$\rho = 0.25$	$\rho = 0.5$	$\rho = 1$	$\rho = 2$
`cons_no`	*Obj*	29.45	880.40	3174.50	7916.65
	PR	617.90	3135.20	6047.75	10802.15
	TTT	588.45	2254.80	2873.25	2885.50
	$\|P^A\|$	5.35	20.50	24.90	25.00
`cons_part`	*Obj*	150.90	1220.35	3557.25	8288.40
	PR	1545.95	3496.05	6043.10	10802.15
	TTT	1395.05	2275.70	2485.85	2513.75
	$\|P^A\|$	15.55	23.20	24.80	25.00
`cons_tot`	*Obj*	169.10	1272.55	3608.80	8334.15
	PR	1663.90	3525.35	6049.05	10802.15
	TTT	1494.80	2252.80	2440.25	2468.00
	$\|P^A\|$	17.85	23.55	24.85	25.00
`nocons_no`	*Obj*	30.35	912.35	3252.05	7994.25
	PR	643.90	3336.05	6054.40	10802.15
	TTT	613.55	2423.70	2802.35	2807.90
	$\|P^A\|$	5.65	21.85	24.95	25.00
`nocons_part`	*Obj*	148.70	1228.20	3585.05	8324.55
	PR	1572.70	3510.55	6050.95	10802.15
	TTT	1424.00	2282.35	2465.90	2477.60
	$\|P^A\|$	15.90	23.15	24.90	25.00
`nocons_tot`	*Obj*	170.85	1272.00	3631.30	8373.30
	PR	1675.50	3543.80	6049.05	10802.15
	TTT	1504.65	2271.80	2417.75	2428.85
	$\|P^A\|$	17.65	23.70	24.85	25.00

TABLE 6.7: Average results for Set 1 instances varying the parameter ρ

Aggregated computational times and percentages gap are shown in Table (6.8).

Table (6.9),presents the average results among policies of the results presented in Table (6.7). Figure (6.5) and Figure (6.6) gives a visual representation of these average results among all policies at the parameter ρ variation.

	Gap (%)	Time (s)
`cons_no`	0.27	537.12
`cons_part`	2.18	3606.62
`cons_tot`	0.71	2375.52
`noCons_no`	0.03	412.36
`noCons_part`	1.93	3509.89
`noCons_tot`	1.01	1820.04

TABLE 6.8: Average MIP Gap (%) and computational times for Set 1 instances.

	$\rho = 0.25$	$\rho = 0.5$	$\rho = 1$	$\rho = 2$
Obj	116.6	1130.9	3468.2	8205.2
PR	1286.6	3424.5	6049.1	10802.2
TTT	1170.1	2293.5	2580.9	2596.9
$\|P^A\|$	13.0	22.7	24.9	25.0

TABLE 6.9: Average results for Set 1 instances varying the parameter ρ

Figure (6.5) shows the variation of the total objective function *Obj* obtained as difference between the PR component and the TTT one. It can be noted how, increasing the parameter ρ, the revenues increase significantly with only a tiny variation of travel time, hence costs. In fact, when $\rho = 0.25$ the average values of PR and TTT are equal to 1286.6 and 1170.1, respectively. Whe ρ is increased to 2, these values equal to PR = 10802.2 and TTT = 2596.9 showing a much relevant increase in the patients' revenue at the cost of only a more minor increase in traveled times. The results obtained in Figure (6.5) reflect the different operational strategies that can be applied by the nurse agencies. A lower value of ρ represents a balanced decision-making process in which revenues and travel times are considered equally while increasing the value shifts the decisions to a more profit-maximization-focused strategy.

Figure (6.6) shows the number of patients that the agency is willing to accommodate when the impact of revenues varies. As expected, increasing the value of ρ, hence of the weight of revenues over the operational costs, leads to a higher acceptance rate for patients 24.9 patients accepted already for $\rho = 1$. To better understand how results are affected by a change in policy we now presents disaggregated results for each case. Table (6.10), Table (6.11) and Table (6.12) present the percentage variation of outcomes compared to our standard policy `cons_no` of patients revenues (PR), travel times (TTT) and objective function (*Obj*), respectively.

The standard policy `cons_no` used in the SMHHP-C is the more rigid and, consequently, the least profitable. Modifying this by either relaxing the consistency constraints or integrating a flexible visit management service enhances the total objective function, yielding higher revenues and reduced

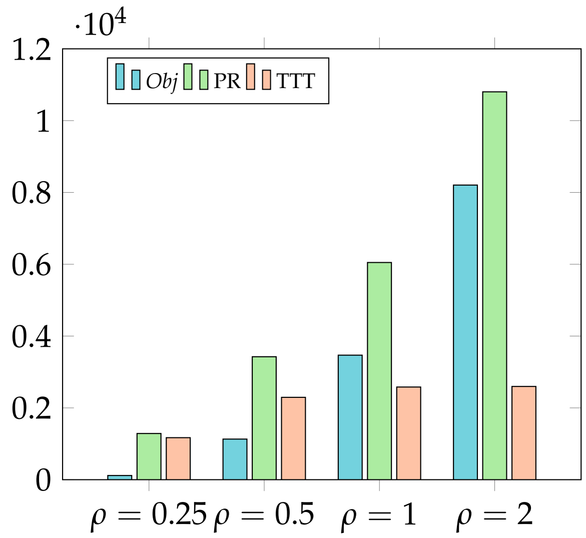


FIGURE 6.5: Grouped bar chart showing the average results among policies of PR, TTT, and Obj values for different ρ parameters.

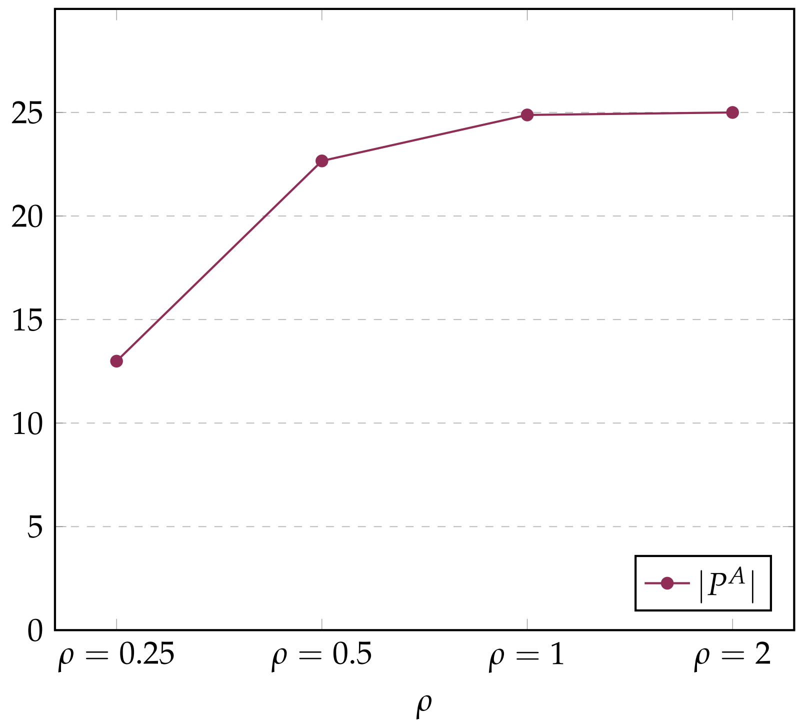


FIGURE 6.6: Average number of accepted patients for different values of parameter ρ.

	$\rho = 0.25$	$\rho = 0.5$	$\rho = 1$	$\rho = 2$
cons_no	**29.45**	**880.40**	**3174.50**	**7916.65**
cons_part	412.39%	38.61%	12.06%	4.70%
cons_tot	474.19%	44.54%	13.68%	5.27%
noCons_no	3.06%	3.63%	2.44%	0.98%
noCons_part	404.92%	39.50%	12.93%	5.15%
noCons_tot	480.14%	44.48%	14.39%	5.77%

TABLE 6.10: Percentage variation of objective function (*Obj*) for different values of ρ and different policies

	$\rho = 0.25$	$\rho = 0.5$	$\rho = 1$	$\rho = 2$
cons_no	**618**	**3135**	**6048**	**10802**
cons_part	150.19%	11.51%	0.00%	0.00%
cons_tot	169.28%	12.44%	0.02%	0.00%
noCons_no	4.21%	6.41%	0.11%	0.00%
noCons_part	154.52%	11.97%	0.05%	0.00%
noCons_tot	171.16%	13.03%	0.02%	0.00%

TABLE 6.11: Percentage variation of Patients Revenues (PR) for different values of ρ and different policies

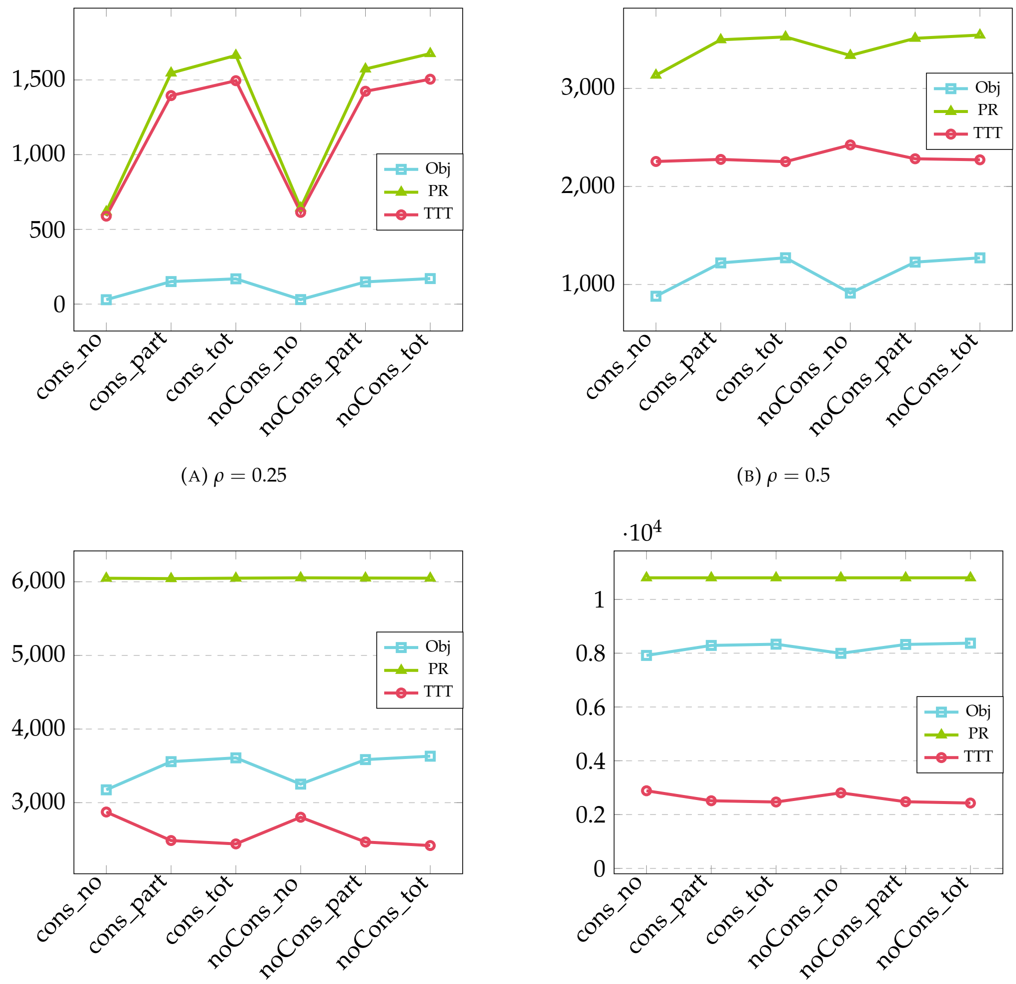
	$\rho = 0.25$	$\rho = 0.5$	$\rho = 1$	$\rho = 2$
cons_no	**588**	**2254.80**	**2873.25**	**2885.50**
cons_part	137.07%	0.93%	-13.48%	-12.88%
cons_total	154.02%	-0.09%	-15.07%	-14.47%
noCons_no	4.27%	7.49%	-2.47%	-2.69%
noCons_part	141.99%	1.22%	-14.18%	-14.14%
noCons_total	155.70%	0.75%	-15.85%	-15.83%

TABLE 6.12: Percentage variation of Total Traveled Times (TTT) for different values of ρ and different policies

travel costs. It's noteworthy, however, that the impact of relaxing consistency is substantially less than that of incorporating flexibility. Table (6.10) shows that, comparing policies with and without consistency (such as the pair `cons_no-noCons_no`), reveals that the improvement in the objective function is relatively modest, ranging from 0.98% at $\rho = 2$ to 3.63% at $\rho = 0.5$. In contrast, adopting a partially or fully flexible policy, while maintaining consistency, results in more significant improvements. This is exemplified by the comparison of `cons_no` with `cons_tot`, where the enhancement in the objective function is 5.27% at $\rho = 2$ and a substantial 474.19% at $\rho = 0.25$. These findings indicate that for an agency, once a patient is accepted, offering a service with flexible visitation days is more advantageous than assigning multiple nurses to a single patient. This approach not only maximizes efficiency but also significantly boosts profitability. This pattern is also confirmed in Figure (6.7) where, for any value of ρ, a growing line between policies with the same consistency approach is reported that decreases when there is a switch between consistency and no consistency.

A notable trend is that as we move from lower to higher ρ values, the revenue (PR) curves tend to level off. This implies that at lower patient profit margins, the model has greater latitude to adjust patient acceptance or rejection in response to policy changes. This is particularly evident in Figure (6.7b), where the PR line exhibits a marked change between policies, indicating that a more flexible model significantly enhances the likelihood of increasing profits accepting more patients. This conclusion is further supported by the behavior of the TTT line, where its relative flatness suggests that increasing patient visits due to flexibility does not correspondingly increase travel times. The higher variation is obtained confronting a total flexible policy with consistency with a rigid one without consistency (pair `cons_total-noCons_no`) with travel times passing from a value of 2252.8 to 2423.70. Conversely, Figure (6.7c) depicts a scenario in which there is no substantial trade-off between policies regarding revenue. The reduction of the objective function is obtained through a variation of the travel times TTT, suggesting that more flexible policies allow for better optimization of routings. Figure (6.7a) and Figure (6.7d) report two extreme situations. In the first, the profit ratio is too low for the agency to practice some analysis; when flexibility is not allowed in most solved instances, no patients are accepted, leading to an objective function equal to 0. On the opposite, when $\rho = 2$, there is no substantial difference between different policies since often all the patients are accepted leaving no space to optimization. Figure (6.8) presents a resume on the number of accepted patients per policy and ρ values.

In the performance analysis of solution methods, we opted to exclude the extreme ρ values, focusing solely on cases where $\rho = \{0.5, 1\}$.



(A) $\rho = 0.25$ (B) $\rho = 0.5$

(C) $\rho = 1$ (D) $\rho = 2$

FIGURE 6.7: Results for different policies at the variation of parameter ρ

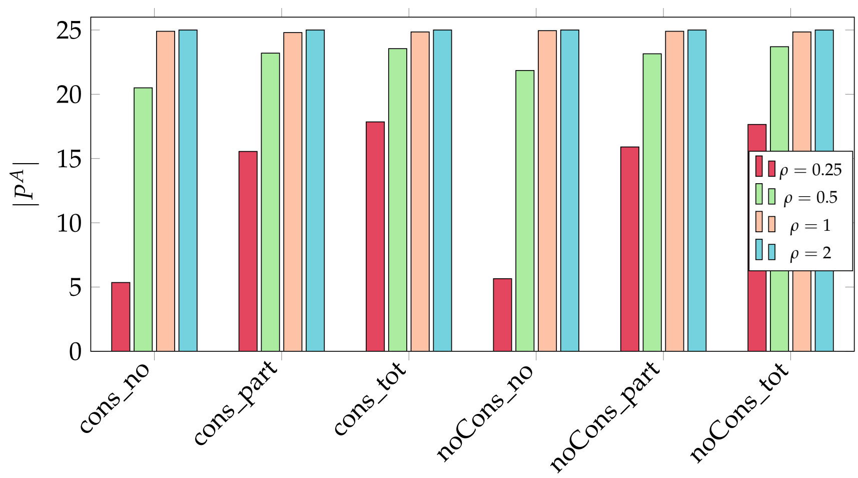


FIGURE 6.8: Number of accepted patients $|P^A|$ for different policies.

TABLE 6.13: Tuning of the `ParallelALNS` parameters.

Parameter	Meaning	Value
$qStep$	Incremental quantity of the degree of destruction q	5
q_{max}	Maximum value of q	$\lfloor 0.75\|P\| \rfloor$
$noImpIter$	Number of iterations without improvement before increasing q	500
$epochIter$	Number of iterations per epoch	10000
$Iter_{init}$	Number of iterations for the initial solution heuristic	100000
τ	Time limit for `ParallelALNS`	300s,10s
$\gamma_1, \gamma_2, \gamma_3$	Weight values in weighted operators	0.9, 0.07, 0.03
T_0	Initial temperature	50
α	Cooling rate	1
T_{iter}	Cooling schedule	100

6.5.3 Resolution Methods Performance Analysis

In this section we present the results obtained by solving the model with the three different teqniques presented in Section (6.4): the Offline Approach, the Myopic Dinamic Heuristic (MDH) and the Multi-Scenario-Based Progressive Fixing (MSB-PF). We conducted preliminary tests using Set 1 instances to assess the quality of our ALNS metaheuristic used to solve the Offline Problem and all the deterministic sub-problems of the dynamic approaches. Based on these tests we fixed nearly all the ALNS parameters, except for the time limit τ which depends on the heuristic applied; we used a time limit in seconds equal to $\tau = 300$ for the Offline Problem and $\tau = 10$ for each sub-problem of MDH and MSB-PF. All parameters values are listed in Table (6.13). We evaluate how well our Adaptive Large Neighborhood Search (ALNS) performs on the first set of problems (Set 1), comparing our Offline Approach's results to those obtained with Gurobi. For the same set, we also explore how the parameter λ, which represents the initial level of knowledge, affects the performance of MDH and MSB-PF. Next, we use the second set of problems (Set 2) to assess the performance of MDH and MSB-PF in terms of computational time and solution quality.

Computational Results on Set 1 instances

The first analysis performed to tune our ALNS parameters has been made on Set 1 instances comparing the exact values found by Gurobi with the ALNS applied to the Offline Problem, hence with all the information known at the beginning. Table (6.14) presents the percentage gap (Δ_{OFF}) for each instance of Set 1 with $\rho = \{0.5, 1\}$. A positive Δ_{gap} indicates that Gurobi performs better than ALNS. Instances are indicated as `ID`-ρ where all share a parameter $\lambda = 100\%$.

It can be noticed how, when $\rho = 1$ the ALNS finds the same solution as the commercial solver Gurobi. However, when $\rho = 0.5$, it finds the optimal solution only in 2 instances out of 20 and has an average gap of the 10%. Set 1 instances have been solved also using MDH and MSB-PF, varying the value of λ. In particular, to evaluate the impact that initial information

$\rho = 0.5$				$\rho = 1$			
Inst.	Δ_{OFF}	Inst.	Δ_{OFF}	Inst.	Δ_{OFF}	Inst.	Δ_{OFF}
0	9	10	33	0	0	10	0
1	9	11	3	1	0	11	0
2	0	12	16	2	0	12	0
3	5	13	7	3	0	13	0
4	13	14	8	4	0	14	0
5	10	15	0	5	0	15	0
6	5	16	3	6	0	16	0
7	28	17	4	7	0	17	0
8	34	18	11	8	0	18	0
9	8	19	17	9	0	19	0

TABLE 6.14: Comparison between ALNS and Gurobi on Set 1 instances.

might have on the overall solution we tested two extreme situations with $\lambda = \{0\%, 100\%\}$. Table (6.15) compares the results obtained with Gurobi for the Offline Problem with the results of the pure dynamic approach and the scenario-based heuristic when the percentage of known information equals the 100%. Columns Δ_{MDH} and Δ_{MSB-PF} report the percentage gap in respect of the Offline Problem (pure ALNS) for the MDH and MSB-PF, respectively. A positive percentage means that the found solution is worse than the Offline one.

From Table (6.15) arises a distinct difference in performance of both the heuristics for different ρ values. For $\rho = 0.5$, the percentage gaps are generally higher, indicating a notable deviation from the optimal solution of the Offline Problem. This could imply that, when the problem has a wider possibility of accepting or rejecting patients approaching it dynamically lead to a lack of solution efficiency. In contrast, when ρ is set to 1, the percentage gaps decrease significantly, indicating a closer alignment with the optimal Offline Problem solutions. This implies that in situations where the problem is primarily focused on routing, rather than dynamic patient selection, having complete information is crucial for achieving solutions that closely approximate the optimum. Furthermore, it's important to emphasize that the notable disparities between the MDH and MSB-PF heuristics observed in instances 5, 6, 7, 14, and 19 may not fully encapsulate the potential differences in performance, primarily due to the smaller size of these instances. In larger and more complex instances, such as those in Set 2, the performance characteristics of these heuristics could be more pronounced and informative. The current dataset, while indicative of specific trends, might not provide a complete picture of the heuristics' capabilities in handling more complex scenarios. In summary, the results presented in Table (6.15) suggest that moving from an exact Offline approach to a dynamic one leads to a loss of solution

$\rho = 0.5$			$\rho = 1$		
Inst	Δ_{MDH}	Δ_{MSB-PF}	Inst	Δ_{MDH}	Δ_{MSB-PF}
0	14.2	14.2	0	1.4	1.4
1	13.8	13.8	1	1.7	1.7
2	3.5	3.5	2	1.0	1.0
3	8.1	8.1	3	0.9	0.9
4	21.2	21.2	4	1.5	1.5
5	**13.6**	**10.6**	5	1.0	1.0
6	**9.8**	**6.4**	6	1.6	1.6
7	**35.0**	**28.9**	7	1.6	1.6
8	39.3	39.3	8	1.8	1.8
9	11.5	11.5	9	1.1	1.1
10	44.9	44.9	10	1.9	1.9
11	7.1	7.1	11	1.1	1.1
12	21.1	21.1	12	1.1	1.1
13	13.0	13.0	13	1.5	1.5
14	**11.5**	**6.6**	14	1.1	1.1
15	4.9	4.9	15	1.2	1.2
16	4.9	4.9	16	0.8	0.8
17	7.3	7.3	17	0.9	0.9
18	15.6	15.6	18	1.1	1.1
19	**19.9**	**16.2**	19	0.9	0.9

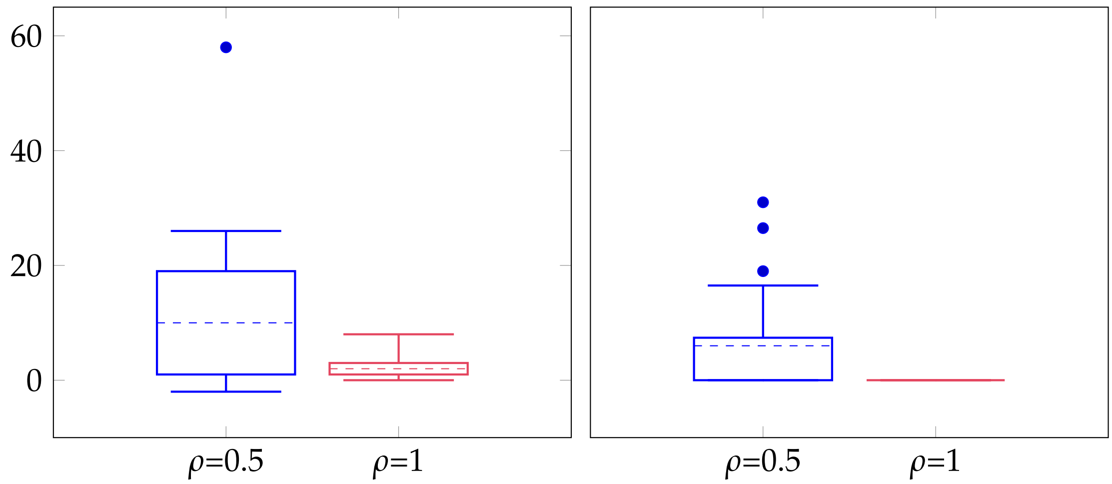
TABLE 6.15: Comparison bewteen the Offline Problem, MDH and MSB-PF on Set 1 instances for $\lambda = 100\%$.

quality already for small size instances. The impact the changing the parameter λ has on the solutions is shown in Figure (6.9) with Figure (6.9a) for the Myopic Dynamic Heurisitic results and Figure (6.9b) for the MSB-PF.

The boxplots represent the percentage improvement that the objective value has when the available initial knowledge pass from the 0% to the 100%. In Figure (6.9a), for example, we see that in the $\rho = 0.5$ case the average improvement when λ is increased is around the 10%, with upgrades in objective functions up to the 26% as represented by the higher whiskers. Confronting Figure (6.9a) and Figure (6.9b) it is evident that the scenario-based heuristic (MDB-PF) presents lower variations of the objective function when the knowledge is increased, with a mean variation of the 5,7% when $\rho = 0.5$ and arriving to no variation in the $\rho = 1$ case. The reason behind this lies in the fact that the use of simulated scenarios to include precision about future events is crucial to improve the quality of solutions during the time horizon. Moreover, both Figure (6.9a) and Figure (6.9b) show a significantly higher variation in results for the $\rho = 0.5$.

Computational Results on Set 2 instances

This section presents the results from applying MDH and MSB-PF to Set 2 instances. We conducted a two-part analysis. First, we examined the benefits of moving from a dynamic to a scenario-based approach, evaluating how future event provisions can influence the solution process. Then, we expanded



(A) MDH (B) MSB-PF

FIGURE 6.9: Distribution of improvements from $\lambda = 0\%$ to $\lambda = 100\%$

on the initial knowledge investigation that we started with Set 1 instances by introducing two additional values of λ, 20% and 50%.

Table (6.16) shows the results for different numbers of patients ($|P|$) and λ values. The columns for MDH and MSB-PF list the results for $\rho = \{0.5, 1\}$ after applying the respective heuristics. The last three columns present the percentage variation between the results of the two methods. Columns $\Delta^{0.5}$ and Δ^{1} show the variation between MDH and MSB-PF when $\rho = 0.5$ and Δ^{1}, respectively. Column Δ_{Avg} is the average of the previous two. A positive value represents improved results, indicating that MSB-PF performs better. The improvements are significantly higher for $\rho = 0.5$ with an average improvements among the all instances equal to 6.7% against the 3.2% of $\rho = 1$. In general, the insertion of scenarios lead to an average improvement of the 4.9%. It is interesting to examine the behaviour when $|P| = 50$. The disparity in variations between the lower ($\lambda = \{0, 20\}$) and higher ($\lambda = \{50, 100\}$) values of λ indicates that, when the number of patients is limited, a higher level of known information is sufficient to obtain good quality results even with a myopic approach. Conversely, for higher number of patients the improvements are evenly distributed among all the λ values with a mean variation of the 4.6% for $|P| = 100$ and 5.1% for $|P| = 200$.

A visual representation of how improvements are distributed among instances is given in Figure (6.10). Figure (6.10a), Figure (6.10b) and Figure (6.10c) present the improvements for different values of λ when $|P|$ is equal to 50,100 and 200, respectively.

Among with the change in objective function, it is interesting to look at the variation in accepted clients to analyze if the choice of the heuristic might lead to the possibility of accommodating more patients. Results are presented in Table (6.17) where, for each combination of patients |P|, ρ and

TABLE 6.16: Computational Results for Set 2 instances

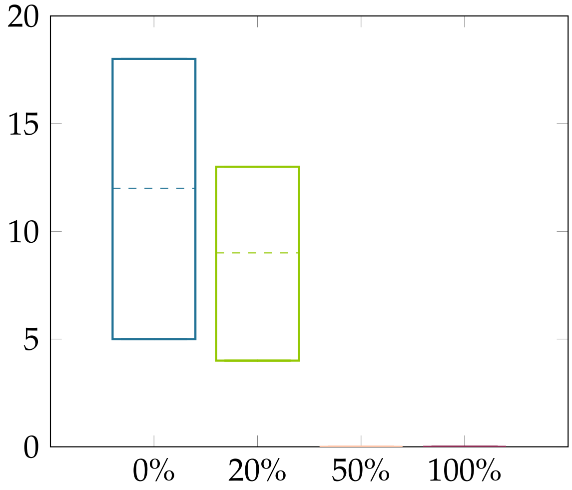
$\lvert P \rvert$	λ (%)	MDH $\rho = 0.5$	MDH $\rho = 1$	MSB-PF $\rho = 0.5$	MSB-PF $\rho = 1$	$\Delta^{0.5}$	Δ^{1}	Δ_{Avg}
50	0	2239.7	8594.3	2650.3	9005.3	18.3	4.8	11.6
	20	2334.7	8690.7	2650.3	9005.3	13.5	3.6	8.6
	50	2649.3	9004.3	2650.3	9005.3	0.0	0.0	0.0
	100	2665.6	9020.6	2665.6	9020.6	0.0	0.0	0.0
100	0	5372.0	16035.6	5696.5	16564.8	6.0	3.3	4.7
	20	5410.6	16314.2	5816.3	16561.6	7.5	1.5	4.5
	50	5629.0	16060.6	5964.5	17089.0	6.0	6.4	6.2
	100	5836.8	16537.1	6035.3	17034.2	3.4	3.0	3.2
200	0	15942.4	39328.4	17019.1	41734.6	6.8	6.1	6.4
	20	16369.1	40742.1	17312.2	42433.9	5.8	4.2	5.0
	50	16662.5	42235.4	17514.9	43926.8	5.1	4.0	4.6
	100	16622.6	43309.1	17922.7	43696.3	7.8	0.9	4.4
					Avg.	6.7	3.2	4.9

λ is indicated the number of rejected patients in the two different heuristics.

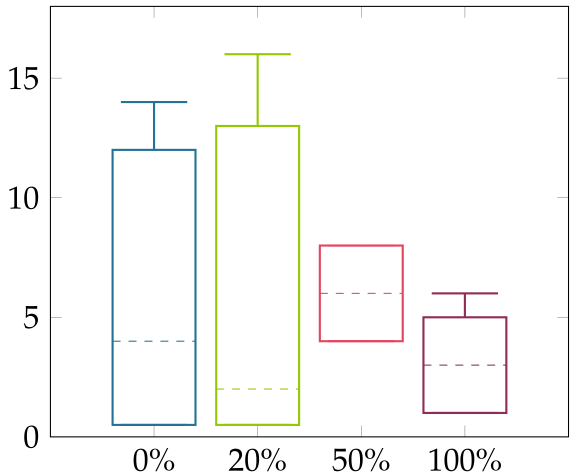
TABLE 6.17: Number of rejected patients in the two heuristics

		MDH $\rho = 0.5$	MDH $\rho = 1$	MSB-PF $\rho = 0.5$	MSB-PF $\rho = 1$
50	0	0.0	0.0	0.0	0.0
	20	0.0	0.0	0.0	0.0
	50	0.0	0.0	0.0	0.0
	100	0.0	0.0	0.0	0.0
100	0	12.0	13.0	11.5	11.5
	20	12.5	11.5	11.0	11.5
	50	12.0	13.0	11.5	10.5
	100	9.0	9.5	10.0	8.0
200	0	57.5	63.5	49.5	53.0
	20	52.5	51.5	49.0	49.5
	50	49.0	50.0	46.5	47.0
	100	47.5	47.5	45.0	47.5

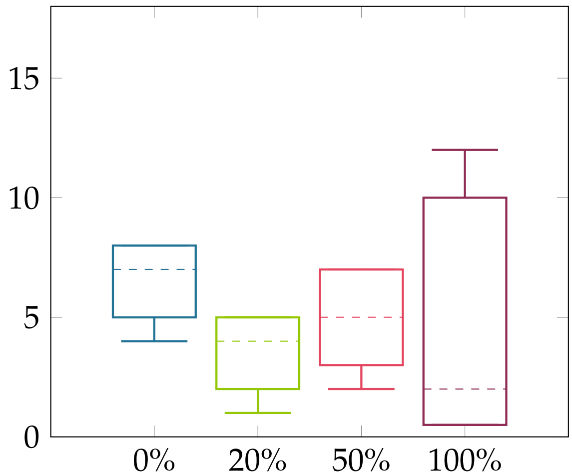
Results display how the MSB-PF heuristic effectively reduces the number of rejected patients in most scenarios. The greatest impact is obtained by increasing the number of patients with an average increment of 4 patients accepted. Specifically, with a patient set size of 200 ($\lvert P \rvert = 200$) and no prior knowledge ($\lambda = 0\%$), MSB-PF manages to serve 11 additional patients at full capacity ($\rho = 1$). Generally, it becomes clear that purely dynamic approaches, which lack foresight into future events, tend to yield inferior results in both objective function quality and the number of patients accommodated. The



(A) $|P| = 50$



(B) $|P| = 100$

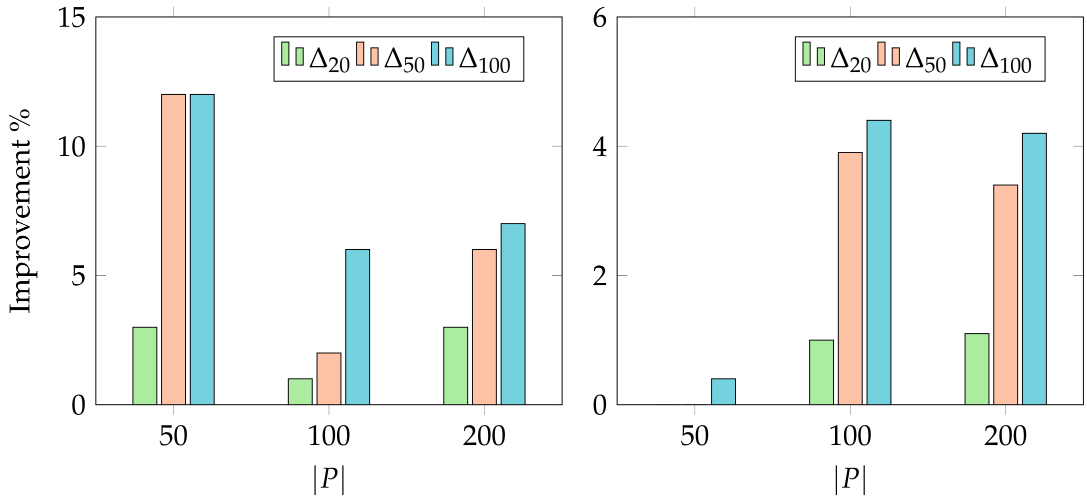


(C) $|P| = 200$

FIGURE 6.10: Percentage improvements of results from MDH to MSB-PF

MSB-PF algorithm outperforms in this regard, enabling the agency to serve a greater number of patients within its routes.

Altering the proportion of information available at the start of the time horizon, namely the patients who have already made requests to the agency, can significantly impact the performance of these heuristics. Indeed, having more certain information at the beginning leads to more effective initial patient assignments, which is crucial for optimizing the results of subsequent days by mitigating the impact of newly arriving information. This is confirmed in Figure (6.11), which depicts the percentage improvement in the objective function for various λ values compared to the baseline scenario of $\lambda = 0\%$, Δ_{50} for example indicates the percentage variation in results when $\lambda = 50\%$. Figure (6.11a) presents the results for MDH, while Figure (6.11b)



(A) MDH (B) MSB-PF

FIGURE 6.11: Percentage improvements of objective function for different λ values.

details the outcomes for MSB-PF.

In both methods, for each number of patients the improvement increases with the increasing of λ value confirming that an increment in known information is directly correlated with an improvement in the overall results, whether scenarios are included or the problem is solved in a myopic approach. In Figure (6.11a), the greatest improvement is obtained for $|P| = 50$ with Δ_{50} equal to 13%. While when the number of patients increases the MDH shows more difficulties in improving the solution even with more information available. The opposite happens in Figure (6.11b), where for $|P| = 50$ the improvement is close to 0%, suggesting that the heuristic obtains a good quality solution for even with little or no information at the beginning. Conversely, for higher values of P, it is evident the contribution of the parameter λ, with increments up to the 4.2% for $|P| = 100$. In general Δ_{50} and Δ_{100} show similar values, suggesting that passing from $\lambda = 50\%$ to $\lambda = 100\%$ does not impact significantly on the resolution.

6.6 Conclusions

In this chapter, we studied the Stochastic Multi-Period Home Healthcare Problem with Consistency Constraints (SMHHP-C) which involves the optimization of a nurse agency patients management system. In a realistic setting, patients requests are dynamically distributed over the time horizon. Under this uncertainty, objective of the SMHHP-C is to determine which patients to accept and to which nurse to assign them maximizing the total profit, computed as the revenue coming from the visits less the travelling costs. We added the service-level constraints of consistency to the model to improve the quality of care provided. We performed an experimental analysis on the role of consistency and visit flexibility inside the problem to assess the impact of relaxing these aspects on the overall solutions. We modelled and solved multiple versions of the original problem modifying the role of consistency and the temporal distribution of visits. Results show how proposing a flexible service in terms of visits days helps in improving the solution. We then approached the dynamic nature of the problem with two different heuristics: the Myopic Dynamic Heuristic and the Multi-Scenario-Based Progressive Fixing. The former solved the problem day by day without any knowledge of the future while the latter exploits the previsions coming from different scenarios to take more future-conscious decisions. Results show how implementing a scenario set is fundamental to obtaining better solutions, especially with low known information at the beginning of the time horizon.

Part II

Rich Vehicle Routing Problems in Logistic Applications

The second part explores the application of Rich Vehicle Routing Problems (RVRPs) in the Logistic sector. The scope of our study is to improve the delivery company's operational efficiency while addressing environmental concerns and adding customer-centric constraints. Initially,in Chapter (7) we present the main literature in the field of logistics application. In Chapter (8) we explore the application of collection lockers in a last-mile delivery service, offering a detailed analysis of its environmental impact. This assessment does not focus solely on the company perspective but extends to include the influence of eco-conscious consumers on the logistic setting. This chapter aims to evaluate how both companies and environmentally aware customers can collectively contribute to reshape the dynamics of delivery services, influencing the overall environmental footprint of the logistic sector. Subsequently, Chapter (9) progresses to investigate an Attended Home Delivery service with recovery options. The problem focuses on the probabilistic nature of customer presence during delivery and the implications of various recovery options in case of missed deliveries. The goal is to achieve transportation efficiency, balancing the minimization of both transportation costs and the penalties incurred with failed deliveries. This dual-focused approach offers a comprehensive understanding of the complexities and challenges inherent in attended home deliveries contributing to both the reduction of costs and the improvement of quality of service.

Chapter 7

Literature Review

Contents

This chapter presents the main literature about RVRPs in the logistic domain. In particular, we focus on works dealing with Last-Mile Delivery problems describing door-to-door and/or locker based delivery models. First, we review the Attended Home Delivery (AHD) to analyze how failed deliveries and alternative deliveries options have been treated in literature. Then, we present the recent literature on delivery problems with a special focus on the ones dealing with the environmental issues.

7.1 Logistic and Operational aspects of Attented Home Delivery

In the scientific literature, there has been a growing interest in AHD as it is one of the most applied trends in last-mile package delivery. The functionality of home delivery is crucial for the economic success of online shopping business models, hence it is fundamental to provide a cost-efficient service while, at the same time, maintaining a good quality of service to meet costumers' expectations. The diverse strategies employed by companies to meet client demands have led to a rich and varied body of literature, especially in the field of AHD. This literature spans several key areas, primarily focusing on tactical management and operational aspects. (Agatz et al., 2008) offers a thorough review of the challenges and opportunities in AHD, presenting an exhaustive perspective on the subject. This work is instrumental in understanding the breadth of issues and potential strategies within the AHD domain.

Two fundamental aspects in AHD are the tactical management of time slots selection and the addressing of failed deliveries. The length of time slots is a critical factor balancing service quality and delivery costs. Narrow time windows preferred by customers may lead to increased delivery costs. Early studies by (Punakivi and Saranen, 2001) highlighted this by showing that unattended services could be up to a third cheaper than attended deliveries with two-hour slots. Similarly, (Boyer et al., 2009) found that a three-hour delivery window is 45% more expensive than unattended delivery. (Gevaers et al., 2011) further illustrated that in low-value customer goods delivery, the cost impact of time windows could negate the benefits of online shopping. From an environmental standpoint, (Manerba et al., 2018) demonstrated how two-hour time slots could increase environmental impacts by up to 400% compared to all-day deliveries. In terms of time slot allocation, (Campbell and Savelsbergh, 2005) pioneered approaches where the service provider controls delivery acceptance and time slot assignments. (Ehmke and Campbell, 2014) expanded this analysis by incorporating variable and stochastic travel times. Cost reductions for strict delivery windows and unpredictable demand can be achieved through dynamic pricing of time slots, as suggested by (Campbell and Savelsbergh, 2005). (Yang et al., 2016) extended this concept by considering potential future demand in routing cost anticipation. (Klein et al., 2019) further explored demand management through differentiated time slot pricing, enabling cost-effective delivery scheduling at an operational level. We present the AHD Problem with Recovery option to distance our research from the existing studies. The problem focuses on dividing the workday into time slots to reflect client delivery preferences, rather than assessing cost impacts in traditional AHD service. Our analysis of varying time window lengths aims to glean managerial insights related to the costs of our recovery options, rather than comparing AHDP-RO with non-attended deliveries. Furthermore, we tackle time slot application from a tactical perspective, excluding aspects such as pricing or allocation. We do not consider request rejections, and all time slots are assumed to have uniform pricing from the client's viewpoint.

From an operational perspective, AHD can be modeled as a basic VRP, where the goal is to dispatch orders to specific locations using a fleet of vehicles while minimizing overall costs. This task becomes more complex as it often falls under the VRP with Time Windows (VRP-TW) category, where deliveries must occur within specified time frames. (Baldacci et al., 2012) offers a review of exact algorithms and model formulations relevant to VRPTW. Given the $\mathcal{NP}$-Hard nature of VRP-TW, where optimal solutions are computationally difficult to derive, researchers have developed various algorithms to approximate delivery costs. To enhance client satisfaction, many studies have introduced flexible delivery options into VRP-TW. (Moccia et al., 2012) were pioneers in formulating a VRP-TW that includes alternative delivery locations, utilizing an incremental Tabu Search approach. (Spliet and Desaulniers, 2015) introduced the Discrete Time Window Assignment VRP, which allows each customer to have a set of potential time windows, from which one must be selected. To address large instances, they developed an exact branch-and-price algorithm along with column generation heuristics. Another extension of VRP-TW is the Vehicle Routing Problem with Delivery Options (VRP-DO), where customers have preferences among various delivery options, and a certain level of client satisfaction is required. (Tilk et al., 2021) introduced a VRP-DO model where requests can be sent to alternative capacitated locations, and carriers need to choose the most effective option for each client. They also presented a new branch-price-and-cut algorithm for exact problem-solving. The most recent application in this field is by (Escudero-Santana et al., 2022), who expanded VRP-TW to include options for clients to suggest combinations of time slots and delivery locations. They also provided a comprehensive analysis of the performance of various tailored heuristics and metaheuristics.

Finally, the last core aspect of AHD is the management of failed deliveries that often results in additional operational costs and negative environmental impacts, as highlighted by (Song et al., 2009). For a detailed discussion on the environmental sustainability of B2C services, (Mangiaracina et al., 2015) provides a comprehensive literature review. Addressing the issue of failed deliveries, much of the research has focused on alternative delivery methods. Automated lockers and collection points have become popular solutions, with recent applications like Grabenschweiger et al., 2021 promoting locker stations via client discounts. An extensive survey of such delivery options, including the use of drones, is provided by Boysen et al., 2020. Innovative methods like leaving packages in customers' car trunks have been introduced by (Reyes et al., 2017) and (Ozbaygin et al., 2017). From a cost implications point of view, some researchers have started to consider the likelihood of customer availability during specific time slots. (Özarık et al., 2021) approached this by calculating the probability of customer presence based on historical delivery success rates, though they did not consider recovery options for failed deliveries. (Florio et al., 2018) offered a more detailed analysis with their *availability profiles* (AP), which predict customer presence at home throughout the day. Their model, aimed at minimizing expected costs of unsuccessful deliveries, allows for multiple visits to the same customer as a recovery strategy. Building on these APs, (Voigt et al., 2023) developed a VRP that balances transportation and failed-delivery costs, with the latter being client-specific and linked to the likelihood of customer presence. While these studies predominantly use deterministic models, a two-stage stochastic approach is explored in (Özarık et al., 2023). Here, routes and schedules are initially planned, and upon actual determination of customer presence, appropriate recovery actions are taken, with penalties applied after two failed attempts in a day. Our research contributes to this body of work by not only considering APs and the cost trade-offs but also introducing multiple recovery options for failed deliveries, with the costs varying based on the chosen policy. For instance, customers can opt for a second delivery attempt or the use of a collection point, adding a novel dimension to the management of failed deliveries in AHD.

7.2 Addressing the environmental issue

Parcel lockers, recognized in specialized literature as an effective way to simultaneously reduce carbon emissions and failed deliveries, serve as automated collection and delivery points (CDPs) where consumers can independently collect their parcels. Tilk et al., 2021 developed a Vehicle Routing Problem with Delivery Options, focusing on consumer-preferred delivery methods. Their objective was to serve all consumers while minimizing route costs, adhering to time window constraints, vehicle capacity, and minimum service levels for each consumer. Grabenschweiger et al., 2021 took a different approach by incorporating consumer satisfaction through heterogeneous locker stations in their VRP, aiming to minimize travel costs and offer compensation to motivate consumers to use lockers. However, these studies did not delve into the environmental side-effects of last-mile delivery systems. From an environmental perspective, Edwards et al., 2010 conducted a carbon audit to assess the impact of failed deliveries. They explored various failure rates in home delivery and calculated the additional carbon cost (in grams of CO_2) for each extra journey. Their comprehensive study showed that CO_2 emissions could be reduced by up to 87% when consumers are, on average, about 1.2 Km from lockers. Jiang et al., 2019 examined cost and emission reductions in a Travelling Salesman Problem that included a pickup cost based on consumer willingness to use CDPs. They found that carbon emissions could be reduced by up to 51.2% in areas where lockers are widely accepted. Schnieder et al., 2021 presented an approach that considers emissions from both the company and consumers. They analyzed the sustainability of integrating CDPs into delivery systems, showing that it heavily depends on environmentally conscious consumer choices. They predetermined the delivery method for each consumer and assumed that consumers always travel to the nearest locker by car when using CDPs. Our work expands on these analyses by offering a comprehensive impact assessment that allows consumers to make sustainable transport choices to reach a locker, regardless of distance. Furthermore, we treat the decision to open or not a locker as a variable, exploring the optimal delivery configuration to reduce overall CO_2 emissions. This approach provides a more nuanced understanding of the environmental implications of delivery choices, considering both company operations and consumer behavior.

Chapter 8

A Location-Routing Problem in Last Mile Delivery with the insertion of Parcel Lockers

Contents

The content of this chapter was presented to the 2^{nd} *IFAC Workshop on Integrated Assessment Modelling for Environmental Systems (IFAC IAMES 2022). This chapter corresponds to the conference paper "V.Bonomi, R.Mansini, R.Zanotti, Last Mile Delivery with Parcel Lockers: Evaluating the environmental impact of eco-conscious consumer behavior" appeared in IFAC-PapersOnLine 2022, 55, 72–77. (DOI: 10.1016/j.ifacol.2022.07.642)*

8.1 Introduction

During the COVID-19 pandemic, the shift towards e-commerce, especially in the business-to-consumer (B2C) sector, and home deliveries marked a significant change, effectively replacing traditional shopping venues like superstores. This adaptation to pandemic-induced lockdowns and movement restrictions signaled a new era in consumer behavior. Statistics from this period reveal a clear consumer preference for avoiding crowded places, opting for online shopping as a safer alternative. The pandemic's influence on online shopping is thoroughly examined in an insightful report released by the

United Nations Conference on Trade and Development (UNCTAD) (*COVID-19 and e-commerce: a global review* 2021). This report comprehensively analyzes data from various economies, including emerging markets and developed countries like Italy. It focuses on the patterns of e-retailers and consumers, shedding light on the significant shift in shopping habits during the pandemic. A key takeaway from the report is the marked increase in online shopping frequency, a trend that starkly contrasts with the pre-pandemic period. The substantial change in customers' behavior and their switch through e-commerce resulted in a tangible impact on urban environments. The increase in delivery vehicles, including vans and commercial trucks, has brought to the attention new environmental issues. Increased air pollution and traffic congestion highlight the adverse side effects of e-commerce growth on the life and safety of people living in urban areas. In response to the pandemic, consumer priorities have shifted significantly, with a newfound emphasis on health, safety, and sustainability. This shift has given rise to a group identified as "reimagined customers," a term coined by analysts like that at (Accenture, 2021). These customers have redefined their consumption patterns in line with these evolving values.

To address these challenges, we study a Last Mile Delivery system in which couriers must deliver packages for clients' houses. In particular, we model a Location-Routing Last Mile Delivery Problem (LR-LMDP) to assess the ecological benefits of a locker-based delivery system compared to traditional door-to-door methods. Central to this analysis is the minimization of total carbon emissions, which involves considering the travel distances of both the delivery company and the consumers, who must retrieve their parcels from the lockers. In performing this analysis, it is fundamental to consider the eco-conscious behavior of consumers, which represents a critical factor in the output of the delivery system. We model various consumer behaviors through two parameters: the maximum distance customers accept to travel to collect packages at parcel lockers and the maximum distance they are willing to travel, generating zero emissions (by foot or bicycle, for example). The computational results obtained by solving the mathematical model on a comprehensive set of instances, provide interesting insights on the problem, emphasizing how the role of consumers is one of the main drivers for controlling the environmental impact.

The chapter is organized as follows. The logistic settings of the problem are defined in Section (8.2), and the mathematical formulation is detailed in Section (8.3). Computational results and managerial insights are presented in Section (8.4), and conclusions are drawn in Section (8.5).

8.2 Problem Description

The LR-LMDP deals with the operational challenges of a delivery company that has to deploy a set of couriers to deliver packages to clients. It can be modeled as a variant of the traditional Vehicle Routing Problem where the vehicles do not have to visit all the nodes. Deliveries can happen at clients' houses or parcel lockers spread over the territory, describing a flexible system that can accommodate consumer preferences while being operationally efficient. The primary objective of the LR-LMDP is to strategically determine which locker stations to open to minimize the overall environmental impact of the delivery process. The impact is measured in terms of carbon emissions (grams per kilometer of CO_2) created by the company's couriers and the consumers who collect their parcels from the lockers. We model the LR-LMDP utilizing a complete directed graph to represent the road network with the consumers and the lockers in the nodes. Let $C = \{1, ..., n\}$ be the set of n consumers and $L = \{n+1, ..., n+m\}$ be the set of m possible lockers' stations. The problem can be defined over a complete directed graph $G = (V, A)$ representing the road network where $V = \{0\} \cup C \cup L$ is the node set with node 0 representing the depot, and $A = \{(i, j) : i, j \in V, i \neq j\}$ is the arc set. We indicate as d_{ij} and t_{ij} the non-negative distance and travel time between any two nodes $i, j \in V$, respectively. We assume that distances satisfy the triangular inequality. Each node i is associated with a service time s_i. We indicate as d_{max} the maximum distance consumers are willing to travel from home to collect their packages. To separately account for arcs traveled to consumers, we need to identify the subset of reachable lockers for each consumer $c \in C$. We define as $L^c \subseteq L$ the subset of potential parcel lockers located at a distance lower than or equal to d_{max} from consumer c. In addition, we define as d_{eco} (Eco-Green Distance) a maximum distance threshold below which we assume any consumer is willing to reach a locker by foot or by bicycle (the emissions to collect packages can be neglected). According to the current EU Regulation on CO_2 emissions, we define a constant emission factor e_v for commercial vehicles and a variable emission factor e_{cl} for private cars depending on the locations of consumer home c and the parcel locker l. In particular,

- $e_{cl} = \begin{cases} e_p, & \text{if } d_{eco} < d_{cl} \leq d_{max} \\ 0, & \text{if } d_{cl} \leq d_{eco}. \end{cases}$

to attribute emissions to a specific consumer c only when the locker is outside his d_{eco} distance. Deliveries are performed by a fleet of homogeneous vehicles $K = \{1, ..., k\}$ with nonrestrictive capacity. Also, locker stations are assumed with unbounded capacity.

8.3 Mathematical Formulation

Let us define, for each arc $(i, j) \in A$, a binary variable x_{ij} taking value one if arc (i, j) is traversed by any vehicle and 0 otherwise. Moreover, for each arc (i, j), we define a continuous variable z_{ij} indicating the time of vehicle arrival at node j when arriving from i. For each set $S \subset V$, let $\delta^+(S) = \{(i, j) \in A : i \in S, j \notin S\}$ and $\delta^-(S) = \{(i, j) \in A : i \notin S, j \in S\}$ be the set of arcs leaving and entering set S, respectively, with $\delta^+(i) = \delta^+(\{i\})$ and $\delta^-(i) = \delta^-(\{i\})$. Each locker $l \in L$ is associated with a binary variable y_l, taking value one if the locker is opened, 0 otherwise. Finally, for each customer $c \in C$ and each locker $l \in L^c$, the binary variable w_{cl} is equal to 1 when consumer c travels to locker l and zero otherwise. If this w_{cl} equals zero customer c receives the package at home. The problem aims to find the optimal subset of lockers to open to minimize the system's total carbon emissions, computed as the sum of emissions produced by the company couriers and the consumers.

The mathematical formulation is as follows:

$$\min \sum_{(i,j)\in A} e_v(d_{ij}x_{ij}) + \sum_{c\in C}\sum_{l\in L^c} e_{cl}(2d_{cl}w_{cl}) \tag{8.1}$$

subject to:

$$\sum_{(i,c)\in\delta^-(c)} x_{ic} = \sum_{(c,i)\in\delta^+(c)} x_{ci} = 1 - \sum_{l\in L^c} w_{cl} \quad c \in C \tag{8.2}$$

$$\sum_{(i,l)\in\delta^-(l)} x_{il} = \sum_{(l,i)\in\delta^+(l)} x_{li} = y_l \quad l \in L \tag{8.3}$$

$$\sum_{(0,j)\in\delta^+(0)} x_{0j} = \sum_{(j,0)\in\delta^-(0)} x_{j0} \leq |K| \tag{8.4}$$

$$w_{cl} \leq y_l \quad c \in C, l \in L^c \tag{8.5}$$

$$\sum_{(i,j)\in\delta^+(i)} z_{ij} - \sum_{(j,i)\in\delta^-(i)} z_{ji} = \sum_{(i,j)\in\delta^+(i)} (t_{ij} + t_i)x_{ij} \quad i \in L \cup C \tag{8.6}$$

$$(t_{0i} + t_{ij} + s_i)x_{ij} \leq z_{ij} \leq (T - t_{j0} - s_j)x_{ij} \quad (i, j) \in A \tag{8.7}$$

$$z_{0i} = t_{0i}x_{0i} \quad i \in L \cup C \tag{8.8}$$

$$z_{ij} \geq 0 \quad (i, j) \in A \tag{8.9}$$

$$x_{ij} \in \{0, 1\} \quad (i, j) \in A \tag{8.10}$$

$$w_{cl} \in \{0, 1\} \quad c \in C, l \in L^c \tag{8.11}$$

$$y_l \in \{0, 1\} \quad l \in L \tag{8.12}$$

The objective function (8.1) minimizes the total environmental impact computed as the total traveled distance by both vehicles and consumers multiplied by the respective emission factors. For the consumers, we calculate the distance at round-trip from their house to the locker, assuming they don't de-tour after collecting the package. Constraints (8.2) regulate the arc flow through the clients' nodes. If a client $c \in C$ is served at home ($\sum_{l\in L^c} w_{cl} = 0$), exactly one arc must enter and leaves node c. The arc flow through locker nodes is set through Constraints (8.3) where an arc traverses nodes l only if the respective locker is open ($y_l = 1$). These constraints are fundamental to impose that a courier route visits a locker only if it is open. Constraints (8.4) ensure that at most K vehicles are used. Constraints (8.5) state that a client $c \in C$ can travel to a locker $l \in L^c$ only if the latter has been previously open, hence included in a vehicle route. Constraints (8.6) determine the arrival time at two consecutive nodes, thus working as sub-tour elimination constraints. In particular, if node j is visited immediately after node i, the time elapsed between the arrival in the two nodes is equal to the time t_{ij} needed to travel between the two nodes plus the service time at note i. The lower and upper bounds of variable z_{ij} are regulated by Constraints (8.7), stating that if arc (i, j) is traversed, then the arrival time at node j must be greater than the time required to leave the depot and serve the consumer i ($t_{0i} + s_i$) and lower than the allowed tour length (T) minus the time required to serve the consumer j and return to the depot ($s_j + t_{j0}$). Constraints (8.8) ensure that the time needed to travel from the depot to any visited node i (when $x_{0i} = 1$) is equal to t_{0i}. Non-negativity and binary conditions on variables are defined in Constraints (8.9) to (8.12).

8.4 Computational Results and Managerial Insights

In this section, we present the results obtained by investigating the impact of the eco-conscious behavior of consumers. In particular, we perform a sensitivity analysis on the parameters d_{max} and d_{eco}, highlighting how consumers' daily choices deeply influence environmental issues. Section (8.4.1) describes the structure of the benchmark instances used in the study, and Section (8.4.2) proposes the computational analysis of the environmental impact of different consumer behaviors. All tests have been run on an Ubuntu 20.04.2 machine with an AMD Ryzen 9 3950x CPU, 16 cores, 32 threads, and 32 GB of RAM. Gurobi 9.1.2 has been used as a mixed integer linear programming solver. A time limit of 8 hours has been set for solving each instance.

8.4.1 Instances Generation

We have solved and tested 60 benchmark instances, differentiating the number of consumers $|C|$, the number of lockers $|L|$, and the number of vehicles $|K|$. The structure and size of the instances are detailed in Table (8.1), where column *#Inst* indicates the number of instances generated per each tuple $(|C|, |L|, |K|)$. Consumers' locations are randomly generated in a geometrical square representing a city area of 15 Km of edge. The travel speed of vehicles is assumed to be uniform and equal to 30 km/h throughout all the area. This space is divided into $|L|$ zones. Each zone is designated to have a potential locker, ensuring that every consumer is within a maximum distance of 5 km from a pick-up station when the number of lockers is set to $|L| = 5$. Additionally, the depot is strategically placed near the area's perimeter to mimic the topical location of big warehouses in industrial zones. The time to serve client and locker nodes varies between 3 and 10 minutes. The working shift of couriers is equal to 8 hours. For the environmental impact assessment, the emissions parameters are equal to $e_v = 161.2g/Km$ for commercial vehicles (European Environment Agency, 2020b) and $e_p = 127.0g/Km$ for private ones (European Environment Agency, 2020a).

$\|C\|$	$\|L\|$	$\|K\|$	$\#Inst$
50	5	2	5
		3	5
50	10	2	5
		3	5
50	15	2	5
		3	5
100	5	2	5
		3	5
100	10	2	5
		3	5
100	15	2	5
		3	5

TABLE 8.1: Structure and size of benchmark instances

We have considered 7 possible values for d_{max} and d_{eco} 0m, 500m, 750m, 1000m, 1500m, 2000m, and 5000m. Combining each value of d_{max} with all the values of d_{eco} equal or lower allowed us to have a total of 28 combinations of the pair d_{max}-d_{eco}. Of these, the 0-0 combination represents the pure home delivery scenario, in which clients are unwilling to travel to collect packages. Considering that each benchmark instance has been solved for every pair combination, we have analyzed 1680 instances in our study. Of these, 1608 have been solved to optimality within the computational time limit. The median optimality gap value is 6%; if we exclude two instances with an average gap more significant than 30%, the average gap to optimality drops to 2% for the remaining ones. Table (8.2) presents, for each pair, the average computational time (s) for 50 and 100 customers. The average percentage gaps are detailed in Table(8.3)

$\|P\|$	d_{max}	d_{eco} 0	500	750	1000	1500	2000	5000
50	0	678.0						
	500	1172.9	1104.0					
	750	1259.4	1173.8	1254.8				
	1000	1154.4	1247.2	1253.8	1014.7			
	1500	1225.7	1225.7	1068.0	1165.1	775.5		
	2000	1254.0	1142.0	1143.2	1125.4	688.3	769.2	
	5000	1529.8	1417.7	1458.5	1335.5	860.7	1179.1	5779.4
100	0	3787.3						
	500	6452.5	5327.4					
	750	6549.4	5474.2	5330.4				
	1000	6781.7	5598.8	5454.2	5214.3			
	1500	6784.5	5587.9	5530.0	5798.5	5994.2		
	2000	6957.1	5703.3	5704.7	5901.9	6031.2	6068.4	
	5000	6794.5	5742.9	5830.6	5918.0	6406.4	6260.3	7001.4

TABLE 8.2: Average computational times (s)

$\|P\|$	d_{max}	d_{eco} 0	500	750	1000	1500	2000	5000
50	0	0						
	500	0	0					
	750	0	0	0				
	1000	0	0	0	0			
	1500	0	0	0	0	0		
	2000	0	0	0	0	0	0	
	5000	0	0	0	0	0	0	12
100	0	0						
	500	1	1					
	750	1	1	1				
	1000	1	1	1	1			
	1500	0	1	1	1	2		
	2000	1	1	1	1	2	3	
	5000	1	1	0	1	2	3	26

TABLE 8.3: Average optimality gap (%)

8.4.2 Environmental Impact

In this analysis, we focus on the effects of changes in eco-conscious consumer behaviors on the last-mile delivery system, specifically examining the impact of variations in the distances consumers are willing to travel to retrieve packages and their inclination towards using eco-friendly transportation methods. Our objective is to understand how these behaviors influence the reduction of environmental emissions in the last-mile delivery process. Our study is divided into two main parts. Initially, we investigate the effects of varying the maximum distance consumers travel (d_{max}) and the ecological distance (d_{eco}), analyzing them separately to discern their individual impacts. The first phase of our computational experiments is designed to calculate the reduction in emissions resulting from alterations in d_{max} and d_{eco}. These parameters are varied one at a time, keeping the other constant, to provide a clear understanding of their individual effects on the solutions. We observe in Figure (8.1) the decrease in carbon emissions when $d_{eco} = 0$ and d_{max} is increased across different consumer and locker combinations. This reduction is compared against a baseline scenario of pure home delivery, where d_{max} - d_{eco} = 0-0. By setting the eco-distance to zero, we can assess the environmental impact when consumers do not utilize green transportation methods. Notably, the behavior of the systems with 50 consumers ($|C| = 50$) differs significantly from those with 100 consumers ($|C| = 100$). When the consumer count increases, leading to more private vehicles for package collection, the difference in carbon emissions compared to the home delivery scenario becomes negligible. A notable impact is observed for a smaller consumer base ($|C| = 50$) when d_{max} exceeds 750 meters. However, the maximum reduction achieved is only about 1.9% for the scenario with 15 lockers, 50 consumers, and d_{max} of 5000 meters, indicating that even if consumers are willing to travel up to 5 km to collect their parcels, the overall system reduction is minimal without the use of green transport methods, questioning the necessity of introducing lockers.

In contrast, Figure (8.2) depicts a different trend, showing the emission reductions compared to home delivery when $d_{max} = 5000$ and d_{eco} increases. No significant improvement is observed for d_{eco} values up to 750 meters. However, emissions are exponentially reduced for $d_{eco} \leq 1000$ meters, peaking at 68% in scenarios with 100 consumers and 15 lockers. This finding suggests that last-mile delivery systems can significantly benefit in terms of environmental impact when consumers actively reduce emissions.

Another critical aspect to consider is the proportion of consumers receiving their parcels at home as d_{max} and d_{eco} increase. Figures (8.3) and (8.4) display the number of consumers served at home for $d_{eco} = 0$ and $d_{max} = 5000m$, respectively. As shown in Figure (8.3), most clients receive home deliveries regardless of d_{max} values, with the percentage hovering around 96.2%

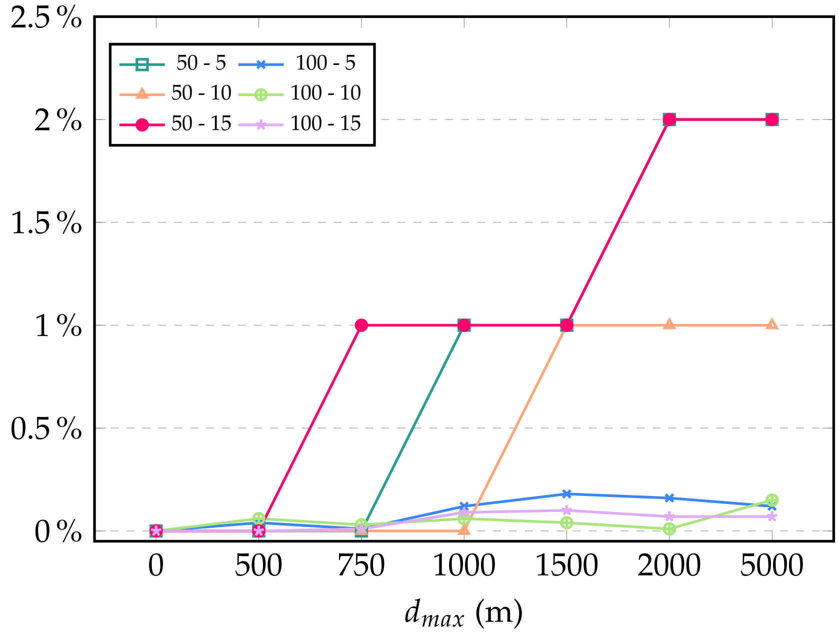


FIGURE 8.1: Decrease of emissions over d_{max} values for each combination of $|C| - |L|$

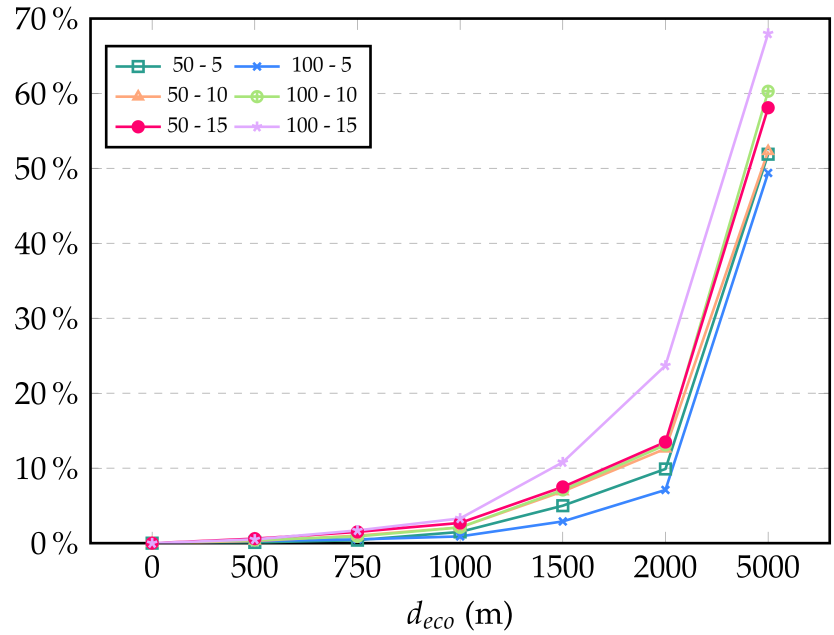


FIGURE 8.2: Decrease of emissions over d_{eco} values for each combination of $|C| - |L|$

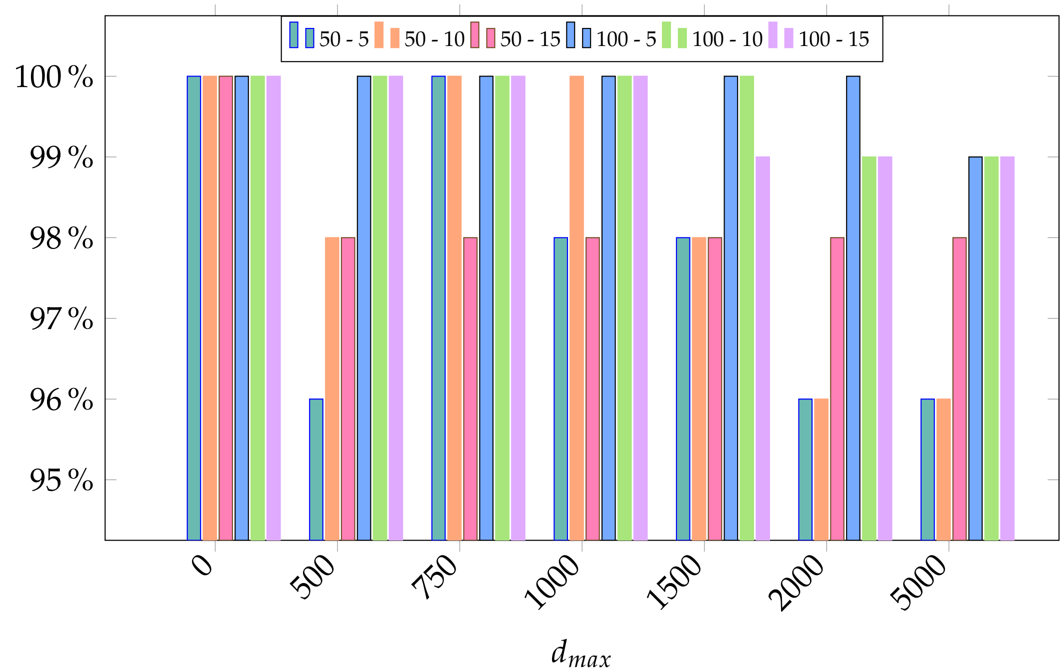


FIGURE 8.3: Consumers served at home over d_{max} values for different combinations of $|C| - |L|$

in optimal scenarios. This high percentage, constant across different d_{max} values, suggests that lockers do not significantly alter the traditional door-to-door delivery model if consumers do not adopt eco-conscious behaviors. Conversely, home deliveries decrease substantially for scenarios where d_{eco} exceeds zero. When $d_{eco} \geq 750m$ meters, the proportion of home deliveries drops exponentially, with a maximum of only 20% of clients receiving home deliveries in the 5000-5000 meter scenario (Figure (8.4)). These results indicate that consumers must be willing to travel more than 1.5 km using zero-emission transportation methods for a locker-based system to be environmentally effective.

In a second round of experiments, we simultaneously varied the parameters d_{max} and d_{eco}. Figure (8.5a) illustrates the impact of different d_{eco} values on emissions reduction, compared to the baseline scenario where d_{eco}=0. This analysis differs from that in Figure (8.2) as it encompasses all potential d_{max} values. For example, with d_{eco} = 1500m, the analysis included all combinations of d_{max}-d_{eco} such as (1500-1500), (2000-1500), and (5000-1500), calculating the emissions reduction percentage relative to the scenarios (1500-0),(2000-0), and (5000-0). This approach provided a comprehensive view of the influence a specific d_{eco} value has across all instances that incorporate it. The box-plot data indicates minimal variability in the results. When d_{eco} is $\leq$ 1000m, the range between the highest and lowest percentage reduction is approximately 5%. The most significant reduction occurs at d_{eco}=5000m, with an average decrease of 56% and a span of 26% between the lower and upper

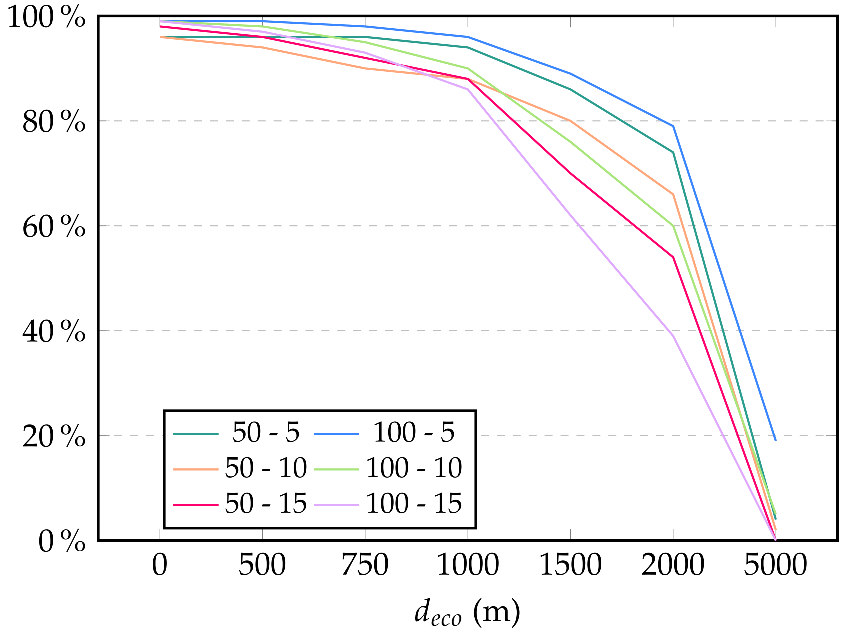
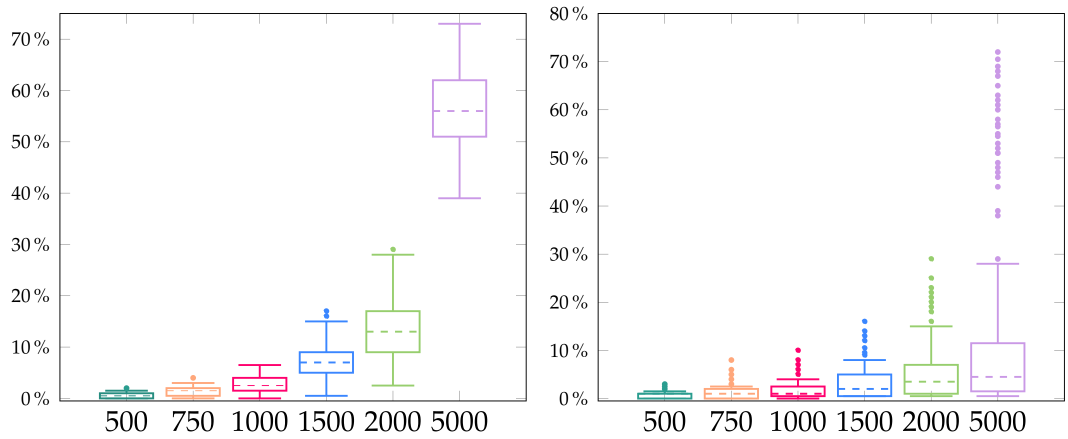


FIGURE 8.4: Consumers served at home over d_{max} values for different combinations of $|C| - |L|$

percentiles. Figure (8.5b) presents a similar analysis on emissions reduction but for different values of d_{max} in respect to the case d_{max}=0. Given a value of d_{max}, we consider all the instances with positive values of $d_{eco} <= d_{max}$. The box-plot shows a similar behavior to Figure (8.5a) with emission decreasing at the increment of the considered parameter and compact boxes representing a low variability in results. However, in Figure (8.5a), a huge numbers of outliers can be highlighted mainly in instances with $d_{max} = d_{eco}$ and for $d_{eco} \geq 1500$; the presence of outliers for higher values of d_{eco} suggests that the key parameter to the study is the eco-conscious distance of clients. In fact, looking at the median percentage values emissions reduction for different d_{max}, it is evident how they are way below the outliers, going from 0.0% when $d_{max} = 500$ to only 3.59% when $d_{max} = 5000$.

8.4.3 Managerial Insights

Purpose of our managerial insights is to have a whole picture of the impact of the consumers and the one of the company. Figure (8.6) shows how, at the increasing of d_{eco}, the total travelled distance is distributed between the company's vehicles (d_v) and the consumers, with (d_c^{eco}) and without (d_c) green means of transport for different values of $|C|$ and $|L|$. It is clear to notice that, for lower values of d_{eco} the majority of the distance is traveled by the company with values always higher than the 90% until $d_{eco} \geq 1500m$. The only exception is the combination for $d_{eco} = 1000m$ with $|C| = 100$ and $|L| = 15$ where the distance travelled by consumers is equal to 87%. In general, the



(A) d_{eco} (B) d_{max}

FIGURE 8.5: Emission reduction for different values of d_{eco} and d_{max}

contribution of d_c is always marginal confirming how the locker-based solution is efficient in reducing emissions only when they adopt an actively green behavior. A conclusion confirmed also by the trend of d_c^{eco} that for values of $d_{eco} \geq 1500m$ starts to decrease reaching values higher than the 30% in the $d_{eco} = 2000m$ case up to the 90% when the d_{eco} distance it's at its maximum. Together with the traveled distances, we are also interested in analyzing how the carbon emissions are divided between the two actors involved. Figure (8.7) presents the distribution of emissions between the company e_v and the clients e_{cl} for each combination of $|C|$ and $|L|$ when d_{max} increases. We do not include the green client since their emissions are always equal to 0. As already concluded, rarely a clients collects a package outside his/her green distance, making the consumers' emissions always lower than the 5%. A clear trend is evident only beyond 750m when $|V_C|$ is equal to 50: the percentage of emissions due to consumers increases from less than 1% to slightly above 4%. For $|V_C| = 100$ an increasing trend could emerge for higher values of d_{max} and $|V_L|$. However, since a value of $|V_L|$ equal to 15 when $|V_C| = 100$ is already realistic, we can infer that consumer emissions for high values of $|V_C|$ can be considered negligible.

Finally, in Figure (8.8) we present the percentage increase in emissions in a scenario in which clients are not included in the objective function ($e_{cl} = 0 \forall c \in C, l \in L$). This is the most common studied scenario in optimization papers when evaluating a locker-based solution of a last-mile system. Not taking consumers emissions into account, optimal solutions show that the introduction of lockers can significantly drop the total traveled distance for the company, decreasing the costs. However, the overall environmental impact can be far greater. As can be seen from the graph, for values greater than $d_{max} = 1000m$ the increase in emissions is exponential, reaching almost the 50% when $d_{max} = 2000m$ and over 250% when $d_{max} = 5000m$.

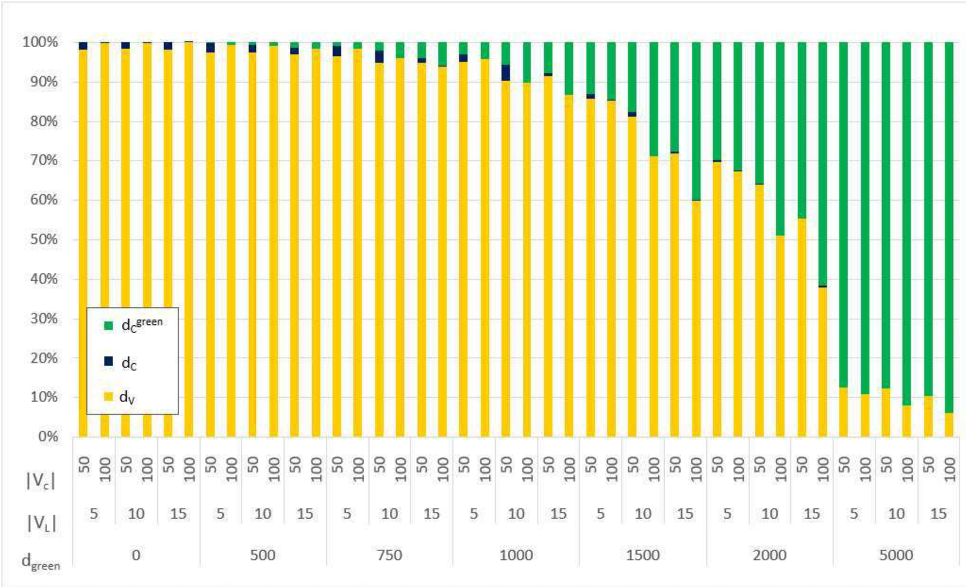


FIGURE 8.6: Travelled distances over d_{eco} values: company vs non-green consumers vs green consumers.

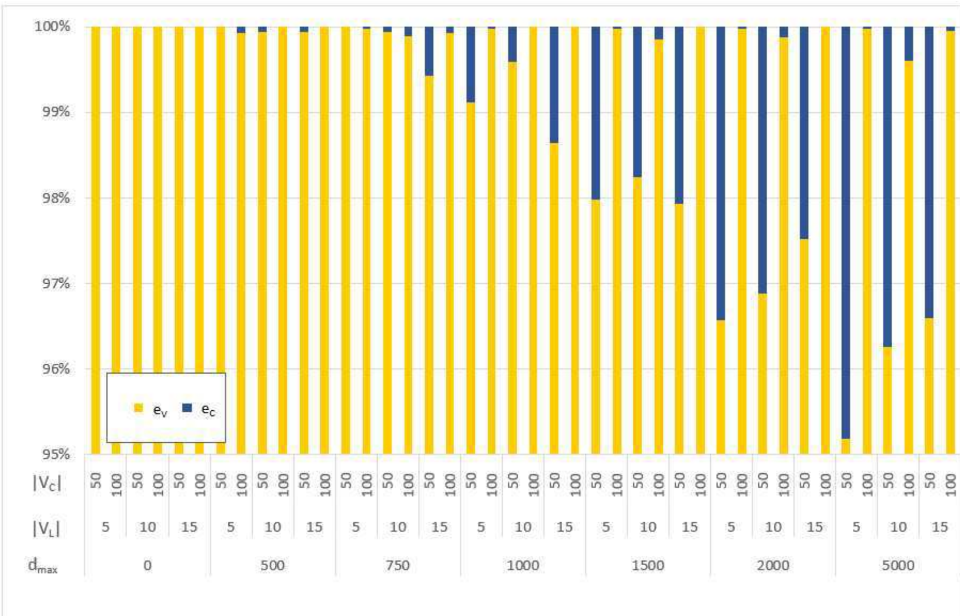


FIGURE 8.7: Emissions distribution: company vs consumers.

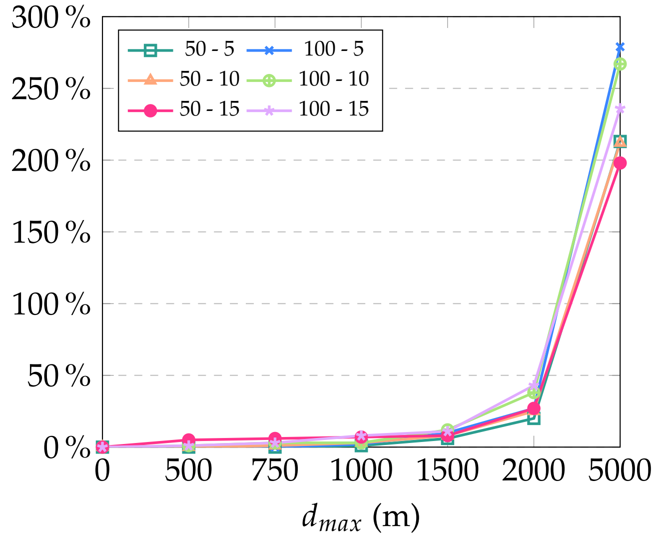


FIGURE 8.8: Emissions increase over d_{max} values when $e_{cl} = 0$, for all $c \in C$ and $l \in L$.

8.5 Conclusions

In this study, we delve into the challenging task of comparing different last-mile transportation models, particularly focusing on their environmental impacts. Our analysis primarily contrasts the traditional door-to-door delivery service, where consumers receive goods directly at their homes, with a parcel locker system, where consumers collect their parcels at a designated hub. The core of our examination is the evaluation of the CO_2 emissions associated with these two models. We propose a mathematical formulation aimed at minimizing the emissions from distances traveled both by the delivery company and consumers who opt to use locker stations. A key aspect of this model is its flexibility in determining the optimal number and locations of lockers to be operational. This decision-making process is ruled by two crucial parameters to simulate consumers behavior: the maximum distance a consumer is willing to travel to a locker using any mode of transport, and the 'green distance'—the furthest a consumer is prepared to go by foot or bicycle, thereby not generating any CO_2 emissions. The computational analysis of this model, applied to over 1600 instances, draws relevant insights on the effectiveness of the locker-based system. One of the key findings is that while the installation of lockers plays a role in emission reduction, it is not the primary factor. Instead, the pivotal element is the eco-conscious behavior of consumers. This revelation shifts the focus towards encouraging environmentally friendly practices among consumers as a more effective strategy for reducing emissions.

Chapter 9

An Attended Home Delivery Problem with recovery options and availability profiles

The content of this chapter was presented to the 1^{st} *ULTRAOPTYMAL Workshop and to the International Conference on Optimization and Decision Science (ODS 2023). This chapter corresponds to the conference paper "V.Bonomi, D.Manerba, R.Mansini, R.Zanotti, Evaluating the impact of recovery options in Attended Home Delivery with Availability profiles", to be submitted*

9.1 Introduction

The COVID-19 pandemic has significantly accelerated the growth of the online retail sector, a trend highlighted in the 2022 European E-Commerce Report. This report indicates a 6% increase in business-to-costumers sales compared to 2019, with an estimated 75% of internet users engaging in online purchases in 2022. The shift from traditional in-store shopping to online platforms, partly driven by the pandemic, has compelled more companies to venture into the market, particularly by offering home delivery services. One prominent model in e-commerce is Unattended Home Delivery, where the recipient's presence isn't required at the time of delivery. However, the recent surge in online shopping has seen a gradual shift towards AHD, especially for products like electronic groceries, high-value electronics, and large furniture items. In AHD, customers schedule a specific delivery time, aligning with their availability to receive the order. As competition intensifies in this space, e-companies are compelled to develop efficient AHD services. A critical component of this efficiency is a well-planned routing and scheduling system to minimize delivery failures. Delivery failures in AHD are typically due to the service being performed outside the customer's preferred time window. These failures are not just costly for the company and burdensome for couriers, but also negatively impact customer satisfaction. However, guaranteeing successful deliveries is challenging, considering the dynamic nature of clients' daily schedules and the operational complexities of last-mile delivery, particularly in traffic-congested urban areas. As the Attended Home Delivery sector expands, the issue of missed deliveries becomes increasingly critical, impacting customer satisfaction and operational costs for companies. To enhance service quality and logistic efficiency, it's crucial to reduce delivery failures. However, this task is non-trivial since it involves the meticulous planning of routes and schedules to face the uncertainty of the dynamic nature of clients' daily behavior. The AHD process is inherently uncertain, and providers must consider this when striving to maintain a high success rate in deliveries and, consequently, service quality. Including probabilistic elements is essential when designing an efficient and reliable AHD system. Moreover, implementing and evaluating recovery strategies for failed deliveries, which we term as *Recover Options* is crucial to the overall effectiveness of the AHD model. Our research centers on an Attended Home Delivery problem with Recovery Options (AHDP-RO). This involves analyzing customer availability during various times of the day and their preferences for handling missed deliveries, such as leaving the package at a secure location, a general collection point, or rescheduling for another delivery attempt. Each of these recovery options entails distinct costs and operational considerations. Since the penalties are probabilistic, the problem can be suitable for stochastic modeling. However, we choose to adopt a deterministic perspective to focus on the strategic and tactical modeling of the AHD settings and the implications of incorporating scenarios where deliveries fail with a certain probability, thus incurring redelivery costs based on the customer's preferred recovery option. To this end, we introduce a Mixed Integer Linear Programming (MILP) model designed to be solved using standard off-the-shelf solvers. The chapter is divided as follows. First, we present the assumptions and the elements of the problem formulation in Section (9.2). Section (9.3) presents the MILP model while Section (9.4) details the results obtained solving to optimality small-size instances. Conclusions are included in Section (9.5).

9.2 Problem Description

In the AHD framework, the operational challenge involves routing a fleet of vehicles with limited capacity to deliver parcels. These deliveries must align with the times when customers are available at home. The primary operational constraints in this context are the delivery time windows at each client's location and the maximum working time length of couriers. Differently from traditional AHD problems, where the delivery is either successful or failed, we specifically address the probability of finding the client at home. Our scope is to describe a realistic setting in which penalties for missed deliveries depend on the probability of finding the client at home as well as on the recovery option that the courier has to perform to deliver the package. We refer to this specific variant as the AHDP-RO.

A specific aspect of our formulation of the AHDP-RO is the division of the operational day into distinct time slots. This structure allows us to apply the concept of *Clients' Availability Profiles* (CAP), as delineated in the study by (Florio et al., 2018), which effectively captures different clients' daily behaviors mapping their probability at home during different times of the day. These profiles are critical in giving a realistic picture of the varied scenarios that might be encountered during delivery operations. For example, a CAP showing higher probabilities in the morning suggests the client's availability earlier in the day but not later. Following the methodology in (Florio et al., 2018), these profiles are originally represented as non-overlapping, continuous, piece-wise linear functions ranging between 0 and 1. In our adaptation, we have chosen to discretize these functions within each interval, intentionally excluding the probability of an absolute certainty (a probability of 1). The data for constructing a CAP for each client can be collected directly at the time of order placement (for instance, through an online portal) or derived from historical data. Figure (9.1) gives some examples of theoretical CAP. It is important to underline that in our implementation, the continuous functions representing the profiles as in Figure (9.1) are discretized in time intervals such that inside each interval the probability of finding the costumer is constant. Taking the example of the *Linear Dec* profile, on a scenario with four sub-intervals, a possible implementation could be:

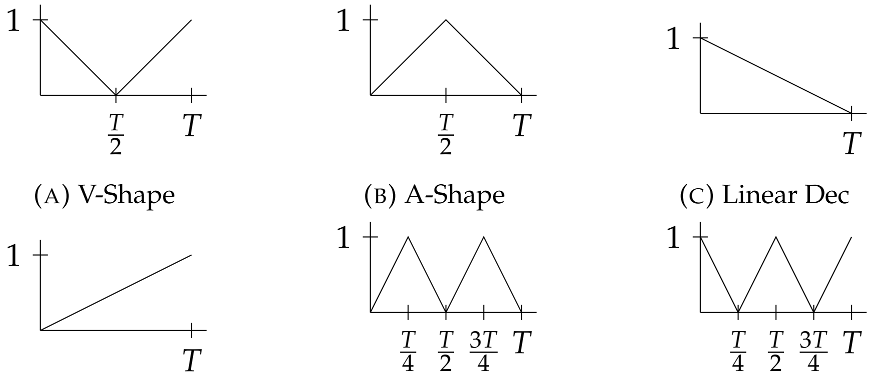
$$\text{Linear Dec} = \begin{cases} 1 & \text{if } 0 \leq t < \frac{T}{4} \\ \frac{3}{4} & \text{if } \frac{T}{4} \leq t < \frac{T}{2} \\ \frac{1}{3} & \text{if } \frac{T}{2} \leq t < \frac{3T}{4} \\ 0 & \text{if } \frac{3T}{4} \leq t \leq T \end{cases}$$

with a decreasing probability through the time horizon T.

The selection of plausible *recovery options* and the implementation of a close-to-reality cost assessment are pivotal in this problem. In our research, we consider three distinct options, each having a unique influence on the total costs:

- *Fixed Penalty* option (FP): This option allows the carrier to leave the parcel at a prearranged secure location (like a garden, garage, private locker, or with a neighbor) if the client is not at home. This choice, not requiring extra operational steps that could disrupt the vehicle schedule, incurs a fixed cost as a penalty. This cost reflects the client's dissatisfaction with receiving the parcel in their absence.

- *Second Attempt* option (SA): If the client is not found at home during the initial attempt, a second delivery on the same day is scheduled. Given



(A) V-Shape (B) A-Shape (C) Linear Dec (D) Linear Inc (E) M-Shape (F) W-Shape

FIGURE 9.1: Examples of customers availability profiles in a time horizon T as implemented by (Florio et al., 2018).

that the probability of the client being home is never 100%, this second attempt is always carried out. The penalty here is a weighted sum of the penalties for both visits, reflecting the necessity of both trips within the model framework. The rationale for using summation, rather than multiplication, is rooted in our understanding that both visits are compulsory. Multiplication would suggest the penalty of the first visit could influence the second's likelihood, which contradicts our model's assumptions. The penalty for a single visit is the round-trip cost from the client's location to the depot.

- *Collection Point* option (CP): In this scenario, if the client is absent, the package is delivered to a shared collection point, like a self-service locker or retail location. We assume there is only one such point for each client. If necessary, it is visited by a vehicle just before returning to the depot at the end of its route. The penalty is the round-trip cost from the client's location to the collection point. Similar to the SA option, we opt for a worst-case analysis to illustrate the potential high costs of selecting an inappropriate time slot.

The options discussed align well with existing company models for last-mile delivery. Nonetheless, the flexibility of both the problem and its mathematical representation allows for the integration of novel recovery strategies, along with their respective actions and associated costs. The main goal of the AHDP-RO centers on reducing the aggregate cost of delivery. This includes the complete cost associated with routing as well as any penalties resulting from undelivered items.

Let us consider a set P of clients requiring deliveries over a day. Each client $p \in P$ is identified by a unique location, a service time st_p, and a delivery demand d_p. Deliveries are performed by a set K of homogeneous vehicles with fixed capacity Q. Each vehicle starts from a common depot and must return within a predefined working time t_{max} which represents the daily planning time horizon of the service. The time horizon t_{max} is partitioned into a set $T = \{1, \dots, \tau\}$ of time windows (a_t, b_t). It is known the probability ρ_{pt} for each client p of being at home during time window $t \in T$. The clients are partitioned according to the selected recovery options (FP,SA and CP), i.e. $P = P_{FP} \cup P_{SA} \cup P_{CP}$ and $P_u \cap P_v = \emptyset \; \forall u, v \in \{FP, SA, CP\}, u \neq v$. AHDP-RO aims to find a set of routes for the vehicles and their relative visiting schedules to perform deliveries while minimizing the overall costs given by the vehicles travelling costs and the penalties associated with the recovery actions computed by considering the probability expressed by the client for each time window.

In the following, we propose a MILP formulation for the AHDP-RO. The problem can be formulated over a set N of nodes, where each node corresponds to a delivery request made by a client. For clients requiring a second delivery attempt, two nodes with the same location are assigned. For each client $p \in P_{SA}$ these are represented by the couple of nodes $N_p = \{n^1_p, n^2_p\}$. To avoid the double impact of the demand of the client on the vehicles capacity, the demand of the second node is always set equal to 0. To formulate option CP, the overall number of drop-off points is represented by the set $C = \{1, \dots, l\}$ of collection points. Each client $p \in P_{CP}$ selects a specific collecting $c_p \in C$ to which his packages need to be delivered. In order to provide a compact formulation, we create a set $N_s = \{|N| + 1, \dots, |N| + |K|\}$ of $|K|$ dummy nodes representing the same depot as a *starting* point for each one of the m vehicles and a set $N_c = \{|N| + |K| + 1, \dots, |N| + |K| + |C| \cdot |K|\}$ of $|C| \cdot |K|$ dummy nodes representing the collection points node specified by delivery option CP and here duplicated for each vehicle. Therefore, each client $p \in P_{CP}$ is associated with a collecting point c_{pk} representing the $c_p \cdot k$-th node in N_c.

The *ending* point for the vehicles is represented by the node $e = |N| + |K| + |C| \cdot |K| + 1$. The problem is defined over a directed graph $G = (V, A)$ where $V = N \cup N_s \cup N_c \cup \{e\}$ and $A = \{(i, j) : i, j \in N\} \cup \{(i, j) : i \in N_s, j \in N\} \cup \{(i, e) : i \in N\} \cup \{(i, j) : i \in N, j \in N_c\} \cup \{(i, j) : i, j \in N_c\} \cup \{(|N| + |K| + c \cdot k, e) : k \in K, c \in C\}$. For each set $S \subset V$, let $\delta^+(S) = \{(i, j) \in A : i \in S, j \notin S\}$ and $\delta^-(S) = \{(i, j) \in A : i \notin S, j \in S\}$ be the set of arcs leaving from and entering set S, respectively. In this expanded graph, the routing cycle for each vehicle $k \in K$ is represented by an oriented path starting at node $|N| + k$ and ending to node e, and visiting (if necessary) one or more collection points represented by nodes $|N| + |K| + |C| \cdot |K|$. This allows us to use two-indexed binary variables representing the selection of an arc without losing the information on which vehicle is visiting which client. The formulation involving the duplicated depots was originally proposed in (Luo et al., 2015). Finally, for each arc $(i, j) \in A$, we define as t_{ij} the positive time required to travel from node i to node j, respectively.

9.3 Mathematical Formulation

Let us define, for each arc $(i, j) \in A$, a binary variable x_{ij} taking value 1 if arc (i, j) is traversed, and 0 otherwise, and a continuous variable z_{ij} representing the arrival time at node j when coming from node i. Moreover, let us define, for each node $i \in N \cup N_s \cup N_c \cup \{e\}$, each time window $t \in T$, and each vehicle $k \in K$, a binary variable y_{it}^k taking value 1 if node i is visited in time window t by vehicle k, and 0 otherwise. Finally, for each $k \in K$, the visit at a collection point $c \in C$ is regulated by the binary variable w_{kc} taking value 1 if the node $|N| + |P| + c \cdot k$ is visited by vehicle k.

The objective function of the AHDP-RO can be formulated as the sum of four different components

$$\min \; f_{WT} + f_{FP} + f_{SA} + f_{CP} \tag{9.1}$$

where

$$f_{WT} := \sum_{(i,j)\in\delta^-(e)} z_{ij} \tag{9.2}$$

$$f_{FP} := \sum_{p\in P_{FP}} \sum_{t\in T} \sum_{k\in K} \alpha \left(1 - \rho_{pt}\right) y_{pt}^k \tag{9.3}$$

$$f_{SA} := \sum_{p\in P_{SA}} \frac{1}{2} \sum_{i\in\{n_p^1, n_p^2\}} \sum_{t\in T} \sum_{k\in K} \beta \left(1 - \rho_{it}\right) y_{it}^k \tag{9.4}$$

$$f_{CP} := \sum_{p\in P_{CP}} \sum_{t\in T} \sum_{k\in K} \gamma_c \left(1 - \rho_{pt}\right) y_{pt}^k \tag{9.5}$$

Subject to:

$$\sum_{(i,j)\in\delta^+(|N|+k)} x_{ij} \le 1 \quad k \in K \tag{9.6}$$

$$\sum_{(i,j)\in\delta^-(e)} x_{ij} = \sum_{k\in K}\sum_{(i,j)\in\delta^+(|N|+k)} x_{ij} \tag{9.7}$$

$$\sum_{(i,j)\in\delta^+(i)} x_{ij} = \sum_{(j,i)\in\delta^-(i)} x_{ji} = \sum_{t\in T}\sum_{k\in K} y^k_{it} \quad i \in N \tag{9.8}$$

$$\sum_{k\in K}\sum_{t\in T} y^k_{it} = 1 \quad i \in N \tag{9.9}$$

$$\sum_{t\in T} y^k_{n^1_p t} = \sum_{t\in T} y^k_{n^2_p t} \quad p \in P_{SA}, k \in K \tag{9.10}$$

$$\sum_{k\in K} y^k_{n^1_p t} + \sum_{k\in K} y^k_{n^2_p t} \le 1 \quad p \in P_{SA}, t \in T \tag{9.11}$$

$$\sum_{(i,j)\in\delta^+(|N|+|K|+k\cdot c)} x_{ij} = \sum_{(j,i)\in\delta^-(|N|+|K|+c\cdot k)} x_{ji} = w_{kc} \quad k \in K, c \in C \tag{9.12}$$

$$\sum_{t\in T}\sum_{i\in N} d_i y^k_{it} \le Q \quad k \in K \tag{9.13}$$

$$\sum_{t\in T} y^k_{|N|+|K|+k\cdot c,t} = w_{kc} \quad k \in K, c \in C \tag{9.14}$$

$$y^k_{pt} \le w_{kc_p} \quad p \in P_{CP}, t \in T, k \in K, c \in C \tag{9.15}$$

$$\sum_{t\in T}\sum_{k\in K} a_t y^k_{it} \le \sum_{(j,i)\in\delta^-(i)} z_{ji} \le \sum_{t\in T}\sum_{k\in K} b_t y^k_{it} \quad i \in N \tag{9.16}$$

$$\sum_{(i,j)\in\delta^+(i)} z_{ij} - \sum_{(j,i)\in\delta^-(i)} z_{ji} \le \sum_{(i,j)\in\delta^+(i)} (t_{ij} + st_h)x_{ij} \quad i \in N \cup C \tag{9.17}$$

$$(t_{|N|+1,i} + t_{ij} + st_i)x_{ij} \le z_{ij} \le (t_{max} - t_{j,e} - st_j)x_{ij} \tag{9.18}$$

$$\sum_{t\in T} y^k_{jt} \ge \sum_{t\in T} y^k_{it} + x_{ij} - 1 \quad (i,j) \in A, k \in K \tag{9.19}$$

$$(i,j) \in A \backslash \{|N|+k, e\}, k \in K$$

$$z_{|N|+k,i} = t_{|N|+k,i} x_{|N|+k,i} \quad i \in N, k \in K \tag{9.20}$$

$$y^k_{|N|+k,0} = \sum_{t\in T} y^k_{e,t} = \sum_{(i,j)\in\delta^+(|N|+k)} x_{ij} \quad k \in K \tag{9.21}$$

$$z_{ij} \ge 0 \quad (i,j) \in A \tag{9.22}$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in A \tag{9.23}$$

$$y^k_{it} \in \{0,1\} \quad i \in N \cup N_s \cup N_c \cup \{e\}, t \in T, k \in K \tag{9.24}$$

$$w_{kc} \in \{0,1\} \quad k \in K, c \in C \tag{9.25}$$

The objective function is divided into four different parts, one for the travelling times and three for the recovery-related penalty. Expression (9.2) represents the sum of arrival times of each vehicle k to the ending depot. Expressions (9.3), (9.4) and (9.5) describe the penalties related to the recovery options FP, SA and CP, respectively. The penalties are related to the probability of not finding a client at home $(1 - \rho_{it})$ and to the cost dependent by the selected recovery option, indicated as α, β or γ_c. Constraints (9.6) – (9.12) regulate the flow on the arcs. In particular, constraints (9.6) and (9.7) impose that from each starting nodes will leave at most one vehicle $k \in K$ that will return to the common ending depot e. Constraints (9.8) force the visit of all the clients, activating exactly one arc entering and exiting each node $i \in N$. Constraints (9.9) impose that the visit is performed by only one vehicle k in a specific time slot t. For clients included in P_{SA} that require a double visit, constraints (9.10) connect the two visits n_p^1 and n_p^2 to the same vehicle k while constraints (9.11) impose that they happen in two different time slots. Constraints (9.12) are collection point flow constraints that force a vehicle $k \in K$ to visit the respective node $|N| + |K| + c \cdot k$ only if it is activated ($w_{kc} = 1$).

The respect of vehicles' capacity is guaranteed by constraint (9.13) where it is imposed that the sum of the clients' demands in a vehicle route does not exceed its capacity. We remind that in the case of nodes $\{n_p^1, n_p^2\}$ belonging to a client $p \in P_{SA}$, $d_{n_p^2}$ is always set equal to 0. Constraints (9.14) and constraints (9.15) regulate the assignment of a visit in a specific time slot. In particular, Constraints (9.14) ensure that the visit of a vehicle $k \in K$ to its collection point $|N| + |K| + c \cdot k$ can occur only if the node is activated. Constraints (9.15) impose the activation of the collection point c_p for the vehicle k ($w_{kc_p} = 1$) if the customers $p \in P_{CP}$ is assigned to the vehicle (y_{pt}^k) in any given time slot t. Constraints (9.16) – (9.18) regulate the arrival time at client nodes. In particular, constraints (9.16) establish that if node $i \in N$ is visited by a vehicle in time slot t the arrival time has to occur inside the corresponding time window $[a_t, b_t]$. The arrival time between two consecutive nodes is regulated in constraints (9.17) ensuring that, if a vehicle visits node j immediately after node i, the time elapsed between the two arrival times is equal to the service time st_i at client i plus the time t_{ij} to travel from i to j. Finally, constraint (9.18) set upper and lower bounds on the duration of each route. Since all the starting depots have the same location, using only the node $|N| + 1$ as starting point for all the vehicles allows us not to duplicate the constraint for each $k \in K$. Constraints (9.19) to constraints (9.21) are logical and linking. In particular, constraints (9.19) guarantee that two consecutive nodes are visited by the same vehicle. The constraints set the variable $x_{ij} = 0$, when only one of nodes i and j is assigned vehicle k. Constraints (9.20) initialize the arrival time at the first node after the starting depot for each vehicle $k \in K$, while constraints (9.21) ensure that the final node e is visited by vehicle $k \in K$ only if it left the starting depot. Finally, constraints (9.22) – (9.25) state binary and non-negative conditions on the variables.

The model formulation in (9.2) – (9.25) is enforced by adding the following more general type of connectivity constraints:

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \geq \sum_{t\in T} y^k_{ht} \quad S \subseteq N, |S| \geq 2, h \in S, k \in K \tag{9.26}$$

Constraints (9.26) strengthened the formulation preventing the generation of sub-tours as in (9.18) imposing that for each subset of nodes $S \subseteq N$ assigned to the same vehicle k ($\sum_{t\in T} y_{htk} = 1$), there is at least one arc $(i, j) \in \delta^+(S)$ connecting the nodes in S with the ones outside S.

9.4 Economic Analysis and Managerial Insights

This section presents an exhaustive economic analysis on the impact of recovery options and penalties in an AHD service. Our objective is to measure the benefits coming in assessing the probability of having failed deliveries in the planning of operational activities of a delivery company. First, we present the benchmark instances used to perform the analysis in Section (9.4.1). Then, we divide our economical analysis in two parts with Section (9.4.2) focused on the cost analysis and Section (9.4.3) centered on the analysis of probabilities and time slots impact. Section (9.4.2) explores, through a two-level optimization, the cost variation and the trade-off coming from prioritizing some elements of the objective function instead of others. The results coming from analyzing each part individually can be beneficial to the decision maker in choosing the more suitable service strategy, knowing the impact on costs of, for example, prioritizing the reduction of penalties over the travelling costs. Section (9.4.3) presents results related to *hit rate*,i.e. the probability of having a successful delivery. In both sections is included an analysis performed on instances with tripled penalties useful to simulate a scenario in which penalties have a larger impact on the costs.

9.4.1 Instances Generation

We have solved and evaluated the problem on a set of 10 benchmark instances characterized by the same number of clients $|P| = 10$ and vehicles $|K| = 4$. In each instance, the recovery options are uniformly distributed among clients with approximately one-third of clients per option. Clients locations are uniformly and randomly generated in a 20x20 km area with a vehicles speed limit constant and equal to 40 km/h. To each client $p \in P$ is associated a service time st_p which is equal to 7 minutes and a demand d_p randomly generated in an interval of $[5, 30]$ units. The subset of collection points $c_p \subseteq C$ that can be selected by client $p \in P_{CP}$ include only the collection point at a maximum travel time of 15 minutes from the client location.

Vehicles capacity is constant among vehicles and equal to $\frac{3\sum_{i\in N} d_i}{|K|}$. The working shift of couriers t_{max} is equal to 6 hours divided into $|T| = \{4, 8\}$ time slots. We implemented 7 different CAPs describing the clients daily behavior, 6 inspired from the ones presented by (Florio et al., 2018) and one random profile. Table (9.1) shows the probabilities of each CAP for $|T| = 4$ and $|T| = 8$ time slots. Recovery options penalties α, β and γ are time dependent and computed as follows:

- $\alpha = st_i \quad \forall p \in P_{FP}$, the fixed penalty is set equal to the service time of each client. In this way, the penalty is not location-dependent and do not requires additional operational or routing activities, symbolizing only the dissatisfaction of the client for not being at home during the delivery.

- $\beta = 2t_{|N|+1,i} \quad \forall p \in P_{SA}$, the penalty is computed as round trip from the depot to the client to accounts the worse-case additional routing necessary to perform the second attempt in case of failed delivery.

- $\gamma_c = 2t_{i,|N|+|K|+c_i \cdot k} \quad \forall p \in P_{CP}$, the penalty is computed as round trip from the client location to the selected collection point to symbolize the cost of the additional routing that the vehicle has to perform to leave the package in a collection point.

The 10 instances have been solved for both values of $|T|$, generating a total of 20 benchmark instances.

	APs	%(t)
$\|T\| = 4$	V-Shape	90 10 10 90
	A-Shape	10 90 90 10
	Linear Dec	90 70 40 10
	Linear Inc	10 40 70 90
	M-Shape	10 90 10 90
	W-Shape	90 10 90 10
$\|T\| = 8$	V-Shape	90 70 30 10 10 30 70 90
	A-Shape	10 40 70 80 80 70 40 10
	Linear Dec	90 90 80 70 60 50 30 10
	Linear Inc	10 30 50 60 70 80 90 90
	M-Shape	10 50 90 50 10 70 90 70
	W-Shape	90 50 10 50 90 70 10 90

TABLE 9.1: Availability Profiles configurations for the $|T| = 4$ and the $|T| = 8$ time slots.

9.4.2 Sensitivity Analysis on Objective Function Components

We conducted a two-level optimization to evaluate how the total costs are influenced by the four components of the objective function. In this process, we solve five different versions of our standard model $AHDP - RO$ prioritizing different elements in each through a two-step optimization. In the first step of this approach, we focused on minimizing only one part of the objective function while keeping the others free. Then, in the second step, we solved the overall problem, including all objective function components, with the constraint that the prioritized component from the first step should not exceed its minimized value. Let us indicate as $AHDP - RO|f^I$ the problem version in which the function f^I is prioritized in the first step and as $z^*_{f^I}$ the optimal value obtained. The second step solves the $AHDP - RO$ standard model with the additional constraint $f^I \leq z^*_{f^I}$ that bounds the value of f^I. Table (9.2) summarizes the function f^I optimized in the first step for each version.

Version	f^I
$AHDP - RO$	$f_{WT} + f_{FP} + f_{SA} + f_{CP}$
$AHDP - RO\|HR$	$f_{FP} + f_{SA} + f_{CP}$
$AHDP - RO\|f_{WT}$	f_{WT}
$AHDP - RO\|f_{FP}$	f_{FP}
$AHDP - RO\|f_{SA}$	f_{SA}
$AHDP - RO\|f_{CP}$	f_{CP}

TABLE 9.2: Prioritized functions in each version.

Except from the standard model $AHDP - RO$, we solved other 5 versions. In particular, $AHDP - RO|HR$ prioritizes the minimization of the three penalties all together that is the equivalent of maximizing the hit rate of the service. Minimizing the penalties without including the times represents a business strategy in which the company is more client-oriented. This can help the decision-maker assess the trade-off that a fully customer oriented model would have on the total costs. On the contrary, $AHDP - RO|f_{WT}$ represents a cost-oriented model that does not include the penalties of missed deliveries. It represents the extreme situation in which the company is not interested in the success of its deliveries but only in optimizing the operational costs. These two versions represent two opposite business strategies that, if confronted, give the trade-off in including penalties in the study. $AHDP - RO|f_{FP}$, $AHDP - RO|f_{SA}$ and $AHDP - RO|f_{CP}$ have been solved, prioritizing one recovery at a time to identify the minimum cost that can be obtained in each. Table (9.3) shows the numerical results obtained divided per version and value of $|T|$. Column f_{tot} indicates the value of the total objective function while the four subsequent columns divides this value into the four components. The last fours columns indicate the percentage of the total objective function that can be attributed to the respective component. For example, for $AHDP - RO$ and $|T| = 4$, the overall costs are equal to

31015 with the traveling costs f_{WT} equal to 26404 that corresponds to an 85% of the total value.

	$\|T\|$	f_{tot}	f_{WT}	f_{FP}	f_{SA}	f_{CP}	$f_{TW}\%$	$f_{FP}\%$	$f_{SA}\%$	$f_{CP}\%$
$AHDP-RO$	4	31015	26404	644	3478	489	85	2	11	2
	8	30259	25588	819	3150	702	85	3	10	2
$\|HR$	4	41982	39756	287	1725	214	95	1	4	1
	8	45140	42540	420	1906	274	94	1	4	1
$\|f_{WT}$	4	32197	25626	1372	4225	974	80	4	13	3
	8	31808	24973	1344	4617	874	79	4	15	3
$\|f_{FP}$	4	33734	29585	273	3269	607	88	1	10	2
	8	34033	29862	350	3182	639	88	1	9	2
$\|f_{SA}$	4	35393	32544	658	1715	476	92	2	5	1
	8	36751	33341	805	1892	713	91	2	5	2
$\|f_{CP}$	4	32982	28884	777	3121	200	88	2	9	1
	8	32861	28779	686	3129	267	88	2	10	1

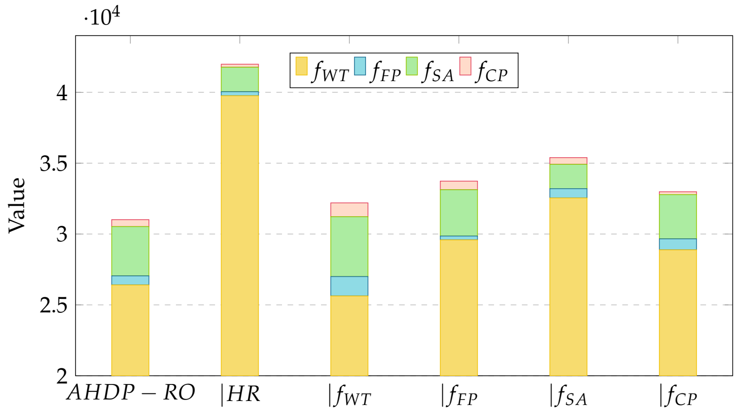
TABLE 9.3: Results of problem versions divided into objective function components

Figure (9.2) summarizes the results of the two-level optimization analysis.

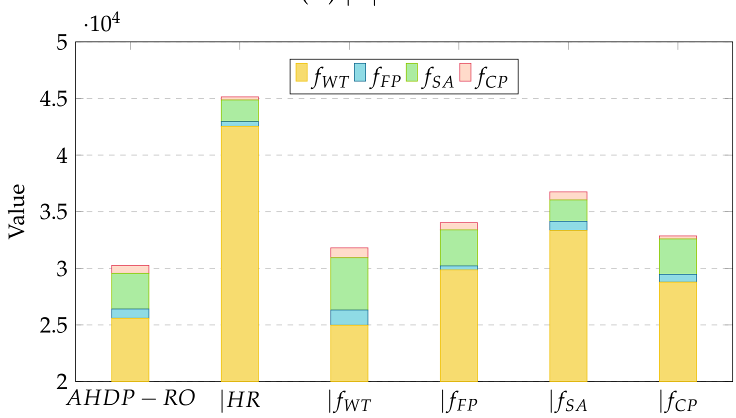
The height of each column represents the average value of the total objective function divided into the four components by different colors. Figure (9.2a) presents the results for instances with $|T| = 4$ and Figure (9.2b) for the ones with $|T| = 8$. In both figures is maintained the proportion of costs among the versions. As expected, $AHDP - RO$ presents the minimum costs since it optimizes all the functions simultaneously with an average objective function of 29863 and 30013 for $|T| = 4$ and $|T| = 8$, respectively. When costs are ignored the costs increases exponentially. $AHDP - RO|HR$ results in being the most expensive options with a total objective function of 62559 for the $|T| = 4$ case and 66323 for the $|T| = 8$ with an average increase of the 115%. This is justified by the fact that f_{WT} has the highest impact on costs with an incidence of the 85% in the $AHDP - RO$ case. Among the recovery options, the most impacting is SA. This is justified by nature of the penalty of this option since it is computed as average on two visits, increasing the probability of visiting clients in a time slot in which they are not at home. The difference in costs between $AHDP - RO$ and $AHDP - RO|f_{WT}$ is minimum with an average increase of the 3%. This suggests that, with this problem structure, prioritizing the travelling costs over penalties do not worsen the objective function significantly. To deepen the analysis on the trade-off between travelling costs and penalties we solved the problem changing the proportion of penalties, multiplying the parameters α, β and γ_c by a factor 3.

Tripled penalties: cost analysis

To delve deeper into the influence of penalties on overall work, we conducted additional model runs with a higher proportion of penalties incorporated



(A) $|T| = 4$

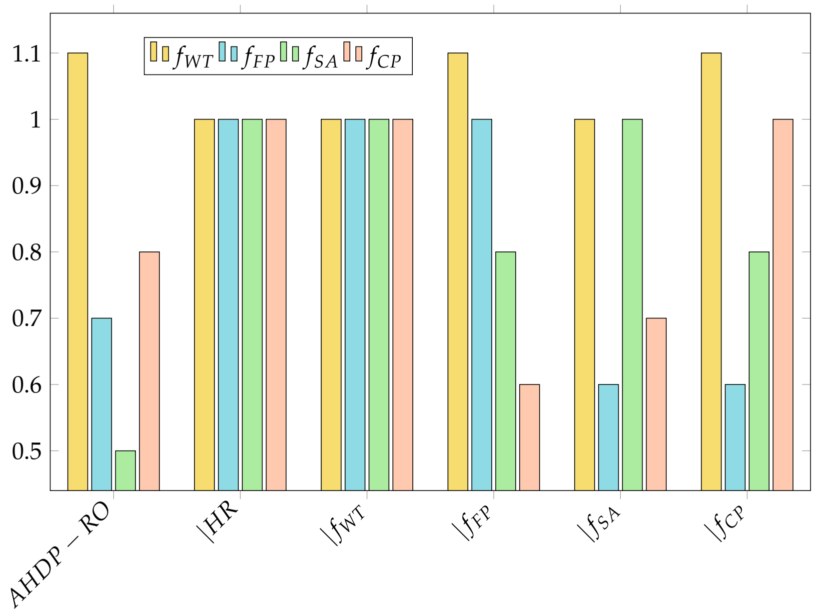


(B) $|T| = 8$

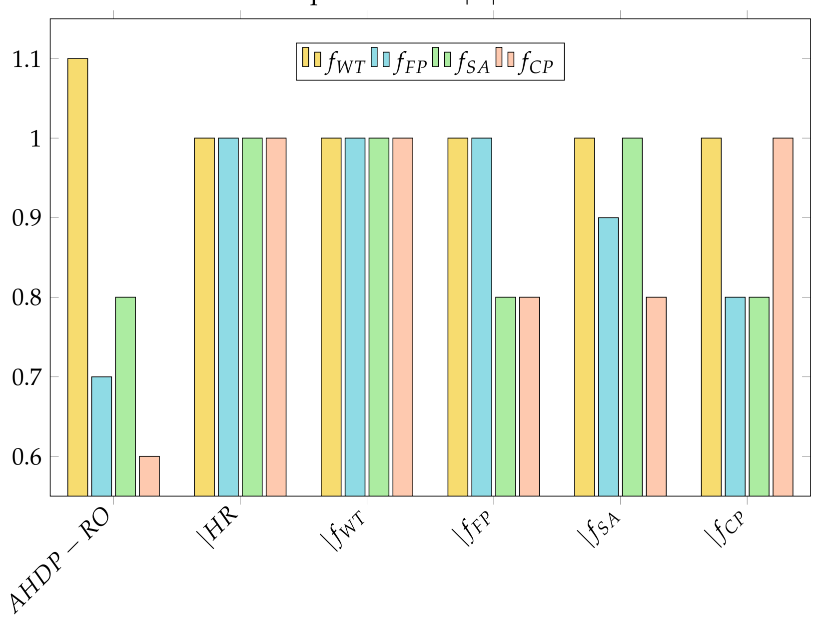
FIGURE 9.2: Objective function division for different values of T

into the objective function. In particular, we multiplied all the penalties by a factor of 3 without changing any other aspect of the problem. Results are presented in Figure (9.3a) and Figure (9.3b) for scenarios with $|T| = 4$ and $|T| = 8$, respectively. Each column's height is determined using the formula $\frac{f^3}{3f}$, where f^3 is the value derived when penalties are multiplied by 3, and f represents the value from the original model. In this way, we can assess the variation in costs between the function components, isolating the increase that comes from the increment of penalties from an actual change in the solution structure. For instance, a column of height 1, i.e. for which $f^3 = 3f$, indicates that the structure of the solution obtained in f^3 is equal to f with the difference in magnitude of penalties. The solutions of f^3 is proportionally better when the height is less than 1 and worse when the column is higher.

Examining the results of $AHDP - RO$ it can be derived how, for both values of T, an increase in penalties leads to solutions in which it is improved the optimization of the penalties in spite of travelling times. Specifically, under the scenario of $|T| = 4$, increasing penalties significantly enhances the effectiveness of recovery option SA. This is illustrated by a column f_{SA} height 0.5, signifying that costs in f^3 are six times less than in f. Exploring both $|T| = 4$ and $|T| = 8$ in $AHDP - RO|HR$ and $AHDP - RO|f_{WT}$ it is evident how the penalty proportion increase has no influence on the overall solution, with all the columns of height equal to 1. This highlights the fact that both the options are penalty-independent. The $AHDP - RO|HR$ in maximizing the hit rate chooses the best time slots in terms of probability for each client, ignoring the travel times but respecting the constraint of the maximum time shift. Multiplying the penalty of a factor three does not impact on the selection of these time slots but just increases proportionally the total penalties. In contrast, $AHDP - RO|f_{WT}$, adopts the opposite tactic, disregarding penalties and time slots initially to concentrate solely on minimizing the total travel times. This approach results in identical optimal routes across both penalty proportions, revealing a proportional relationship in costs, uninfluenced by penalty variations. This trend extends to other versions where specific functions are prioritized. In each case, the corresponding column consistently reaches a height of 1, indicating that the initial choice of time slots remains unaffected by escalated penalties. However, the role of f_{WT} evolves with variations in T. For instance, Figure (9.3a) reveals that at $|T| = 4$, in both $AHDP_RO|f_{FP}$ and $AHDP_RO|f_{CP}$ the ratio between f and f^3 is around 1.1. This suggests that increasing penalties slightly diminishes the efficiency of travel time optimization in these scenarios. Yet, this trend does not hold for $|T| = 8$, where the columns reach a uniform height of 1, indicating improved routing optimization capability at higher time slot numbers, even with increased penalties. However, it is important to highlight how the deterioration of f_{WT} never exceeds a ratio of 1.1. This finding implies that prioritizing



(A) Comparison of objective function components when penalties are tripled versus standard penalties for $|T| = 4$



(B) Comparison of objective function components when penalties are tripled versus standard penalties for $|T| = 8$

penalties does not correspondingly escalate travel times. This insights suggests that, even in contexts where clients are more impacting than routing operations, penalties can be effectively optimized with only a marginal increase in routing costs.

9.4.3 Hit-Rate Analysis

In the context of AHD, the hit rate indicator is as crucial as the minimization of costs. The hit rate is a performance metric employed to determined the frequency of successful deliveries and having a high value corresponds to an higher client satisfaction. In this section, we present the economical analysis of the $AHDP - RO$ regarding the hit rates of each recovery option. To this purpose we computed the average hit rate amongst instances for each recovery option, calculated as average probability of finding clients at home. This value is then compared with the theoretical best, worst and average rates that could be obtained from the data available. In particular, the best and worst case are obtained visiting all the clients in their best and worst time slots, respectively. Table (9.4) shows the results of the hit rate (HR) for each recovery option and different values of T as well as the best, average and worst values.

	$\lvert T\rvert$	HR	HR_{worst}	HR_{avg}	HR_{best}
FP	4	73.0	10.0	51.4	89.5
	8	69.9	10.3	54.7	88.3
SA	4	67.2	12.3	51.8	87.9
	8	69.3	9.8	54.2	87.0
CP	4	67.4	10.0	51.4	90.0
	8	60.3	9.7	55.0	87.5

TABLE 9.4: Hit rates values for each recovery option and different values of T

Figure (9.4) illustrates the difference in value that the obtained hit rate has from these three values. An upward bar indicates a positive difference, hence an obtained hit rate higher than the one it is compared to.

For instance, looking at the FP option for $|T| = 4$, the obtained hit rate is the 62% higher than the worst-case (FP_{worst}), 21% higher than the expected average (FP_{avg}) and 18% lower than the best case scenario (FP_{best}). In general, it is positive that in all the three recovery options the obtained hit rate is higher than the expected average. This indicates that including the minimization of penalties in the optimization process improves the results that could be obtained only averaging the hit rates. It is also promising the fact that our hit rates are closer to the best case than the worst. However, hit rates are not the only indicator that we can obtained to measure the client-oriented delivery efficiency. In fact, the average hit rate does not give any indication of the quality of the visited time slots for individual clients. Therefore, we

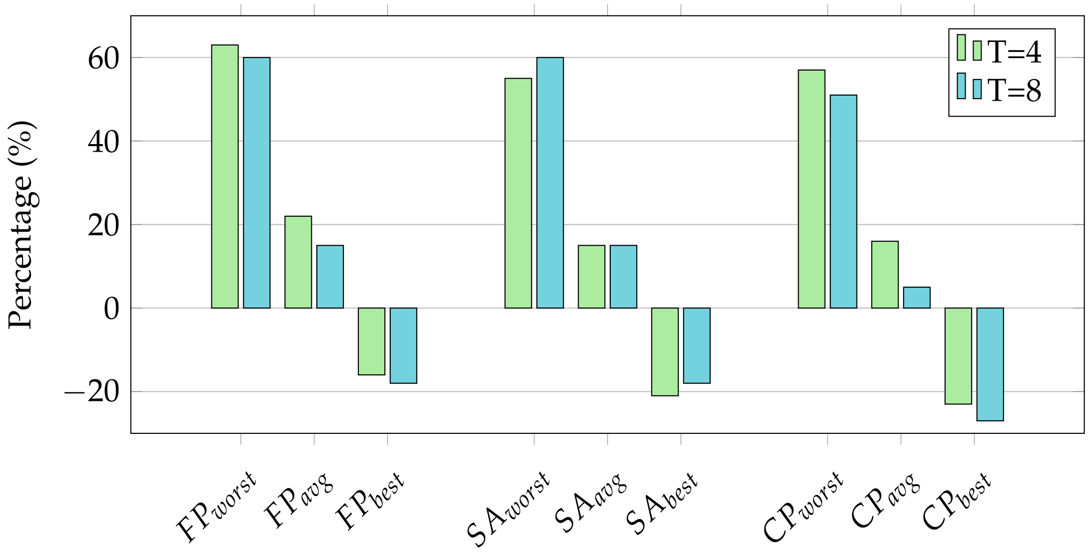


FIGURE 9.4: Comparison of Hit-Rate percentages with the max, min, and mean Hit-Rate values

computed the average *score* for each recovery option. This score is obtained arranging clients' time slots in descending orders of probability and assigning to each a numeric value ranging from $|T|$ to 1 with 1 representing the worst time slot. Figure (9.5) depicts the average scores of each recovery option obtained normalizing the scores of each client. A higher score implies that clients have been visited in the higher probability time slots. The best score belongs to the option FP went $|T| = 4$ while SA presents always the lowest one. This means that the model promotes visits in higher probability time slots for clients with less expensive options. The low value of SA it is also justified by the double visits required by the option that lowers the average score of the client.

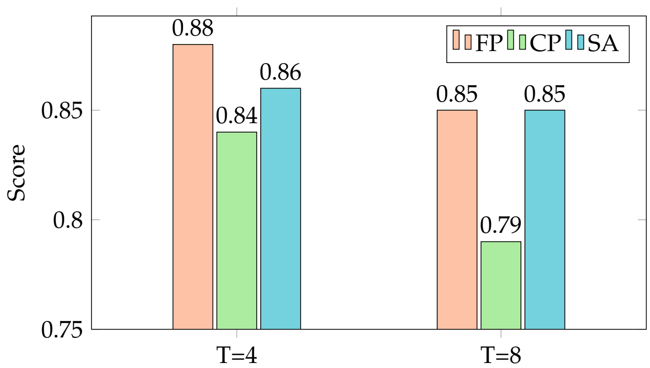


FIGURE 9.5: Quality of the visited time slots

Tripled penalties: Hit-Rate analysis

As in section (9.4.2), we performed a sensitivity analysis on the magnitude of penalties. As shown in Section (9.4.2), an increase in penalties leads to a better optimization of clients penalties with a low increase in travelling costs. Here, Figure (9.6) and Figure (9.7) present the same analysis performed in Figure (9.4) and Figure (9.5). In general, both the hit rates and the clients' time slots scores significantly improve. Figure (9.6) shows a significant increase in the obtained Hit-Rate in all the options. FP_{worst}, for example, has a value of 71.3%, showing an increase of the 15% compared to Figure (9.4). For clarity, Table (9.5) shows the hit rates (HR) of each option and the value of the worst, average and best possible hit rates of the instance.

	$\|T\|$	HR	HR_{worst}	HR_{avg}	HR_{best}
FP	4	81.3	10.0	51.4	89.5
	8	75.7	10.3	54.7	88.3
SA	4	80.9	12.3	51.8	87.9
	8	76.1	9.8	54.2	87.0
CP	4	74.2	10.0	51.4	90.0
	8	75.7	9.7	55.0	87.5

TABLE 9.5: Hit rates values for each recovery option and different values of T

It is interesting to point out how, in Figure (9.7), SA passes from being the worse option to the best when penalties are tripled. This is justified by the fact that SA is the most expensive option, hence the model tries to mitigate the costs of these clients assigning to them better probability time slots.

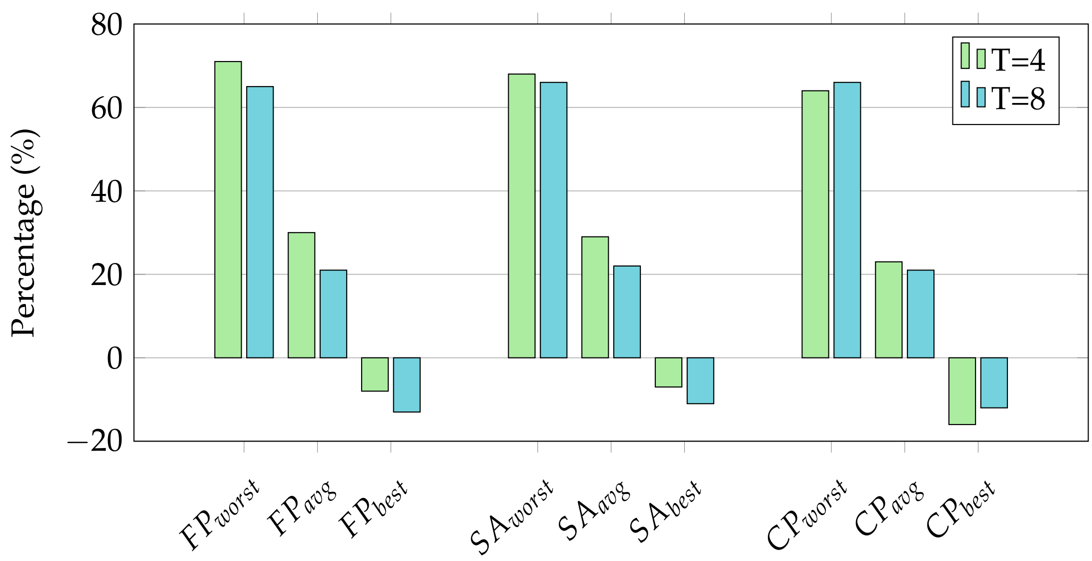


FIGURE 9.6: Comparison of Hit-Rate percentages with the max, min, and mean Hit-Rate values for tripled penalties

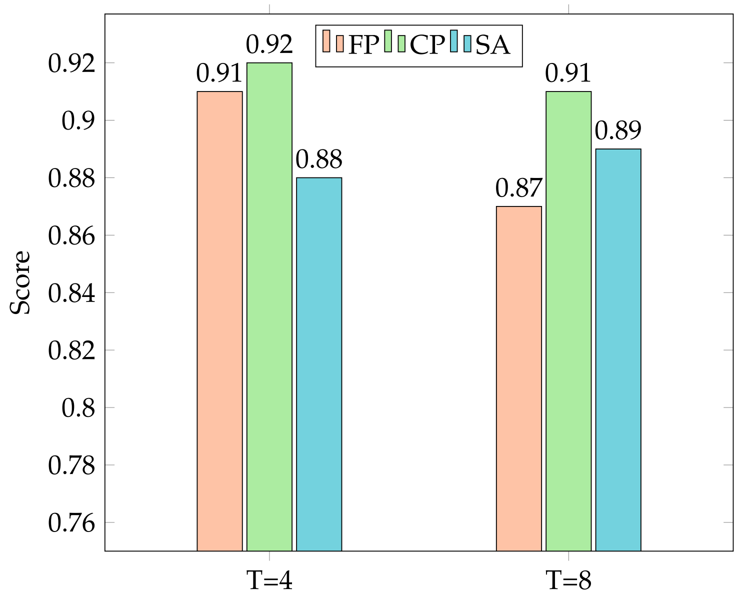


FIGURE 9.7: Quality of the visited time slots with tripled penalties

9.5 Conclusions

In this chapter, we have studied the complexity and challenges of the Attended Home Delivery Problem with Redelivery Options (AHDP-RO), a critical issue in the contemporary logistic and delivery sector. Our research was aimed at developing a more efficient and customer-centric approach to managing attended home deliveries, a domain that significantly impacts customer satisfaction and operational costs. We proposed a Mixed Integer Linear Problem in which we try to minimize the operational costs composed of both traveled times and penalties for missed deliveries. To give a realistic setting, we modeled these probabilities dependent on both clients' availability during the day and the chosen recovery option. Computational results highlight the trade-offs involved in prioritizing different elements of the objective function, such as balancing penalty reduction against travel costs. With travel times being the most expensive part, they show how neglecting their minimization in a pure maximization of hit rate scenario could increase the overall costs exponentially. Moreover, the managerial insights on the probability of successful deliveries for different options is crucial in understanding how delivery success rates are influenced by factors like time slots and customer availability. In conclusion, we propose an economic analysis to deepen the theoretical understanding of various factors impacting AHD services providing valuable insights for logistic companies.

Chapter 10

Conclusions

In this thesis, we have extensively studied Rich Vehicle Routing Problems (RVRPs) in the domain of home healthcare and logistic applications. Through compact Mixed Integer Linear Programming formulations (MILP), exhaustive sensitivity analysis and meaningful managerial insights, and efficient exact and heuristic solution approaches, we captured the complexity and the variety inherent in real-world applications of VRPs. Precisely, we first addressed Nurse Routing Problems (NRPs) in which nurses must provide care directly at patients' houses. Through detailed mathematical modeling, we have presented several fairness functions to capture the needs and goals of the TOC (the central healthcare provider), the nurses and the patients. By developing both single-objective and multi-objective approaches, we provided insights into the complex interplay of objectives among various stakeholders, offering a more comprehensive understanding of their needs and priorities. To efficiently solve large-size instances of the problem, we develop an innovative concept of ALNS that can deal with multiple objective functions when designing destroy and repair operators and acceptance ones. Computational results reveal interesting managerial insights on different equity measures defined for nurses and patients and highlight their relationship with goals imposed by the territorial center in charge of home healthcare services. The developed method is efficient and effective in small and large instances. The comparison with a state-of-the-art MIP solver is highly favorable. On the last part of our analysis on NRP, we proposed the SMHHP-C, a multi-period stochastic and dynamic model with consistency constraints in which we performed a sensitivity analysis on the consistency constraint proposing a new set of patient-management policies differing from the degree of flexibility in nurse-patient assignments and time of visit. The realistic-size instances of the SMHHP-C have been solved using a dynamic approach and a multi-scenario based approach to highlight the importance of including sample scenarios of future information in a dynamic problem. Finally, we have delved into the logistical and environmental aspects of last-mile delivery problems. We proved that a locker-based delivery system is ecologically efficient only when customers adopt an eco-conscious behavior. Then, in an Attended Home Delivery (AHD) setting, we draw an operational model to assess the cost of the delivery system when failed delivery penalties are dependent on both the probability of clients' availability during the time horizon and on the recovery option performed by couriers to deliver the package to an alternative location. In conclusion, our research has highlighted the importance of flexibility and adaptability in VRP models to meet the changing demands of the transportation sector. Integrating fairness considerations, advanced heuristic approaches, and environmental consciousness into these models reflects a forward-thinking approach to logistical challenges. The solutions we have proposed are not only efficient but also mindful of the broader societal and environmental impacts.

Bibliography

Accenture (2021). *Life Reimagined: Mapping the motivations that matter for today's consumers*.

Agatz, Niels et al. (Mar. 2008). "Challenges and Opportunities in Attended Home Delivery". In: *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer US, 379–396. ISBN: 9780387777788. DOI: 10.1007/978-0-387-77778-8_17.

Al Chami, Zaher et al. (June 2021). "A GRASP-ALNS combination for robust pickup and delivery problem". In: *International Journal of Production Research* 60.12, 3809–3828. ISSN: 1366-588X. DOI: 10.1080/00207543.2021.1933236.

Al Hasan, Hasan et al. (Sept. 2018). "Surgical case scheduling with sterilising activity constraints". In: *International Journal of Production Research* 57.10, 2984–3002. ISSN: 1366-588X. DOI: 10.1080/00207543.2018.1521015.

Allen, Michael et al. (Apr. 2020). "Organising outpatient dialysis services during the COVID-19 pandemic. A simulation and mathematical modelling study". In: DOI: 10.1101/2020.04.22.20075457.

Baldacci, Roberto, Aristide Mingozzi, and Roberto Roberti (Apr. 2012). "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints". In: *European Journal of Operational Research* 218.1, 1–6. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2011.07.037.

Belhor, Mariem et al. (Mar. 2023). "Multi-objective evolutionary approach based on K-means clustering for home health care routing and scheduling problem". In: *Expert Systems with Applications* 213, p. 119035. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2022.119035.

Bent, Russell W. and Pascal Van Hentenryck (Dec. 2004). "Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers". In: *Operations Research* 52.6, 977–987. ISSN: 1526-5463. DOI: 10.1287/opre.1040.0124.

Bhattarai, Sudhan, Yaneth Correa-Martinez, and Leonardo Bedoya-Valencia (July 2022). "A multi-objective home healthcare routing problem". In: *International Journal of Healthcare Management* 16.2, 311–325. ISSN: 2047-9719. DOI: 10.1080/20479700.2022.2102111.

Bolsi, Beatrice et al. (Nov. 2022). "Heuristic algorithms for integrated workforce allocation and scheduling of perishable products". In: *International Journal of Production Research* 61.20, 7048–7063. ISSN: 1366-588X. DOI: 10.1080/00207543.2022.2144525.

Bonomi, V., R. Mansini, and R. Zanotti (2022). "Fairness in Home Healthcare: can patient-centered and nurse-centered measures concur to the same goals?" In: *IFAC-PapersOnLine* 55.10, 3136–3141. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2022.10.211.

Boyer, Kenneth K., Andrea M. Prud'homme, and Wenming Chung (Mar. 2009). "THE LAST MILE CHALLENGE: EVALUATING THE EFFECTS OF CUSTOMER DENSITY AND DELIVERY WINDOW PATTERNS". In: *Journal of Business Logistics* 30.1, 185–201. ISSN: 2158-1592. DOI: 10.1002/j.2158-1592.2009.tb00104.x.

Boysen, Nils, Stefan Fedtke, and Stefan Schwerdfeger (Sept. 2020). "Last-mile delivery concepts: a survey from an operational research perspective". In: *OR Spectrum* 43.1, 1–58. ISSN: 1436-6304. DOI: 10.1007/s00291-020-00607-8.

Campbell, Ann Melissa and Martin W. P. Savelsbergh (Aug. 2005). "Decision Support for Consumer Direct Grocery Initiatives". In: *Transportation Science* 39.3, 313–327. ISSN: 1526-5447. DOI: 10.1287/trsc.1040.0105.

Cappanera, Paola and Maria Grazia Scutellà (Apr. 2021). "Addressing consistency and demand uncertainty in the Home Care planning problem". In: *Flexible Services and Manufacturing Journal* 34.1, 1–39. ISSN: 1936-6590. DOI: 10.1007/s10696-021-09412-z.

Carello, Giuliana, Ettore Lanzarone, and Sara Mattia (Mar. 2018). "Trade-off between stakeholders' goals in the home care nurse-to-patient assignment problem". In: *Operations Research for Health Care* 16, 29–40. ISSN: 2211-6923. DOI: 10.1016/j.orhc.2017.12.002.

Chen, Katherine L. et al. (June 2021). "How is the COVID-19 pandemic shaping transportation access to health care?" In: *Transportation Research Interdisciplinary Perspectives* 10, p. 100338. ISSN: 2590-1982. DOI: 10.1016/j.trip.2021.100338.

COVID-19 and e-commerce: a global review (2021). Tech. rep.

Dantzig, G. B. and J. H. Ramser (Oct. 1959). "The Truck Dispatching Problem". In: *Management Science* 6.1, 80–91. ISSN: 1526-5501. DOI: 10.1287/mnsc.6.1.80.

Decerle, Jérémy et al. (Feb. 2019). "A memetic algorithm for multi-objective optimization of the home health care problem". In: *Swarm and Evolutionary Computation* 44, 712–727. ISSN: 2210-6502. DOI: 10.1016/j.swevo.2018.08.014.

Demirbilek, Mustafa, Juergen Branke, and Arne Strauss (Jan. 2018). "Dynamically accepting and scheduling patients for home healthcare". In: *Health Care Management Science* 22.1, 140–155. ISSN: 1572-9389. DOI: 10.1007/s10729-017-9428-0.

Demirbilek, Mustafa, Juergen Branke, and Arne K. Strauss (Apr. 2019). "Home healthcare routing and scheduling of multiple nurses in a dynamic environment". In: *Flexible Services and Manufacturing Journal* 33.1, 253–280. ISSN: 1936-6590. DOI: 10.1007/s10696-019-09350-x.

Edwards, Julia et al. (Jan. 2010). "Carbon Dioxide Benefits of Using Collection–Delivery Points for Failed Home Deliveries in the United Kingdom". In: *Transportation Research Record: Journal of the Transportation Research Board* 2191.1, 136–143. ISSN: 2169-4052. DOI: 10.3141/2191-17.

Ehmke, Jan Fabian and Ann Melissa Campbell (Feb. 2014). "Customer acceptance mechanisms for home deliveries in metropolitan areas". In: *European Journal of Operational Research* 233.1, 193–207. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2013.08.028.

Escudero-Santana, Alejandro et al. (Apr. 2022). "Improving E-Commerce Distribution through Last-Mile Logistics with Multiple Possibilities of Deliveries Based on Time and Location". In: *Journal of Theoretical and Applied Electronic Commerce Research* 17.2, 507–521. ISSN: 0718-1876. DOI: 10.3390/jtaer17020027.

European Environment Agency (2020a). *Monitoring of CO2 emissions from passenger cars – Regulation (EU) 2019/631*.

— (2020b). *Monitoring of CO2 emissions from vans – Regulation (EU) 2019/631*.

Florio, Alexandre M., Dominique Feillet, and Richard F. Hartl (Nov. 2018). "The delivery problem: Optimizing hit rates in e-commerce deliveries". In: *Transportation Research Part B: Methodological* 117, 455–472. ISSN: 0191-2615. DOI: 10.1016/j.trb.2018.09.011.

Gevaers, Roel, Eddy Van de Voorde, and Thierry Vanelslander (Aug. 2011). "Characteristics and Typology of Last-mile Logistics from an Innovation Perspective in an Urban Context". In: *City Distribution and Urban Freight Transport*. Edward Elgar Publishing. ISBN: 9780857932747. DOI: 10.4337/9780857932754.00009.

Ghannadpour, Seyed Farid and Abdolhadi Zarrabi (Feb. 2019). "Multi-objective heterogeneous vehicle routing and scheduling problem with energy minimizing". In: *Swarm and Evolutionary Computation* 44, 728–747. ISSN: 2210-6502. DOI: 10.1016/j.swevo.2018.08.012.

Gobbi, A. et al. (2019). "A Kernel Search for a Patient Satisfaction-oriented Nurse Routing Problem with Time-Windows". In: *IFAC-PapersOnLine* 52.13, 1669–1674. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2019.11.440.

Gobbi, Alessandro et al. (Feb. 2022). "Hybridizing adaptive large neighborhood search with kernel search: a new solution approach for the nurse routing problem with incompatible services and minimum demand". In: *International Transactions in Operational Research* 30.1, 8–38. ISSN: 1475-3995. DOI: 10.1111/itor.13127.

Goeke, Dominik, Roberto Roberti, and Michael Schneider (July 2019). "Exact and Heuristic Solution of the Consistent Vehicle-Routing Problem".

In: *Transportation Science* 53.4, 1023–1042. ISSN: 1526-5447. DOI: 10.1287/trsc.2018.0864.

Grabenschweiger, Jasmin et al. (Jan. 2021). "The vehicle routing problem with heterogeneous locker boxes". In: *Central European Journal of Operations Research* 29.1, 113–142. ISSN: 1613-9178. DOI: 10.1007/s10100-020-00725-2.

Grenouilleau, Florian et al. (May 2019). "A set partitioning heuristic for the home health care routing and scheduling problem". In: *European Journal of Operational Research* 275.1, 295–303. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2018.11.025.

Grieco, Luca, Martin Utley, and Sonya Crowe (June 2020). "Operational research applied to decisions in home health care: A systematic literature review". In: *Journal of the Operational Research Society* 72.9, 1960–1991. ISSN: 1476-9360. DOI: 10.1080/01605682.2020.1750311.

Groër, Chris, Bruce Golden, and Edward Wasil (Oct. 2009). "The Consistent Vehicle Routing Problem". In: *Manufacturing &; Service Operations Management* 11.4, 630–643. ISSN: 1526-5498. DOI: 10.1287/msom.1080.0243.

Gunantara, Nyoman (Jan. 2018). "A review of multi-objective optimization: Methods and its applications". In: *Cogent Engineering* 5.1. Ed. by Qingsong Ai, p. 1502242. ISSN: 2331-1916. DOI: 10.1080/23311916.2018.1502242.

Hanafi, Saïd, Renata Mansini, and Roberto Zanotti (2020). "The multi-visit team orienteering problem with precedence constraints". In: *European Journal of Operational Research* 282.2, pp. 515–529.

Hvattum, Lars M., Arne Løkketangen, and Gilbert Laporte (Nov. 2006). "Solving a Dynamic and Stochastic Vehicle Routing Problem with a Sample Scenario Hedging Heuristic". In: *Transportation Science* 40.4, 421–438. ISSN: 1526-5447. DOI: 10.1287/trsc.1060.0166.

Hvattum, Lars Magnus, Arne Løkketangen, and Gilbert Laporte (Mar. 2007). "A branch-and-regret heuristic for stochastic and dynamic vehicle routing problems". In: *Networks* 49.4, 330–340. ISSN: 1097-0037. DOI: 10.1002/net.20182.

Ishizaka, Alessio, Craig Pearman, and Philippe Nemery (Sept. 2012). "AHPSort: an AHP-based method for sorting problems". In: *International Journal of Production Research* 50.17, 4767–4784. ISSN: 1366-588X. DOI: 10.1080/00207543.2012.657966.

Jiang, Haoqiang, Paulo Gomes, and Debra Vander Meer (Apr. 2023). "Promoting continuity of care in nurse-patient assignment: A multiple objective heuristic algorithm". In: *Decision Support Systems* 167, p. 113926. ISSN: 0167-9236. DOI: 10.1016/j.dss.2023.113926.

Jiang, Li et al. (2019). "A Travelling Salesman Problem With Carbon Emission Reduction in the Last Mile Delivery". In: *IEEE Access* 7, 61620–61627. ISSN: 2169-3536. DOI: 10.1109/access.2019.2915634.

Khodabandeh, Pouria et al. (Jan. 2021). "A Bi-Objective Home Health Care Routing and Scheduling Model with Considering Nurse Downgrading

Costs". In: *International Journal of Environmental Research and Public Health* 18.3, p. 900. ISSN: 1660-4601. DOI: 10.3390/ijerph18030900.

Klein, Robert et al. (Feb. 2019). "Differentiated Time Slot Pricing Under Routing Considerations in Attended Home Delivery". In: *Transportation Science* 53.1, 236–255. ISSN: 1526-5447. DOI: 10.1287/trsc.2017.0738.

Kovacs, Attila et al. (Nov. 2015a). "The Generalized Consistent Vehicle Routing Problem". In: *Transportation Science* 49.4, 796–816. ISSN: 1526-5447. DOI: 10.1287/trsc.2014.0529.

Kovacs, Attila A., Sophie N. Parragh, and Richard F. Hartl (Dec. 2015b). "The multi-objective generalized consistent vehicle routing problem". In: *European Journal of Operational Research* 247.2, 441–458. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2015.06.030.

Laarhoven, Peter J. M. van and Emile H. L. Aarts (1987). "Simulated annealing". In: *Simulated Annealing: Theory and Applications*. Springer Netherlands, 7–15. ISBN: 9789401577441. DOI: 10.1007/978-94-015-7744-1_2.

Liu, Ran, Biao Yuan, and Zhibin Jiang (July 2016). "Mathematical model and exact algorithm for the home care worker scheduling and routing problem with lunch break requirements". In: *International Journal of Production Research* 55.2, 558–575. ISSN: 1366-588X. DOI: 10.1080/00207543.2016.1213917.

Lone, S. and J.W. Weltevreden (2022). *2022 European E-commerce Report*. Tech. rep. Amsterdam University of Applied Sciences and Ecommerce Europe.

Long, Qiang, Xue Wu, and Changzhi Wu (2021). "Non-dominated sorting methods for multi-objective optimization: Review and numerical comparison". In: *Journal of Industrial &; Management Optimization* 17.2, 1001–1023. ISSN: 1553-166X. DOI: 10.3934/jimo.2020009.

Luo, Zhixing et al. (June 2015). "On service consistency in multi-period vehicle routing". In: *European Journal of Operational Research* 243.3, 731–744. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2014.12.019.

Malagodi, Laura, Ettore Lanzarone, and Andrea Matta (Jan. 2021). "Home care vehicle routing problem with chargeable overtime and strict and soft preference matching". In: *Health Care Management Science* 24.1, 140–159. ISSN: 1572-9389. DOI: 10.1007/s10729-020-09532-2.

Mancini, Simona, Margaretha Gansterer, and Richard F. Hartl (Sept. 2021). "The collaborative consistent vehicle routing problem with workload balance". In: *European Journal of Operational Research* 293.3, 955–965. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2020.12.064.

Manerba, Daniele and Renata Mansini (2016). "The Nurse Routing Problem with Workload Constraints and Incompatible Services". In: *IFAC-PapersOnLine* 49.12, 1192–1197. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2016.07.670.

Manerba, Daniele, Renata Mansini, and Roberto Zanotti (2018). "Attended Home Delivery: reducing last-mile environmental impact by changing customer habits". In: *IFAC-PapersOnLine* 51.5, 55–60. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2018.06.199.

Mangiaracina, Riccardo et al. (July 2015). "A review of the environmental implications of B2C e-commerce: a logistics perspective". In: *International Journal of Physical Distribution &; Logistics Management* 45.6, 565–591. ISSN: 0960-0035. DOI: 10.1108/ijpdlm-06-2014-0133.

Mansini, Renata and Roberto Zanotti (July 2019). "Optimizing the physician scheduling problem in a large hospital ward". In: *Journal of Scheduling* 23.3, 337–361. ISSN: 1099-1425. DOI: 10.1007/s10951-019-00614-w.

Moccia, L, J-F Cordeau, and G Laporte (Feb. 2012). "An incremental tabu search heuristic for the generalized vehicle routing problem with time windows". In: *Journal of the Operational Research Society* 63.2, 232–244. ISSN: 1476-9360. DOI: 10.1057/jors.2011.25.

Mosquera, Federico, Pieter Smet, and Greet Vanden Berghe (Mar. 2019). "Flexible home care scheduling". In: *Omega* 83, 80–95. ISSN: 0305-0483. DOI: 10.1016/j.omega.2018.02.005.

Özarık, Sami Serkan et al. (Apr. 2021). "Optimizing e-commerce last-mile vehicle routing and scheduling under uncertain customer presence". In: *Transportation Research Part E: Logistics and Transportation Review* 148, p. 102263. ISSN: 1366-5545. DOI: 10.1016/j.tre.2021.102263.

Özarık, Sami Serkan et al. (Apr. 2023). "An Adaptive Large Neighborhood Search heuristic for last-mile deliveries under stochastic customer availability and multiple visits". In: *Transportation Research Part B: Methodological* 170, 194–220. ISSN: 0191-2615. DOI: 10.1016/j.trb.2023.02.016.

Ozbaygin, Gizem et al. (June 2017). "A branch-and-price algorithm for the vehicle routing problem with roaming delivery locations". In: *Transportation Research Part B: Methodological* 100, 115–137. ISSN: 0191-2615. DOI: 10.1016/j.trb.2017.02.003.

Pacheco, Joaquín and Manuel Laguna (Aug. 2020). "Vehicle routing for the urgent delivery of face shields during the COVID-19 pandemic". In: *Journal of Heuristics* 26.5, 619–635. ISSN: 1572-9397. DOI: 10.1007/s10732-020-09456-8.

Psaraftis, Harilaos N. (Dec. 1995). "Dynamic vehicle routing: Status and prospects". In: *Annals of Operations Research* 61.1, 143–164. ISSN: 1572-9338. DOI: 10.1007/bf02098286.

Punakivi, Mikko and Juha Saranen (Apr. 2001). "Identifying the success factors in e-grocery home delivery". In: *International Journal of Retail &; Distribution Management* 29.4, 156–163. ISSN: 0959-0552. DOI: 10.1108/09590550110387953.

ReportLinker (2023). *Global Transportation Services Industry*.

Reyes, Damián, Martin Savelsbergh, and Alejandro Toriello (July 2017). "Vehicle routing with roaming delivery locations". In: *Transportation Research Part C: Emerging Technologies* 80, 71–91. ISSN: 0968-090X. DOI: 10.1016/j.trc.2017.04.003.

Ropke, Stefan and David Pisinger (Nov. 2006). "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows". In: *Transportation Science* 40.4, 455–472. ISSN: 1526-5447. DOI: 10.1287/trsc.1050.0135.

Saaty, Thomas L. (Sept. 1990). "How to make a decision: The analytic hierarchy process". In: *European Journal of Operational Research* 48.1, 9–26. ISSN: 0377-2217. DOI: 10.1016/0377-2217(90)90057-i.

Saaty, Thomas L. and Luis G. Vargas (2013). "The Analytic Network Process". In: *Decision Making with the Analytic Network Process: Economic, Political, Social and Technological Applications with Benefits, Opportunities, Costs and Risks*. Vol. 195. Boston, MA: Springer US, pp. 1–40.

Schnieder, Maren, Chris Hinde, and Andrew West (June 2021). "Sensitivity Analysis of Emission Models of Parcel Lockers vs. Home Delivery Based on HBEFA". In: *International Journal of Environmental Research and Public Health* 18.12, p. 6325. ISSN: 1660-4601. DOI: 10.3390/ijerph18126325.

Shaw, Paul (1998). "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems". In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 417–431. ISBN: 9783540494812. DOI: 10.1007/3-540-49481-2_30.

Sirikijpanichkul, Ackchai, Sarintorn Winyoopadit, and Anchalee Jenpanitsub (2017). "A multi-actor multi-criteria transit system selection model: A case study of Bangkok feeder system". In: *Transportation Research Procedia* 25, 3736–3755. ISSN: 2352-1465. DOI: 10.1016/j.trpro.2017.05.228.

Song, Liying et al. (Jan. 2009). "Addressing the Last Mile Problem: Transport Impacts of Collection and Delivery Points". In: *Transportation Research Record: Journal of the Transportation Research Board* 2097.1, 9–18. ISSN: 2169-4052. DOI: 10.3141/2097-02.

Spliet, Remy and Guy Desaulniers (July 2015). "The discrete time window assignment vehicle routing problem". In: *European Journal of Operational Research* 244.2, 379–391. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2015.01.020.

Tilk, Christian, Katharina Olkis, and Stefan Irnich (June 2021). "The last-mile vehicle routing problem with delivery options". In: *OR Spectrum* 43.4, 877–904. ISSN: 1436-6304. DOI: 10.1007/s00291-021-00633-0.

Tirkolaee, Erfan Babaee et al. (Mar. 2020). "A novel hybrid method using fuzzy decision making and multi-objective programming for sustainable-reliable supplier selection in two-echelon supply chain design". In: *Journal of Cleaner Production* 250, p. 119517. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2019.119517.

Tversky, Amos (July 1972). "Elimination by aspects: A theory of choice." In: *Psychological Review* 79.4, 281–299. ISSN: 0033-295X. DOI: 10.1037/h0032955.

Voigt, Stefan et al. (Mar. 2023). “The Vehicle Routing Problem with Availability Profiles”. In: *Transportation Science* 57.2, 531–551. ISSN: 1526-5447. DOI: 10.1287/trsc.2022.1182.

Yang, Xinan et al. (May 2016). “Choice-Based Demand Management and Vehicle Routing in E-Fulfillment”. In: *Transportation Science* 50.2, 473–488. ISSN: 1526-5447. DOI: 10.1287/trsc.2014.0549.