Intra-Hospital Transportation Problem: A Two-Phase Bayesian Deep Reinforcement Learning Framework

by

Hossein Torkinezhadirani

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Industrial Engineering

1 July 2025

Intra-Hospital Transportation Problem: A Two-Phase Bayesian Deep Reinforcement Learning Framework

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Hossein Torkinezhadirani

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. F. Sibel Salman (Advisor)

_____

Assoc. Prof. Barış Yıldız

_____

Assoc. Prof. Eda Yücel

Date: _____

To Bonitam, for your love and support, and to my family, for your constant care and encouragement.

This work carries traces of your support.

ABSTRACT

Intra-Hospital Transportation Problem: A Two-Phase Bayesian Deep Reinforcement Learning Framework
Hossein Torkinezhadirani
Master of Science in Industrial Engineering
1 July 2025

The rapid growth in intra-hospital transportation demand, driven by increasing patient flows and operational complexity, has placed critical pressure on hospital logistics systems. Transportation requests, ranging from time-sensitive patient transportation to the delivery of medical supplies and equipment, must be fulfilled with high reliability while managing human resource constraints and emerging automation technologies. This thesis addresses the Real-Time Intra-Hospital Transportation Problem (RIHTP), aiming to optimize the assignment and routing of both human porters and service robots under dynamically arriving and heterogeneous transportation requests. We first develop a detailed problem formulation that integrates recharging requirements for robots, compatibility constraints, workload balancing, and urgency-sensitive delay penalties. For the static setting, the problem is modeled as a Dial-a-Ride Problem (DARP) and solved using a Mixed Integer Linear Programming (MILP) formulation to benchmark solution quality. For the dynamic setting, we propose a novel two-phase solution methodology: Phase 1 employs a Bayesian Deep Reinforcement Learning framework to assign requests in real time under uncertainty, and Phase 2 applies a Best Insertion heuristic followed by Adaptive Variable Neighborhood Search (AVNS) for route construction and consolidation. Using a suite of realistically simulated hospital scenarios, our computational experiments demonstrate that the proposed Bayesian-based assignment policy outperforms existing baseline policies in the heterogeneous environment of hospital transportation, achieving a 9.26% improvement over the current hospital policy. Furthermore, the proposed two-phase solution approach yields a 10.88% reduction in total objective value compared to the conventional one-phase reoptimization method widely adopted in the literature, while simultaneously reducing average runtime from 954.97 seconds to 299.56 seconds. Furthermore, experiments investi-

gating other operational and tactical decisions highlight the substantial benefits of integrating service robots into the transportation system: the average workload of human porters decreases by approximately 15%, delay penalties are reduced by up to 85%, and overall objective function values drop by 13%. These findings collectively underscore the efficacy and scalability of our proposed framework in enhancing responsiveness, equity, and efficiency within real-time hospital logistics. The proposed methods are shown to be effective across diverse operational conditions, including varying request arrival rates, transporter fleet configurations, and robot eligibility scenarios. Additionally, the framework captures the complex interplay between assignment policies, route optimization, and automation integration, offering practical insights into transporter coordination and workload management. In summary, this thesis contributes scalable, cost-efficient, and policy-aware solutions for real-time intra-hospital transportation, providing actionable guidance for smart hospital logistics, automation planning, and the future integration of learning-based decision support systems in healthcare environments.

ÖZETÇE

Hastane İçi Taşımacılık Problemi: İki Aşamalı Bayesyen Derin Pekiştirmeli Öğrenme Çerçevesi
Hossein Torkinezhadirani
Endüstri Mühendisliği, Yüksek Lisans
1 Temmuz 2025

Hastane içi taşımacılık talebindeki hızlı artış, artan hasta akışları ve operasyonel karmaşıklığın etkisiyle hastane lojistik sistemleri üzerinde önemli bir baskı oluşturmuştur. Zaman hassasiyetine sahip hasta taşımalarından tıbbi malzeme ve ekipman teslimatına kadar uzanan taşıma taleplerinin, hem insan kaynağı kısıtları hem de gelişmekte olan otomasyon teknolojileri göz önünde bulundurularak yüksek güvenilirlikle karşılanması gerekmektedir. Bu tez, Real-Time Intra-Hospital Transportation Problem (RIHTP) üzerine odaklanmakta ve dinamik olarak gelen, heterojen özellikler taşıyan taşıma talepleri altında insan portörler ile servis robotlarının atama ve rotalama kararlarının optimize edilmesini amaçlamaktadır. İlk olarak, robotlar için şarj gereksinimleri, görev uyumluluğu kısıtları, iş yükü dengesi ve aciliyet bazlı gecikme cezaları gibi unsurları içeren ayrıntılı bir problem formülasyonu geliştirilmiştir. Statik ortam için problem, Dial-a-Ride Problem (DARP) olarak modellenmiş ve çözüm kalitesini değerlendirmek amacıyla Mixed Integer Linear Programming (MILP) formülasyonu ile çözülmüştür. Dinamik ortamda ise iki aşamalı yenilikçi bir çözüm yöntemi önerilmiştir: Birinci aşamada, gerçek zamanlı ve belirsizlik altındaki atama kararlarını vermek üzere Bayesian Deep Reinforcement Learning çerçevesi kullanılmaktadır. İkinci aşamada ise, rota oluşturma ve birleştirme işlemleri için Best Insertion sezgiseli uygulanmakta, ardından Adaptive Variable Neighborhood Search (AVNS) yöntemi kullanılmaktadır. Gerçekçi hastane senaryoları kullanılarak yürütülen hesaplamalı dene-yler, önerilen Bayesyen tabanlı atama politikasının, hastane içi taşıma sistemlerinin heterojen yapısında mevcut temel politikalara kıyasla üstün performans gösterdiğini ve mevcut hastane politikasına göre %9.26 oranında iyileşme sağladığını ortaya koymaktadır. Ayrıca, önerilen iki aşamalı çözüm yaklaşımı, literatürde yaygın olarak kullanılan tek aşamalı yeniden optimizasyon yöntemine kıyasla toplam amaç fonksiyonu değerinde %10.88 azalma sağlarken,

ortalama çözüm süresini 954.97 saniyeden 299.56 saniyeye düşürmektedir. Buna ek olarak, farklı operasyonel ve taktik kararları inceleyen deneyler, taşıma sistemine servis robotlarının entegre edilmesinin önemli kazanımlar sağladığını göstermektedir: İnsan portörlerin ortalama iş yükü yaklaşık %15 oranında azalmakta, gecikme cezaları %85'e varan oranlarda düşmekte ve toplam amaç fonksiyonu değeri %13 oranında iyileşmektedir. Elde edilen bulgular, önerilen çerçevenin gerçek zamanlı hastane lojistiğinde duyarlılık, adalet ve verimlilik bakımından etkili ve ölçeklenebilir olduğunu göstermektedir. Geliştirilen yöntemlerin farklı operasyonel koşullar, talep geliş hızları, taşıyıcı filo yapıları ve robot uygunluğu senaryoları, altında da başarılı performans sergilediği gözlemlenm-iştir. Ayrıca bu çerçeve, atama politikaları, rota optimizasyonu ve otomasyon entegrasyonu arasındaki karmaşık etkileşimi dikkate alarak taşıyıcı koordinasyonu ve iş yükü yönetimine dair değerli çıkarımlar sunmaktadır. Sonuç olarak bu tez, gerçek zamanlı hastane içi taşımacılık için ölçeklenebilir, maliyet etkin ve politika duyarlı çözümler sunmakta; akıllı hastane lojistiği, otomasyon planlaması ve öğrenme tabanlı karar destek sistemlerinin sağlık hizmetlerine entegrasyonu için uygulanabilir öneriler ortaya koymaktadır.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. F. Sibel Salman, for her unwavering support, insightful guidance, and continuous encouragement throughout my master's journey. Her mentorship has been instrumental not only in the development of this thesis but also in shaping my academic perspective and research skills.

I would also like to thank my thesis committee members for their valuable time and constructive feedback, which helped me refine and strengthen this work.

My deepest appreciation goes to the members of our research group, Dr. Ali Hassanzadeh and Prof. Özgür M. Araz, for their helpful discussions, support, and collaboration during this research.

Finally, I extend my heartfelt thanks to those who have quietly and consistently stood by me throughout this journey. Their presence, whether near or far, has been a source of strength and balance during the most demanding times.

TABLE OF CONTENTS

List of Tables xii

List of Figures xiv

Abbreviations xvii

Chapter 1: Introduction 1

Chapter 2: Literature Review 5

2.1 Hospital Transportation 5

2.1.1 Intra-hospital Transportation 6

2.1.2 Inter-hospital Transportation 10

2.2 Dial-A-Ride 13

2.2.1 Dial-A-Ride in Hospital Transportation 17

2.3 Applications of Autonomous Robots in Hospitals Transportation . . . 21

2.4 Impacts of RFID Technology on Healthcare Logistics and Patient Care 24

2.5 Comparison of Our Study with Related Studies and Our Contributions 27

Chapter 3: Problem Definition 31

3.1 Transportation Requests 33

3.2 Porters, Service robots, and Dispatcher 37

3.3 Hospital Network 40

3.4 Dynamic Events 42

3.5 An Illustrative Example 43

Chapter 4: Optimized Frameworks for Static Intra-Hospital Transportation 45

4.1 Mathematical Modes for Static Approaches to Assign and Schedule Requests . . . 45
4.1.1 Human-Operated Dial-A-Ride Mathematical Model . . . 45
4.1.2 Human-Robot Dial-A-Ride Mathematical Model . . . 52

Chapter 5: Real-Time Optimization of Intra-hospital Transportation 60

5.1 Phase 1 Decision: Assignment Problem . . . 63
5.1.1 Sequential Decision Process Model . . . 64
5.1.2 Deep Q-Learning . . . 66
5.2 Phase 2 Decision: Routing Problem . . . 71

Chapter 6: Numerical Experiments 78

6.1 Design of Test Instances . . . 79
6.2 Numerical Results: Assessing the Best Insertion and AVNS Heuristic for Phase 2 . . . 82
6.3 Numerical Results: Assessing the BNN-based Policy for Phase 1 . . . 95
6.3.1 Baseline Policies . . . 95
6.3.2 Features . . . 96
6.3.3 Training Settings . . . 97
6.3.4 Learning $\pi_{DQN}^{Bayesian}$ . . . 99
6.3.5 Comparison of Policies Under Different Settings . . . 104
6.4 Numerical Results: Assessing the Two-Phase Solution Approach . . . 113
6.5 Numerical Results: Impact of Utilizing Service Robots in Intra-Hospital Transportation . . . 119
6.5.1 System-Wide Performance Implications of Service Robot Integration . . . 121
6.5.2 Overall Insights . . . 125

Chapter 7: Conclusion 128

Bibliography 137

Appendix A: Summary Table for Previous Studies in Intra-Hospital transportation 151

Appendix B: Baseline Policies 156

Appendix C: Benchmark 159

Appendix D: Scatter and Box Plots for Different Fleet Configurations 165

Appendix E: Scatter and Box Plots for Different Service Robots Speed Variation Factors 169

Appendix F: Notation Summary 172

LIST OF TABLES

3.1 Summary of requests by type, capacity, urgency, and difficulty 35

4.1 Table of notations for sets. 46
4.2 Table of notations for parameters and constants. 46
4.3 Table of notations for decision variables. 47
4.4 Table of notations for additional sets. 53
4.5 Table of notations for compatibility and battery parameters. 53
4.6 Table of notations for robot-specific variables and parameters. 53

6.1 The dynamic routing problem results: impact of maximum iteration on heuristic's performance. 87
6.2 The dynamic routing problem results: impact of maximum iteration on heuristic's performance.-*Cont'd* 88
6.3 Summary: impact of maximum iteration on heuristic's performance. . . . 89
6.4 The dynamic routing problem results: impact of neighborhood size (h) on heuristic's performance. 90
6.5 The dynamic routing problem results: impact of neighborhood size (h) on heuristic's performance.-*Cont'd* 91
6.6 Summary: impact of neighborhood size (h) on heuristic's performance. . . 91
6.7 Comparison of MILP with time limit and BI+VNS performance across problem instances 94
6.8 Statistical comparison of BI + VNS and MILP with time limit solutions across different problem sizes. 95
6.9 Summary of baseline policies and their characteristics. 96
6.10 Tested and selected values for training parameters. 99
6.11 Performance summary of BNN architectures. 105

6.12 Detailed performance statistics for 25 test instances across various policies, and transporter fleet. . . . 106
6.13 Detailed performance statistics for 25 test instances across various policies, and speed variation factor. . . . 109
6.14 Comparison of one-phase and two-phase approaches in terms of objective and percentage difference with respect to $\pi_{reopt} + ATS$ ($\varnothing$). . . . 115
6.15 Comparison of one-phase and two-phase approaches in terms of runtime for each instance. . . . 116
6.16 Percentage distribution of request types for each test instance. . . . 119
6.17 Percentage of requests that can and cannot be handled by service robots in each test instance. . . . 120
6.18 Average objective function terms across different service robot integration levels. . . . 124

A.1 Previous survey comparison in intra-hospital transportation. . . . 152
A.2 Previous survey comparison in intra-hospital transportation.-*Cont'd* . . . 153
A.3 Previous survey comparison in intra-hospital transportation.-*Cont'd* . . . 154
A.4 Previous survey comparison in intra-hospital transportation.-*Cont'd* . . . 155

C.1 Summary statistics (min, max, mean) for Objective Difference (%) and Runtime Ratio of ATS across different numbers of requests. . . . 162

F.1 Table of notations for sets. . . . 172
F.2 Table of notations for parameters. . . . 173
F.3 Table of notations for parameters.-*Cont'd* . . . 174
F.4 Table of notations for variables. . . . 175
F.5 Table of notations (others) . . . 176
F.6 Table of notations (others).-*Cont'd* . . . 177

LIST OF FIGURES

2.1 (Color online) A service robot designed by FDATA for delivering supplies efficiently and safely in the hospital [Fdata Co., 2025]. 21

2.2 (Color online) An RFID system consists of an RFID reader that receives data from one or several tags and transfers this information to a host system. [Profetto et al., 2022]. 25

3.1 Transportation request types and their node representations 35

3.2 Penalty function to compute the inconvenience associated with a request (delay). (a) patient transportation (b) medical suppliers, medical equipment, and administrative documents 37

3.3 Flowchart of the current dispatching system 38

3.4 (Color online) Overview of the (BL, RL) policy for recharging [Jeong and Moon, 2024]. 40

3.5 (Color online) Illustration of Real-Time Intra-Hospital Transportation Problem (RIHTP) 44

5.1 The flowchart of the two-phase solution methodology for dynamic version 63

5.2 Illustration of the sequential decision-making process of the problem in a Sequential Decision Process Model. 66

5.3 Architecture of the Deep Q-Learning (DQL) model for Phase 1 decision in RIHTP. 69

5.4 Standard training process in a RL setting [Sutton, 2018]. 70

6.1 (Color online) 1000-episode moving average 101

6.2 (Color online) 1000-episode moving average 102

6.3 95% confidence intervals for objective means for each BNN architecture 105
6.4 (a) Objective values across instances when porters are faster than service robots; (b) Objective values across instances with mixed speeds of porters and service robots. 111
6.5 Comparison of transport type usage across different service robot integration scenarios. 121
6.6 Average performance metrics of each transporter type (V_1, V_2, and R) across varying service robot integration levels. 122
6.7 Comparison of normalized objective function terms across different service robot integration levels. 124

D.1 Scatter and box plots for 25 test instances across various policies, and 30 porters and 0 service robots. 165
D.2 Scatter and box plots for 25 test instances across various policies, and 30 porters and 5 service robots. 166
D.3 Scatter and box plots for 25 test instances across various policies, and 30 porters and 10 service robots. 166
D.4 Scatter and box plots for 25 test instances across various policies, and 40 porters and 0 service robots. 167
D.5 Scatter and box plots for 25 test instances across various policies, and 40 porters and 5 service robots. 167
D.6 Scatter and box plots for 25 test instances across various policies, and 40 porters and 10 service robots. 168

E.1 Scatter and box plots for 25 test instances across various policies, and speed variation factor 1. 169
E.2 Scatter and box plots for 25 test instances across various policies, and speed variation factor 1.25. 170
E.3 Scatter and box plots for 25 test instances across various policies, and speed variation factor 1.5. 170

E.4 Scatter and box plots for 25 test instances across various policies, and speed variation factor 2. 171

ABBREVIATIONS

RFID	Radio Frequency Identification
DARP	Dial-A-Ride Problem
PMSP-SDST	Parallel Machine Scheduling Problem With Sequence-dependent Setup Time
FPTP	Ergonomic Patient Transportation Problem
MILP	Mixed-integer Linear Programming
SP	Scheduling Problem
MTDARP	Multi-Trip Dial-A-Ride Problem
ER-DARP	Extended Realistic Dial-A-Ride Problem
MDLS	Multi-directional Local Search
VND	Variable Neighborhood Descent
MD-DARP	The Multi-Depot Dial-A-Ride Problem
R-DARP	Realistic Dial-A-Ride Problem
PDVRP	Pickup and Delivery Vehicle Routing Problem
MTSPM	Multiple Steps Traveling Salesman Problem Model
VRPM	Vehicle Routing Problem Model
TRM	Temporary Replenishment Mode
CRM	Concentrate Replenishment Mode
AR	Autonomous Robot
AMR	Autonomous Mobile Robot
CG	column generation
PM	Patient Mover
ACO	Ant Colony Optimization
OR Wing	Operating Room Wing

AURORA	Autonomous Self-Navigating Robotic OR Assistance
AGV	Automated Guided Vehicle
FSRPS-AGV	Fleet Sizing and Routing Problem with Synchronization for Autonomous Guided Vehicles
ALNS	Adaptive Large Neighborhood Search
RL	Reinforcement Learning
DQL	Deep Q-Learning
BNN	Bayesian Neural Network
MILP	Mixed Integer Linear Programming
RIHTP	Real-Time Intra-Hospital Transportation Problem
FMS	Fleet Management System
RTLS	Real-time locating systems
AVNS	Adaptive Variable Neighborhood Search
VNS	Variable Neighborhood Search
HPAH	Hospital's Proximity Assignment Heuristic
MC	Monte Carlo
MILP-TL	Mixed Integer Programming Model with Time Limit
TS	Tabu Search
ATS	Adaptive Tabu Search

Chapter 1

INTRODUCTION

Effective transportation management serves a pivotal function in modern healthcare delivery systems. According to [Landry and Philippe, 2004], logistics-related operations represent roughly 46% of a hospital's operational budget, highlighting transportation's significant financial and operational impact within healthcare settings. Furthermore, owing to integrating a growing variety of tests, assessments, and expert consultations, the landscape of clinical diagnostics is advancing. Consequently, patients interact with an extensive network of medical professionals which creates complex pathways that require frequent transportation across different hospital departments and services. This aspect is crucial not only for patient mobility but also for the seamless movement of supplies and equipment essential for daily operations. Inefficiencies or delays in transportation in such materials can lead to a cascade of negative outcomes, including underutilization of valuable health resources, disruption of scheduled services, and a detrimental impact on patient satisfaction due to the need for rescheduling and extended waiting times. Thus, an integrated approach to the management of transportation, for patients, supplies and equipment, with a focus on timely and coordinated movement both within an individual hospital and across different facilities, is emerging as a vital strategy for improved delivery of patient care, optimization of resource use, and achievement of a smoother healthcare delivery system.

Nowadays, advancements in medical technology, the rising volume of patient admissions, and the need for faster diagnostic and treatment processes have rendered intra-hospital transportation, defined as the transportation of a patient and medical items between any two services within the same hospital, is crucial for maintaining smooth hospital workflows. This aspect is of paramount importance in large healthcare facilities since the services are usually distributed among several buildings or

even a whole campus. Intra-hospital transportation affects whether the hospital can provide timely and effective treatment. An efficient intra-hospital transportation system should exist in facilities where departments are widely distributed, necessitating internal departmental visits and inter-building transfers. The coordination of those movements should be effectively performed to create no bottlenecks delaying the delivery of treatment, which leads to overloads of resources.

The significance of intra-hospital transportation goes beyond the operational domain and borders on patient safety and quality of care. Poor transportation systems in hospitals may lead to delays in critical diagnostics and treatment, which in turn may culminate in worsening health conditions or leading to missed therapeutic opportunities. It is actually not only a problem of logistics but also an issue of patient care activity that requires considerations for safety, comfort, and dignity.

Recent technology advancements have markedly enhanced intra-hospital patient transportation. Comprehensive scheduling systems, seamlessly integrated with transportation services, are now essential for effective care coordination. The implementation of technology-driven solutions has improved real-time logistics management, enabling increased efficiency and flexibility in patient transport. These advances enable hospitals to flexibly respond to evolving patient care requirements while embodying overarching healthcare trends centered on innovation, efficiency, and patient-centric care.

Radio Frequency Identification (RFID) has become an example of technology-driven innovative solutions in intra-hospital transportation. According to [Awawdeh and Abu-Shanab,], "RFID is an emerging technology that is rapidly becoming a required standard for hospitals and continues to grow in the healthcare industry to track inventory, positive patient identification, and manage personnel". The technology not only dramatically cuts the time and effort of internal logistics, thereby optimizing operational workflows, but also give the ability to obtain information in real-time. Their ability to reduce operating costs and minimizing errors contributes to the efficiency of patient transport within healthcare facilities.

Another advantageous technology that can be incorporated into hospital transportation is Autonomous Mobile Robots (AMRs), which are promising owing to

their elevated autonomy and capacity to navigate the surroundings [Bernhard et al., 2023]. As posited in the study by [Rossetti and Selandari, 2001], hospitals need to thoroughly evaluate the implementation of mobile robots, which are called service robots, in their delivery systems. Mobile robotic solutions are economically efficient and can match or surpass the performance of human-operated delivery systems. Despite the advantages of using service robots, there are some challenges in integrating such technologies in the hospital. To shed light on this issue, the interaction between patients and robots needs careful consideration, since patients may suffer anxiety from repeated encounters with robots in the medical settings. Furthermore, utilizing such beneficial technology requires infrastructure, including dedicated lanes and spaces in the corridors and elevators, in addition to constructing recharging stations. Additionally, service robots require a resilient telecommunication infrastructure to provide connectivity and interaction with external elements such as elevators and smart gadgets [Fanti et al., 2020]. Nonetheless, the advantages of such implementation in the hospital outweigh the disadvantages. Although the initial investment for implementation is considerable, this strategy may markedly reduce the number of delivery personnel required in the hospital, thereby drastically lowering variable operating expenses and improving overall efficiency.

In this thesis, we treat Real-Time Intra-Hospital Transportation Problem (RIHTP) with consolidation of requests by integrating a Deep Q-Learning (DQL) approach with heuristic algorithms to make robust, real-time, and fast decisions in the unpredictable and dynamic environment of hospital transportation. The primary problem is to efficiently assigning and routing porters and service robots in response to dynamically arriving transportation requests, focusing on minimizing total travel and service time, balancing the workload of porters while minimizing the total penalty associated with delays. In contrast to traditional scheduling techniques which assume full knowledge of transportation requests beforehand, our methodology considers the dynamic and stochastic characteristics of transportation requests where new requests arrive unpredictably throughout the day. This requires real-time decision-making to enhance fleet utilization while accommodating real-time factors such as patient priority, porter availability, and hospital-specific workflow conditions.

To tackle this problem, we utilize a reinforcement learning framework based in Deep Q-Learning, employing a Bayesian Neural Network (BNN) as a function approximator. The BNN-based agent predicts expected future rewards for state-action pairs to facilitate a robust decision-making system under uncertainty. The model is trained offline to produce near-optimal assignment decisions in real-time. The decision-making process entails choosing the most suitable porter or robot for an newly-arrived request, considering the present system conditions and expected future demand. Furthermore, a heuristic routing algorithm optimizes the routes of assigned porters and robots, ensuring efficient completion of requests. Together these methods, DQL-based assignment and heuristic-based routing, provide a scalable and adaptable solution for complex hospital transportation system.

The remainder of the thesis is organized as follows. Chapter 2 provides a survey of the related literature. Chapter 3 defines the problem that we have encountered comprehensively and introduce the Mixed Integer Linear Programming (MILP) model. Chapter 4 describes and formulates the intra-hospital transportation. Chapter 5 details the implementation of the reinforcement learning model, feature engineering, and heuristic-based scheduling for real-time optimization in addition to the benchmarks. Chapter 6 provides empirical evidence of the effectiveness of the solution approach. Chapter 7 summarizes the work and suggests future research directions.

Chapter 2

LITERATURE REVIEW

Initially, the review address hospital transportation, categorizing it into two main sections: intra-hospital (within the same hospital) and inter-hospital (between different hospitals) transportation. This segment is crucial as efficient patient transportation directly influences the timeliness and quality of care received, and supports healthcare professionals in delivering optimal services. Subsequently, attention is directed towards the Dial-A-Ride Problem (DARP), specifically in hospital transportation, which stands at the core of our investigation. DARP encompasses strategies for scheduling and routing transportation in a way that aligns with patient needs and healthcare demands. The discussion is bifurcated into two types of DARP: static, where transportation requirements are known beforehand, and dynamic, which deals with requests that arise spontaneously, necessitating real-time solutions. Finally, we examine the recent papers on applications of autonomous robots as well as RFID technology in hospitals.

2.1 Hospital Transportation

Patient transport can primarily be classified into two categories, which are inter-hospital transportation and intra-hospital transportation. Inter-hospital transportation is the transport of patients from their homes to the hospital or between two different hospital infrastructures. On the other hand, intra-hospital transportation is the transport of patients between the different facilities in the same hospital. The intra and inter-hospital patient transport problems are often studied using the Dial-A-Ride Problem [Ton et al., 2024]. This is a model whose implications and specific applications in optimizing the logistics of patient transport is more profoundly presented in the rest of the literature review. Although there are a lot of similarities,

there are differences between intra and inter-hospital transportation claims for the use of proper techniques in the case of every problem. As already explained, the concept of hospital transportation is extended beyond the movement of patients and includes the movement of medical equipment, drugs, blood samples, and other important resources. This elaborated scope allows the significance of logistics in the healthcare environment to guarantee efficient and timely access to resources as an integral part of patient service. The current section, hence, is meant to review the related literature for each of the problems, although more concentrated would be the problem of intra-hospital transportation.

2.1.1 Intra-hospital Transportation

A centralized patient movement system is important for the intra-hospital transportation process so that the movement of patients and the material that is crucial to them, including documents, blood samples, and medical supplies, is done effectively and in a coordinated manner. Therefore, improvement in total flow management in hospitals dramatically increases their operation efficiency, decreases queues for diagnostics results, and ensures that all the appropriate information and resources concerning patient care are distributed to healthcare professionals on time in order to maximize the outcome of patients and hospital productivity.

To the best of our knowledge, the first centralized patient movement system referenced is by [Schall, 1988], which proposes for the first time the efficiencies gained by implementing such a system. The system, which utilizes dedicated personnel called Patient Movers (PMs) in a strategic way, is thus demonstrated to drastically reduce patient transport times within health facilities. This pilot study, therefore, open a path for future research into patient transportation logistics streamlining within a hospital.

Similarly, there are studies on improving quality, such as [Dershin and Schaik, 1993], which deal with the organizational and staffing aspects of a hospital-wide patient transport system as a quality improvement measure. The approach is prepared for a hospital quality improvement program based on a centralized communication

structure with a queuing model for monitoring the number of people needed. The inclusion of a digital system for data collection and surveillance details the role of technology in further increasing the efficiency of the system. As the presented case study shows, wait time improvements, savings, and customer satisfaction gains can be substantial and very significant for proper hospital logistics.

In an efficient centralized patient movement dispatch system, [Chen et al., 2005] proceed to discuss the issues and the procedural requirements. The authors emphasize the need for exceptional communication between the clinics, the dispatch system, and the patient movers. They advocate for technological support, proposing that all patient transport efforts be centralized in one unit to ensure meticulous monitoring of medical records and patient movements.

In the later part of the 2010s, there is a notable increase in the focus on intra-hospital transportation. Recent research increasingly focuses on employing advanced logistical strategies and mathematical modeling to address the complexities of intra-hospital patient transportation. [Hanne et al., 2009] apply an optimization approach to relieve the logistical issues that exist in patient transportation within hospitals in Germany. The study presents the system Opti-TRANS, a comprehensive transportation planning system that robustly improves patient flow through strategic management of transport requests. This system exemplifies how logistical efficiency improvements can significantly impact healthcare, advocating for the adoption of advanced logistical strategies and technological innovations.

Several studies have dealt with this challenge of optimally managing the intra-hospital patient transportation system toward optimizing system service performance and quality. In the work by [Kuchera and Rohleder, 2011], optimization of staffing schedules for fluctuating requests is performed through a queuing theory-based modeling of the problem. The study reveals the fact that the maintenance of service quality requires operational practices to be adaptable to daily and hourly fluctuations of transportation demands. Furthermore, [Bärmann et al., 2024] examine the optimization of intra-hospital patient transportation scheduling, emphasizing the difficulty of accommodating time-sensitive and diverse transportation requests within limited resources. To address the numerous, frequently competing priorities

in hospital logistics, they construct the issue as a lexicographic optimization model, whereby urgent, delayed, and non-urgent requests are prioritized in a hierarchical manner. The authors suggest an exact solution method based on column generation. Their aim is to prioritize the minimization of delays for urgent requests, thus reducing overall delays, and ultimately minimizing the total distance traveled. [Hamdi et al., 2024] address the stretcher-based intra-hospital transportation problem, aiming to reduce appointment delays and unassigned missions. They formulate a mixed-integer linear programming (MILP) model based on real hospital data, solved via CPLEX, to optimize daily transporter assignments. The objective combines minimizing cumulative delay and mission cancellations while ensuring workload balance and synchronization.

[Séguin et al., 2019] draw from this line of inquiry and propose a mixed-integer program for improved patient transportation efficiencies in hospitals. The methodology, using past service call data, has the objective of finding optimal resources to minimize operational costs and maximize the service level without interfering with the already scheduled employee shifts. The strength of this study is that it tends to make advancements in operations that are clearly possible through the use of advanced mathematical modeling techniques.

The dynamic and complex nature of intra-hospital patient transport is found to be addressed by technological innovations in scheduling and real-time management systems. For instance, in the study by [Beaudry et al., 2010], an adaptive model for scheduling is designed to make the hospital transport service more flexible and more responsive, therefore creating a balance between patient convenience and operational costs. In the area of intra-hospital transport scheduling, [Fiegl and Pontow, 2009] solve the problem through real-time scheduling with an always-adjusting decision framework to new requirements for transportation with agility. This paper highlights the flexibility and benefits of online scheduling systems in the swift health facility environment and discusses their role in rendering services more responsive and patient-oriented. Furthermore, [Sun et al., 2025] examine the inefficiencies in intra-hospital patient transportation systems, specifically the delays resulting from first-in-first-out assignment strategies that overlook transporter prox-

imity. To enhance system performance, they devise and execute a proximity-based transporter-to-request allocation technique, in which each request is scored based on its time in the queue, clinical priority, and whether a transporter is located in the same proximity group. Requests with the highest scores are prioritized for assignment. The objective is to minimize overall transport completion times and reduce the percentage of transports exceeding a 45-minute threshold. This intervention is developed and assessed utilizing discrete-event simulation (DES) and subsequently implemented via an established electronic medical record (EMR) system at a major university hospital. The methodology integrates simulation-based optimization with real-world execution, eliminating the need for workforce alterations or external technologies.

Dynamic patient transportation is operationalized by [Kergosien et al., 2011] at the Hospital Complex of Tours, France, through the mobilization of demand, which oscillates daily with the need to be transported by specific vehicles and under a particular operational protocol. With over 46,000 transportations every year, this particular approach and the realization of the importance of real-time management in cost reduction and the transfer of patients in a prompt manner is attributed. Another work, but an in-depth study using simulation, is performed by [Bozer and Eamrungroj, 2020] at the University of Michigan Hospital, indicating the operational inefficiencies related to dispatching and proposing improving strategies. This work speaks to the needed coordination of dispatching and equipment management to support a well-functioning patient flow in hospitals.

Moreover, the recent attention to managing transport requests in real-time, discussed by [Ton et al., 2024], raises the importance of advanced scheduling and rescheduling strategies toward improving hospital logistics. The paper describes an advanced real-time management approach for intra-hospital patient transport requests using a Parallel Machine Scheduling Problem with Sequence-Dependent Setup Times (PMSP-SDST). Its solution involves a constructive heuristic for initial scheduling, a local search heuristic for constant improvement, and dynamic policies for rescheduling, all tested with intensive simulations for optimal patient flow and resource utilization within hospital settings. This paper, therefore, makes impor-

tant progress in the use of real-time data and mathematical modeling for patient transportation optimization.

Sharing of strategic capacity among services helps improve efficiency and response in emergency medical services, as integrated by [Kergosien et al., 2023]. The routing and scheduling aspect of this problem is addressed by [Turan et al., 2011], who take into consideration this unique characteristic of intra-hospital patient transportation, so as to cater for the travel and preparation times crucial for patient care. The emphasis here is on ensuring that porters are effectively used within the operational framework of the hospital.

Finally, studies by [Elmbach et al., 2015] and [von Elmbach et al., 2019] address the ergonomic aspect of patient transportation, which highlights the importance of decreasing physical pressure on porters. These studies integrate ergonomic considerations into logistical planning so as to promote a healthier work environment and make improvements with operational efficiency.

2.1.2 Inter-hospital Transportation

In our research, we primarily focus on intra-hospital transportation, meaning the logistics of moving patients inside a single healthcare facility. Nonetheless, we also present the review of literature on inter-hospital transportation with the intention to make the topic clearer and to point out the principal differences between inter and intra-hospital transportation. as we mentioned before, inter-hospital transportation consists of the transport of patients from their homes to the hospital or between two different hospital infrastructures. More specifically, inter-hospital transportation consists of moving patients to and from their residences to healthcare facilities (outbound requests) or from the healthcare facilities to their residences (inbound requests). This process is intended to be focused mainly on the transfers that take place between the patients' homes and hospitals and vice versa.

The genesis of advances in the field of patient transportation logistics is the development of strong optimization models that would be the basis for overcoming challenges that are highly intricate in providing the effective and safe transportation

of patients. In particular, papers by [Tóth et al., 2024] and [Detti et al., 2017] emphasize the novelty in mixed-integer linear programming (MILP) models that are expected to bring significant uplift in the efficiency and cost-effectiveness of non-urgent patient transport systems. The work of [Tóth et al., 2024] in introducing a MILP model is commendable not only for the ambitious levels of vehicle utilization, the number of patients to serve, and the total number of kilometers traveled, but also for considerations based on the nuanced needs of an oncoming diversity of patient demographics, a critical analysis based on the aging populations globally. The work of [Detti et al., 2017] advances the Multi-Depot Dial-A-Ride Problem (MTDARP) by integrating heterogeneous vehicles with complex patient-vehicle compatibility constraints for the first time. This innovation marks a significant step forward, aligning logistical efficiencies with the crucial aspects of patient safety and comfort. In the work by [Molenbruch et al., 2017b], this optimization is formulated through a combination of restrictions in ride-sharing based on health safety and restrictions due to patient-specific needs. It is reported, in the case of the study of a Belgian transportation provider, that tailors transportation logistics and systematic organization of routes can greatly enhance efficiency and safety in the delivery of non-urgent hospital transport services. All of these models demonstrate approaches of deep analytics toward the optimization of healthcare logistics while emphasizing the need to include flexibility and responsiveness to patient needs in such frames.

Beyond the core optimization models, the development of the associated algorithms is the most fertile ground for innovation and the development of solutions to the highly computational aspects of coordinating patient transportation logistics. Research efforts pursued by [Qu and Bard, 2015], [Zhang et al., 2015], and [Luo et al., 2019] represent the state of the art in algorithmic methods that have been crafted to enhance the scheduling, routing, and overall management of healthcare transportation services. The development by [Qu and Bard, 2015] of a branch-and-price-and-cut algorithm for vehicles with configurable capacities denotes a concerted strategic approach toward enhancing the efficiency of patient transport operations. In a related manner, the research by [Zhang et al., 2015] in which they develop a memetic algorithm for the Multi-Trip Dial-A-Ride Problem (MTDARP) repre-

sents not only the improvement in operational efficiency but also proposes a novel framework for navigating the complexity of constraints related to patient transportation. Moreover, the two-phase branch-and-price-and-cut algorithm of the innovative model by [Luo et al., 2019] for the Extended Realistic Dial-A-Ride Problem (ER-DARP) underlines that even more detailed features regarding the patient requests and vehicle travel optimization are taken into consideration, making a huge leap in the development of algorithms. These advances underline the potential transformation that algorithmic solutions bring toward the redefinition of the standards of efficiency and responsiveness in healthcare logistics.

The translation of theoretical research into actionable strategies within healthcare systems represents a critical milestone in the movement toward operational efficiencies in patient transportation logistics. The works of [Lim et al., 2017], [Chane-Haï et al., 2020], [Kiechle et al., 2007], and [Melachrinoudis et al., 2007], which are focused on implementation, are just a few shining examples of how academic research can be marshaled to realize practical gains in healthcare delivery. [Lim et al., 2017] design a model for vehicle and healthcare staff scheduling in the Hong Kong public healthcare system. This model is very comprehensive and can achieve the twin targets of maximum service coverage and minimum operational costs. Another similar example is that of [Chane-Haï et al., 2020], which deals with the non-urgent patient transportation problem using a Multi-Trip Dial-A-Ride framework. They find a balance between the much-appreciated, patient-centered care and cost-efficient operations. The innovative technique they apply, truly based on the reality of the patient transportation logistics landscape, enables the realization of the profound impacts that research can make on enhancing the everyday experience for the patient and healthcare provider.

In their paper, [Kiechle et al., 2007] look at this problem by studying the management of a fleet by the Austrian Red Cross for both regular and emergency patient transfers within a non-profit framework. The paper deals with the issue of maintaining solid scheduling in the face of the unpredictable demands of emergency services, which, in turn, require sudden, dynamic reallocation of vehicles. The assumptions here are that the fleet is able to meet both regular as well as emergency needs on an

efficient basis and that vehicles can be reassigned quickly post-emergency use. Another paper, [Melachrinoudis et al., 2007], studies optimization strategies for CAB Health and Recovery Services, Boston, on efficient management of client transportation. Its main focus is on minimizing transportation costs and client inconveniences, such as excessive ride times, by operating under assumptions of predictable vehicle availability and pre-scheduled client times. The study includes a case study comparing the benefits of centralized dispatch against independent scheduling, thereby highlighting the practical significance of theoretical models in enhancing healthcare logistics and patient transportation services.

Together, these comprehensive studies of inter-hospital and intra-hospital transportation represent multidimensional problems and solutions necessary to ensure efficient, timely, and patient-centered transportation between the different healthcare facilities. These studies obviously highlight an interface between logistical efficiency and quality of care: centralized systems, improvements in quality, strategic management, and adaptive planning of schedules, highlighting the importance of innovative logistical solutions. In short, this body of work emphasizes the dynamic interaction of all aspects of healthcare logistics and repeats the importance of the role played by transportation in the improvement of hospital operations and other logistics in the general healthcare delivery system. In the next part of our discussion, we will thoroughly examine the Dial-A-Ride problem (DARP), delving into its complexities and applications in the context of hospital transportation.

2.2 Dial-A-Ride

In this section, we conduct a thorough literature review on the Dial-A-Ride Problem (DARP) to include its static and dynamic aspects to bring out its fundamentals and the methodologies being developed over time. Our search cuts across the DARP applications and focuses on how this problem has been addressed in varied ways within the transportation logistics domain. Thereafter, we narrow our focus to the area of DARP implementation in the context of hospital transportation. This particular area of study is crucial for improving efficiency and responsiveness in healthcare

logistics, where timely and reliable transportation can significantly impact patient care and operational efficacy. By examining DARP within hospital transportation, mostly patient transportation, we aim to uncover insights into the challenges and solutions that researchers and practitioners have identified, emphasizing the adaptations and innovations required to address the unique needs of healthcare facilities and patient services.

The Dial-A-Ride Problem is one of the landmark problems in transportation logistics research. According to [Cordeau and Laporte, 2003a], a Dial-A-Ride system is a type of collective transportation that depends on demand. Each user requests a trip between an origin and a preferred destination, which is coupled to a set of service level requirements. The service provider aims to optimize vehicle routes and schedules while adhering to technical limits for collection and delivery [Parragh et al., 2006].

Nonetheless, a standard definition of DARP is proposed by [Cordeau and Laporte, 2003b]. The problem involves creating a number of low-cost vehicle routes in a complete graph of nodes and arcs, assuming that all customer requirements are met. Nodes relate to users' pickup and delivery locations, supplemented by the vehicle depot. Each directed link between two nodes is represented by an arc, which has a travel time and an associated cost if it is part of the solution. Each route begins and ends at the depot at predetermined time intervals and has a maximum route duration. For those users served in between, service at each location begins within a time window. The maximum user ride time and a vehicle's load cannot exceed its capacity. To create an accurate physical route, users should visit their origin and destination in the correct order and with the same vehicle. A service duration is the time required to load and unload users.

In DARP, the goal is to minimize operational costs while satisfying demand and meeting side limitations. However, service level requirements can also be optimized. Balancing human and economic views in DARP solutions is crucial for providing excellent and efficient transportation for users with specific requirements, including door-to-door services for the elderly and disabled. The Dial-A-Ride systems are becoming increasingly popular in light of the ageing population. They

also serve a social function by reducing the isolation of vulnerable groups in society. As demand grows, however, service providers struggle to manually create vehicle routes and schedules. Planning algorithms are necessary to maintain cost efficiency and service quality [Molenbruch et al., 2017a]. Major contributions to the significant foundational work on DARP are by [Psaraftis, 1980], [Psaraftis, 1983], and [Desrosiers et al., 1986], who develop dynamic programming algorithms to achieve optimal solutions in scenarios that involve a single vehicle. Their pioneering contributions have been majorly influential in further developments, such as broadening our understanding of dynamic routing problems that eventually led to innovations in public and specialized transportation services.

The distinction between static and dynamic Dial-A-Ride Problems could be articulated as follows: A DARP is considered static (off-line) when all the information required to make the decisions lies within the reach of the planner before operations actually start. In such a case, although decisional data may evolve with time, one supposes the planner to be capable of elaborating an exhaustive plan to serve a given user base during the planning horizon. Such a global strategy, which can be materialized as a set of routing guidelines or a list of designated routes and schedules, is predetermined and will not be changed after being applied. On the other hand, the problem is called dynamic (online) when, under the progress of operations, the appearance of information that was not foreseen and the plan may be altered in light of this new information. The problem is, for example, called dynamic if the planner can re-update the existing plans due to the appearance of new users, the changes in available information about current users, or because of unexpected events such as delays and vehicle breakdowns. The measure to which plans can be adapted to the light of new or developing information captures the dynamic feature of the DARP [Ho et al., 2018].

The majority of research on the DARP has been focused on the static variant of the problem, in which all user requests are known ahead of time, and vehicle routes are planned accordingly.

Several studies employ exact methods to solve DARP optimally. Initially, exact solutions were derived via pure dynamic programming [Psaraftis, 1980, Psaraftis,

1983, Desrosiers et al., 1986]. Furthermore, [Cordeau, 2006] and [Ropke et al., 2007] present exact formulations of the Dial-A-Ride Problem (DARP) and introduce branch-and-cut algorithms to find optimal solutions. Column generation, nested within a Branch-and-Price framework, is one of the most effective exact approaches identified in the literature for dealing with the DARP and its different variants. Several studies, including [Bard and Jarrah, 2013], [Bärmann et al., 2024], [Gschwind and Irnich, 2015], and [Bettinelli et al., 2011], utilize this algorithm. Nonetheless, since the DARP is an NP-hard problem [Baugh Jr et al., 1998], the majority of problems of practical size are addressed utilizing approximate solution methods. This classification emphasizes the enormous computing hurdles associated with developing efficient DARP solutions. For the static Dial-A-Ride Problem (DARP), metaheuristic techniques, particularly Tabu Search, are among the most effective solution strategies. Tabu search is versatile and effective in addressing the Dial-A-Ride Problem as shown in [Cordeau and Laporte, 2003b], [Aldaihani and Dessouky, 2003], and [Melachrinoudis et al., 2007], among others. Building on the foundational techniques, [Toth and Vigo, 1996] propose an insertion procedure, which was later improved by a Tabu thresholding algorithm proposed by [Toth and Vigo, 1997] to refine initial solutions. Exploring alternative heuristics broadens the scope of strategies against DARP's challenges, with simulated annealing and genetic algorithms making significant contributions, as evidenced by the work of [Baugh Jr et al., 1998] and reflected in studies by [Uchimura et al., 2002] and [Jorgensen et al., 2007].

Early research on DARP primarily concentrated on its dynamic aspect, as highlighted in seminal works by [Wilson et al., 1971] and [Wilson and Colvin, 1977]. Despite the initial focus, the dynamic variation of DARP has not received as much attention as its static cousin in recent years. The insertion algorithm developed by [Madsen et al., 1995], inspired by the foundational work of [Jaw et al., 1986], marks a significant advancement by applying this method to handle a complicated, real-world dynamic DARP with multiple objectives. Building on these achievements, [Coslovich et al., 2006] introduce a revolutionary two-phase insertion technique to speed the process of reviewing user requests, thus facilitating rapid decision-making regarding acceptance or rejection.

In experimental studies, dynamic DARPs are frequently examined through simulation models in which decisions influence subsequent system states [Häll et al., 2015, Quadrifoglio et al., 2008]. Numerous studies emphasize the accommodation of new user needs as the main driver for the replanning process [Hanne et al., 2009, Berbeglia et al., 2012, Häll and Peterson, 2013]. Nonetheless, certain studies investigate alternate triggers, like vehicle breakdowns or driver rest periods [Beaudry et al., 2010].

In dynamic environments, where conditions can change quickly, the need for faster algorithmic response times becomes critical, distinguishing operational requirements from those in static contexts, where algorithms can operate for extended periods of time. To meet this desire for speed, parallel computing emerges as an intuitively appealing alternative, utilizing the capacity of numerous processors to do computations concurrently, dramatically lowering overall computing time. In this vein, [Attanasio et al., 2004] make a substantial contribution by offering various parallel variants of the Tabu Search algorithm first proposed by [Cordeau and Laporte, 2003b]. This unique adaptation illustrates a purposeful shift toward using computational advances to improve the responsiveness and efficiency of solutions in the continually shifting terrain of DARP challenges.

For a detailed examination of various solution strategies and their implementations, we direct readers to the surveys conducted by [Ho et al., 2018],[Molenbruch et al., 2017a], and [Cordeau and Laporte, 2007].

2.2.1 Dial-A-Ride in Hospital Transportation

The problem examined in this study falls under the general area of pickup and delivery challenges, notably the Dial-A-Ride Problem (DARP). DARP plays an important role in improving hospital transportation systems, which has a substantial impact on patient care quality and operating efficiency. By optimizing vehicle scheduling and routing for hospital transfers, DARP allows healthcare organizations to manage their transportation resources better, ensuring that patients and medical suppliers are moved quickly and safely across departments or hospital networks.

This optimization is critical for minimizing patient wait times, which has a direct impact on patient satisfaction and the overall treatment experience. Furthermore, strategic adoption of DARP solutions aids in the reduction of transportation costs and the optimum utilization of available porters, which is especially significant in resource-constrained healthcare settings.

According to [Ho et al., 2018], using DARP in intra-hospital transportation efficiently handles the complicated logistics of transferring patients, supplies, and equipment, addressing specific healthcare constraints, and prioritizing care requirements. The DARP distinguishes itself from traditional pickup and delivery scenarios by emphasizing the least amount of inconvenience experienced by users. This aim is often met by limiting each user's travel time (the amount of time spent in the vehicle), the additional ride time (the difference between the actual and shortest possible ride time for a user), and any deviations from recommended pickup and delivery times. Balancing service quality improvement against the reduction of fleet operational expenditures is critical, with the latter encompassing issues such as the number of porters used, the total duration of routes, and the cumulative distance covered. [Beaudry et al., 2010].

The papers [Melachrinoudis et al., 2007], [Kiechle et al., 2007], and [Hanne et al., 2009] are among the first to examine the DARP in patient transportation, both within and between hospitals, setting the groundwork for future research in this area. Their pioneering work has made a substantial contribution to the development of DARP applications in healthcare settings, highlighting the potential for efficient transportation logistics to improve patient care and hospital operating efficiency.

Several studies on Static Dial-A-Ride Problem applications in hospital transportation offered separate but complementary techniques. [Molenbruch et al., 2017b] and [Melachrinoudis et al., 2007] investigate DARP in terms of patient comfort and cost efficiency. [Molenbruch et al., 2017b] use a Multi-directional Local Search (MDLS) algorithm in a Variable Neighborhood Descent (VND) framework to improve patient transportation logistics by combining limits for health safety and specialized medical demands. A case study with a Belgian transportation service demonstrates the efficiency of this technique. [Melachrinoudis et al., 2007] examine

a static DARP model, offering a double request Dial-A-Ride model with soft time windows targeted at minimizing transportation expenses and customer discomfort for CAB Health and Recovery Services, Inc. Both studies emphasize the value of personalized logistical techniques that address unique operational and patient-centered restrictions.

[Tóth et al., 2024] uses the DARP framework to improve patient transportation logistics to hospitals in non-urgent circumstances. It presents four Mixed-Integer Linear Programming (MILP) models that optimize the routing of a single vehicle by minimizing travel distance while adhering to patient-specific constraints such as journey durations and vehicle capacity. The models are evaluated using real-world situations and a commercial solver, with results demonstrating varying efficiency among the models, pointing to possible advances in integrating public transportation affordability with the convenience of private taxi services.

Moreover, The Multi-depot Dial-A-Ride problem (MD-DARP) model developed by [Detti et al., 2017], employing a Mixed Integer Linear Programming (MILP) approach, efficiently optimizes the non-emergency transportation of patients using heterogeneous vehicles, considering compatibility constraints and operational requirements, which significantly improves real-world patient logistics.

[Zhang et al., 2015] and [Chane-Haï et al., 2020] use DARP to effectively schedule non-emergency transportation services. [Zhang et al., 2015] addresses the Multi-Trip Dial-A-Ride Problem (MTDARP), improving ambulance logistics by permitting multiple trips per shift while handling constraints such as time windows, route lengths, and obligatory disinfection periods between rides. In contrast, [Chane-Haï et al., 2020] create a mathematical model concentrating on a Multi-Trip Dial-A-Ride Problem, where each patient's transportation is separated into morning and evening trips connected by a maximum daily ride time limit, seeking to minimize both cost and excess travel time. These studies show novel approaches to improving service quality and operational efficiency using careful scheduling and shared constraints.

Dynamic Dial-A-Ride Problems, on the other hand, pose particular problems in hospital transportation because of their real-time operating demands. Several researchers have made major contributions to tackling these difficulties using novel

methods and approaches. [Hanne et al., 2009] explores intra-hospital patient transportation in the German healthcare system, creating Opti-TRANS. This transportation planning system employs a dynamic Dial-A-Ride approach to manage the logistical complexities of patient transport demands, resulting in significant operational gains and strategic enhancements to patient care quality through logistical innovation.

In an analogous method concentrating on the real-time demands of patient transportation, [Kergosien et al., 2011] use a dynamic DARP formulation at the hospital complex of Tours. This work uses a Tabu Search algorithm with adaptive memory to dynamically improve transportation logistics, efficiently managing unexpected transportation demands by prioritizing urgent requests and including vehicle disinfection times.

[Kiechle et al., 2007] investigates the dynamic DARP within the Austrian Red Cross, which addresses both regular and emergency patient transportation demands. Their research emphasizes the significance of a flexible and dynamic scheduling system that can adjust to vehicles' unexpected disappearance and reappearance, intending to balance routing costs and emergency service response times.

[Luo et al., 2019] suggests a mathematical model to optimize patient transportation. Their adaptation of the Realistic Dial-A-Ride Problem (R-DARP) includes a two-phase branch-and-price-and-cut strategy that accounts for dynamic operational constraints such as time windows and changeable vehicle capacity. This strategy efficiently prioritizes transportation requests while shortening trip durations, outperforming earlier strategies, especially in the dynamic hospital transportation scenario.

Finally, in [Beaudry et al., 2010], a dynamic DARP model is improved for hospital settings, taking into account various logistical elements like patient isolation, disinfection processes, and heterogeneity in vehicle fleet management. Their two-phase technique, which includes initial request insertion and Tabu Search-based optimization, exemplifies developments in dynamic scheduling tactics for increasing the efficiency and responsiveness of hospital patient transportation systems.

2.3 Applications of Autonomous Robots in Hospitals Transportation

The incorporation of ARs into healthcare presents a significant potential for addressing challenges related to patient transportation, drugs delivery [Søraa and Fostervold, 2021, BačÍK et al., 2017]. This robots has the potential to reduce wait times, enhance patient safety, minimize the spread of disease, increase staff productivity, substantial cost savings and redefines the landscape of modern healthcare [Marmaglio et al., 2023, Khalid et al., 2021, Holland et al., 2021]. One example of such robots is illustrated in Figure 2.1.

Figure 2.1: (Color online) A service robot designed by FDATA for delivering supplies efficiently and safely in the hospital [Fdata Co., 2025].

Although utilizing ARs in delivery is a relatively new concept, many studies of research have been undertaken on the application of robots for daily delivery requests (see, e.g., [Ermagan et al., 2023, Jeong and Moon, 2024, Yu et al., 2020]). Nonetheless, the research on optimization of transportation systems integrated with ARs, especially AMRs, is relatively limited. Some of the papers that have directly addressed the optimization of the transportation system within hospitals using robots

are as follows.

[Cheng et al., 2023] and [Fanti et al., 2020] examine route planning for heterogeneous service robots. [Cheng et al., 2023] take heterogeneous service robots with battery consideration into account in Singapore Changi General Hospital. They consider a stochastic mixed-integer programming model with the aim to optimize the routes for charging and service requests, including medication distribution, meal deliveries, and waste collection. The findings demonstrate that the efficient use of service robots may substantially decrease total hospital expenses while satisfactorily fulfilling medical needs. Likewise, [Fanti et al., 2020] emphasize the optimization challenges in integrating robots for in-hospital drug delivery, focusing on efficiency and minimizing human involvement. They use a Dynamic Pickup and Delivery Problem model, resolved by a rolling horizon technique integrated with a column generation (CG) algorithm. The article reports a reduction of around 35% in travel distance, hence improving operational efficiency while maintaining service quality.

In the study by [Jin et al., 2021], a novel mixed refilling model combining a multiple steps traveling salesman problem model (MTSPM) and a vehicle routing problem model (VRPM) are proposed to optimize the refilling routes in medical logistics systems. This model is tested under two scenarios, temporary replenishment mode (TRM) and concentrate replenishment mode (CRM). Using Ant Colony Optimization (ACO), results indicate substantial reductions in route durations, achieving 32.18% and 58.32% in temporary and concentrate replenishment modes, respectively.

Furthermore, [Bernhard et al., 2023] and [Aziez et al., 2022] focus on the fleet management and sizing of robots in hospital settings. In order to determine the optimal fleet size that balances operational costs and workload efficiency in operating room wings (OR wings), [Bernhard et al., 2023] present Autonomous Self-Navigating Robotic OR Assistance (AURORA), focusing on managing a fleet of homogeneous robots that are tasked to execute generic delivery tasks within the non-sterile areas of operating room wings (OR wings). This study utilizes simulation to evaluate 150 scenarios, including diverse fleet compositions, robot velocities, and workloads, based on the real arrangement of a German university hospital's OR wings. Building on these insights, [Aziez et al., 2022] introduce a fleet sizing and routing problem

with synchronization for Autonomous Guided Vehicles (AGVs) with dynamic demands (FSRPS-AGV) to optimize the number and types of carts and AGVs needed to execute all daily requests within a hospital. This problem handles different requests consisting of several pickup and delivery tasks with operation synchronization among the tasks of the same request and movement synchronization with respect to AGVs and carts. The authors propose an even-based rolling horizon approach with Adaptive Large Neighborhood Search (ALNS). The research indicates that minor adjustments in scheduling limitations and less conservative speed estimations might provide substantial enhancements in performance and cost reductions, therefore increasing the efficiency and quality of patient care in hospital logistics.

Furthering the discourse on fleet management, [Guzmán Ortiz et al., 2021] outline the development of a Fleet Management System (FMS), incorporating a routing engine, task scheduler, Endorse Broker, controller, and backend API. This research seeks to optimize logistics operations in a practical hospital environment by the strategic planning and management of task execution with a fleet of mobile robots. Task allocation and trolley loading challenges are addressed by Integer Linear Programming (ILP) for smaller problem sizes and a Genetic Algorithm (GA) for larger scales, with dynamic path planning identified as a significant obstacle, validating the importance of using robots in hospitals.

Lastly, [COURIER, 2000] leverages simulation to analyze the cost and performance trade-offs associated with deploying a fleet of robotic couriers in hospital settings, particularly for clinical laboratory and pharmaceutical delivery. The simulation results indicate that the robotic fleet not only meets but substantially improves performance, decreasing turnaround times by 34% and delivery variability by 38%.

The aforementioned studies support the integration of robots into hospital operations, which can not only make improvements in operational efficiency but also reduce operational costs. Despite the contributions of the papers in this domain, to the best of our knowledge, the deployment of service robots alongside humans in the optimization of intra-hospital transportation has not been studied comprehensively, identifying areas for further research.

For additional information surrounding integrating robots in the hospitals, please

see [Cruz et al., 2024].

2.4 Impacts of RFID Technology on Healthcare Logistics and Patient Care

Healthcare professionals face significant challenges in mitigating adverse events and improving patient safety [Yao et al., 2010]. Adverse events refer to complications arising during a patient's hospitalization that are unrelated to their prior disease [Martinez Perez et al., 2018]. Such circumstances can produce substantial consequences for patients, their families, and the healthcare system. Implementing traceability, which involves tracking information from origin to delivery, can enhance patient safety. In healthcare, this involves the accurate identification of patients, medications, and their interactions, thereby reducing adverse events and improving overall treatment outcomes.

Radio Frequency Identification (RFID) technology provides an efficient approach for real-time tracking and automatic identification of patients and medications, thereby reducing errors and improving patient safety. RFID is a technology that uses radio waves for the efficient, automatic, and real-time collecting and transfer of data, eliminating the need for human engagement. This technology, whether utilized independently or alongside other technologies, has been seen as an opportunity to mitigate issues that threaten public health or to enhance its management [Profetto et al., 2022].

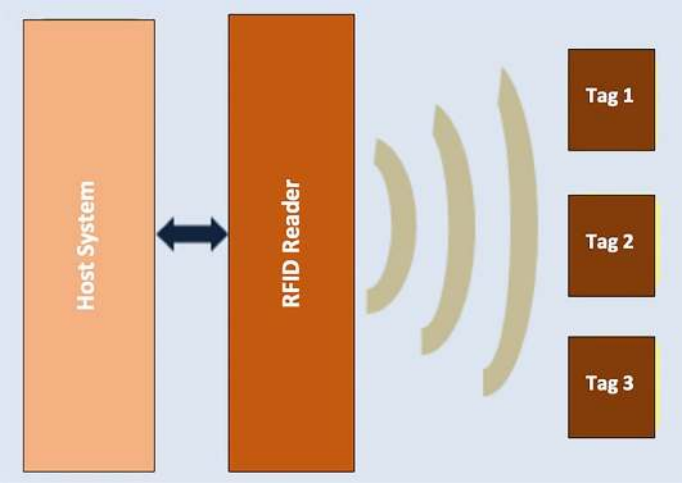


Figure 2.2: (Color online) An RFID system consists of an RFID reader that receives data from one or several tags and transfers this information to a host system. [Profetto et al., 2022].

According to [Profetto et al., 2022], RFID devices vary in electromagnetic transmission configurations tailored for specific applications, comprising components such as RFID tags, readers with antennas and transceivers, and host systems or connections to business systems (Figure 2.2). The tags, embedded with electronic circuits and antennas, attach to items and carry unique identifiers alongside memory that stores product-related data. The RFID reader retrieves and decodes data from tags, interacting via RF waves facilitated by its antenna, and can both read from and write to tags. This data management is handled through software that coordinates the reader and tag operations and stores information in databases [Kumari et al., 2015]. The system connects to host computers or middleware, linking the RFID setup with organizational systems [Camacho-Cogollo et al., 2020]. Tags come in active, passive, and semi-passive forms [Kumari et al., 2015]. Active tags, which include sensors and have their power sources, boast large memories and operate over longer distances but are costlier and bulkier. Passive tags, powered by the reader's magnetic field, are cheaper, lighter, and smaller, yet have limited range and storage capacities. Semi-passive tags, powered by batteries, remain dormant until activated by the reader, prolonging their lifespan [Camacho-Cogollo et al., 2020].

RFID technology in hospitals significantly enhances operational efficiency and patient safety. Real-time locating systems (RTLS) monitor the movements of patients, personnel, and equipment, guaranteeing the efficient utilization of medical resources and their availability when required. RFID is crucial for managing medical inventories, reducing equipment theft or loss, and improving the accuracy of medical procedures by ensuring that the correct tools are available at the exact location and time. Moreover, RFID technology reduces medical errors in operating rooms by tracking surgical supplies and instruments, improves patient identification processes to minimize medication errors, and aids in infection prevention and control by monitoring changes in wound conditions through advanced sensors [Profetto et al., 2022].

Recent breakthroughs in RFID technology have profoundly influenced hospital logistics and patient care. In their study, [Bochem et al., 2022] develop a comprehensive RFID system alongside a smartphone application to enhance the management of medical devices in hospitals that allows medical personnel to quickly locate, reserve, and manage medical equipment. Moreover, [Geary et al., 2022] evaluate the implementation of RFID technology in enhancing the transport of critical laboratory samples in a hospital, using the CUBE socio-technical systems analysis. Their results demonstrate that RFID has improved the reliability and traceability of sample transport. [Ruan et al., 2018] illustrate how Hartford Hospital in the United States employs passive RFID tags for the identification of telemetry transmitters and critical care items like prefilled syringes and beds. This RFID integration markedly enhances the rapid identification of surgical instruments, hence improving patient safety and minimizing repair durations. Furthermore, [Chit et al., 2021] examines the utilization of RFID in specimen logistics to improve the tracking of blood tubes and guarantee the precision of environmental conditions throughout transport, which results in increased operational efficiency. Similarly, [Tsai et al., 2019] create an RFID-based system utilizing location-based services for continuous, real-time monitoring of medical equipment. The designed system significantly improves the accessibility and utilization of hospital resources. These examples collectively demonstrate the transformative potential of RFID technology in enhancing hospital

operations and elevating the quality of patient services.

Despite all the advantages brought by RFID technology to the hospital system, before the implementation of an RFID-based smart hospital system, several limitations regarding the use of RFID technology in hospital settings must be acknowledged. According to [Haddara and Staaby, 2018], the first constraint to be considered is that RFID may produce electromagnetic interference (EMI) with specific medical devices, potentially compromising patient safety in certain hospital environments. Medical devices may disrupt the accurate reading of RFID tags due to electromagnetic interference (EMI) or the presence of substantial metallic objects. Furthermore, implementing such a system poses concern in the security and privacy of data [Munoz-Ausecha et al., 2021]. This integration raises vulnerabilities, including unauthorized access and data tampering, that threaten the confidentiality and integrity of sensitive medical information. These vulnerabilities must be rigorously addressed through enhanced security protocols and privacy safeguards, such as advanced encryption, strong authentication, and privacy-enhancing technologies, to ensure the safe deployment of RFID technologies in healthcare settings [Abdulghani et al., 2022].

2.5 Comparison of Our Study with Related Studies and Our Contributions

The literature review on real-time intra-hospital transportation presented in the previous section underscores critical gaps where most recent studies seem to have sacrificed optimal responsiveness and efficiency in hospital transport systems. A notable limitation is that the majority of existing research focuses solely on patient transportation. However, other items that are routinely transported as part of the hospital's daily operations, such as medications, paperwork, and other kinds of medical supplies, are not handled with the same level of caution. Moreover, most studies have focused primarily on two objectives: minimizing costs or travel times and enhancing patient satisfaction. However, a clear and significant gap exists in considering fairness among hospital porters. Addressing this aspect could help bal-

ance porters' workloads, fostering a more productive work environment where staff are engaged and discrimination is actively mitigated. Additionally, the literature highlights a lack of effective penalty functions for delays. Most research merely incorporates time windows for each request, which may fail to capture the complexities of real-world operations, particularly in large urban hospitals. One of the complicating aspects is formulating specific time windows poses significant complications. Moreover, due to hospitals' inherent need to fulfill all requests without refusal, this can lead to infeasible solutions and increasing the number of workers. Therefore, this assumption may be unrealistic.

Another significant gap in the literature is the lack of study targeting the incorporation of service robots into hospital transportation systems and the enhancement of their operations. Service robots have the potential to enhance efficiency and reduce human labor. Nevertheless, their application in intra-hospital logistics remains largely unstudied. Furthermore, most studies do not include consolidation, wherein a porter or service robot can simultaneously handle many transportation requests rather than completing one before starting another. This limitation may lead to inefficient scheduling, increased delay time, and suboptimal resource utilization. Finally, whereas optimization methods in this field often rely on either exact or approximate solution methods, there has been a research gap into the application of Reinforcement Learning (RL) for real-time decision-making in intra-hospital transportation. Reinforcement learning techniques have considerable adaptability and scalability, particularly in dynamic environments characterized by high uncertainty. Addressing these issues could significantly improve the timeliness, efficiency, and automation of intra-hospital transportation systems.

So, the contributions we hope to make in this thesis are as follows:

- We present a comprehensive mathematical formulation for the static version of the intra-hospital transportation problem that explicitly integrates both service robots and human porters, considering diverse and complex transportation requests and specific recharging requirements for robots. Unlike previous models, our approach uniquely incorporates detailed characteristics of differ-

ent request types, including on-demand patient transportation, scheduled patient transportation, medical supplies, medical equipment, and administrative documents, allowing for a realistic and holistic representation of hospital operations.

- We introduce an effective workload balancing mechanism designed specifically to enhance job satisfaction among porters and reduce potential operational discrimination. This consideration is pivotal in practical hospital settings, as equitable distribution of tasks is known to directly impact employee motivation and service quality.

- We implement a novel convex, non-decreasing delay penalty function, tailored specifically to address the urgency and criticality of different request types. By doing so, our model accurately reflects the real-world implications of delays within hospital transportation, emphasizing the urgency and prioritization inherent in patient care contexts.

- Our research introduces a novel hybrid solution approach, combining advanced heuristic methods with Deep Reinforcement Learning (DRL). Specifically, we develop a two-phase framework that separates initial assignment decisions and subsequent routing optimization, leveraging Bayesian Deep Q-Learning in the first phase to dynamically and robustly handle real-time assignment decisions, followed by heuristic routing strategies, including Best Insertion and Adaptive Variable Neighborhood Search (AVNS), to efficiently address the routing problem in the second phase.

- Our study introduces a novel hybrid intra-hospital transportation system that integrates both Autonomous Mobile Robots (AMRs) and human porters within a unified operational framework. Unlike prior work that focuses exclusively on either automated or manual systems, we explicitly model and optimize the coordinated use of both agents. This hybrid structure leverages the efficiency and consistency of AMRs for standardized tasks while preserving the adaptability

and patient-centered capabilities of human porters. Furthermore, the incorporation of Radio Frequency Identification (RFID) technology enables real-time tracking and dynamic dispatching, significantly enhancing the responsiveness and reliability of the system.

- We address consolidation by allowing transporters to handle multiple requests simultaneously without stringent sequence constraints, thus significantly improving operational efficiency and reducing unnecessary movements. This feature is particularly novel, enhancing flexibility and resource utilization within the transportation system.

Appendix A provides a comparative literature analysis according to some key characteristics, which helps to clarify the research gap and further contributions of this study.

Chapter 3

PROBLEM DEFINITION

This chapter describes RIHTP. The overall goal is to improve the efficiency of porter and service robot circulation within the hospital, consequently streamlining the assignment process. This ensures that transportation requests are routed to the most appropriate porters or service robots, thereby harmonizing with numerous aims.

Objectives for hospital transportation are divided into two main areas: one of them is the operational cost, which is affected by costs on porter and service robot travel time and the number of porters and service robots taken in by shift, making it necessary for the management to improve the efficiency by reducing the actual number needed. The second category pertains to the effectiveness of customer service by cutting down on the time each request spends before being transported, making it available for them, and the riding time for a request.

The trade-off between two competing objectives measures route effectiveness. The first one is to reduce as much as possible both direct and indirect operating costs. The other one is that patient inconvenience needs to be minimized. Major factors that contribute to the operating costs of the transportation department are travel times. Travel time for patient trips is the sum of the actual movement time and time spent assisting the patient, such as time in boarding the transporter. The travel time is considered even for empty trips, that is, without a patient. Breaches of time windows do not only cause inconvenience to patients but also negatively impact costs for hospitals; late pick-ups at the origin cause patient waiting time, while early arrivals mean idle time for transporter crews, thus increasing direct expenses. At the point of arrival, early arrivals are prevented by the time-window settings, but late deliveries worsen patient inconvenience and increase hospital indirect costs. This is even more critical if the delivery location is a service department (e.g., an

operating theater), as it directly translates into underutilization of equipment and staff. Additionally, late arrivals disturb the planned schedule at the service unit, potentially leading to the patient suffering from very long waiting times.

This research presents an adaptable model and technique designed to improve intra-hospital mobility in diverse healthcare environments. The methodology is scalable and adjustable, guaranteeing alignment with the distinct operational requirements of various healthcare facilities. However, to show the efficiency of our methodology, the case study is conducted at one of the hospitals located in Turkey, which is an extremely large medical facility with 373 distinct zones that handle around 1,000 transportation requests per day, which consist of a wide range of transportation requests, including patient, medical equipment, medical suppliers, and administrative document delivery. Each zone within the hospital is connected by varying distances measured in time units, which affect the travel time necessary for each transportation request.

In the hospital studied in this thesis, we have the subsequent goals. First and foremost, we want to lower the overall amount of time porters and service robots must walk, which has a direct influence on operational costs. Secondly, we seek to create a fair distribution of requests throughout the entire planning period, promoting equity among the porters. Third, our goal is to reduce the penalties associated with starting a transportation request later than a fixed allowed time, which is critical for maintaining high levels of satisfaction with patients. Achieving these objectives could not only significantly reduce operational inefficiencies but also make an excellent contribution to balancing the workload of porters and improving the overall patient experience.

To make this operation effective, the hospital incorporates RFID technology to track the movement of porters within the hospital. This technology assists the dispatcher in determining and appointing the porter closest to the pickup point of a request. Although request assignment by dispatcher can be effected manually, it is inefficient and often makes terrible assignments, resulting in long completion times. In this light, further improvement of the assignment process is crucial to raising general efficiency and reducing operational stress on porters and dispatchers.

In the subsequent sections, we delve into the important components of RIHTP. The primary question for each of these components is to choose appropriate assumptions and how implement them for our experiments. These elements are as follows:

- Specifying request features
- Characterizing the porter, service robots, and dispatcher's operational features
- Modeling the hospital network
- Defining dynamic Events

3.1 Transportation Requests

A transportation request, as in the standard Dial-A-Ride Problem (DARP), includes an origin point for patient pickup, a destination point for patient drop-off, a preferred time for pickup or delivery, and the style of transportation required (e.g., wheelchair or stretcher). However, in RIHTP, we have faced a variety of transportation requests, such as medical suppliers, medical equipment, and documents, as well as patients.

Transportation requests at this hospital are classified according to the nature of the request, which includes transporting patients (both scheduled and on-demand), administrative documents, medical equipment, and medical supplies such as blood tubes and pharmacy items. Every request is issued an Event Parameter ID, which corresponds to the event type. Each request includes a pickup point and one or two delivery points. A request is considered complete when the assigned porter has visited all of the designated pickup and delivery points for that request. However, in order to make the model more effective, we categorize transportation requests into three classes. Type 1 requests are for requests that do not require any paperwork or waiting time, meaning they have just one pickup and one delivery point. For example, deliver medical supplies between two locations. Porters or service robots can handle these requests while still managing other transportation requests; that is called consolidation, allowing them to optimize their workload. Type 2 requests,

on the other hand, contain paperwork, which requires handling at both the pickup and delivery locations. Similar to Type 1 requests, they can be taken on additional transportation tasks between pickup and delivery places, as long as the same porter controls the entire procedure at both ends. Type 3 requests require the porter to wait at the initial destination. A common example is transferring patients to treatments such as MRI scans, where the porter must wait until the procedure is completed before returning the patient to the original or another location. The porter is unable to conduct any other duties while waiting. Note that this request type can only be fulfilled by porters, not service robots.

Figure 3.1 shows that each request has an origin point (pickup) and one or two drop-off points (delivery). To accomplish a Type 1 request, the porter must go from the pickup point to the delivery point. This category usually comprises delivering patients or administrative paperwork in addition to medical equipment, which is a simple task that requires only one trip from source to destination. Type 2 requests are slightly more sophisticated. For example, either porter must go to the pickup point to retrieve a blood tube and associated paperwork. The porter then goes to the initial delivery location, obtains a signed document, and returns it to the original pickup point. This sort of request includes an intermediary phase that necessitates more coordination and time management. Type 3 requests are much more complicated. In this scenario, the porter delivers a patient from the original pickup location to the first delivery destination. In this case, the porter must wait until the patient's procedure is completed before returning the patient to the original. This type of request requires the porter to remain inactive during the procedure, limiting their capacity to perform other activities during that time.

In addition to the aforementioned features, there is a capacity type associated with each request that restricts the number of requests that a porter can handle simultaneously or can be consolidated from each of these classes. We divide the requests into two classes based on their capacity type. Class 1 contains the request to transport a patient or medical equipment, while Class 2 includes transporting administrative documents or medical supplies. This categorization will help us better formulate the problem. The information regarding each request is summarized in

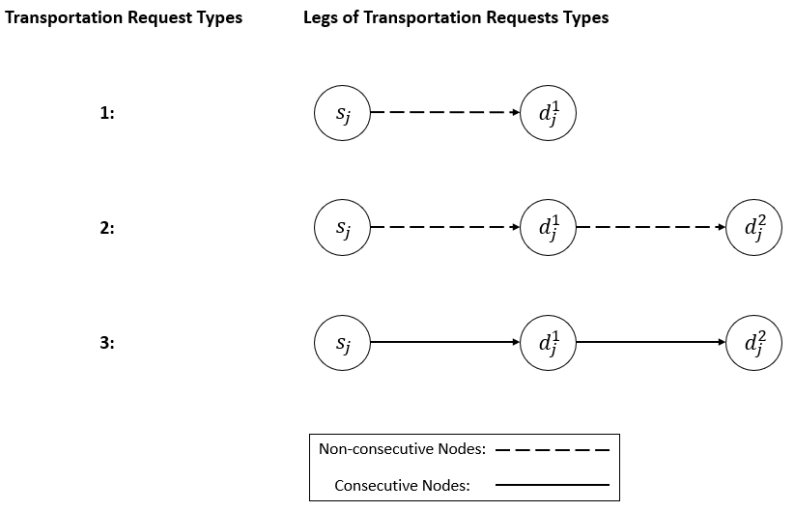


Figure 3.1: Transportation request types and their node representations

Table 3.1.

Table 3.1: Summary of requests by type, capacity, urgency, and difficulty

Request Name	Type	Capacity Type	Request Urgency	Request Difficulty
On-demand Patient Transportation	Type 1	Class 1	High	High
Administrative Document	Type 1	Class 2	Low	Low
Medical Equipment	Type 1	Class 2	Medium	Medium
Medical Supplies	Type 2	Class 1	Low	Low
Scheduled Patient Transportation	Type 3	Class 1	High	High

The hospital's protocol states that the delay is only attributed to the arrival time at the pickup point of request $j \in J$. In accordance with the policy, it is required to arrive at the pickup points for patient transportation requests within 10 minutes and for other types of requests within 15 minutes. Consequently, should it fail to reach the pickup point within the stipulated times, a penalty is incurred. This penalty increases linearly in proportion to the time that passes after these time limits.

We establish a complete set of attributes for each transportation request j, specifying the task's requirements and constraints. A generic request is represented as a tuple $j = (s_j, d_j^1, d_j^2, c_j, i_j, k_j, t_j)$, where $s_j \in P$, , $d_j^1 \in D_1$ and $d_j^2 \in D_2$, $c_j \in \{1, 2, 3\}$, $i_j \in \{1, 2, 3\}$, $k_j \in \{1, 2\}$, and $t_j \geq 0$ are the pickup point, the delivery points, the

request type, the priority type, the capacity type of that request, and the occurrence time or arrival time of the request, respectively. Note that for some requests, d_j^2 might be empty.

Incurred inconvenience of request $j \in J$ started at time $A_{s_j v}$, which is the arrival time at the pickup point of that request is expressed by the non-decreasing and convex penalty function $p(t_j, b_j, A_{s_j v}, \beta_j, \alpha, \tau)$. To accurately model the penalty function, we have selected a generalized piecewise function that can consist of k intervals. During these periods, the penalty associated with delay exhibits a linear increase. In the following intervals, the slope of the linear function increases by a constant percentage of its preceding coefficient value. Considering the particular attributes and operating protocols of the hospital in issue, it has been concluded that modeling three periods sufficiently represents the situation at this healthcare facility. This function is dependent on β_j, the penalty for each time unit of delay, which is considered based on the priority type, b_j, the duration of time after which the penalty begins to be counted, and τ is the breakpoint where the slope of the linear function increases by $\alpha\beta_j$ as proposed by [Ghiani et al., 2022]. Figure 3.2 shows a graphical representation of the penalty functions. Also, the mathematical formulation of the penalty function is shown below:

$$p(t_j, b_j, A_{s_j v}, \beta_j, \alpha, \tau) = \begin{cases} 0 & t_j \leq A_{s_j v} < t_j + b_j \\ \beta_j(A_{s_j v} - (t_j + b_j)) & t_j + b_j \leq A_{s_j v} < t_j \\ & +b_j + \tau \\ \beta_j \tau + (1+\alpha)\beta_j(A_{s_j v} - (t_j + b_j + \tau)) & A_{s_j v} \geq t_j + b_j + \tau \end{cases}$$

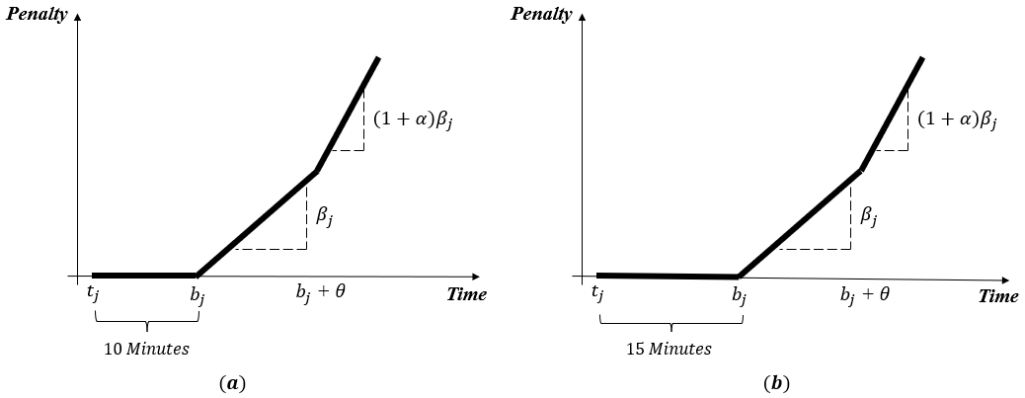


Figure 3.2: Penalty function to compute the inconvenience associated with a request (delay). (a) patient transportation (b) medical suppliers, medical equipment, and administrative documents

3.2 Porters, Service robots, and Dispatcher

In the studied hospital, a homogeneous fleet of workers was responsible for transportation tasks. Porters are employees in hospitals who are responsible for delivering related items or patients based on the request. In other words, they collect the material or patient from the first node, marked as the pickup point, and deliver it to the end node(s), denoted as the delivery point(s). Each porter begins his day at the depot and returns to it at the end of the day. The current system employs roughly 100 porters. Each porter is available for a specific time period of the day or shift, with one or more programmed service interruptions of a specified duration (the hospital has three shifts). As a result, each shift has less than 100 porters working. Furthermore, since the studied hospital is enormous, we have two separate groups of porters who start at distinct depot nodes. As mentioned before, when a request is assigned to porters, they have 10 minutes to be at the pickup location for patient transfer and 15 minutes for other requests. In the case of arriving late at the pickup location, the porters will be penalized by the central monitoring system.

A central dispatcher manages the assignment process, which includes giving transportation assignments to porters all around the hospital. Leveraging RFID technology, the hospital improves the efficiency of this process. RFID readers are

positioned at the entrances to each zone, collecting porter arrival times as they pass through those spots. In other words, when a porter enters a zone in the hospital, this technology collects the passing time. This data is archived in the hospital's detection system, allowing for real-time tracking of porters' whereabouts. When a new transportation request is generated, the system identifies porters in proximity to the pickup point, displaying this information to the dispatcher. The dispatcher then manually assigns the request to a porter, relying on their judgment and the real-time data provided by the RFID system. In the current operational framework, dispatchers follow a sequential order for request assignments, ensuring that porters complete their current tasks before proceeding to the next. This approach, however, does not account for dynamic reallocation based on evolving conditions and demands. The dispatcher's decisions are generally intuitive and based on experience rather than an optimal analytical method. This strategy, while functional, frequently results in inefficient task allocation and increased operational inefficiencies. Figure 3.3 indicated the current process of assigning a new request to a porter.

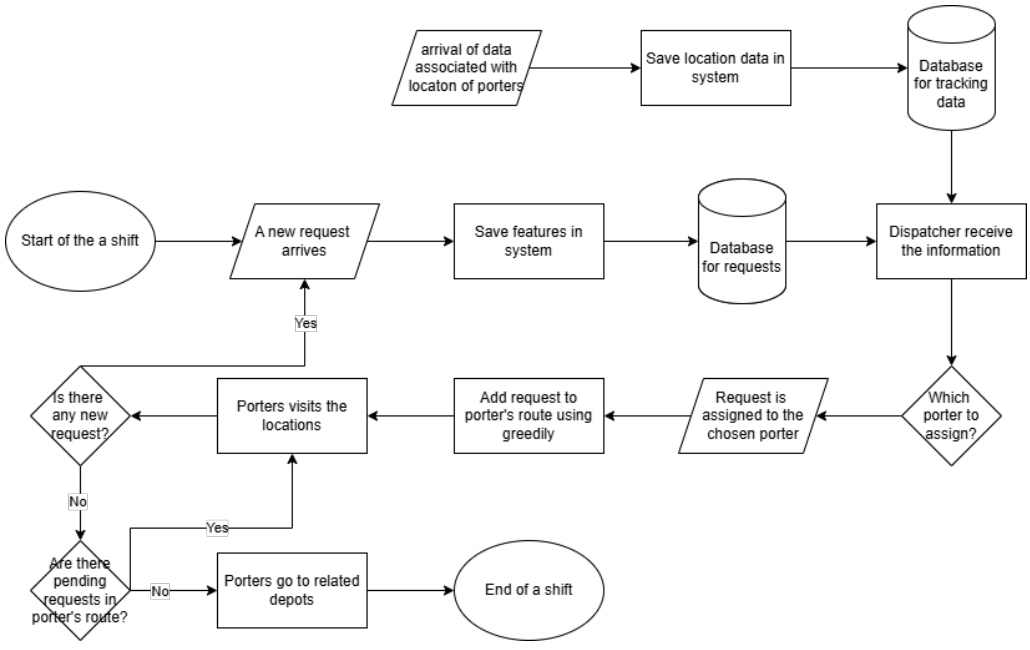


Figure 3.3: Flowchart of the current dispatching system

Given that there are plenty of requests in a day in the hospital, this system

must incorporate an assignment system that takes into account the porters' closest path in terms of duration between points as well as their capacity for transporting hospital materials and patients, rather than just their availability. As a result, the current approach is inefficient for porters in terms of time management and task balance, particularly during the hospital's peak hours.

The current system only relies on porters, excluding robotic assistance. According to recent advancements and the empirical evidence presented in Chapter 2, emphasizing the significant benefits of incorporating service robots into the hospital's transportation system, we consider it more effective to integrating service robots in the current system, introducing a human-robot operational model.

The service robots under consideration are equipped with secured storage boxes specialized to handle administrative documents and medical supplies only. The service robots are considered to function as porters, with the difference that they possess a greater capacity to manage these requests. Additionally, these service robots require periodic recharging, demanding proactive planning to maintain operational continuity without disrupting healthcare services.

For the recharging strategy, we apply a newly devised partial recharging strategy to perform battery management proposed by [Jeong and Moon, 2024]. This strategy is similar to the well-known (r, Q) policy used in inventory management. A specified amount RL is incurred if the battery level falls below a designated threshold BL following the fulfillment of an order, where BL represents the base level and RL denotes the recharge level. The battery level in this context refers to the duration a service robot can operate before completely shutting down. This parameterized policy is referred to as (BL, RL) policy. Figure 3.4 illustrates how the policy operates. Nevertheless, we have implemented a modification to the recharging policy. Specifically, due to the service robots' inability to alter their destinations mid-route, we introduce a condition whereby if a service robot's battery level falls below the threshold BL while going to the next location, it diverts to a recharging station instead of proceeding to the location. This adjustment ensures continuous operation without risking the service robot's shutdown mid-task. Moreover, for simplicity, since the discharging rate of service robots is significantly lower when they

are stationary compared to when they are moving, we disregard the periods when the service robots are not moving. Consequently, the battery levels are calculated based on the time they are traveling and serving.

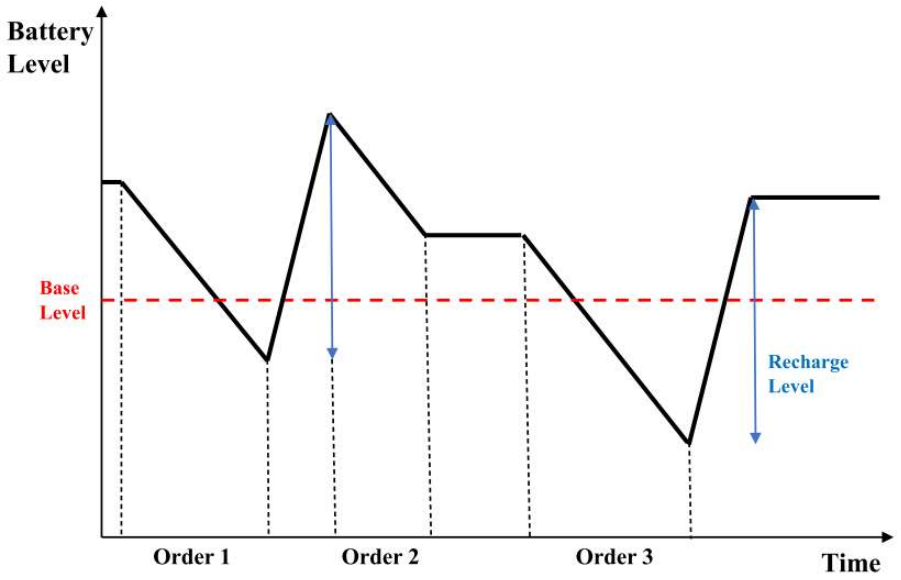


Figure 3.4: (Color online) Overview of the (BL, RL) policy for recharging [Jeong and Moon, 2024].

3.3 Hospital Network

There are many different locations, such as 373 unique locations or zones in the hospital, that need to be visited in case of any request requirement as a pickup or delivery location. Consequently, we have confronted a huge network that could not be optimized without a scientific approach. The zones in the hospital act as both source and target locations, with their responsibilities changing based on the specifics of each request. This diversity of source and target locations emphasizes the importance of an optimal assignment procedure to ensure efficient and timely transportation services within the hospital. Furthermore, The shortest travel time between any two points is assumed to be known and accounts for walking inside the buildings.

To formally define RIHTP, let $G = (N, A)$ be a complete directed graph, where

$N = \{0, 1, \ldots, N\}$ is the node set, and $A = \{(m, n) : m, n \in V\}$ is the arc set. The nodes $P = \{m, \ldots, n\}$ are pickup points, $D_1 = \{n + 1, \ldots, 2n\}$, and $D_2 = \{2n + 1, \ldots, N\}$ are first and second delivery points, respectively, the nodes $Q = \{0, 1, 2\}$ represent the depots, and the $RS = \{3, \ldots, m\}$ indicate the recharging stations. We assume a transporter fleet denoted by $V = V_1 \cup V_2 \cup R$ is located at the depot 0 for $v \in V_1$, 1 for $v \in V_2$, and 2 for $v \in R$ at the beginning of the planning horizon, whose length is denoted by T. The fleet must service a number of requests that arrive in the $[0, T]$ interval, but may be served after T if necessary. There are a set of requests $J = \{j_1, j_2, \ldots, j_m\}$ that need to be served. The request set is composed of there subsets J_1, J_2, and J_3 that indicates the different types of requests. Each transportation request $j \in J$ can be denoted by a pickup point and one or two delivery point(s) in N. For each arc (m, n), there is a corresponding time d_{mn}, indicating the time that is required to travel from node m to node n by using the shortest route. Furthermore, each node in the network is given a weight q_{mk} based on the request it is associated with. These weights are incorporated into the set defined for each request, influencing the overall logistics plan. Moreover, there is a score γ_m related to each node, which indicates the difficulty level of the request for which the node serves as either the pickup or delivery point. Each transporter is assigned a capacity, denoted by c_k, that indicates the maximum load or number of requests they can manage for each type of request according to the capacity. For ease of use, we call both porters and service robot transporters. This capacity constraint must be followed to ensure that transporters are not overworked and the transportation process remains efficient and effective. Moreover, a service time s_m is assigned to each node based on their role as a pickup or delivery point in the network. In such a network, the aim is that transporters travel through the nodes and complete the requests in the minimum amount of time while the the total delay is kept at the minimum level in addition to balancing the workload of porters.

Routes in this network are subject to the following assumptions: (a) Each transporter must commence and conclude its route at the designated depot; (b) Both the pickup and delivery points associated with a single request must be serviced by the same transporter; (c) The sequence of stops must ensure that a pickup point precedes

its corresponding delivery point(s); (d) For certain types of requests, consolidation is permissible, allowing indirect routing between pickup and delivery locations; (e) Once a transporter has commenced travel towards a destination, diversion from this path is not permitted.

3.4 Dynamic Events

In this context of hospital transportation, dynamic events are crucial in defining the efficiency and effectiveness of transporters routing and scheduling. This fact can be strongly evidenced by the incidence in most hospitals, where only a smaller proportion of transport requests are booked in advance. Consequently, the transportation problem for assignments evolves in a dynamic environment, and information becomes known throughout the day.

As a result, the planned route is, in this context, a sequence of requests assigned to a transporter that is scheduled but not yet serviced. As it is described by [Beaudry et al., 2010], several typical events require revisions to these planned routes, including patient-related events, including the arrival of new transportation requests, especially those needing short-term service; changes in the details of unserviced requests due to erroneous information; and cancellations of requests. Moreover, vehicle-related issues could affect the overall scheduling and routing. Examples of such vehicle-related events are the early entry into a service location by a vehicle, vehicle failure, and unscheduled periods of rest. Obviously, under such dynamic conditions, actions in dispatching must be done promptly as soon as new information becomes available. However, in our problem, the only dimension of dynamics is the arrival of new requests over time.

This makes the transportation problem relatively more complex because it must satisfy two opposing objectives: minimization of operating costs and provision of convenience to the patient. The operating cost is a function of travel time, which may be moving time or time spent with a patient assisting them into the transporter or travel time without the patient. These have resulted in increased direct costs when patients are either kept waiting or the transporter crew spends time in

idleness because of late pickups and early arrivals. Late deliveries, mainly to service departments like operating theaters, could cause underutilization of equipment and staff, further increasing hospital indirect costs and disrupting planned schedules.

In a dynamic environment, porters stay in touch with the dispatcher all the while and inform the dispatcher that all the steps have been completed. This dispatcher ensures flexibility in making necessary changes to the route. This dynamic characteristic of hospital transport is such that it requires an enormously flexible system that is able to cope with the continuous flow of new information while adjusting routes very quickly. Integrating real-time data and executing swift dispatch actions remain paramount for maintaining an effective transportation service with minimal cost implications, taking care of patient comfort, and timely arrivals at the destination points of services.

3.5 An Illustrative Example

Figure 3.5 illustrates RIHTP at a specific time of the day upon arrival of a new request, $j+5$, where there are five pending transportation requests, indexed as $j, j+1, j+2, j+3$, and $j+4$. The depots, represented by blue squares (DEP_0, DEP_1, DEP_2), act as starting and ending locations for transporters, while black circles indicate the corresponding pickup and delivery nodes for each request. Each request consists of a pickup location s_i and one or two possible delivery destinations, d_i^1 and d_i^2 with specific characteristics explained in previous sections. Three transporters, one from each fleet, porters and service robots, are currently operating, traveling along their assigned routes. The solid black arcs in the figure represent the fixed paths that have already been traversed, while the black dashed arcs denote planned but yet-to-be-executed movements before arrival of the new request. The red dashed arcs represent the updated route which is assigned to the new request. The figure demonstrates how transporters dynamically respond to new requests while adhering to precedence constraints, ensuring that pickups occur before corresponding deliveries. At the given time snapshot, some transporters are in the middle of completing deliveries, while others are en route to pick up new requests.

If no additional assignments are made, transporters will return to their respective depots; otherwise, they will continue to serve new requests as they arrive. This illustration captures the continuous and adaptive nature of the RIHTP, where real-time decision-making plays a crucial role in determining the optimal routing and scheduling of transporters to ensure efficient intra-hospital transportation services.

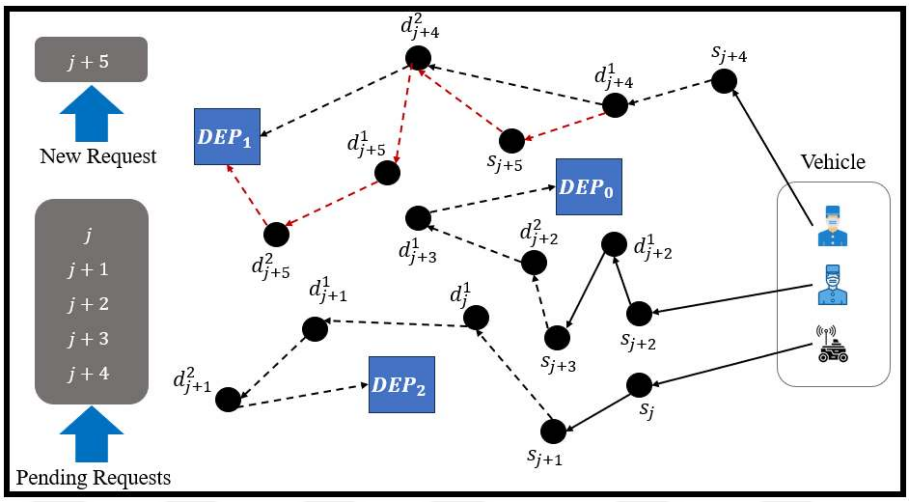


Figure 3.5: (Color online) Illustration of Real-Time Intra-Hospital Transportation Problem (RIHTP)

Chapter 4

OPTIMIZED FRAMEWORKS FOR STATIC INTRA-HOSPITAL TRANSPORTATION

The Dial-A-Ride problem, particularly in intra-hospital transportation, has been examined via different variants in the existing literature, as discussed in Chapter 2. In this chapter, we address the static (offline) version of the problem by developing two mathematical models for the human-operated and human-robot intra-hospital transportation problems using the DARP framework, in which all transportation requirements are predetermined, allowing transporter routes to be strategically arranged at the beginning of each day.

4.1 Mathematical Modes for Static Approaches to Assign and Schedule Requests

The Dial-A-Ride Problem (DARP) is defined by a set of clients or users, each with a pickup origin and one or two drop-off destinations. In more traditional versions, identical vehicles are stationed at a joint depot to serve all customers. The goal is to create a set of vehicle routes that minimize overall costs while accommodating the greatest number of requests possible under a predefined set of constraints. The Dial-A-Ride Problem (DARP) framework has also been applied to the Pickup and Delivery Vehicle Routing Problem (PDVRP) and the Vehicle Routing Problem with Time Windows (VRPTW) [Cordeau and Laporte, 2007]. In the subsequent part of this section, we propose a mathematical model for the static version of our problem.

4.1.1 Human-Operated Dial-A-Ride Mathematical Model

We utilize the three-index formulation for the Dial-A-Ride Problem (DARP) proposed by [Cordeau, 2006]. Nevertheless, some constraints require modification due

to the specific assumptions and unique characteristics inherent to RIHTP.

The following sets, parameters, and decision variables are defined:

Table 4.1: Table of notations for sets.

Notation	**Explanation**
D	Set of delivery nodes, $D = D_1 \cup D_2$.
P	Set of pickup nodes.
Q	Set of depot nodes.
N	Set of all nodes, $N = P \cup D \cup Q$.
J	Set of all requests, $J = J_1 \cup J_2 \cup J_3$.
V	Set of all available transporters, $V = V_1 \cup V_2$.
K	Set of capacity types.

Table 4.2: Table of notations for parameters and constants.

Notation	**Explanation**
c_{mnkv}	Capacity-related parameter equal to $\min\{c_k, c_k + q_{mk}\}$, where c_k is the capacity of type k for transporter v.
a^o_{mj}	Binary parameter equal to 1 if node $m \in N$ is the pickup point ($o = 0$), first delivery point ($o = 1$), or second delivery point ($o = 2$) of request $j \in J$; 0 otherwise.
d_{mn}	Travel time between nodes $m \in N$ and $n \in N$.
q_{mk}	Required capacity of type $k \in K$ at node $m \in N$.
s_m	Service time at node $m \in N$.
w_1, w_2, w_3	Weights associated with each term in the objective function.
t_j	Arrival time of request $j \in J$.
b_j	Time after which the delay penalty begins for request $j \in J$.
T	End of the planning horizon.
β	Penalty coefficient for delay in completing request j.
γ_m	Score assigned to node $m \in N$.
M	A sufficiently large constant used in constraint formulations.
l	Maximum allowed ride time.

Table 4.3: Table of notations for decision variables.

Notation	**Explanation**
X_{mnv}	Equal to 1 if porter $v \in V$ travels from node $m \in N$ to node $n \in N$; 0 otherwise.
L_{jv}	Ride time of request $j \in J$ when handled by porter $v \in V$.
C_v	Equal to 1 if porter $v \in V$ is assigned to at least one request; 0 otherwise.
W_{nvk}	Load of type $k \in K$ carried by porter $v \in V$ upon arriving at node $n \in N$.
A_{nv}	Arrival time of porter $v \in V$ to node $n \in N$.
U_{nv}	MTZ (Miller–Tucker–Zemlin) subtour elimination variable for node $n \in N$ and porter $v \in V$.
E_j^1, E_j^2	Delay values associated with the start time of request $j \in J$.
B^1	Maximum total working time among all porters.
B^2	Minimum total working time among all porters.

The proposed mathematical model for the static version of our problem is as follows.

$$Min\ Z = w_1 Z_1 + w_2 Z_2 + w_3 Z_3 \tag{4.1}$$

$$Z_1 = \sum_{m\in N}\sum_{n\in N}\sum_{v\in V}(d_{mn} + s_m)X_{mnv} \tag{4.2}$$

$$Z_2 = (B^1 - B^2) \tag{4.3}$$

$$Z_3 = \beta_j(\sum_{j\in J} E_j^1 + \alpha \sum_{j\in J} E_j^2) \tag{4.4}$$

subject to:

$$\sum_{v\in V}\sum_{m\in N\setminus\{n\}} X_{mnv} = 1 \qquad \forall n \in P \tag{4.5}$$

$$a_{nj}^0 A_{nv} - (t_j + b_j) \leq E_j^1 \qquad \forall n \in P, v \in V, j \in J \tag{4.6}$$

$$a_{nj}^0 A_{nv} - (t_j + b_j + \tau) \leq E_j^2 \qquad \forall n \in P, v \in V, j \in J \tag{4.7}$$

$$a_{nj}^0 A_{nv} \geq t_j \sum_{m\in N\setminus\{n\}} a_{nj}^0 X_{mnv} \qquad \forall n \in P, v \in V, j \in J \tag{4.8}$$

$$\sum_{n\in N\setminus\{0\}} X_{0nv} = C_v \qquad v \in V_1 \tag{4.9}$$

$$\sum_{m\in N\setminus\{0\}} X_{m0v} = C_v \qquad v \in V_1 \tag{4.10}$$

$$\sum_{n\in N\setminus\{1\}} X_{1nv} = C_v \qquad v \in V_2 \tag{4.11}$$

$$\sum_{m\in N\setminus\{1\}} X_{m1v} = C_v \qquad v \in V_2 \tag{4.12}$$

$$MC_v \geq \sum_{m\in P}\sum_{n\in N\setminus\{m\}} X_{nmv} \qquad \forall v \in V \tag{4.13}$$

$$C_v \leq \sum_{n \in N \backslash \{m\}} \sum_{m \in P} X_{nmv} \qquad \forall v \in V \qquad (4.14)$$

$$\sum_{m \in N \backslash \{n\}} X_{mnv} - \sum_{m \in N \backslash \{n\}} X_{nmv} = 0 \qquad \forall n \in N, v \in V \qquad (4.15)$$

$$\sum_{n \in N \backslash \{m\}} a^0_{mj} X_{nmv} - \sum_{n \in N \backslash \{m'\}} a^1_{m'j} X_{nm'v} = 0 \qquad \forall m \in P, m' \in D_1, j \in J \backslash J_3, v \in V \qquad (4.16)$$

$$\sum_{n \in N \backslash \{m\}} a^0_{mj} X_{nmv} - \sum_{n \in N \backslash \{m'\}} a^2_{m'j} X_{nm'v} = 0 \qquad \forall m \in P, m' \in D_2, j \in J_2, v \in V \qquad (4.17)$$

$$\sum_{n \in N \backslash \{m\}} a^0_{mj} X_{mnv} - \sum_{n \in N \backslash \{m'\}} a^1_{m'j} X_{m'nv} = 0 \qquad \forall m \in P, m' \in D_1, j \in J \backslash J_3, v \in V \qquad (4.18)$$

$$\sum_{n \in N \backslash \{m\}} a^0_{mj} X_{mnv} - \sum_{n \in N \backslash \{m'\}} a^2_{m'j} X_{m'nv} = 0 \qquad \forall m \in P, m' \in D_2, j \in J_2, v \in V \qquad (4.19)$$

$$a^0_{mj} a^1_{nj} (X_{mnv} - \sum_{n' \in N \backslash \{m\}} X_{n'mv}) = 0 \qquad \forall j \in J_3, v \in V, n \in D_1, m \in P \qquad (4.20)$$

$$a^0_{mj} a^1_{nj} a^2_{m'j} (X_{nm'v} - X_{mnv}) = 0 \qquad \forall j \in J_3, v \in V, m \in P, n \in D_1, m' \in D_2 \qquad (4.21)$$

$$A_{mv} + (d_{mn} + s_m) X_{mnv} - T(1 - X_{mnv}) \leq A_{nv} \qquad \forall n \in N \backslash \{0,1\}, m \in N \backslash \{0,1\}, v \in V, m \neq n \qquad (4.22)$$

$$A_{mv} \geq d_{0m} X_{0mv} \qquad \forall m \in N \backslash \{0,1\}, v \in V_1 \qquad (4.23)$$

$$A_{mv} \geq d_{1m} X_{1mv} \qquad \forall m \in N \backslash \{0,1\}, v \in V_2 \qquad (4.24)$$

$$U_{nv} - U_{mv} + (|N| + 1) X_{mnv} \leq |N| \qquad \forall n \in N \backslash \{0,1\}, m \in N \backslash \{0,1\}, v \in V, m \neq n \qquad (4.25)$$

$$a^1_{mj} U_{mv} - a^0_{nj} U_{nv} \geq 1 \qquad \forall j \in J, v \in V, n \in N, m \in N, m \neq n \qquad (4.26)$$

$$a^1_{mj}U_{mv} - a^2_{nj}U_{nv} \geq 1 \qquad \forall j \in J, v \in V, n \in N, m \in N, m \neq n \quad (4.27)$$

$$W_{mkv} + q_{nk} - c_{mnkp}(1 - X_{mnv}) \leq W_{nkv} \qquad \forall n \in N, m \in N, v \in V, k \in K, m \neq n \quad (4.28)$$

$$L_{jv} = a^1_{mj}A_{mv} - (a^0_{nj}A_{nv} + s_n \sum_{m' \in N \setminus \{n\}} X^{'}_{m}nv \qquad \forall n \in P, m \in D_1, v \in V, j \in J_1 \quad (4.29)$$

$$L_{jv} = a^2_{mj}A_{mv} - a^0_{nj}a^1_{oj}(A_{nv} + (s_n + s_o) \sum_{i \in N \setminus \{n\}} X_{inv}) \qquad \forall n \in P, m \in D_2, v \in V, j \setminus J_1 \quad (4.30)$$

$$a^1_{mj}a^0_{nj}d_{nm} \sum_{o \in N \setminus \{n\}} X_{onv} \leq L_{jv} \leq a^0_{nj}l \sum_{o \in N \setminus \{n\}} X_{onv} \qquad \forall n \in P, m \in D_1, v \in V, J \in J_1 \quad (4.31)$$

$$a^2_{m'j}a^1_{mj}a^0_{nj}(d_{nm} + d_{mm'}) \sum_{o \in N \setminus \{n\}} X_{onv} \leq L_{jv} \leq a^0_{nj}l \sum_{o \in N \setminus \{n\}} X_{onv} \qquad \forall n \in P, m \in D_1, m^{'} \in D_2, v \in V, J \setminus J_1 \quad (4.32)$$

$$\sum_{m \in N} \sum_{n \in N \setminus \{m\}} \gamma_m(d_{mn} + s_m)X_{mnv} \leq B^1 \qquad \forall v \in V \quad (4.33)$$

$$\sum_{m \in N} \sum_{n \in N \setminus \{m\}} \gamma_m(d_{mn} + s_m)X_{mnv} \geq B^2 \qquad \forall v \in V \quad (4.34)$$

$$\sum_{m \in N} (X_{m1p} + X_{1mp}) = 0 \qquad \forall v \in V_1 \quad (4.35)$$

$$\sum_{m \in N} (X_{m0p} + X_{0mp}) = 0 \qquad \forall v \in V_2 \quad (4.36)$$

$$X_{mnv} \in \{0, 1\} \qquad \forall v \in V, m \in N.n \in N \quad (4.37)$$

$$W_{nkv} \in \mathbb{N} \qquad \forall v \in V, n \in N, k \in K \quad (4.38)$$

$$A_{nv} \geq 0 \qquad \forall v \in V, n \in N \quad (4.39)$$

$$L_{jv} \geq 0 \qquad \forall j \in J, v \in V \quad (4.40)$$

$$C_v \in \{0,1\} \qquad \forall v \in V \tag{4.41}$$

$$U_{np} \in \mathbb{N} \qquad \forall v \in V, n \in N \tag{4.42}$$

$$D_j \geq 0 \qquad \forall j \in J \tag{4.43}$$

$$B^1 \geq 0 \tag{4.44}$$

$$B^2 \geq 0 \tag{4.45}$$

The model aims to minimize the weighted sum (4.1), which contains travel time and service time (4.2), (ii) the difference between the maximum and minimum workload of a porter (4.3), (iii) the total penalty associated with the lateness from the arrival time of a request (4.4). Constraints (4.5) ensure that a request can be handled just by a porter. Constraints (4.6) and (4.7) are used to calculate the delays related to each request. Constraints (4.8) ensure that a porter that is assigned to a request cannot start it unless the request has arrived. Constraints (4.9) and (4.11) ensure that at the beginning of the day, if a porter is assigned to at least one request, he should leave the location where it originates from. Similarly, Constraints (4.10) and (4.12) ensure that at the end of the day, such porters go back to the location where they originate from. Constraints (4.13) and (4.14) express the necessary conditions for variables C_p. Constraints (4.15) guarantee that porters visiting a location should leave that location. The purpose of constraints (4.16) - (4.21) is to enforce that if a request is handled by a porter, that porter needs to visit the nodes associated with each request. By imposing constraints (4.22) - (4.24), we calculate the arrival time of each porter to the visited nodes. Constraints (4.25) - (4.27) prevent sub-tours by ensuring porters follow a proper sequence in their route. Constraints (4.28) are formulated to address the load of each type carried by a porter upon arriving at a node. Constraints (4.29) and (4.30) define the ride time of each user, which is bounded by constraints (4.31) and (4.32). Constraints (4.33) are critical for calculating the maximum working time of a porter in the planning horizon. Similarly, constraints (4.34) are used for calculating the minimum working time of a porter in the planning horizon. Constraints (4.35) - (4.36) prohibit each type of porter from visiting the depot node designated for the other type of porters. Constraints (4.37) - (4.45) define the domains of the variables.

In logistics optimization, the deployment of a homogeneous fleet of vehicles, each with similar capabilities and operating between shared points of origin and destination, creates mathematical symmetry. This symmetry allows for the interchangeable assignment of routes among any permutation of vehicles. Although these solutions differ mathematically, they lead to comparable practical consequences, resulting in duplicate computational efforts. To reduce inefficiencies, two sets of symmetry-breaking constraints are used, inspired by [Dahle et al., 2019] research on the Pickup and Delivery Problem. These lexicographical limitations effectively discriminate apparently symmetrical solutions, shortening the computing process and improving route allocation efficiency in such systems that are stated as:

$$\sum_{n\in P} X_{0nv} \geq \sum_{n\in P} X_{0n(v+1)} \qquad \forall v \in V_1 \backslash \{|V_1|\} \tag{4.46}$$

$$\sum_{n\in P} X_{1nv} \geq \sum_{n\in P} X_{1n(v+1)} \qquad \forall v \in V_2 \backslash \{|V_2|\} \tag{4.47}$$

$$\sum_{n\in N} \sum_{m\in N\backslash\{n\}} (d_{nm} + s_n) X_{nmv} \geq \sum_{n\in N} \sum_{m\in N\backslash\{n\}} (d_{nm} + s_n) X_{nm(v+1)} \qquad \forall v \in V_1 \backslash \{|V_1|\} \tag{4.48}$$

$$\sum_{n\in N} \sum_{m\in N\backslash\{n\}} (d_{nm} + s_n) X_{nmv} \geq \sum_{n\in N} \sum_{m\in N\backslash\{n\}} (d_{nm} + s_n) X_{nm(v+1)} \qquad \forall v \in V_2 \backslash \{|V_1|\} \tag{4.49}$$

Constraints (4.46) and (4.47) are intended to prioritize the employment of porters with lower indices, assuring their deployment before others. Meanwhile, constraints (4.48) and (4.49) require that the length of routes increase with the porters index so that porters with lower indices have greater route length.

4.1.2 Human-Robot Dial-A-Ride Mathematical Model

We employ the identical parameters, indices, and decision variables defined above, together with the following additional ones:

Table 4.4: Table of notations for additional sets.

Notation	Explanation
R	Set of all available service robots.
V	Set of all available transporters, $V = V_1 \cup V_2 \cup R$.
RS	Set of recharging station nodes.
N	Set of all operational nodes (excluding recharging stations), $N = P \cup D \cup Q$.

Table 4.5: Table of notations for compatibility and battery parameters.

Notation	Explanation
f_{jv}	Binary parameter equal to 1 if request $j \in J$ can be handled by transporter $v \in V$; 0 otherwise.
r	Discharging rate of battery per unit of time or distance.
ρ	Fixed recharging amount at each recharging visit.
λ	Recharging rate of the battery during charging.
h_{max}	Maximum battery capacity of each service robot.
π	Minimum allowed charge level (safety threshold).

Table 4.6: Table of notations for robot-specific variables and parameters.

Notation	Explanation
F_{mnlv}	Binary parameter equal to 1 if service robot $v \in R$ visits recharging station $l \in RS$ between its path from node $m \in N$ to node $n \in N$.
X_{mnv}	Equal to 1 if node $n \in N$ is directly or indirectly visited after node $m \in N$ by transporter $v \in V$; 0 otherwise.
H_{nv}	Battery level of service robot $v \in R$ upon leaving node $n \in N$.

Consequently, the new mathematical model is as follows:

$$Min\ Z = w_1 Z_1 + w_2 Z_2 + w_3 Z_3 \tag{4.50}$$

$$Z_1 = \sum_{v\in V}\sum_{m\in N}\sum_{n\in N}(d_{mn}+s_m)X_{mnv} - \sum_{v\in R}\sum_{m\in N}\sum_{n\in N}\sum_{l\in L} d_{mn}F_{mnlv} + \sum_{n\in N}\sum_{l\in L}(d_{ml}+s_l+\frac{\rho}{\lambda}+d_{ln})F_{mnlv} \tag{4.51}$$

$$Z_2 = \ B^1 - B^2 \tag{4.52}$$

$$Z_3 = \beta_j(\sum_{j\in J} E_j^1 + \alpha \sum_{j\in J} E_j^2) \tag{4.53}$$

subject to:

$$\sum_{v\in V} f_{nv} \sum_{m\in N\setminus\{n\}} X_{mnv} = 1 \qquad \forall n \in P \tag{4.54}$$

$$a_{nj}^0 A_{nv} - (t_j + b_j) \le E_j^1 \qquad \forall n \in P, v \in V, j \in J \tag{4.55}$$

$$a_{nj}^0 A_{nv} - (t_j + b_j + \tau) \le E_j^2 \qquad \forall n \in P, v \in V, j \in J \tag{4.56}$$

$$a_{nj}^0 A_{nv} \ge t_j \sum_{m\in N\setminus\{n\}} a_{nj}^0 X_{mnv} \qquad \forall n \in P, v \in V, j \in J \tag{4.57}$$

$$\sum_{n\in N\setminus\{0\}} X_{0nv} = C_v \qquad v \in V_1 \tag{4.58}$$

$$\sum_{m\in N\setminus\{0\}} X_{m0v} = C_v \qquad v \in V_1 \tag{4.59}$$

$$\sum_{n\in N\setminus\{1\}} X_{1nv} = C_v \qquad v \in V_2 \tag{4.60}$$

$$\sum_{m\in N\setminus\{1\}} X_{m1v} = C_v \qquad v \in V_2 \tag{4.61}$$

$$\sum_{m\in N\setminus\{2\}} X_{m2v} = C_v \qquad v \in R \tag{4.62}$$

$$\sum_{n\in N\setminus\{2\}} X_{2nv} = C_v \qquad v \in R \tag{4.63}$$

$$MC_v \geq \sum_{m\in P}\sum_{n\in N\setminus\{m\}} X_{nmv} \qquad \forall v \in V \tag{4.64}$$

$$C_v \leq \sum_{n\in N\setminus\{m\}}\sum_{m\in P} X_{nmv} \qquad \forall v \in V \tag{4.65}$$

$$\sum_{m\in N\setminus\{n\}} X_{mnv} - \sum_{m\in N\setminus\{n\}} X_{nmv} = 0 \qquad \forall n \in N, v \in V \tag{4.66}$$

$$f_{jv}\sum_{n\in N\setminus\{m\}} a^0_{mj}X_{nmv} - \sum_{n\in N\setminus\{m'\}} a^1_{m'j}X_{nm'v} = 0 \quad \forall m \in P, m' \in D_1, j \in J\setminus J_3, v \in V \tag{4.67}$$

$$f_{jv}\sum_{n\in N\setminus\{m\}} a^0_{mj}X_{nmv} - \sum_{n\in N\setminus\{m'\}} a^2_{m'j}X_{nm'v} = 0 \quad \forall m \in P, m' \in D_2, j \in J_2, v \in V \tag{4.68}$$

$$f_{jv}\sum_{n\in N\setminus\{m\}} a^0_{mj}X_{mnv} - \sum_{n\in N\setminus\{m'\}} a^1_{m'j}X_{m'nv} = 0 \quad \forall m \in P, m' \in D_1, j \in J\setminus J_3, v \in V \tag{4.69}$$

$$f_{jv}\sum_{n\in N\setminus\{m\}} a^0_{mj}X_{mnv} - \sum_{n\in N\setminus\{m'\}} a^2_{m'j}X_{m'nv} = 0 \quad \forall m \in P, m' \in D_2, j \in J_2, v \in V \tag{4.70}$$

$$a^0_{mj}a^1_{nj}(X_{mnv} - \sum_{n'\in N\setminus\{m\}} X_{n'mv}) = 0 \qquad \forall j \in J_3, v \in V\setminus R, n \in D_1, m \in P \tag{4.71}$$

$$a^0_{mj}a^1_{nj}a^2_{m'j}(X_{nm'v} - X_{mnv}) = 0 \qquad \forall j \in J_3, v \in V\setminus R, m \in P, n \in D_1, m' \in D_2 \tag{4.72}$$

$$A_{mv} + (d_{mn} + s_m)X_{mnv} - T(1 - X_{mnv}) \leq A_{nv} \qquad \forall n \in N\setminus\{0,1,2\}, m \in N\setminus\{0,1,2\}, v \in V\setminus R, m \neq n \tag{4.73}$$

$$A_{mv} + (d_{mn} + s_m)X_{mnv} - T(1 - X_{mnv}) - d_{mn} \sum_{l \in RS} F_{mnl} + \sum_{linRS} (d_{ml} + s_l + d_{ln} + \rho/\lambda)F_{mnl} - T(1 - X_{mnv}) \leq A_{nv} \quad \forall n \in N \backslash \{0,1,2\}, m \in N \backslash \{0,1,2\}, v \in R, m \neq n \tag{4.74}$$

$$A_{mv} \geq d_{0m} X_{0mv} \quad \forall m \in N \backslash \{0,1,2,3\}, v \in V_1 \tag{4.75}$$

$$A_{mv} \geq d_{1m} X_{1mv} \quad \forall m \in N \backslash \{0,1,2,3\}, v \in V_2 \tag{4.76}$$

$$A_{mv} \geq d_{2m} X_{2mv} \quad \forall m \in N \backslash \{0,1,2,3\}, v \in R \tag{4.77}$$

$$U_{nv} - U_{mv} + (|N| + 1)X_{mnv} \leq |N| \quad \forall n \in N \backslash \{0,1,2\}, m \in N \backslash \{0,1,2,3\}, v \in V, m \neq n \tag{4.78}$$

$$a^1_{mj} U_{mv} - a^0_{nj} U_{nv} \geq 1 \quad \forall j \in J, v \in V, n \in N, m \in N, m \neq n \tag{4.79}$$

$$a^1_{mj} U_{mv} - a^2_{nj} U_{nv} \geq 1 \quad \forall j \in J, v \in V, n \in N, m \in N, m \neq n \tag{4.80}$$

$$W_{mkv} + q_{nk} - c_{mnkp}(1 - X_{mnv}) \leq W_{nkv} \quad \forall n \in N, m \in N, v \in V, k \in K, m \neq n \tag{4.81}$$

$$L_{jv} = f_{jv} \left(a^1_{mj} A_{mv} - (a^0_{nj} A_{nv} + s_n \sum_{m' \in N \backslash \{n\}} X'_{mnv}) \right) \quad \forall n \in P, m \in D_1, v \in V, j \in J_1 \tag{4.82}$$

$$L_{jv} = f_{jv}\left(a^2_{mj}A_{mv} - a^0_{nj}a^1_{oj}(A_{nv} + (s_n + s_o)\sum_{i\in N\setminus\{n\}} X_{inv})\right) \quad \forall n \in P, m \in D_2, v \in V, j\setminus J_1 \tag{4.83}$$

$$a^1_{mj}a^0_{nj}d_{nm}\sum_{o\in N\setminus\{n\}} X_{onv} \le L_{jv} \le a^0_{nj}l\sum_{o\in N\setminus\{n\}} X_{onv} \quad \forall n \in P, m \in D_1, v \in V, J \in J_1 \tag{4.84}$$

$$a^2_{m'j}a^1_{mj}a^0_{nj}(d_{nm} + d_{mm'})\sum_{o\in N\setminus\{n\}} X_{onv} \le L_{jv} \le a^0_{nj}l\sum_{o\in N\setminus\{n\}} X_{onv} \quad \forall n \in P, m \in D_1, m^{'} \in D_2, v \in V, J \setminus J_1 \tag{4.85}$$

$$\sum_{m\in N}\sum_{n\in N\setminus\{m\}} \gamma_m(d_{mn} + s_m)X_{mnv} \le B^1 \quad \forall v \in V\setminus R \tag{4.86}$$

$$\sum_{m\in N}\sum_{n\in N\setminus\{m\}} \gamma_m(d_{mn} + s_m)X_{mnv} \ge B^2 \quad \forall v \in V\setminus R \tag{4.87}$$

$$\sum_{m\in N}(X_{m1p} + X_{1mp} + X_{m2p} + X_{2mp}) = 0 \quad \forall v \in V_1 \tag{4.88}$$

$$\sum_{m\in N}(X_{0mp} + X_{m0p} + X_{m2p} + X_{m2p}) = 0 \quad \forall v \in V_2 \tag{4.89}$$

$$\sum_{m\in N}(X_{0mp} + X_{m0p} + X_{1mp} + X_{m1p}) = 0 \quad \forall v \in R \tag{4.90}$$

$$H_{mv} \le H_{nv} - rd_{nm}(X_{nmv} - \sum_{l\in RS} F_{nmlv}) - rs_m + \sum_{l\in RS}(\rho - r(d_{nl} + d_{lm} - s_l))F_{mnlv} + h_{max}(1 - X_{mnv}) \quad \forall n \in N\setminus\{0,1,2\}, m\setminus\{0,1,2\}, v \in R, m \ne n \tag{4.91}$$

$$H_{nv} \leq h_{max} - r(d_{2n} + s_n)X_{2nv} + h_{max}(1 - X_{2nv}) \quad \forall n \in N\backslash\{0, 1, 2\}, v \in R \tag{4.92}$$

$$H_{nv} \geq h_{max} - r(d_{2n} + s_n)X_{2nv} - h_{max}(1 - X_{2nv}) \quad \forall n \in N\backslash\{0, 1, 2\}, v \in R \tag{4.93}$$

$$H_{nv} \leq h_{max} \quad \forall n \in N\backslash\{0, 1, 2\}, v \in R \tag{4.94}$$

$$H_{nv} \geq \pi \quad \forall n \in N\backslash\{0, 1, 2\}, v \in R \tag{4.95}$$

$$\sum_{n\in P} X_{0nv} \geq \sum_{n\in P} X_{0n(v+1)} \quad \forall v \in V_1\backslash\{|V_1|\} \tag{4.96}$$

$$\sum_{n\in P} X_{1nv} \geq \sum_{n\in P} X_{1n(v+1)} \quad \forall v \in V_2\backslash\{|V_2|\} \tag{4.97}$$

$$\sum_{n\in P} X_{1nv} \geq \sum_{n\in P} X_{1n(v+1)} \quad \forall v \in R\backslash\{|R|\} \tag{4.98}$$

$$\sum_{n\in N}\sum_{m\in N\backslash\{n\}} (d_{nm} + s_n)X_{nmv} \geq \sum_{n\in N}\sum_{m\in N\backslash\{n\}} (d_{nm} + s_n)X_{nm(v+1)} \quad \forall v \in V_1\backslash\{|V_1|\} \tag{4.99}$$

$$\sum_{n\in N}\sum_{m\in N\backslash\{n\}} (d_{nm} + s_n)X_{nmv} \geq \sum_{n\in N}\sum_{m\in N\backslash\{n\}} (d_{nm} + s_n)X_{nm(v+1)} \quad \forall v \in V_2\backslash\{|V_1|\} \tag{4.100}$$

$$\sum_{n\in N}\sum_{m\in N\backslash\{n\}} (d_{nm} + s_n)X_{nmv} \geq \sum_{n\in N}\sum_{m\in N\backslash\{n\}} (d_{nm} + s_n)X_{nm(v+1)} \quad \forall v \in R\backslash\{|R|\} \tag{4.101}$$

$$F_{mnlv} \in \{0,1\} \qquad \forall v \in V, l \in RS, m \in N, n \in N \tag{4.102}$$

$$X_{mnv} \in \{0,1\} \qquad \forall v \in V, m \in N.n \in N \tag{4.103}$$

$$W_{nkv} \in \mathbb{N} \qquad \forall v \in V, n \in N, k \in K \tag{4.104}$$

$$H_{nv} \geq 0 \qquad \forall v \in R, n \in N \tag{4.105}$$

$$A_{nv} \geq 0 \qquad \forall v \in V, n \in N \tag{4.106}$$

$$Y_{nv} \in \{0,1\} \qquad \forall v \in R, n \in N \backslash \{0,1,2,3\} \tag{4.107}$$

$$L_{jv} \geq 0 \qquad \forall j \in J, v \in V \tag{4.108}$$

$$Q_{nv} \geq 0 \qquad \forall v \in V, n \in N \backslash \{0,1,2,3\} \tag{4.109}$$

$$C_v \in \{0,1\} \qquad \forall v \in V \tag{4.110}$$

$$U_{np} \in \mathbb{N} \qquad \forall v \in V, n \in N \tag{4.111}$$

$$D_j \geq 0 \qquad \forall j \in J \tag{4.112}$$

$$B^1 \geq 0 \tag{4.113}$$

$$B^2 \geq 0 \tag{4.114}$$

Similar to the previous mathematical model, this model seeks to minimize the weighted sum of (i) the total time porters spend to fulfill all requests 4.50, encompassing both travel and service time (4.51), (ii) the difference between the maximum and minimum workload of a porter (4.52), and (iii) the cumulative penalty linked to lateness relative to the arrival time of a request (4.53). Constraints (4.54 - 4.90) and (4.96 - 4.114) are identical to the previous mathematical model. Constraints (4.91 - 4.93) provide a balance equation for the battery using the modified (BL, RL) recharging policy. Constraints (4.94) and (4.95) define the upper and lower bounds for battery level, respectively.

Chapter 5

REAL-TIME OPTIMIZATION OF INTRA-HOSPITAL TRANSPORTATION

This chapter presents our proposed solution approach, which is an integration of the Deep Q-Learning (DQL) algorithm with other heuristic algorithms. RIHTP is inherently difficult to solve due to its dynamic, high-dimensional, and combinatorial nature. In fact, RIHTP, modeled within the Dial-A-Ride Problem (DARP) framework, presents significant challenges that stem from its complexity and the demanding operational constraints of healthcare environments. Firstly, the problem is characterized by several restricted constraints and assumptions, such as the capacity of transports, specific pickup and drop-off points, and predefined ride time durations for each request. These conditions significantly restrict the feasible solution space and increase the computational burden, making it impractical to find optimal solutions within a reasonable timeframe using traditional methods. Moreover, the real-time aspect of the problem introduces additional complexity. Decisions regarding the assignment of transportation requests and the determination of transporters' routes must be made instantaneously to maintain a seamless flow of operations within the hospital. This is crucial in preventing delays that could adversely affect patient care and hospital efficiency.

In large healthcare facilities, especially in the examined hospital, which typically handle a high volume of transportation requests daily, the situation is further complicated. Each request varies in terms of urgency, destination, and required resources, adding layers of complexity to the decision-making process. Consequently, attempting to solve the entire problem each time a decision is needed is not feasible. The computational load would be excessive, and the time taken to arrive at a decision would be incompatible with the rapid pace required for effective intra-hospital

logistics. Thus, traditional approaches to solving such complex, dynamic problems are often inadequate in these high-pressure, real-time environments where speed and adaptability are paramount.

To effectively address these challenges, we decompose the RIHTP into two critical and interrelated decision-making phases at each decision point: assignment and routing. We designate these as Phase 1 and Phase 2 decisions, respectively. In Phase 1, the decision involves determining the most suitable transporter for each newly arrived transportation request. This decision focuses on efficiently matching transport resources to demands based on current system states and the specifics of the request, such as urgency, destination proximity, and transporter workload. This phase is driven by three primary objectives: minimizing the total travel and service time, balancing the workload among transporters, and reducing the delay penalty associated with pickup points. Phase 2, on the other hand, focuses on reoptimizing the route for the assigned transporter. This decision adjusts the previously planned route to incorporate the new assignment, optimizing the travel route to minimize the total travel and service time and reduce the delay penalty. Nonetheless, these decisions are closely interconnected. For each decision point, the outcome of Phase 1, assigning the optimal transporter, must be determined before Phase 2 can begin, where routes are reoptimized based on these assignments. Moreover, the results of Phase 2 are crucial for informing the next set of Phase 1 decisions. Without the updated routing information from Phase 2, it would be difficult to make effective assignments in the following round. Consequently, the decision should be make successively for each phase. This cycle ensures that each phase informs and enhances the other, maintaining a continuous flow of updated decision-making.

In Phase 1 decision, we employ reinforcement learning, namely Deep Q-Learning, to tackle the distinct challenges associated with this phase. The effectiveness of an assignment decision in Phase 1 can only be assessed once the routing in Phase 2 has been established. This necessitates a solution approach that can predict the future impacts of the assignment decision on the overall objectives. Reinforcement learning is optimal for this goal as it emphasizes the selection of actions that maximize cumulative future rewards, therefore predicting and enhancing the long-term outcomes

of decisions made at each decision point. Deep Q-Learning is utilized to develop a robust and near-optimal policy for assignment decision-making. This technique accurately evaluates the potential future consequences of current decisions. It employs an extensive array of features to predict the results of each decision over the remaining planning horizon. Furthermore, Deep Q-Learning is effective at developing robust policies that can be used with high speed, making it suitable for the real-time requirements of hospital operations.

In Phase 2 decision, which focuses on optimizing the routing of a particular transporter after the insertion of a new request into its schedule, we employ a two-stage heuristic composed of Best Insertion (BI) followed by Adaptive Variable Neighborhood Search (AVNS). This approach is well-suited for real-time hospital logistics, as it delivers high-quality routing decisions within a limited computational time. The BI heuristic provides a rapid and feasible initial solution by deterministically identifying the least-cost insertion position for the new request within the existing route. Subsequently, AVNS is applied to iteratively improve this initial solution by exploring a dynamic sequence of increasingly complex neighborhoods. This combination balances intensification and diversification, enabling effective local improvements without excessive computation.

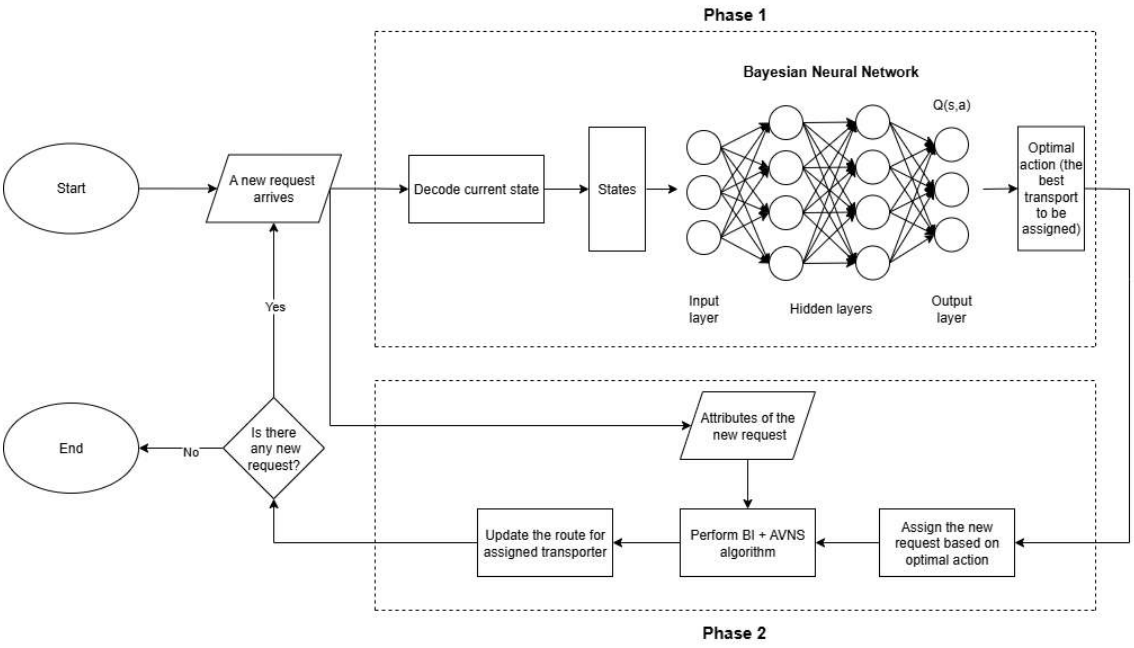


Figure 5.1: The flowchart of the two-phase solution methodology for dynamic version

As depicted in Figure 5.1, our two-phase solution methodology is activated upon the arrival of a new request. We first utilize the policy produced using reinforcement learning to assign the request to the suitable transporter. Subsequent to the assignment, the BI + AVNS algorithm proceeds to include the newly assigned request into the current route. This systematic methodology guarantees that each phase complements the others, enhancing both the assignment and routing decisions in real-time to dynamically adjust to changes in the hospital's transportation requests. The details of this solution methodology are explained in the next sections.

5.1 Phase 1 Decision: Assignment Problem

This section examines Phase 1 of the Real-Time Intra-Hospital Transportation Problem (RIHTP), assignment decision, which is crucial for improving operational workflows within the hospital. This phase is organized as a sequential decision process model, where each decision point aligns with the arrival of a new transportation request, requiring prompt and strategic assignment to suitable transporters. We analyze the sequential decision process model, detailing the formulation of deci-

sion epochs, states, actions, and rewards that facilitate the management of assignment complexities in real-time. Furthermore, we examine the utilization of Deep Q-Learning (DQL) to enhance these decisions. This includes discussions on the DQN architecture and training setting to guarantee the system's effective adaptation to the evolving requirements of hospital logistics.

5.1.1 Sequential Decision Process Model

We model Phase 1, assignment decision, as a sequential decision process, comprising a series of states linked by actions and transitions as follows:

Decision Epoch: A decision epoch is represented by a time instant, during the planning horizon, when a decision must be taken (arrival of a new request). We denote the k^{th} transportation request as j_k and the time of the k^{th} decision epoch as t_k. Therefore, the set of decision epoch is represented as $T_{action} = \{t_k : k \in 1, 2, \ldots, \mid J \mid\}$

State: The state at a decision epoch encapsulates the information needed for making the decision. In our problem, the state S_k at decision epoch k includes the information about the customers, the porters and service robots, as well as the planned routes. Since the planned routes encompass the pertinent details for the fleets, the fleet information can be excluded from the state information. Mathematically, a state has four components:

1. t_k: Time of decision epoch.

2. j_k = New request.

3. $J_k = (J_{P,k}, J_{R,k})$: Set of assigned transportation requests still not started. The set is split between requests planned with porters $J_{P,k}$ and service robots $J_{R,k}$.

4. $\Gamma_k = (\Gamma_{P,k}, \Gamma_{R,k})$: Set of planned routes for porters $\Gamma_{P,k}$ and service robots $\Gamma_{R,k}$ when the request j_k is received.

To wrap it up, we represent the state S_k as a tuple $S_k = (t_k, j_k, J_k, \Gamma_k)$. In the initial state $t = 0$, all planned routes are empty.

Action and Reward: At every decision epoch, we determine an action x_k from the set of possible actions $X(S_k)$. In RIHTP, an action incorporates to which transporter the new request j_k should be assigned. The action is represented as a tuple $\Gamma_k^x = (\Gamma_{P,k}^x, \Gamma_{R,k}^x)$, where $\Gamma_{P,k}^x$ ($\Gamma_{R,k}^x$) is the updated set of porter (or service robot) planned routes. In addition, $J_k^x = (J_{P,k}^x, J_{R,k}^x)$ represents the updated set of not yet started transportation requests.

The reward of an action x_k is $R(S_k, x_k) = 100 \cdot \frac{R(\Gamma_k) - R(\Gamma_k^x)}{R(\Gamma_k) + \epsilon}$, where the first reward in the objective value of the current routes and the subsequent is the objective value of the post-decision routes. In other words, this reward function represents the percentage increase in the objective function after the assignment. The ϵ is added to the denominator to prevent division by zero in cases where the objective function evaluates to zero. To calculate $R(\Gamma_k^x)$, we need to solve the intra-hospital transportation problem based on the system states and assignment action x_k using a heuristic that we will introduce in Section 5.2.

Stochastic Information and Transition: Transitions arise from the acknowledgment of arbitrary transportation requests. These requests are exogenous to the decision-making process. For example, in our computations, we utilize a random distribution derived from hospital data to generate requests. The realized information W_{k+1} either reveals a new request and the time of the request $W_{k+1} = \{j_{k+1}, t_{k+1}\}$ or leads to termination of the process because the operation periods ended, $W_{k+1} = \{\}$. In the latter case, the process terminates in the final state, $S_{k+1} = S_k$. Otherwise, the new state S_{k+1} is a combination of post-decision state S_{k+1}^x and stochastic information W_{k+1} with the following updates.

1. The time of the decision epoch is t_{k+1}, which is the time of the arrival of the new request j_{k+1}.

2. The porters and service robots may have started new requests in the time

between j_k and j_{k+1}. We denote the set of such requests $J^x_{k.k+1}$. These requests are no longer part of the sate, and therefore, $J_{k+1} = J^x_k \backslash J^x_{k,k+1}$.

3. Similarly, the routes Γ_{k+1} are updated by removing the visited nodes.

Objective Function: A solution to Phase 1 is a policy $\pi \in \Pi$ that assigns an action to each state. The optimal solution is a policy π^* that minimize the total expected reward and can be expressed by

$$\pi^* = \arg\min_{\pi \in \Pi} \mathbb{E}\left[\sum_{k=0,\ldots,K} R(S_k, X^\pi(S_k)) \mid S_0\right] \tag{5.1}$$

Where $X^\pi(S_k)$ is the action selected by policy π at state S_k.

Figure 5.2 summarizes the information of Sequential Decision Process Model.

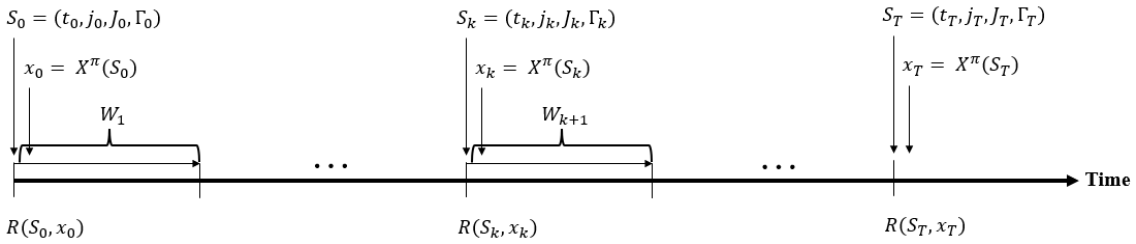


Figure 5.2: Illustration of the sequential decision-making process of the problem in a Sequential Decision Process Model.

5.1.2 Deep Q-Learning

It is known that the solution to a sequential decision process model can be derived via backward induction applied to the Bellman equation 5.2 :

$$V(S_k) = \max_{x_k \in X(S_k)} \{R(S_k, x_k) + \mathbb{E}[V(S_{k+1})|S_k]\} \tag{5.2}$$

However, the size of the state space in this problem is too large to enumerate. This problem involves routing multiple porters and service robots to serve numerous requests. Furthermore, the state space is continuous and encompasses multiple dimensions. Thus, even attempting to estimate the value function with tabular reinforcement learning approaches (such as tabular Q-Learning) is impractical because

storing the value for each possible state-action pair results in significant memory usage, rendering frequent value assessments and, consequently, learning unfeasible. In contrast, deep reinforcement learning algorithms (such as Deep Q-Learning) address these practical challenges because neural networks can learn the relationships between various states and actions, using this knowledge to extend the value from known state-action pairs to those that have not been explored. Consequently, we require deep reinforcement learning algorithms to estimate the value of each state-action pair. We choose Deep Q-Learning from the deep reinforcement algorithms owing to its off-policy approach, which allows for the learning of optimal policies from exploratory data without restriction to the current strategy, hence improving its adaptability and efficacy in complicated contexts.

Deep Q-Learning (DQL) is an advanced modification of the traditional Q-learning technique, incorporating deep neural networks to learn the value of taking an action at each given state. In fact, DQL theoretically uses a neural network as a function approximator to estimate so-called Q-values, Deep Q-Network (DQN), representing the expected future rewards for a specific state-action pair. Let θ denote the current set of weights defining the DQN that models the agent's behavior. For any state-action pair (S_k, x), the Q-value corresponds to the value assigned by the DQN with weights θ to that particular pair. The Q-values approximate Equation 5.3 as follows:

$$V(S_k) \approx \max_{x_k \in X(S_k)} Q_\theta(S_k, x_k) \tag{5.3}$$

Additionally, by conducting the training phase of the DQN offline and retaining only its weights θ, the agent can swiftly calculate, for any given state S_k, the Q-value for each potential action x_k, thereby readily determining the action that offers the highest expected reward. At decision epoch k, the action x_k chosen by θ for execution is the one that maximizes the expected future rewards, that is,

$$x_k = \arg \max_{x_k \in X(S_k)} Q_\theta(S_k, x_k), \quad \forall k \tag{5.4}$$

Given the dynamic nature of the problem, where requests with different types and requirements arrive over time, altering the plans significantly, a Bayesian Neural

Network (BNN) is preferred over a traditional neural network. BNNs are particularly suited for environments with high uncertainty and variability because they not only predict outcomes but also estimate the reliability of these predictions. This probabilistic approach allows for better decision-making under uncertainty, which is critical in hospital settings where decisions must adapt to changing conditions on the fly. We thus model the agent with a multi-layer perceptron structured as a BNN, featuring an input layer, one hidden layers, consisting of a defined number of neurons, and an output layer. The input layer receives the encoded state S_k and propagates it to the hidden layer. The hidden layer in this network processes the output from the input layer by first calculating the dot product of the incoming weights and the layer's inputs. These results are then modified by a Bayesian inference method rather than a simple activation function, allowing the model to account for uncertainty in the weights. The network utilizes the leaky rectified linear unit function (LeakyReLU) to introduce non-linearity before forwarding the outputs to the subsequent layer. The final hidden layer outputs predictions to the output layer, which then estimates the Q-values $Q_\theta(S_k, x)$ for each possible action x from set $X(S_k)$, incorporating probabilistic reasoning at every step. The architecture of the Deep Q-Learning (DQL) model employed in Phase 1 for assignment decision-making is shown in the Figure 5.3.

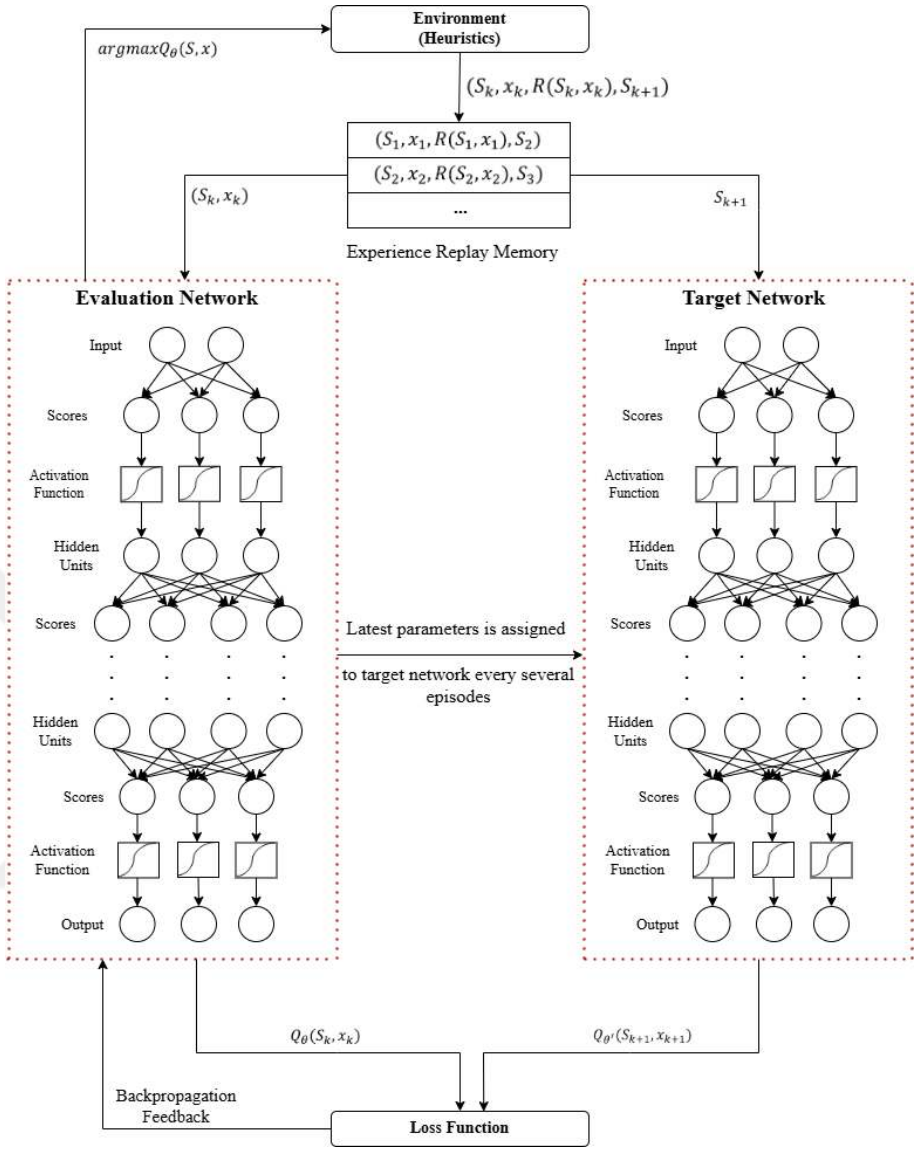


Figure 5.3: Architecture of the Deep Q-Learning (DQL) model for Phase 1 decision in RIHTP.

The parameters θ (including both weights and biases as distributions characterized by μ and σ) of the Bayesian Neural Network (BNN) are optimized during a training phase that consists of simulating a total of $N_{episodes}$ instances, each referred to as an episode. Each episode represents a working shift in a day of hospital operations, encapsulating a sequence of patient transportation requests that arrive over time, each with varying requirements.

To enhance the agent's ability to handle diverse scenarios and make robust deci-

sions under uncertainty, at the start of each episode, a daily patients request instance p is randomly sampled from a set of possible instances P. Each profile is a set of customer requests made during a working shift, including all the information required to define request j.

During each episode, the agent evaluates the current state S_k at every decision epoch k using the current distribution parameters. The agent then decides the optimal action x_k, determining the assignment of requests. After taking an action, the routing heuristic is triggered and the agent receives a reward $R(S_k, x_k)$ and transitions to the next state S_{k+1} (see Figure 5.4 for an illustration of this process). Once the post-decision state is observed, each experience tuple $(S_k, x_k, R(S_k, x_k), S_{k+1})$ is stored for later use in updating the BNN parameters.

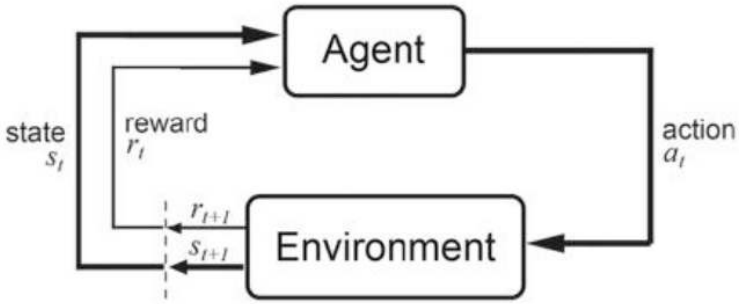


Figure 5.4: Standard training process in a RL setting [Sutton, 2018].

At each decision epoch, the agent typically chooses the action that maximizes the expected total reward for the current state S_k, a strategy known as exploitation (see Equation 5.4). However, to prevent the model from converging prematurely on suboptimal policies and to encourage exploration of the state-action space, exploration is regulated by Thompson Sampling, executed by Monte Carlo (MC) Dropout in the Bayesian Q-network. At each decision epoch k, the network executes T stochastic forward passes with distinct dropout masks, producing T sampled Q-value vectors. A sample is randomly selected to represent the current Q-value estimate, and the action with the greatest value among valid actions is chosen. This uncertainty-aware sampling technique allows the agent to equilibrate exploration and exploitation by

automatically incorporating epistemic uncertainty in its value estimates, without necessitating an explicit exploration parameter like ε.

At the end of an episode, instead of immediately using the collected experience tuples to train the BNN, they are first stored in a replay buffer with size of $\mathcal{M}$. The distribution parameters are then updated using tuples randomly sampled from this buffer, a method known as experience replay [Lin, 1992]. This technique helps to use a range of both recent and past experiences to train the BNN, reducing the correlation among sequential observations and thereby stabilizing parameter updates and enhancing the training efficacy.

Upon completion of the training phase, the optimized parameters θ^* guide the BNN policy $\pi_{DQN}^{Bayesian}$ to select actions that maximize the Q-value $Q_{\theta^*}(S_k, x)$ for any given state S_k.

5.2 Phase 2 Decision: Routing Problem

Phase 2 of the RIHTP involves a critical post-decision step where the newly assigned transportation request is integrated into the existing route of the transporter. This process utilizes a combination of Best Insertion and Variable Neighborhood Search [Mladenović and Hansen, 1997] with some adaptive mechanism to ensure optimal insertion.

We utilize the Best Insertion algorithm to generate an initial feasible solution. The solution found serves as a basis for future improvement employing the Adaptive Variable Neighborhood Search (AVNS) metaheuristic. This algorithm's inputs comprise a list of previously assigned requests, newly assigned requests, visited nodes, the current route, and all the required parameters, including service times, travel time matrix, capacities, etc. The Best Insertion algorithm is selected for its efficiency in quickly generating a feasible route by progressively integrating a new request at the position that has the minimum increase in the route's total travel time and delay penalty while adhering to defined constraints. Following the generation of an initial route, the AVNS algorithm is employed to explore a diverse range of neighborhoods. In this method, we systematically modify the initial solution by

varying the neighborhood size and type in order to have a thorough exploration of the solution space.

We represent the solution for the routing problem after assigning the newly arrived request j_k at decision epoch k just by a list of subsequent nodes $\Gamma_{v,k}$ with $J_v \subseteq J$, i.e., the assignment of the requests to the transporter $v \in V$ and its execution order. Let S be a solution of the routing problem, $C(S)$ the objective function value consists of the weighted sum of the overall travel time and the total delay penalty associated with the solution. A route S is feasible if all the constraints are satisfied and optimal if it is feasible and has the smallest value $C(S)$ among all feasible possible routes. We restrict the search to feasible solutions and directly use objective value $C(S)$ as an evaluation function. We, therefore, guide the search to remain in the feasible region by modifying the search to generate only feasible solutions.

In the first step of the algorithm, a Best Insertion algorithm, insert the request j_k to the best position in the current route of the selected transporter $v \in V$ to generate the initial route $\Gamma_{v,k}$. Nonetheless, Type 3 transportation requests, which necessitate the successive visit of related nodes, coupled with the maximum ride time constraint, provide considerable challenge in the optimal insertion of nodes. In most cases, the solution found by the standard Best Insertion algorithm is either infeasible or suboptimal. Thus, in this step, we presume that all nodes in a request must be visited sequentially, indicating the absence of request consolidation. That is, instead of adding the nodes, we seek to identify the optimal position of a request in the route. Consequently, solution S in this step is represented as J_v which shows the ordered requests to be performed by transporter $v \in V$. We use objective function $C(S)$ to evaluate the solutions, ultimately selecting the position that results in the minimum increase in $C(S)$, i.e., $Min\ \Delta C(S') = C(S) - C(S')$, where S' is the solution before inserting the new request in the route. Note that the insertion can only occur for positions in the current solution such that for such positions $A_{p_j v} < t_k$. This means that the pickup point of request j, s_j, have not been visited (fixed) upon the arrival of the new request j_k at the decision epoch k. Subsequent to this step, a local search is initiated to integrate the consolidation assumption into

the initial solution obtained. The local search utilizes the Swap operator over nodes. The algorithm involves continuously searching for a better solution by attempting to swap two nodes n and $n^{'}$ concurrently within the solution array S. Each node $n = s_{j_1}, d^1_{j_1}, d^2_{j_1}, \ldots, s_{j_{|J_v|}}, d^1_{j_{|J_v|}}, d^2_{j_{|J_v|}}$, where $j_1, \ldots, j_{|J_v|} \in J_v$, can be swapped with each node $n^{'}$, if $A_{nv} > t_k$ and $A_{n'v} > t_k$, i.e., none of the nodes has been visited before. Moreover, if a selected node belongs to a Type 3 request, all corresponding nodes associated with that request (s_j, d^1_j, d^2_j) are moved together to preserve feasibility. The operator functions by progressively enhancing a specified solution until no additional enhancements are possible. Furthermore, the swaps are permitted provided they do not lead to an infeasible solution. A solution is considered better if it has a smaller objective function value compared to the current solution.

Following the aforementioned steps, an AVNS algorithm is triggered to improve the current solution. The AVNS algorithm contains three main steps; shaking, local search, and changing neighborhood. In our study, an Exchange move is performed for shaking, and Swap and Relocate operators are implemented during local search.

The Swap operator exchanges functions according to the previous explanation. The Relocate operator relocates the position of all corresponding nodes of a request (i.e., s_j, d^1_j, d^2_j) to to a new position within the solution S. The algorithm starts with an Exchange move. Among the allowed nodes, the ones that have not been fixed, two random nodes are selected and their positions are exchanged. If one of the nodes are associated with a request of Type 3, we also change the positions of other corresponding nodes. This Exchange operator is performed for h iterations, where h is the neighborhood size. Shaking terminates at this step, after which the algorithm transitions to local search. Local search randomly implements one of the two operators. As in the first Swap operator, these operators also continue until there is no enhancement in the solution. After completing the local search, the obtained solution $S^{'}$ is compared to the current best solution S^*. If it is not a better solution, then the iteration counter (*iter*) is increased by one, the best solution is not changed, and the cycle is rerun. If a better solution is found, the best solution is updated, the iteration counter (*iter*) is increased by one, the non-improving counter *non_improvement_count* is reset, and the cycle is rerun. The loop continues until

($iter$) or $non\_improvement\_count$ exceeds the defined stopping limits, max_iter and $max\_non\_improvement$, respectively.

We have considered an adaptation in the algorithm, which makes it faster. This adaptation involves dynamically adjusting the neighborhood size h based on the success rates of previous attempts. By not persisting in ineffective neighborhoods and instead modifying its search strategy to explore potentially more beneficial areas, the algorithm optimizes the search process and prevents stagnation within less promising regions of the solution space. Also, note that due to supplementary assumptions for service robots, the heuristic for service robots includes visits to recharging stations as required; hence, the objective value of each solution is computed after integrating the recharging stations into the existing solution. In other words, throughout the entire algorithm, including insertion, local search, and shaking, we exclude the recharge station from the solution. This is because altering the position of nodes significantly affects the battery level of the robots, which in turn changes both the necessity and optimal placement of recharging stations. These changes can lead to infeasible routes that cannot be effectively handled by standard neighborhood operators. To address this challenge, we design this simple yet effective mechanism that preserves route feasibility by maintaining compatibility with battery constraints and recharging requirements. Upon identifying the solution, we incorporate the nearest recharging station into the route following the designated nodes and subsequently compute the related objective value of the solution. The pseudo-code of the algorithms are provided in Algorithm 1 and Algorithm 2.

Algorithm 1 Best Insertion Algorithm

Input: $distance\_matrix$, $service\_times$, $assigned\_request$, $new\_assigned\_request$, $fixed\_requests$, $fixed\_nodes$, $current\_route$, $current\_list$, $depot$, $trasnporter$, $robots\_list$.

Output: $best\_route$, $best\_cost$.

if $current\_route$ is empty **then**
  $current\_route.append(depot)$
  Append $pickup[new\_assigned\_request]$ to $current\_route$
  Append $delivery1[new\_assigned\_request]$ to $current\_route$
  if $delivery2[new\_assigned\_request]$ is not NaN **then**
    Append $delivery2[new\_assigned\_request]$ to $current\_route$
  end if
else
  $best\_cost \leftarrow \infty$
  $best\_route \leftarrow \{\}$
  for $i \in (len(fixed\_requests) - 1, len(current\_list) + 1)$ **do**
    $temp\_route \leftarrow \{\}$
    $temp\_list \leftarrow current\_list$
    Insert $new\_assigned\_request$ at position i in $temp\_list$
    Append $depot$ to $temp\_route$
    for $req \in temp\_list$ **do**
      Append $pickup[new\_assigned\_request]$ to $temp\_route$
      Append $delivery1[new\_assigned\_request]$ to $temp\_route$
      if $delivery2[new\_assigned\_request]$ is not NaN **then**
        Append $delivery2[new\_assigned\_request]$ to $temp\_route$
      end if
    end for
    Append $depot$ to $temp\_route$
    $temp\_cost \leftarrow w_1 * compute\_travel\_time(temp\_route) + w_3 * compute\_delay\_penalty(temp\_route)$
    if $temp\_cost < best\_cost$ **then**
      $best\_route \leftarrow temp\_route$
      $best\_cost \leftarrow temp\_cost$
    end if
    $i \leftarrow i + 1$
  end for
end if
$best\_route, best\_cost \leftarrow Swap(best\_route, fixed\_nodes)$
if $transporter \in robots\_list$ **then**
  $best\_route \leftarrow GetRouteWithRechargingStations(best\_route)$
  $best\_cost \leftarrow w_1 * compute\_travel\_time(best\_route) + w_3 * compute\_delay\_penalty(best\_route)$
end if
return $best\_route, best\_cost$

Algorithm 2 Adaptive Variable Neighborhood Search (AVNS) Algorithm - Routing Problem

Input: *distance_matrix*, *service_times*, *robots_list* , *fixed_nodes*, *initial_route*, *initial_cost*, *depot*, *trasnporter*, *max_iter*, *max_h*, *min_success_rate*, *min_neighborhood_attempts*, *max_non_improvement*.

Output: *improved_solution*, *improved_cost*.

Initialize *neighborhood_success* with *max_h* zeros
Initialize *neighborhood_attempts* with *max_h* zeros
curren_route $\leftarrow$ *initial_route*, *improved_route* $\leftarrow$ *initial_route*, *improved_cost* $\leftarrow$ *initial_cost*
iter $\leftarrow 0$, *no_improvement_count* $\leftarrow 0$
while *iter* $\leq$ *max_iter* **do**
    $h \leftarrow 1$
    $improvement\_found = False$
    while $h \leq max\_h$ **do**
        $neighborhood\_attempts[h-1] \leftarrow neighborhood\_attempts[h-1] + 1$
        $new\_route \leftarrow Shake(current\_route, h, fixed\_nodes)$
        Randomly choose one of the local search operators (*relocate or swap*)
        $new\_solution \leftarrow LocalSearch(new\_solution, fixed\_nodes)$
        if *transporter* $\in$ *robots_list* **then**
            $new\_route\_with\_station \leftarrow \{\}$
            $(new\_route\_robot \leftarrow GetRouteWithRechargingStations(new\_route)$
            $new\_cost \leftarrow w_1 * compute\_travel\_time(new\_route\_robot) + w_3 * compute\_delay\_penalty(new\_route\_robot)$
        end if
        $new\_cost \leftarrow w_1 * compute\_travel\_time(new\_route) + w_3 * compute\_delay\_penalty(new\_route)$
        if $FeasibilityConditions(new\_route) = True \wedge new\_cost < best\_cost$ **then**
            $improved\_route \leftarrow new\_solution$
            $improved\_cost \leftarrow new\_cost$
            $current\_route \leftarrow new\_route$
            $neighborhood\_success[h-1] \leftarrow neighborhood\_success[h-1] + 1$
            $improvement\_found = True$
            $h = 1$
            $no\_improvemenr\_count = 0$
        else
            $success\_rate = neighborhood\_success[h-1]/max(1, neighborhood\_attempts[h-1])$
            if $(success\_rate < min\_success\_rate) \wedge (neighborhood\_attempts[h-1] > min\_neighborhood\_attempts)$ **then**
                $h \leftarrow h + 2$
            else
                $h \leftarrow h + 1$
            end if
        end if
    end while
    if $improvement\_found = False$ **then**
        $no\_improvement\_count \leftarrow no\_improvement\_count + 1$
    end if
    if $no\_improvement\_count \geq max\_non\_improvement$ **then**
        break
    end if
    $iter \leftarrow iter + 1$
end while

45: **if** $transporter \in robots\_list$ **then**
46: $improved\_route \leftarrow GetRouteWithRechargingStations(improved\_route)$
47: $improved\_cost \leftarrow w_1 * compute\_travel\_time(improved\_route) + w_3 * compute\_delay\_penalty(improved\_route)$
48: **end if**
49: **return** $improved\_route$, $improved\_cost$

Chapter 6

NUMERICAL EXPERIMENTS

In this chapter, we provide a series of computational experiments aiming to assess the proposed RIHTP solution approaches. To simulate the unpredictable and dynamic nature of transportation demands, test instances are generated informed by real hospital data. Moreover, we utilize the Poisson process only for data generation in the routing problem to more accurately reflect the reality of the problem. The first part outlines the dataset generation procedure. Subsequently, to show the efficiency of the proposed routing heuristic, Best Insertion + VNS, we compare the performance of the heuristic to the MILP model. Next, we evaluate the Bayesian Neural Network (BNN)-based policy for dynamic assignment decisions by benchmarking it against baseline policies introduced in Appendix B, with particular attention to transporter utilization and learning progression during training. The integrated two-phase approach, which combines BNN-based assignment and heuristic routing, is then compared to established benchmarks, defined in Appendix C, to determine its effectiveness under real-time operating constraints. Finally, operational insights are examined, including the advantages of introducing service robots.

The computational runs were conducted on a PC equipped with an 11th Gen Intel® Core™ i7-11700K CPU @ 3.60 GHz, utilizing a single core, with up to 32 GB of RAM, running Windows 11 Pro (version 24H2, 64-bit operating system). Gurobi 12.0.1 was used to solve the proposed MILP model, and all other solving procedures were programmed in Python and compiled on the same computer. For the subset of experiments where runtime performance was not critical and only the quality of solutions was evaluated, computations were also conducted on Google Colab using an NVIDIA A100 GPU.

6.1 Design of Test Instances

We limit our experiments to the day shift (7:00 a.m. to 4:00 p.m.). The main set of instances was generated using real data collected from a Turkish hospital, focusing on request arrivals during regular working hours. Based on the observed distribution of demand timings within this interval, we fit an appropriate random distribution to model interarrival times. This baseline set of instances is denoted by $\mathcal{I}^{HP}$. Moreover, to better simulate the situation for routing problem assessment in Section 6.2, We generate instances using a Poisson process. The inter-arrival times were modeled using an exponential distribution with mean time value of 10 minutes. This instance set is donated as $\mathcal{I}^{PP}_{\lambda=1/10}$.

The depots have been chosen as the actual sites within the hospital. For the recharging station, ten stations are picked based on the outcomes of the p-center facility location problem. We create a full list of requests for each instance by randomly shuffling from the available request types to imitate the randomness seen in real-time demand.

For each request, the nodes were chosen from a total of 372 locations that represented possible pickup and delivery points in the hospital environment, using a layered random selection process. For each case, an origin-destination pair was utilized, and therefore, every request was individually related to potential operation locations.

In this study, the travel times between each pair of locations within the hospital are methodically calculated using empirical travel times provided by advanced RFID technology. This method entails averaging travel time obtained from prior porter movements between multiple locations, hence obtaining the average travel time under normal operational conditions. To refine these empirical estimates and improve the structural integrity of our travel times data, we use a standard technique that finds the shortest path between each location pair. The resultant matrix satisfies the triangular inequality property; that is, the travel time between any two locations directly is never more than the travel time between them through another location. Furthermore, this matrix is symmetric, showing the same travel time consistently

between locations regardless of travel direction.

Moreover, the study involves service times for the requests executed by each of the nodes in the hospital logistics network. The exact service time is carefully selected based on the nature of the request being done and whether the node serves as a pickup or delivery point. Especially for pickup points, it takes into account necessary preparations. For example, in patient transport, service time incorporates the time required to prepare and load the patient from the ward within the hospital into a transport vehicle. Conversely, for intermediate nodes that are not endpoints, service times are adjusted for each request currently under consideration. For example, nodes processing Type 3 demands have an expected waiting time that can be compared to services given a patient in a hospital. On the other hand, nodes concerned with Type 2 requests to health facilities have a service time to process the items and order before any extra traveling is allowed. Consequently, a uniform distribution is assumed for service times, with parameters selected according to the type of each request.

Various studies have adopted different assumptions regarding the speed of service robots, with reported values ranging from 0.2 to 2.1 m/s (meters per second), whereas the typical walking speed of humans in hospital environments ranges from 0.6 to 1.3 m/s [Rondoni et al., 2024, Aziez et al., 2022, Fragapane et al., 2020]. Nevertheless, to maintain the model's simplicity, we assume distinct travel speeds for service robots in order to evaluate their relative efficiency under varying speed scenarios.

Regarding the energy consumption of service robots, idle energy usage is excluded from the route optimization model. This simplification is justified by the primary focus of the study, which is on optimizing assignment and routing decisions rather than assessing energy efficiency. Furthermore, the assumption is supported by the practical implementation of energy-saving strategies, such as low-power standby modes and the deactivation of non-essential subsystems during periods of inactivity. The operational duration of service robots prior to the need for recharging is a critical factor that affects the modeling framework. Empirical evidence indicates that service robots typically maintain functionality for several hours when operated

continuously. The proposed model assumes a standardized operational duration of 240 minutes for the battery capacity of service robots (h_{max} = 240 minutes), as the analysis is restricted to a single working shift and idle energy consumption is disregarded. For the BL (π) and RL (ρ) of the (BL, RL) policy, 2% and 75% of the total capacity have been considered, respectively. In addition, the discharging rate (r) is assumed to be proportional to the time spent in operation, while the recharging rate (λ) is set at 5% of the total battery capacity per minute.

The slope of the penalty function used to penalize delays in request start time is assumed to be different across all request types based on their priority or urgency. That is, three urgency groups have been considered. The first group consists of patient transportation, where it is crucial to ensure timeliness and comfort, as it directly affects patient care and satisfaction. The second group involves medical equipment transportation, which, while important, does not carry the same immediate impact on individual well-being as patient transportation. Lastly, the third group includes the delivery of medical supplies and administrative documents, where the priority is lower, allowing for more flexibility in timing without compromising the overall functionality of healthcare services. Additionally, the model incorporates a scoring mechanism for each node, which is designed to reflect the relative effort required for different types of requests. Specifically, nodes involved in patient or medical equipment transport are assigned a significantly higher score compared to those handling other types of logistical requests. This differentiation recognizes the increased complexity and resource commitment necessary for patient and medical equipment transportation, acknowledging the critical nature of these activities in maintaining patient care standards. Given the considerable size of the hospital being analyzed, establishing an optimal maximum ride time presents substantial challenges. To address this issue, we define the maximum permissible ride time as a specific percent more than the value of the longest direct travel time recorded across all requests, varying based on the request types. This strategy ensures that the maximum ride time is both practical and operationally feasible while also providing a clear benchmark that takes into account the huge hospital's spatial dimensions and layout difficulties. Finally, for the weight in the objective function associated

with each objective, we consider a high value for the total travel time and total delay objectives since these are the crucial objectives considered by every hospital.

6.2 Numerical Results: Assessing the Best Insertion and AVNS Heuristic for Phase 2

This section systematically assesses the effectiveness and solution quality of the proposed routing heuristic. To this end, we conduct a comparative analysis between the heuristic method and the exact solutions obtained through the MILP model proposed in Chapter 4. The MILP models are solved using Gurobi Optimizer to create reliable benchmarks. In this regard, the standard Gurobi solver operates for a complete hour, and the obtained solutions are reported as our benchmark solutions.

We consider four distinct sets of test instances generated using hospital data, including 4, 6, 8, and 12 requests, each containing five randomly created instances. Thus, a total of 20 instances are generated to thoroughly assess the convergence and efficacy of the suggested routing heuristic under varying conditions. These numbers are selected since they can cover the potential number of requests that may be assigned to a transporter during a work shift. It is essential to note that these instances are challenging because they are subsequent requests and have close arrival times, which could complicate the optimization of the route. Nevertheless, Phase 2 could be more easily resolved in the main problem due to assigning requests to various transporters. We take these challenging instances into account in order to demonstrate the heuristic's effectiveness. Moreover, since the routing heuristic is the same for service robots and porters, differing only in considering recharging stations, which only impacts objective function calculations, each instance is addressed under the assumption that a single porter manages all transportation operations.

Additionally, due to the distinction between nodes' arrival time calculation in static and dynamic versions, which can significantly influence the solutions obtained, the experimental instances are conducted using dynamic assumptions to better simulate real-time intra-hospital transportation settings. Transportation requests are

assumed to arrive over time, indicating that the porter is unable to start traveling to a pickup point unless the relevant request has been properly received and known to the system. Nonetheless, this dynamic arrival assumption is not applicable to Type 3 requests, which refer to pre-scheduled transportation requests. In particular, instead of using constraint sets 4.8 and 4.57, which were designed under static assumptions, we introduce a new constraint set 6.1 that explicitly enforces the dynamic arrival conditions while preserving the correct handling of pre-scheduled (Type 3) requests. Note that for Type 3 requests, we consider the previous constraint set.

$$a^0_{nj} A_{nv} \geq \sum_{m \in N \setminus \{n\}} a^0_{nj}(t_j + d_{mn}) X_{mnv} \qquad \forall n \in P, v \in V, j \in J \setminus J_3 \tag{6.1}$$

In addition, to facilitate greater flexibility and exploration in the route optimization process, no visited nodes are fixed during this series of experiments. This elimination enables the algorithm to access a broader search space, allowing for better examination of its effectiveness, which is the main purpose of this section.

Prior to comparing the performance of the proposed heuristic with the MILP benchmark, a hyperparameter tuning procedure is conducted. The heuristic is governed by three primary hyperparameters: the maximum number of iterations, the neighborhood size parameter (h), and the maximum number of iterations allowed without improvement. We implement a hierarchical tuning approach. As the most important hyperparameter, the optimal value for the maximum number of iterations is initially determined, and subsequently, the neighborhood size is explored. Due to the influence of problem size on solution quality and computational effort, these parameters are defined as proportional to the problem size, specifically the number of requests and the total number of associated nodes. The maximum iteration limit is determined by evaluating values that are 10, 15, 20, and 30 times the number of nodes. Moreover, the neighborhood size parameter (h) is ranging from of 1, 2, and 3 times the number of assigned requests. However, the non-improving iteration threshold is established at a fixed proportion, one-fifth of the number of iterations. For instance, an instance involving four transportation requests and ten nodes results in iteration limits of 100, 150, 200, and 300, with corresponding neighborhood sizes of 4, 8, and 12 and non-improving iteration thresholds of 20, 30, 40, and 60.

The results of different maximum iterations and neighborhood sizes (h) across four problem scales, including 4, 6, 8, and 12 requests, each evaluated using 5 instances and 10 replications are reported in Tables 6.1, 6.2, 6.4 and 6.5. Moreover, Tables 6.3 and 6.6 present the average results aggregated across each problem scale individually, as well as the overall average across all scales. For each instance, the objective function value found by Best Insertion, average, standard deviation and percentage standard deviation of solutions improved by VNS over 10 replications, average and maximum runtime, and average percentage improvement over solution found by Best Insertion are shown in columns *Objective*, *Objective Avg.*, *Objective Std.*, *Objective % Std.*, *Runtime Avg.*, *Runtime Max.*, and *Objective % Difference*, respectively. These metrics evaluate the heuristic in terms of computational efficiency, robustness, and solution quality by systematically varying the hyperparameters.

The results reveal important patterns that inform the selection of an appropriate maximum iteration. At the aggregate level, increasing the maximum iteration consistently results in a reduction in the average objective values, showing higher quality of the solution. In particular, the average objective value decreases from 280.42 at 10× to 278.58 at 30×, resulting in a 0.65% gain. This moderate reduction is accompanied by a significant increase in computational time, with the average runtime increasing from 56.85 seconds to 140.73 seconds, which is more than double the computational effort involved. The reduction rate diminishes beyond the threshold of $20 \times |Nodes|$. Likewise, although the standard deviation and percentage deviation of objective values decrease gradually as the maximum iteration increases, suggesting that the solution is more stable, the rate of reduction also levels off after $20 \times |Nodes|$. As a result, the $20 \times |Nodes|$ setting encompasses nearly the entire range of achievable improvements in consistency and quality, without the substantial runtime penalty associated with the more advanced configuration.

This claim is further supporter by individual instance observations. In instance 9 with 8 requests, the heuristic obtains an average objective value of 219.51 under the 10× configuration. However, the standard deviation is quite high at 9.45 (4.31%), signifying considerable variability. Conversely, the identical case assessed with 20×

iterations consistently yields objective values under 215 and has no fluctuation, underscoring a significant enhancement in robustness and convergence. A similar trend is observed in instance 16, which is one of the most complicated test cases and contains 12 requests. $30 \times |Nodes|$ configuration yields an objective of 766.82 with a standard deviation of 0.13 (0.02%). Nonetheless, it necessitates 438.66 seconds for computation. On the other hand, the 20× arrangement achieves a roughly equivalent objective (about 766.79) in 30% reduced time. The significant increase in runtime is not justified by the marginal improvement of less than 0.03 units, which underscores that beyond a specific iteration threshold, additional computational effort only provides minimal practical benefit.

The results also indicate advantageous scalability with the rise in problem size. The average objective values increase proportionately with the number of requests, indicating the escalating complexity of the solution space as expected. Meanwhile, the stability of solutions significantly enhances with larger instances. In scenarios involving 4 requests, percentage deviations often vary from 1.6% to 4%, but in situations with 12 requests, they usually remain below 0.1%, even at significant iteration levels. This indicates that the heuristic demonstrates a type of self-regularization in larger-scale applications. Moreover, while the computational cost inherently escalates with problem size, it remains below acceptable limits under the 20× configuration. These patterns collectively confirm that $20 \times |Nodes|$ provides a strong and efficient equilibrium. It provides substantial improvements in quality and stability compared to lesser iteration counts, while avoiding the inefficiencies associated with additional iterations. The uniformity of this behavior across various problem sizes confirms its broad applicability. Consequently, the default value of maximum iteration for all subsequent heuristic assessments is established as $20 \times |Nodes|$.

Complementing this analysis, the neighborhood size parameter h is examined through controlled experiments using three settings: $|Requests|$, $2 \times |Requests|$, and $3 \times |Requests|$. All experiments are performed with the examined iteration configuration of $20 \times |Nodes|$ over the same set of 20 instances. Results demonstrate that increasing h does not yield significant enhancements in solution quality. The average objective stays largely unchanged throughout the three configurations:

279.28 for $|Requests|$, 279.85 for $2 \times |Requests|$, and 279.88 for $3 \times |Requests|$. The standard deviation for convergence stability diminishes from 2.88 at the baseline setting to 2.47 for the intermediate configuration, subsequently increasing to 3.09 when h increases to $3 \times |Requests|$. Percentage deviations exhibit a same pattern, attaining their minimum value (1.42%) at the intermediate neighborhood size while being elevated at both the lower (1.86%) and higher (2.06%) configurations. These results indicate that a moderate expansion in neighborhood size might improve consistency, but excessive growth may result in instability or inefficient search behavior.

Runtime analysis reveals a clear and consistent trend: computational effort grows substantially with neighborhood size. The average runtime increases from 74.31 seconds at $|Requests|$, to 107.34 seconds at $2 \times |Requests|$, and 126.60 seconds at $3 \times |Requests|$. Maximum runtimes show a parallel trend. This expected rise is caused by the requirement to evaluate supplementary potential solutions in each iteration due to the expansion of larger neighborhoods. Yet, the higher computational burden does not provide proportional improvements in outcomes. Numerous cases, including instance 3 (consisting of 4 requests) and instance 17 (comprising 12 requests), provide objective values that are either equivalent to or somewhat inferior to those obtained with lower neighborhood sizes, while requiring significantly more processing time. These cases illustrate that above a certain threshold, larger neighborhoods may reduce the efficacy of local development and postpone convergence owing to excessive exploration.

Consequently, the fundamental configuration of $k = |Requests|$ offers the optimal balance of solution quality, convergence stability, and computing efficiency. The intermediate setting $k = 2 \times |Requests|$ offers a little enhancement in variability; nevertheless, the benefits are insufficient to justify the increased runtime. The larger neighborhood size imposes extra computational burden without much benefit and, in certain instances, it underperforms. Consequently, the baseline neighborhood size is preserved in the final heuristic configuration.

Overall, for future examinations, we select $20 \times |Nodes|$ as maximum number of iterations, $|Requests|$ as neighborhood sizes, and $4 \times |Nodes|$ as the maximum allowed iterations without improvement.

Table 6.1: The dynamic routing problem results: impact of maximum iteration on heuristic's performance.

Instance Number	Request Numbers	Heuristic Hyperparameters: Maximum Iterations	Heuristic Hyperparameters: Neighborhoods (h)	Heuristic Hyperparameters: Maximum NonImproving	BI: Objective	BI + VNS: Objective: Avg.	BI + VNS: Objective: Std.	BI + VNS: Objective: % Std.	BI + VNS: Runtime: Avg.	BI + VNS: Runtime: Max	Objective % Difference
1	4	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	23.83	23.56	0.42	1.78%	1.14	1.75	1.10%
2	4	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	77.63	77.63	0.00	0.00%	0.98	1.27	0.00%
3	4	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	66.06	50.72	7.38	14.55%	3.31	4.79	23.22%
4	4	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	26.46	21.50	1.74	8.09%	1.33	1.75	18.74%
5	4	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	31.61	31.58	0.09	0.28%	0.98	1.81	0.09%
6	6	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	59.76	55.28	4.04	7.31%	5.40	7.95	7.50%
7	6	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	170.68	170.78	0.00	0.00%	5.32	6.44	0.00%
8	6	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	142.22	141.84	0.49	0.35%	12.82	21.42	0.27%
9	6	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	58.82	57.74	3.23	5.59%	5.52	9.52	1.83%
10	6	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	116.54	111.44	0.00	0.00%	5.67	7.64	4.38%
11	8	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	241.71	219.51	9.45	4.31%	34.07	57.30	7.50%
12	8	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	245.56	238.87	2.35	0.98%	24.46	33.87	2.72%
13	8	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	313.12	391.83	4.79	1.22%	41.14	53.28	5.15%
14	8	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	127.10	124.61	0.61	0.49%	17.84	23.20	1.96%
15	8	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	294.70	275.91	0.00	0.00%	18.75	22.33	6.38%
16	12	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	843.59	766.76	0.00	0.00%	179.90	243.11	9.11%
17	12	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	1019.54	957.94	18.43	1.92%	248.02	474.18	6.04%
18	12	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	799.83	777.09	11.05	1.42%	265.68	462.47	2.84%
19	12	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	448.10	435.41	6.34	1.46%	114.41	166.18	2.83%
20	12	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	705.25	678.41	0.00	0.00%	150.26	194.66	3.81%
1	4	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	23.83	23.39	0.46	1.98%	1.79	2.95	1.84%
2	4	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	77.63	77.63	0.00	0.00%	1.50	1.98	0.00%
3	4	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	66.06	50.28	7.52	14.97%	4.06	4.82	23.89%
4	4	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	26.46	23.16	2.85	12.29%	41.76	2.61	12.49%
5	4	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	31.61	31.55	0.12	0.37%	1.5	2.02	0.18%
6	6	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	59.76	51.47	3.84	7.46%	8.43	11.32	13.87%
7	6	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	170.68	170.68	0.00	0.00%	7.96	10.18	0.00%
8	6	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	142.22	141.68	1.11	0.78%	18.16	30.14	0.38%
9	6	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	58.82	58.22	1.63	2.81%	6.94	9.79	1.02%
10	6	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	116.54	111.44	0.00	0.00%	7.40	8.78	4.38%
11	8	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	241.71	217.16	5.53	2.53%	49.04	83.75	10.16%
12	8	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	245.56	238.13	0.00	0.00%	35.00	44.10	3.03%
13	8	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	413.12	389.80	3.66	0.94%	51.43	63.46	5.64%
14	8	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	127.10	112.16	4.08	3.34%	29.69	41.15	3.89%
15	8	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	294.70	276.34	1.39	0.50%	25.56	31.10	6.23%
16	12	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	843.56	766.88	0.15	0.02%	263.22	326.20	9.09%
17	12	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	1019.54	971.06	23.04	2.37%	284.19	372.18	4.76%
18	12	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	799.83	771.17	16.15	2.09%	332.87	473.11	3.58%
19	12	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	448.10	428.67	4.87	1.14%	157.51	191.40	4.33%
20	12	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	705.25	678.41	0.00	0.00%	198.10	241.96	3.81%
1	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	23.83	23.39	0.46	1.97%	2.17	3.15	1.84%
2	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	77.63	77.63	0.00	0.00%	1.91	2.30	0.00%
3	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	66.06	47.58	4.94	10.38%	4.98	7.62	27.97%
4	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	26.46	21.5	1.74	8.09%	2.14	2.97	18.75%
5	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	31.61	31.53	0.14	0.44%	1.84	2.70	0.25%
6	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	59.76	52.80	2.62	4.96%	9.66	14.44	11.65%
7	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	170.68	170.68	0.00	0.00%	10.77	13.87	0.00%
8	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	142.22	140.44	1.52	1.08%	32.36	49.97	1.25%
9	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	58.82	58.26	1.43	2.45%	9.63	14.54	0.95%
10	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	116.54	111.44	0.00	0.00%	9.73	11.71	4.38%
11	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	241.71	214.95	0.21	0.10%	60.98	109.06	11.07%
12	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	245.56	238.13	0.00	0.00%	45.79	65.18	3.03%
13	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	413.12	390.89	4.62	1.18%	72.52	100.18	5.38%
14	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	127.10	123.64	0.94	0.76%	42.12	60.13	2.72%
15	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	294.70	275.91	0.00	0.00%	33.38	46.4	6.38%

Table 6.2: The dynamic routing problem results: impact of maximum iteration on heuristic's performance.-*Cont′d*

Instance Number	Request Numbers	Heuristic Hyperparameters			BI	BI + VNS					Objective % Difference
		Maximum Iterations	Neighborhoods (h)	Maximum NonImproving	Objective	Objective			Runtime		
						Avg.	Std.	% Std.	Avg.	Max	
16	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	843.59	766.79	0.09	0.01%	322.58	496.40	9.10%
17	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	1019.54	962.07	20.73	2.15%	425.25	609.74	5.64%
18	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	799.83	763.19	17.31	2.27%	513.43	766.87	4.58%
19	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	448.10	436.36	0.88	0.20%	229.01	314.44	2.62%
20	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	705.25	678.41	0.00	0.00%	244.39	272.52	3.81%
1	4	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	23.83	23.12	0.37	1.60%	3.10	3.91	2.95%
2	4	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	77.63	77.63	0.00	0.00%	3.09	3.77	0.00%
3	4	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	66.06	45.60	0.00	0.00%	6.63	8.93	30.96%
4	4	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	26.46	20.95	0.00	0.00%	3.10	3.84	20.82%
5	4	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	31.61	31.39	0.12	0.38%	3.24	3.89	0.71%
6	6	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	59.76	50.44	3.14	6.23%	14.07	22.94	15.59%
7	6	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	170.68	170.68	0.00	0.00%	15.81	17.40	0.00%
8	6	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	142.22	141.74	0.50	0.35%	37.40	53.54	0.33%
9	6	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	58.82	55.49	4.62	8.33%	18.59	53.50	5.66%
10	6	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	116.54	111.44	0.00	0.00%	13.62	16.03	4.38%
11	8	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	241.71	214.56	0.30	0.14%	107.50	174.09	11.23%
12	8	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	245.56	238.13	0.00	0.00%	58.68	64.42	3.03%
13	8	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	413.12	387.04	2.40	0.62%	101.90	126.87	6.31%
14	8	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	127.10	122.51	4.25	3.47%	56.13	174.98	3.61%
15	8	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	294.70	276.34	1.39	0.50%	47.42	152.69	6.23%
16	12	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	843.59	766.82	0.13	0.02%	438.66	485.00	9.10%
17	12	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	1019.54	962.07	20.73	2.15%	553.62	721.24	5.64%
18	12	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	799.83	766.54	12.21	1.59%	696.76	1037.24	4.16%
19	12	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	488.10	430.69	5.86	1.36%	271.51	322.38	3.88%
20	12	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	705.25	678.41	0.00	0.00%	363.75	411.14	3.81%

Table 6.3: Summary: impact of maximum iteration on heuristic's performance.

Request Numbers	Heuristic Hyperparameters			BI + VNS					Avg. Objective % Difference
	Maximum Iterations	Neighborhoods (k)	Maximum NonImproving	Objective			Runtime		
				Avg.	Std.	% Std.	Avg.	Max	
4	10×| *Nodes* |	| *Requests* |	3×| *Nodes* |	41.00	1.93	4.94%	1.55	2.27	8.63%
4	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	41.20	2.19	5.92%	2.12	2.88	7.68%
4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	40.33	1.46	4.18%	2.61	3.75	9.76%
4	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	39.74	0.10	0.40%	3.83	4.87	11.09%
6	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	107.42	1.55	2.65%	6.95	10.59	2.80%
6	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	106.70	1.32	2.21%	9.78	14.04	3.93%
6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	106.72	1.11	1.70%	14.43	20.91	3.65%
6	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	105.96	1.65	2.98%	19.90	32.68	5.19%
8	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	250.15	3.44	1.40%	27.25	38.00	4.74%
6	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	248.72	2.93	1.46%	38.14	52.71	5.79%
8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	248.70	1.15	0.41%	50.96	76.19	5.72%
8	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	247.72	1.67	0.95%	74.33	138.61	6.08%
12	10×| *Nodes* |	| *Requests* |	3×| *Nodes* |	723.12	7.16	0.96%	191.65	308.12	4.93%
12	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	723.24	8.84	1.12%	247.18	320.99	5.11%
12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	721.36	7.80	1.16%	346.93	491.99	5.15%
12	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	720.91	7.79	1.03%	464.86	595.54	5.32%
Total	10×| *Nodes* |	| *Requests* |	2×| *Nodes* |	280.42	3.52	2.49%	56.85	89.74	5.28%
Total	15×| *Nodes* |	| *Requests* |	3×| *Nodes* |	279.97	3.82	2.68%	103.73	148.21	5.63%
Total	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	279.28	2.88	1.86%	74.31	97.66	6.07%
Total	30×| *Nodes* |	| *Requests* |	6×| *Nodes* |	278.58	2.8	1.34%	140.73	192.92	6.92%

Table 6.4: The dynamic routing problem results: impact of neighborhood size (h) on heuristic's performance.

Instance Number	Request Numbers	Heuristic Hyperparameters			BI	BI + VNS					Objective Percentage Difference
		Maximum Iterations	Neighborhoods (h)	Maximum NonImproving	Objective	Objective			Runtime		
						Avg.	Std.	% Std.	Avg.	Max	
1	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	23.83	23.39	0.46	0.00%	2.17	3.15	1.84%
2	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	77.63	77.63	0.00	0.00%	1.91	2.30	0.00%
3	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	66.06	47.58	4.94	0.00%	4.98	7.62	27.97%
4	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	26.46	21.5	1.74	0.00%	2.14	2.97	18.75%
5	4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	31.61	31.53	0.14	0.00%	1.84	2.70	0.25%
6	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	59.76	52.80	2.62	0.00%	9.66	14.44	11.65%
7	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	170.68	170.68	0.00	0.00%	10.77	13.87	0.00%
8	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	142.22	140.44	1.52	0.00%	32.36	49.97	1.25%
9	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	58.82	58.26	1.43	0.00%	9.63	14.54	0.95%
10	6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	116.54	111.44	0.00	0.00%	9.73	11.71	4.38%
11	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	241.71	214.95	0.21	0.00%	60.98	109.06	11.07%
12	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	245.56	238.13	0.00	0.00%	45.79	65.18	3.03%
13	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	413.12	390.89	4.62	0.00%	72.52	100.18	5.38%
14	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	127.10	123.64	0.94	0.00%	42.12	60.13	2.72%
15	8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	294.70	275.91	0.00	0.00%	33.38	46.4	6.38%
16	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	843.59	766.79	0.09	0.00%	322.58	496.40	9.10%
17	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	1019.54	962.07	20.73	0.00%	425.25	609.74	5.64%
18	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	799.83	763.19	17.31	0.00%	513.43	766.87	4.58%
19	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	448.10	436.36	0.88	0.00%	229.01	314.44	2.62%
20	12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	705.25	678.41	0.00	0.00%	244.39	272.52	3.81%
1	4	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	23.83	23.21	0.42	1.83%	2.96	4.68	2.58%
2	4	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	77.63	77.63	0.00	0.00%	2.62	3.31	0.00%
3	4	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	66.06	45.60	0.00	0.00%	6.47	8.45	30.96%
4	4	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	26.46	21.50	1.74	8.10%	2.84	3.60	18.74%
5	4	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	31.61	31.53	0.14	0.43%	2.33	3.51	0.27%
6	6	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	59.76	51.06	3.12	6.11%	11.52	14.16	14.56%
7	6	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	170.68	170.68	0.00	0.00%	13.13	15.15	0.00%
8	6	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	142.22	141.14	1.35	0.96%	27.95	35.76	0.76%
9	6	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	58.82	85.58	0.32	0.55%	10.72	17.98	0.40%
10	6	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	116.54	111.44	0.00	0.00%	11.45	13.32	4.38%
11	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	241.71	214.87	0.24	0.11%	68.66	120.19	11.10%
12	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	245.56	238.13	0.00	0.00%	51.25	72.91	3.03%
13	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	413.12	391.21	5.31	1.36%	79.86	118.82	5.30%
14	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	127.10	122.53	4.22	3.45%	46.31	75.58	3.60%
15	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	294.70	276.34	1.39	0.50%	38.14	50.93	6.23%
16	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	843.59	766.79	0.09	0.01%	320.48	410.37	9.10%
17	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	1019.54	966.77	22.51	2.33%	347.65	395.65	5.18%
18	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	799.83	774.62	14.89	1.92%	547.86	1090.22	3.15%
19	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	448.10	434.87	3.36	0.77%	268.13	465.56	2.95%
20	8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	705.25	4678.32	0.19	0.03%	276.50	314.03	3.82%
1	4	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	23.83	23.21	0.42	1.81%	3.32	4.16	2.60%
2	4	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	77.63	77.63	0.00	0.00%	3.26	4.09	0.00%
3	4	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	66.06	47.16	4.91	10.41%	6.75	12.21	28.61%
4	4	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	26.46	21.50	1.74	8.10%	3.44	4.23	18.75%
5	4	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	31.61	31.47	0.15	0.48%	3.17	4.70	0.44%
6	6	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	59.76	51.08	2.87	5.61%	13.14	21.47	14.52%
7	6	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	170.68	170.68	0.00	0.00%	14.69	18.09	0.00%
8	6	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	142.22	141.58	1.10	0.78%	31.03	51.02	0.45%
9	6	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	58.82	58.40	0.90	1.54%	14.08	20.39	0.71%
10	6	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	116.54	111.44	0.00	0.00%	14.10	16.04	0.71%
11	8	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	241.71	215.00	0.16	0.07%	66.26	91.35	11.05%
12	8	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	245.56	238.87	2.35	0.98%	57.36	63.05	2.72%
13	8	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	413.12	387.57	4.10	1.06%	89.53	132.29	6.19%
14	8	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	127.10	122.74	4.30	3.50%	45.38	57.70	3.43%
15	8	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	294.70	275.91	4.30	1.55%	44.63	51.35	6.38%

Table 6.5: The dynamic routing problem results: impact of neighborhood size (h) on heuristic's performance.-*Cont'd*

Instance Number	Request Numbers	Heuristic Hyperparameters: Maximum Iterations	Heuristic Hyperparameters: Neighborhoods (h)	Heuristic Hyperparameters: Maximum NonImproving	BI: Objective	BI + VNS: Objective: Avg.	BI + VNS: Objective: Std.	BI + VNS: Objective: % Std.	BI + VNS: Runtime: Avg.	BI + VNS: Runtime: Max	Objective Percentage Difference
16	12	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	843.59	766.85	0.14	0.02%	381.30	539.66	9.10%
17	12	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	1019.54	962.32	21.11	2.19%	585.20	674.90	5.61%
18	12	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	799.83	766.77	15.44	2.01%	595.11	1076.08	4.13%
19	12	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	448.10	434.96	4.93	1.13%	267.31	381.29	2.93%
20	12	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	705.25	678.41	0.00	0.00%	286.86	327.89	3.81%

Table 6.6: Summary: impact of neighborhood size (h) on heuristic's performance.

Request Numbers	Heuristic Hyperparameters: Maximum Iterations	Heuristic Hyperparameters: Neighborhoods (k)	Heuristic Hyperparameters: Maximum NonImproving	BI + VNS: Objective: Avg.	BI + VNS: Objective: Std.	BI + VNS: Objective: % Std.	BI + VNS: Runtime: Avg.	BI + VNS: Runtime: Max	Avg. Objective % Difference
4	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	40.33	1.46	4.18%	2.61	3.75	9.76%
4	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	39.89	0.46	2.07%	3.44	4.71	10.51%
4	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	40.99	1.44	4.16%	3.99	5.88	10.48%
6	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	106.72	1.11	1.70%	14.43	20.91	3.65%
6	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	106.58	0.96	1.52%	14.95	19.27	4.02%
6	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	106.24	1.17	1.59%	17.81	25.80	3.28%
8	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	248.70	1.15	0.41%	50.96	76.19	5.72%
8	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	248.62	2.23	1.08%	58.84	87.69	5.85%
8	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	248.82	3.44	1.43%	60.63	79.75	5.15%
12	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	721.36	7.80	1.16%	346.93	491.99	5.15%
12	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	724.32	8.21	1.01%	352.12	535.17	4.84%
12	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	721.46	8.32	1.07%	423.96	599.96	5.12%
Total	20×| *Nodes* |	| *Requests* |	4×| *Nodes* |	279.28	2.88	1.86%	74.31	97.66	6.07%
Total	20×| *Nodes* |	2×| *Requests* |	4×| *Nodes* |	279.85	2.47	1.42%	107.34	161.21	6.31%
Total	20×| *Nodes* |	3×| *Requests* |	4×| *Nodes* |	279.88	3.09	2.06%	126.60	177.85	6.01%

Next, we assess the performance of the suggested heuristic by comparing it with the optimal solutions derived from the MILP model. We use the same 15 instance that we employed to tune the hyperparameters. In addition to the 15 instances analyzed in the preceding part, we generate and evaluate an additional set of 15 instances to further verify the robustness and generalizability of the heuristic. These additional instances are generated with an exponential distribution to represent interarrival times between transportation requests, with mean value of 10 minutes. The second set of instances more accurately simulates the assignment situation in real-time with several transporters, where requests are assigned to a transporter

with longer intervals between arrivals.

We do not report the results for instances with 12 requests, as the MILP model fails to produce competitive solutions or, in some cases, even a feasible one. Despite allocating a time limit of two hours and providing warm-start solutions derived from the NoRel heuristic, mainly designed for integer programming models in which solving the root linear relaxation poses significant difficulty [Gurobi Optimization, LLC, 2024], the solver often fails to converge to feasible or high-quality outcomes. Moreover, our computational experiments reveal that the linear relaxation provides very loose lower bounds, resulting in a substantial optimality gap throughout most of the branch-and-bound process. In several instances, the optimality gap remains above 30% even in the final iterations. Given these persistent computational challenges and the fact that the 12-request instances represent the largest-scale settings we tested, many of which lead to such problematic assignments, we exclude these cases from our reported results.

Tables 6.7 and 6.8 present a comprehensive comparison between the heuristic solution approach and the exact MILP model, with a focus on both solution quality and computational efficiency. The evaluation metrics include the absolute and percentage differences in objective values, as well as the runtime ratio, which is computed as the ratio of the MILP runtime to the BI+VNS runtime. These differences are calculated relative to the MILP outcomes and are reported using minimum, maximum, and average values across individual instance sets as well as the entire dataset.

Across small-scale $\mathcal{I}^{HP}$ instances with 4 requests, the BI+VNS heuristic achieves high solution fidelity, with an average objective value deviation of just 1.91%, ranging from exact matches to a maximum of 4.34%. Similarly, for $\mathcal{I}^{PP}_{\lambda=1/10}$ instances of the same size, the heuristic attains nearly identical solutions in all cases, with an average gap of only 0.35% and several instances reporting a 0.00% difference. These results highlight the accuracy of BI+VNS in smaller problem settings.

In terms of computational efficiency, BI+VNS offers substantial speed advantages in this regime. Runtime ratios for 4-request instances average 0.19 ($\mathcal{I}^{HP}$) and 0.23 ($\mathcal{I}^{PP}_{\lambda=1/10}$), indicating that MILP executes slightly faster in absolute time for these

small problems. However, the heuristic's performance remains acceptable, particularly considering that solution quality is not compromised and runtime remains in the sub-second to few-second range.

As the problem size increases to 6 requests, the relative advantages of BI+VNS become more apparent. In the $\mathcal{I}^{HP}$ group, the average objective gap increases to 7.94%, primarily driven by a few outliers, such as Instance 9, where the heuristic deviates by 23.54%. Nevertheless, the heuristic maintains high accuracy in the majority of instances, with the smallest gap at just 1.24%. For $\mathcal{I}^{PP}_{\lambda=1/10}$ instances with 6 requests, BI+VNS matches the MILP solutions exactly across all tested cases, reflecting a 0.00% average gap. Notably, MILP runtimes increase dramatically in this setting, with one instance requiring over 32 minutes (Instance 6), while BI+VNS solves all instances in under 18 seconds. The resulting runtime ratios average 7.96 ($\mathcal{I}^{HP}$) and 42.28 ($\mathcal{I}^{PP}_{\lambda=1/10}$), with peaks exceeding 198× in the most extreme case. These results emphasize the superior scalability of the heuristic.

For 8-request instances, MILP's limitations become more severe. While it manages to solve all $\mathcal{I}^{HP}$ instances and four of the five $\mathcal{I}^{PP}_{\lambda=1/10}$ problems, it requires the full 3600-second time limit in most cases. Despite this, the heuristic achieves an average gap of just 3.51% ($\mathcal{I}^{HP}$) and 1.75% ($\mathcal{I}^{PP}_{\lambda=1/10}$), with one instance even slightly outperforming the MILP solution (Instance 15, $\mathcal{I}^{HP}$: -0.45%; Instance 11, Exp: -3.06%). Runtime ratios escalate drastically in this group, with averages of 76.12 ($\mathcal{I}^{HP}$) and 63.39 ($\mathcal{I}^{PP}_{\lambda=1/10}$), and maximum values reaching 107.85 and 117.92, respectively. These figures underscore the practical infeasibility of using MILP for medium-to-large problems and the significant computational benefit of BI+VNS.

From a statistical perspective, BI+VNS achieves an overall average objective difference of 4.45% for $\mathcal{I}^{HP}$ instances and just 0.70% for $\mathcal{I}^{PP}_{\lambda=1/10}$, demonstrating greater robustness on non-uniform request distributions. The maximum observed difference (23.54%) is isolated, and the heuristic performs within 3% of optimality in most cases. The corresponding average runtime ratios—28.09 for $\mathcal{I}^{HP}$ and 35.30 for Exp—illustrate the extreme computational advantage of the heuristic, especially for larger instances.

Overall, the BI+VNS heuristic demonstrates strong performance across a wide

range of problem settings. It achieves near-optimal solutions in most small- and medium-scale cases and consistently outperforms MILP in terms of runtime. The heuristic is particularly effective on stochastic instance sets, where MILP struggles with numerical and combinatorial complexity. These findings reinforce the practical applicability of BI+VNS in our real-time transportation contexts, where timely and reliable decisions are critical and exact methods are computationally infeasible.

Table 6.7: Comparison of MILP with time limit and BI+VNS performance across problem instances

Type	**Number of Requests**	**Instance**	**MILP-TL**		**BI+VNS**		**Obj % Diff**	**Runtime Ratio**
			Objective	**Runtime**	**Objective**	**Runtime**		
$\mathcal{I}^{HP}$	4	1	22.94	0.46	23.39	2.17	1.96%	0.21
		2	77.63	0.25	77.63	1.91	0.00%	0.13
		3	45.60	0.23	47.58	4.98	4.34%	0.05
		4	20.95	0.40	21.50	2.14	2.63%	0.19
		5	31.33	0.71	31.53	1.84	0.64%	0.39
	6	6	47.90	87.11	52.80	9.66	10.23%	9.02
		7	165.96	119.07	170.68	10.77	2.84%	11.06
		8	138.72	26.31	140.44	32.36	1.24%	0.81
		9	47.16	139.43	58.26	9.63	23.54%	14.48
		10	109.42	43.15	111.44	9.73	1.85%	4.43
	8	11	212.99	3600.00	214.95	60.98	0.92%	59.04
		12	232.40	3600.00	238.13	45.79	2.47%	78.62
		13	382.48	3600.00	390.89	72.52	2.20%	49.64
		14	109.98	3600.00	123.64	42.12	12.42%	85.47
		15	277.15	3600.00	275.91	33.38	-0.45%	107.85
$\mathcal{I}^{PP}_{\lambda=1/10}$	4	1	24.58	0.79	25.01	1.60	1.73%	0.49
		2	42.68	0.81	42.68	2.64	0.00%	0.31
		3	24.77	0.11	24.77	1.52	0.00%	0.07
		4	16.93	0.31	16.93	1.63	0.00%	0.19
		5	56.30	0.29	56.30	2.75	0.00%	0.11
	6	6	71.32	1971.36	71.32	9.96	0.00%	198.01
		7	63.23	213.96	63.23	17.04	0.00%	12.55
		8	73.63	0.65	73.63	10.79	0.00%	0.06
		9	27.59	34.69	27.59	10.59	0.00%	3.27
		10	78.96	2.49	78.97	14.63	0.00%	0.17
	8	11	119.38	3600.00	115.73	30.53	-3.06%	117.92
		12	141.23	3600.00	145.05	63.77	2.71%	56.46
		13	98.57	50.95	98.62	44.89	0.05%	1.13
		14	39.80	3600.00	40.81	32.81	2.55%	109.73
		15	86.79	3600.00	92.43	59.40	6.50%	31.73

Table 6.8: Statistical comparison of BI + VNS and MILP with time limit solutions across different problem sizes.

Instance Set	**Performance Metric**	**Stats**	**Number of Requests**			**Total %**
			4 requests	**6 requests**	**8 requests**	Diff.
$\mathcal{I}^{HP}$	Objective Value % Difference	Avg.	1.91%	7.94%	3.51%	4.45%
		Min.	0.00%	1.24%	-0.45%	0.26%
		Max.	4.34%	23.54%	12.42%	13.43%
	Runtime Ratio	Avg.	0.19	7.96	76.12	28.09
		Min.	0.05	0.81	49.64	16.83
		Max.	0.39	14.48	107.85	40.91
$\mathcal{I}^{P}_{\lambda=1/10}$	Objective Value % Difference	Avg.	0.35%	0.00%	1.75%	0.70%
		Min.	0.00%	0.00%	-3.06%	-1.02%
		Max.	1.73%	0.00%	6.50%	2.74%
	Runtime Ratio	Avg.	0.23	42.28	63.39	35.30
		Min.	0.07	0.06	1.13	0.42
		Max.	0.31	198.01	117.92	105.41

6.3 Numerical Results: Assessing the BNN-based Policy for Phase 1

This section illustrates the progression of learning that takes place during the training of the DQN and provides the results of a comparison between the learned policies and the baseline policies in terms of transporter utilization and performance.

6.3.1 Baseline Policies

To evaluate the effectiveness of the learned assignment policy $\pi^{Bayesian}_{DQN}$, we establish a comparative framework based on a set of representative baseline policies. These baseline policies serve as benchmarks, providing reference points against which the relative performance of $\pi^{Bayesian}_{DQN}$ can be measured. Specifically, we define three baseline policies: a *randomized policy*, a *heuristic policy*, *NN-based policy*, and *hospital's operational policy*, denoted by $\pi_{randomized}$, $\pi_{heuristic}$, $\pi^{Deterministic}_{DQN}$, and $\pi_{hospital}$, respectively. These policies are selected to reflect varying levels of operational sophistication, from naive allocation to context-aware assignment strategies, and collectively function as straw man baselines for empirical evaluation. The explanations of these policies are summarized in Table 6.9. However, thorough explanations are

provided in Appendix B.

Table 6.9: Summary of baseline policies and their characteristics.

Policy Name	**Notation**	**Brief Explanation**
Randomized Policy	$\pi_{randomized}$	Assigns a transporter to each request uniformly at random, providing a naive benchmark.
Heuristic Policy	$\pi_{heuristic}$	Uses a weighted cost function based on workload, travel time, and delay to choose the best transporter.
NN-based Policy	$\pi_{DQN}^{Deterministic}$	Employs a standard DQN using the same features as the BNN, assigning the request to the transporter with the highest estimated Q-value.
Hospital Policy	$\pi_{hospital}$	Reflects the actual hospital strategy; assigns requests based on proximity and current transporters states.

6.3.2 Features

A structured representation of the system state is employed to train the DQN in our experimental setup. This representation includes three categories of features: temporal, request-specific, and transporter-related.

Given a state S_k, the temporal component is captured by the scalar feature t_k, representing the current time at which the decision must be made. This inclusion enables the agent to acquire policies that are responsive to the dynamics of request accumulation and service constraints over the planning horizon, as well as to the progression of time.

To characterize the operational status of the transporters in state S_k, we extract and encode a collection of aggregate performance metrics consisting of the cumulative travel time, total workload, and accumulated delay penalties that are linked to the current and future assignments of each transporter. These features are essential for the agent to make decisions that are equitable in the distribution of workloads, route efficiency, and delay minimization.

In addition to temporal and transporter-specific information, features that characterize the properties of the incoming service request are encoded in the state representation. In order to achieve this, we have implemented a categorical feature that specifies the nature of the received request. This feature enables the agent to distinguish between request priorities, service time distributions, and capacity constraints.

6.3.3 Training Settings

To have an uncertainty-aware agent that is able to capture stochastic conditions, a Deep Q-Network (DQN) modeled as a multilayer perceptron within a Bayesian Neural Network (BNN) framework is employed. The DQN architecture consists of Bayesian linear layers, each containing N_{nodes} neurons across N_{layers} hidden layers, followed by an output layer with $|X(S_k)|$ neurons. In contrast to deterministic DQN, which generates fixed estimates of Q-values, BNNs maintain a distribution of weights and biases, enabling the model to determine its confidence in the value estimates. Each layer in the BNN is implemented using a Bayesian linear transformation, where the parameters are modeled as independent Gaussian random variables. Specifically, for each weight w and bias b, the posterior distribution is approximated as $q(w) = \mathcal{N}(\mu_w, \sigma_w^2)$ and $q(b) = \mathcal{N}(\mu_b, \sigma_b^2)$. However, to guarantee that the standard deviation σ_w remains positive and numerically stable, we reparameterize it via softplus transformation $\sigma_w = \log(1 + e^{\rho_w})$, where ρ_w is an unfixed learnable parameter. This technique is a common formulation employed in variational BNNs, as it enables direct optimization through gradient-based methods [Blundell et al., 2015].

During the forward pass, weights are sampled via the reparameterization trick, $\tilde{w} = \mu_w + \sigma_w \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$, which allows the sampling process to be differentiable and thus compatible with backpropagation [Kingma et al., 2013]. The sampled weights are then used to compute linear activations, which are passed through a nonlinear activation function. In order to preserve gradient flow in negative activation regimes and to decrease the prevalence of inactive ("dead") neurons, which can diminish uncertainty propagation in deep Bayesian networks, LeakyReLU is selected

over the standard ReLU [Tempczyk et al., 2022].

To further promote generalization and capture model uncertainty, we apply Monte Carlo (MC) Dropout after each layer during training, following the Bayesian approximation perspective proposed in [Gal and Ghahramani, 2016]. In fact, at training time, the network performs a stochastic forward passes, with a stochastic realization of weights and dropout masks. The Q-value for a state-action pair (S_k, x_k) is then estimated. This ensemble-like behavior helps calibrate confidence intervals around the predicted Q-values. However, due to the computational expense of calculating all potential actions in a state, we opt to greedily consider N actions in each state instead. This selection examines a weighted sum of the state vector associated with each transporter, denoted as $w_1 \cdot TT_v(t_k) + w_2 \cdot WL_v(t_k) + w_3 \cdot DP_v(t_k)$, where $WL_v(t_k)$ represents the current workload, $TT_v(t_k)$ signifies the travel time component, and $DP_v(t_k)$ indicates the delay penalty of the transporter's current route v. Based on our preliminary results, this technique may lead to improved long-term decision-making.

The objective in training the Bayesian Neural Network (BNN) is to minimize a loss function that balances predictive accuracy with model regularization. The loss consists of two components: the mean-squared temporal difference (TD) error [Lee and He, 2019] and a Kullback-Leibler (KL) divergence [Zhang et al., 2023] that penalizes deviation from a predefined prior distribution. For a minibatch of transitions $\{(S_k, x_k, R(S_k, x_k), S_{k+1})\}_{k=1}^{B}$, the loss is expressed as:

$$\mathcal{L}(\theta) = \frac{1}{B} \sum_{i=1}^{B} \left(Q_\theta(S_k, x_k) - y_i\right)^2 + \lambda_{KL} \cdot KL\left[q(\theta) \,\|\, p(\theta)\right], \tag{6.2}$$

where B is the minibatch size, y_i denotes the target Q-value with parameter θ^- for the i-th transition, and λ_{KL} is the KL weight. To accelerate learning and improve temporal credit assignment, we employ n-step bootstrapping for target estimation. Instead of using only the immediate reward, the target incorporates the discounted sum of the first n rewards, followed by a bootstrapped Q-value from the target network at step $k+n$. Formally, the target is defined as:

$$y_i = R_k^{(n)} + \gamma^n \cdot \max_{x' \in X(S_{k+n})} Q_{\theta^-}(S_{k+n}, x') \cdot \mathbb{1}_{\{\neg \text{Done}_{k+n}\}}, \tag{6.3}$$

where $R_k^{(n)} = \sum_{j=0}^{n-1} \gamma^j R(S_{k+j}, x_{k+j})$ is the accumulated discounted reward, γ is the discount factor, and $\mathbb{1}_{\{\neg \text{Done}_{k+n}\}}$ is an indicator function that disables bootstrapping if the episode terminates before step $k+n$. This formulation enables more informative target signals and supports faster convergence during training.

The model is trained using the Adam optimizer proposed by [Kingma and Ba, 2014] with a learning rate η and a KL weight λ_{KL} that increases periodically until a maximum threshold, and a minibatch size B over $N_{episodes}$ episodes, where each episode covers a working shift in the hospital ($T = 540$ minutes) and is randomly sampled from a set of 500 possible instances. Training samples are drawn from a replay buffer of size M, and a target network is updated periodically to stabilize Q-learning. Moreover, exploration is governed by Thompson Sampling, implemented via Monte Carlo (MC) Dropout in the Bayesian Q-network. At each decision epoch k, the network performs T stochastic forward passes using different dropout masks, generating T sampled Q-value vectors. One of these samples is then randomly selected to represent the current Q-value estimate, and the action with the highest value among valid options is chosen. This uncertainty-aware sampling strategy enables the agent to balance exploration and exploitation by implicitly capturing epistemic uncertainty in its value estimates, without requiring an explicit exploration parameter such as ε.

The training procedure for the BNN with Thompson Sampling and n-step bootstrapping over hospital instances $\mathcal{I}^{HP}$ is summarized in Algorithm 3.

6.3.4 Learning $\pi_{DQN}^{Bayesian}$

Table 6.10: Tested and selected values for training parameters.

Parameter	**Tested Values**	**Selected Value**
$N_{episodes}$	$\{5, 10, 15, 20\} \cdot 10^3$	$10 \cdot 10^3$
N_{layers}	$\{1, 3, 5\}$	5
N_{nodes}	$\{32, 64\}$	32

Note: Final values are selected based on the best average objective function obtained in the testing phase.

Algorithm 3 BNN Training with Thompson Sampling and n-step Bootstrapping

Input: Bayesian Q-network Q_θ, target network Q_{θ^-}, optimizer O, learning rate η, discount factor γ, initial KL weight $\lambda_{KL,0}$, max KL weight $\lambda_{KL}^{\max}$, number of training episodes $N_{episodes}$, minibatch size B, replay buffer size M, n-step horizon n, hospital instance set $\mathcal{I}^{HP}$.

Output: Trained BNN weights θ.

Initialize replay buffer $\mathcal{M} \leftarrow \emptyset$
Initialize weights $\theta \leftarrow \theta_0$, $\lambda_{KL} \leftarrow \lambda_{KL,0}$
for $e = 1$ to $N_{episodes}$ **do**
  Sample instance $\mathcal{I}_e^{HP} \sim \mathcal{I}^{HP}$ and load environment
  Initialize environment $S_0 \leftarrow env.reset()$, return $R_e \leftarrow 0$
  for each request $x_k \in X(S_k)$ **do**
    Sample T Q-value vectors using Thompson Sampling: $Q_\theta^{(1:T)}(S_k, \cdot)$
    Sample one realization $Q^{sample}(S_k, \cdot) \sim \{Q_\theta^{(t)}\}_{t=1}^{T}$
    Select action $x_k \leftarrow \arg\max_{x \in X(S_k)} Q^{sample}(S_k, x)$
    Execute x_k, observe $R(S_k, x_k)$, S_{k+1}, and Done_{k+1}
    Store $(S_k, x_k, R(S_k, x_k), \text{Done}_{k+1}, S_{k+1})$ in $\mathcal{M}$
    Update state $S_k \leftarrow S_{k+1}$, accumulate $R_e \leftarrow R_e + R(S_k, x_k)$
  end for
  if $|\mathcal{M}| \geq B + n$ **then**
    Sample B transitions: $\{(S_j, x_j, R_j, \text{Done}_j, S'_j)\}_{j=1}^{B}$
    for $j = 1$ to B **do**
      Compute n-step return:
      $R_j^{(n)} = \sum_{i=0}^{n-1} \gamma^i R_{j+i}$
      $y_j \leftarrow R_j^{(n)} + \gamma^n \cdot \max_{x'} Q_{\theta^-}(S_{j+n}, x') \cdot \mathbb{1}_{\{\neg \text{Done}_{j+n}\}}$
      Predict $Q_\theta(S_j, x_j)$ using current BNN and one stochastic realization from MC Dropout
      Compute TD Loss: $\mathcal{L}_{MSE} = \frac{1}{B} \sum_j (Q_\theta(S_j, x_j) - y_j)^2$
      Compute KL Loss: $\mathcal{L}_{KL} = \frac{1}{B} \cdot KL[q(\theta) \,\|\, p(\theta)]$
      Total Loss: $\mathcal{L}(\theta) = \mathcal{L}_{MSE} + \lambda_{KL} \cdot \mathcal{L}_{KL}$
      Compute gradient: $g_t \leftarrow \nabla_\theta \mathcal{L}(\theta_t)$
      Update first moment: $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$
      Update second moment: $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$
      Bias correction: $\hat{m}_t \leftarrow \frac{m_t}{1 - \beta_1^t}$, $\hat{v}_t \leftarrow \frac{v_t}{1 - \beta_2^t}$
      Parameter update: $\theta \leftarrow \theta - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$
    end for
  end if
  Update KL weight: $\lambda_{KL} \leftarrow \min(\lambda_{KL}^{\max}, \frac{e}{N_{episodes}} \cdot \lambda_{KL}^{\max})$
  if $e \bmod 25 = 0$ **then**
    Sync target network: $Q_{\theta^-} \leftarrow Q_\theta$
  end if
end for
return θ

We next examine the evolution of the BNN-based policy's performance throughout the training process, with a particular focus on its behavior across episodes for different BNN architectures, as shown in Table 6.10. Moreover, their performance is evaluated under distinct instances. To eliminate the variation caused by the VNS component in Phase 2, we consider only the Best Insertion (BI) heuristic for reoptimizing the routes after assignment. This ensures that the same action in a given state produces identical results across different replications. Consequently, any differences observed between policies will be solely due to their structural differences, rather than variations in the objective function values introduced by VNS. Furthermore, at this stage, we consider that the service robots are slower compared to porters.

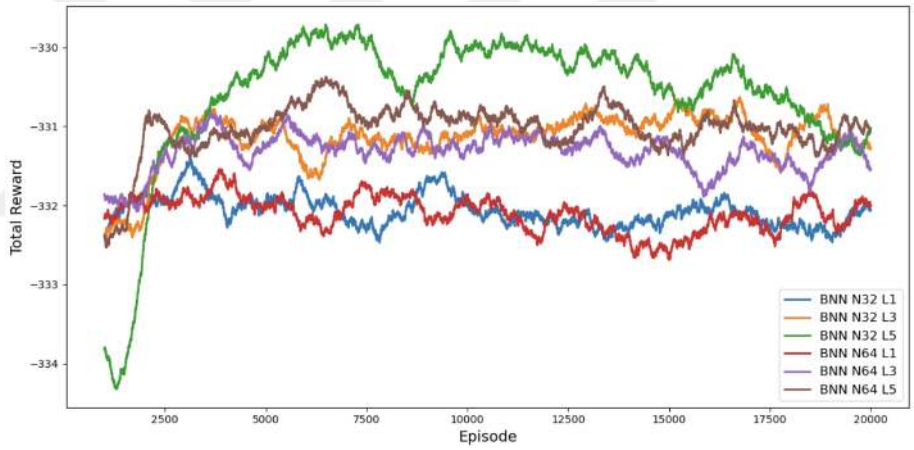


Figure 6.1: (Color online) 1000-episode moving average

Figures 6.1 and 6.2 illustrate the evolution of the average total rewards and objective function value for six Bayesian Neural Network (BNN) architectures evaluated over 20,000 episodes, where each episode is randomly drawn from a set of 500 distinct test instances. Specifically, each data point represents the average of 1000 consecutive episodes. We select a large window size due to the variability in the data stemming from the selection of diverse instances and the inherent uncertainty in Bayesian Neural Networks. In all configurations, deeper architectures surpass shallower ones, with the 32-neuron, 5-layer model (N32 L5) having the highest average total reward

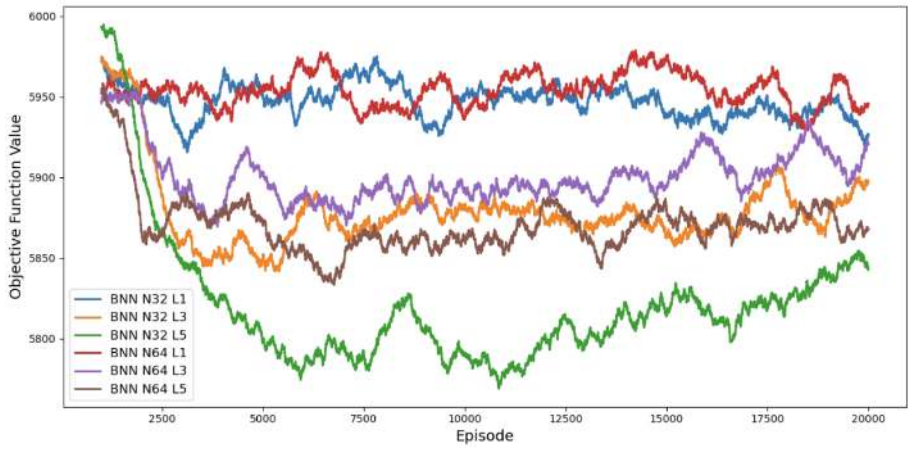


Figure 6.2: (Color online) 1000-episode moving average

(approximately -330.1) and demonstrating rather constant performance. Although broader networks (N64) gain from enhanced depth, their performance trajectories are often unstable, indicating more susceptibility to the variability of test instances. The second figure confirms this, as BNN N32 L5 attains the lowest and most continuously decreasing objective function value throughout training, signifying both successful optimization and alignment between the optimization objective and policy quality. The learning behavior shown in Figure 6.2 supports these findings from a task-specific optimization perspective. Here, BNN N32 L5 again outperforms other models, achieving the lowest and most consistently declining objective function values throughout training. This not only confirms the agent's ability to minimize costs effectively but also reflects strong alignment between the learned value function and the true performance objective.

According to the plots, architectural depth plays a critical role in model performance. In other words, deeper configurations, particularly BNN N32 L5, consistently achieve superior results in both total reward and objective value. This architecture demonstrates stable learning and outperforms its shallower counterparts across the entire training horizon. In contrast, shallow networks such as BNN N32 L1 and BNN N64 L1 plateau early and fail to reach competitive performance, reflecting insufficient capacity to approximate complex value functions. Nonetheless,

deeper or wider models do not always guarantee improved results. For instance, although BNN N64 L5 initially performs well, its total reward begins to deteriorate after episode 10,000. A similar pattern is seen for BNN N32 L5 and BNN N32 L3. This degradation may be attributed to compounding uncertainty in posterior sampling or a gradual shift in the action distribution as the model becomes increasingly confident in potentially suboptimal predictions. Since Thompson Sampling selects actions based on stochastic Q-value estimates generated through Monte Carlo Dropout, overly confident but poorly calibrated uncertainty estimates can mislead policy decisions, especially in later stages of training.

Unlike ε-greedy strategies, Thompson Sampling inherently balances exploration and exploitation by drawing samples from the model's approximate posterior at each decision point. In the early training stages, the posterior distribution remains broad, promoting exploration across diverse actions. As training progresses and the posterior concentrates, the agent becomes more deterministic. This dynamic works effectively in models with well-calibrated uncertainty (e.g., BNN N32 L5), but may cause premature convergence or declining performance in unstable models like BNN N64 L5, where uncertainty is underestimated or erratic.

The overall trends confirm that moderate-width, deep BNNs, specifically N32 L5, are better suited to learning robust and generalizable dispatching policies under uncertainty. These models offer a favorable trade-off between expressiveness and Bayesian regularization, maintaining strong performance across diverse instances. Moreover, the use of randomized sampling from a pool of 500 problem instances ensures that learned policies are not overfitted and retain strong generalization capacity in stochastic environments.

To conduct a more reliable and accurate evaluation of the learned Bayesian policies, we perform an extensive performance analysis across 25 distinct test instances, each replicated 10 times to account for stochasticity in the model's posterior sampling. This results in a total of 250 evaluation runs per architecture. The aggregated results, including the mean and standard deviation of the objective function values, are summarized in Table 6.11. The results in the table confirm and complement the trends observed in the learning curves. Among all configurations, the model with

$N_{nodes} = 32$, $N_{layers} = 5$, and $N_{episodes} = 10{,}000$ training episodes (N32 L5) stands out, achieving the lowest mean objective value (2458.72) and travel time (5811.04), which aligns with its consistently superior learning trajectory shown in Figures 6.1 and 6.2. While BNN N32 L1 achieves the lowest delay (94.88) and BNN N64 L1 achieves the lowest workload (388.15), these gains come at the expense of higher overall objective values, suggesting a trade-off rather than a holistic improvement. Wider networks such as N64 L5 perform reasonably well but display instability, as seen in the degradation of reward after 10,000 episodes.

The bar chart presented in Figure 6.3 provides a statistical perspective on the performance of various BNN architectures by displaying their objective values alongside 95% confidence intervals, aggregated over 250 evaluation runs per configuration. This visual evidence reinforces the findings from Table 6.11 and Figures 6.1 and 6.2. Notably, the configuration BNN N32 L5 trained over 10000 episodes achieves the lowest mean objective value and exhibits one of the narrowest confidence intervals, indicating both superior optimization performance and low variance across diverse test instances. In contrast, shallow models such as BNN N32 L1 and BNN N64 L1 consistently perform worse, with higher objective values and wider confidence intervals, suggesting inadequate capacity and unstable generalization.

These results support the conclusion that architectural depth plays a more decisive role than width in BNN-based policy learning under uncertainty. In particular, moderately wide and sufficiently deep models like N32 L5 offer the best trade-off between expressiveness, stability, and generalization. The training with 10,000 episodes appears sufficient for convergence in this configuration, beyond which deeper networks may become prone to overfitting or poorly calibrated confidence. Thus, N32 L5 emerges as the most reliable architecture for robust and scalable dispatching under stochastic dynamics.

6.3.5 Comparison of Policies Under Different Settings

In this section, we compare the performance of the trained policy $\pi_{DQN}^{Bayesian}$ to the performance of the baseline policies $\pi_{randomized}$, $\pi_{heuristic}$, $\pi_{DQN}^{Deterministic}$, and $\pi_{hospital}$

Table 6.11: Performance summary of BNN architectures.

N_{nodes}	N_{layers}	N_{episodes}	**Mean Obj. Value**	**Mean Travel Time**	**Mean Delays**	**Mean Workloads Diff.**
32	1	5000	2505.65	5972.52	**94.88**	393.47
		10000	2527.73	6016.27	108.85	388.39
		15000	2522.35	5986.14	123.39	391.90
		20000	2525.66	5998.73	118.32	394.20
	3	5000	2494.38	5905.75	129.47	401.48
		10000	2496.49	5929.11	113.24	397.77
		15000	2493.83	5923.03	111.83	399.08
		20000	2488.40	5905.86	115.60	399.09
	5	5000	2467.31	5826.60	137.37	408.60
		10000	**2458.72**	**5811.04**	130.31	410.89
		15000	2475.36	5845.31	138.67	408.85
		20000	2482.80	5881.82	123.78	402.79
64	1	5000	2531.42	6031.54	102.74	388.54
		10000	2518.09	6004.41	95.83	389.92
		15000	2528.44	6024.88	102.15	**388.15**
		20000	2517.89	5897.28	111.36	392.16
	3	5000	2503.28	5952.79	107.11	396.59
		10000	2497.04	5943.07	100.60	397.85
		15000	2502.25	5929.45	126.61	399.16
		20000	2500.24	5956.01	119.29	398.02
	5	5000	2489.44	5897.19	125.53	401.74
		10000	2484.84	5919.54	118.05	398.99
		15000	2488.64	5898.64	121.33	403.24
		20000	2497.02	5917.75	124.66	400.27

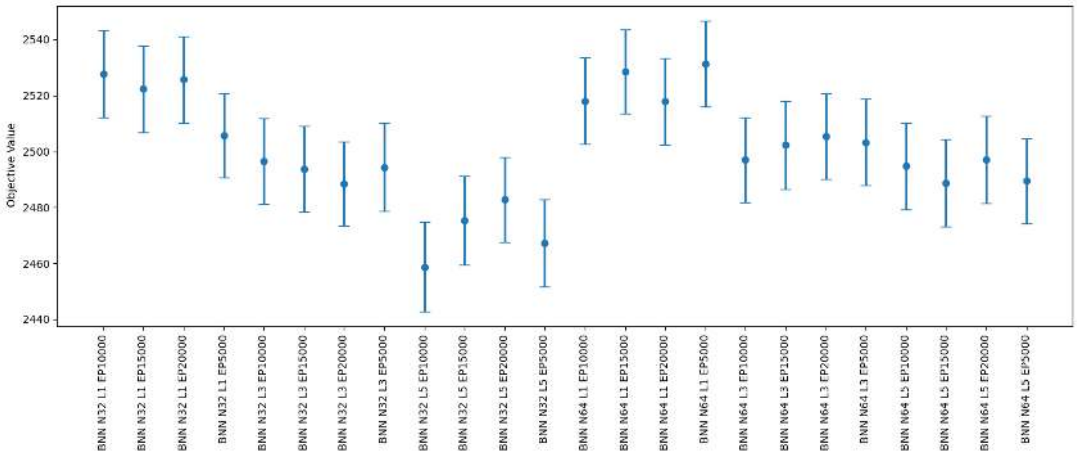


Figure 6.3: 95% confidence intervals for objective means for each BNN architecture

Table 6.12: Detailed performance statistics for 25 test instances across various policies, and transporter fleet.

N_{porters}	N_{robots}	Policy	Objective Value		Travel Time		Delays		Workload Diff.		$P(\pi_a)$
			Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	
30	0	$\pi_{DQN}^{Bayesian}$	2443.06	141.21	5705.12	261.71	283.66	141.78	237.77	55.18	39.33%
		$\pi_{DQN}^{Deterministic}$	2807.20	157.35	5697.67	267.72	1049.34	236.10	541.98	73.97	30.28%
		$\pi_{heuristic}$	2352.43	121.36	5704.68	261.79	79.69	75.81	193.42	44.49	**41.58%**
		$\pi_{hospital}$	2356.57	117.15	5704.04	262.85	88.78	66.73	197.24	43.92	41.47%
		$\pi_{randomized}$	4026.53	570.14	5687.48	262.48	4031.87	1265.45	693.97	113.64	0.00%
	5	$\pi_{DQN}^{Bayesian}$	2515.18	140.29	5741.19	264.00	287.21	150.52	519.11	39.57	**37.02%**
		$\pi_{DQN}^{Deterministic}$	2840.91	213.56	5716.66	271.33	1101.39	339.21	568.44	44.52	28.86%
		$\pi_{heuristic}$	2563.91	131.24	6067.45	278.66	109.48	81.59	465.70	36.07	35.80%
		$\pi_{hospital}$	2568.94	133.63	6060.77	276.21	127.77	89.18	467.63	36.58	35.67%
		$\pi_{randomized}$	3993.68	536.35	5784.62	267.67	3796.70	1192.05	805.78	107.94	0.00%
	10	$\pi_{DQN}^{Bayesian}$	2522.32	136.52	5788.83	262.11	260.84	139.31	512.27	39.27	**36.42%**
		$\pi_{DQN}^{Deterministic}$	2836.82	208.23	5717.95	269.48	1089.09	330.39	570.02	42.87	28.49%
		$\pi_{heuristic}$	2621.89	129.56	6258.49	283.53	74.78	54.50	442.90	35.85	33.91%
		$\pi_{hospital}$	2621.37	127.39	6258.00	285.53	73.37	47.94	444.08	36.68	33.92%
		$\pi_{randomized}$	3967.22	546.14	5865.26	268.17	3654.67	1248.99	796.23	110.18	0.00%
40	0	$\pi_{DQN}^{Bayesian}$	2379.77	118.41	5721.05	262.58	125.97	86.51	204.80	41.18	31.03%
		$\pi_{DQN}^{Deterministic}$	2586.31	145.01	5718.32	267.14	538.72	149.05	417.45	48.39	25.04%
		$\pi_{heuristic}$	2347.58	116.22	5721.30	263.99	53.73	49.07	187.82	38.31	31.96%
		$\pi_{hospital}$	2346.31	114.91	5719.51	264.43	51.98	46.57	188.56	38.69	**32.00%**
		$\pi_{randomized}$	3450.33	430.64	5700.24	264.90	2606.89	924.60	637.39	104.81	0.00%
	5	$\pi_{DQN}^{Bayesian}$	2437.55	123.97	5759.41	265.26	127.33	87.39	414.25	34.82	**29.04%**
		$\pi_{DQN}^{Deterministic}$	2579.43	135.94	5738.31	268.76	491.83	129.57	436.90	36.82	24.90%
		$\pi_{heuristic}$	2506.84	121.76	6009.78	275.76	64.17	49.08	386.32	33.84	27.02%
		$\pi_{hospital}$	2502.68	119.90	6006.63	275.26	57.30	37.22	385.57	33.64	27.14%
		$\pi_{randomized}$	3434.89	392.05	5780.22	267.16	2467.40	822.58	679.19	104.49	0.00%
	10	$\pi_{DQN}^{Bayesian}$	2452.96	124.15	5811.49	266.28	116.10	85.97	409.61	33.74	**27.69%**
		$\pi_{DQN}^{Deterministic}$	2585.33	126.99	5740.58	270.62	502.52	121.69	440.45	39.46	23.79%
		$\pi_{heuristic}$	2586.62	123.22	6224.60	285.08	58.61	46.72	366.69	33.01	23.75%
		$\pi_{hospital}$	2584.80	123.19	6223.93	284.70	54.27	40.45	367.60	34.32	23.80%
		$\pi_{randomized}$	3392.28	379.03	5852.34	264.00	2291.87	802.58	672.98	100.53	0.00%

introduced in Appendix B. As in the previous section, we evaluate 25 distinct test instances, each examined 10 times, and report the aggregated results across all 250 runs. To compare the performance of different policies, we consider the $\pi_{randomized}$ as the main benchmark. Then, we define the percentage improvement of π_a over $\pi_{randomized}$ as:

$$P(\pi_a) = 100 \times \frac{C(S_{\pi_{randomized}}) - C(S_{\pi_a})}{C(S_{\pi_{randomized}})} \tag{6.4}$$

Where $C(\pi_a)$ is the objective function value obtained under policy π_a.

6.3.5.1 Analysis for Different Transporter Configurations

Table 6.12 summarizes detailed performance statistics for the evaluated policies across different configurations of transporter fleet. Moreover, figures in Appendix D displays scatter and box plots illustrating the mean values for each test instance. According to the results, $\pi_{DQN}^{Bayesain}$ consistently provides superiority and robustness as the heterogeneity of the system increases. In fact, we observe that in the purely human-operated environments (e.g., $N_{robots} = 0$), $\pi_{heuristic}$ and $\pi_{hospital}$ consistently achieve the best performance, with average objective values approximately 41–42% lower than $\pi_{randomized}$ for 30 transporters and 31-32% for 40 transporters. This is visually confirmed by the box plots in figures in Appendix D, where these policies show the lowest median objective values. However, although they significantly outperform $\pi_{DQN}^{Deterministic}$, their performance is only marginally better than that of $\pi_{DQN}^{Bayesian}$. For example, in the scenario with 40 porters and no service robots, $\pi_{heuristic}$ achieves a 0.93% greater reduction compared to the base policy ($\pi_{randomized}$), while $\pi_{hospital}$ achieves a 0.97% reduction.

We note that the performance of these policies deteriorates as heterogeneity in the fleet configuration increases. Once service robots are introduced and the system's heterogeneity increases, creating an environment in which the outcomes of each action (assignment) vary significantly due to the differences in the speeds of service robots and porters, the heuristic-driven decision policies underperform. In these scenarios, the BNN-based policy $\pi_{DQN}^{Bayesian}$ not only exceeds the deterministic DQN and other baseline policies in average objective value reduction but also consistently attains small standard deviations (approximately 5% of the average objective function), signifying improved stability across the 250 replications. This trend is clearly observable box plots, where the $\pi_{DQN}^{Bayesian}$ policy's box plots remain narrow and centered on lower medians (2450–2550) compared to the wider and higher-distribution ranges of the deterministic DQN, heuristic, and hospital policies. Moreover, by examining the mean for each test instance in the scatter plot, we observe that as heterogeneity increases, particularly in cases with $N_{robots} = 10$, $\pi_{DQN}^{Bayesian}$ achieves a lower mean objective values. This indicates that $\pi_{DQN}^{Bayesian}$ not only outperforms the other

policies overall but also consistently attains better performance and the minimum objective function value for all test instances. For instance, for a fleet configuration of $N_{porters} = 40$ and $N_{robots} = 10$, representing the most heterogeneous combination, $\pi_{DQN}^{Bayesian}$ attains an average objective value reduction of 27.69%, outperforming the deterministic DQN policy by nearly 4% and the heuristic and hospital-native policies by almost 3.9–4%. Notably, the box plots in this configuration show $\pi_{DQN}^{Bayesian}$ with a narrower interquartile range and fewer outliers, reinforcing its ability to consistently adapt to the increased complexity introduced by service robots. Furthermore, the scatter points demonstrate that $\pi_{DQN}^{Bayesian}$ outperforms other policies across the 25 test instances individually, indicating that its robustness is not simply a byproduct of averaging across replications but is consistently observable across individual problem instances. Moreover, in this environment, characterized by increased system heterogeneity, $\pi_{DQN}^{Deterministic}$ manages to sustain performance levels that are comparable to the other baseline policies, such as $\pi_{hospital}$ and $\pi_{heuristic}$. This suggests that while $\pi_{DQN}^{Deterministic}$ generally does not outperform the Bayesian variant, it remains robust enough to match the performance of heuristic and hospital-based policies under these complex operational conditions.

This superior performance of $\pi_{DQN}^{Deterministic}$ in heterogeneous environments stems from its ability to distinguish between assignments that lead to worse outcomes. In other words, since in this examination the service robots are slower, assigning too many requests to them can significantly increase the objective function value. Policies such as $\pi_{hospital}$ and $\pi_{heuristic}$, which only consider the current state of transporters, fail to capture this insight. By contrast, $\pi_{DQN}^{Deterministic}$ can learn these differences and make more informed decisions. On the other hand, in homogeneous environments where no difference in the transporters' speeds, simply assigning based on the current state can already yield good outcomes. In these cases, $\pi_{hospital}$ and $\pi_{heuristic}$ slightly outperform $\pi_{DQN}^{Deterministic}$. This also indicate the efficiency of our state definition that can capture the information required to make an informed decision. Nonetheless, although $\pi_{DQN}^{Deterministic}$ can still learn the best actions using our state definition, occasional suboptimal assignments can occur due to uncertainty inherent in its action selection process.

Additionally, the results in table and figure reveal that as the number of transporters decreases, the relative importance of actions, and thus policy decisions, becomes more pronounced. Specifically, when 50 transporters are available, the observed improvements over the randomized policy range from 23% to 27%, with the $\pi_{DQN}^{Bayesian}$ policy achieving the highest improvement of 27.69%. In contrast, when only 30 transporters are present, the best policies, such as the heuristic and hospital, achieve objective value reductions of up to 41.58%. These observations underscore that as transporter resources become more constrained, the impact of each action taken by the policy grows significantly, leading to sharper performance gains and making the choice of policy even more critical to achieving optimal outcomes.

Table 6.13: Detailed performance statistics for 25 test instances across various policies, and speed variation factor.

Speed Variation Factor	**Policy**	**Objective Value**		**Travel Time**		**Delays**		**Workload Diff.**		$P(\pi_a)$
		Mean	**Std.**	**Mean**	**Std.**	**Mean**	**Std.**	**Mean**	**Std.**	
1	$\pi_{DQN}^{Bayesian}$	2422.88	118.11	5762.31	262.91	111.51	79.57	366.76	35.11	28.10%
	$\pi_{DQN}^{Deterministic}$	2464.69	127.33	5727.92	267.34	226.66	106.18	414.31	32.98	26.86%
	$\pi_{heuristic}$	2394.55	111.38	5770.87	262.19	33.57	32.26	363.85	33.63	28.94%
	$\pi_{hospital}$	2393.48	111.91	5769.87	263.66	32.08	31.52	363.52	32.76	**28.97%**
	$\pi_{randomized}$	3369.87	379.81	5728.05	264.16	2357.21	813.30	678.84	103.98	0.00%
1.25	$\pi_{DQN}^{Bayesian}$	2406.45	117.37	5757.04	262.54	54.20	49.17	409.79	35.16	**29.31%**
	$\pi_{DQN}^{Deterministic}$	2463.65	126.24	5733.12	267.57	219.65	98.52	412.72	30.32	27.63%
	$\pi_{heuristic}$	2493.48	119.63	6007.43	275.54	43.56	42.16	365.41	33.65	26.76%
	$\pi_{hospital}$	2492.24	116.41	6007.98	274.54	39.91	34.05	365.42	33.40	26.79%
	$\pi_{randomized}$	3404.40	386.33	5788.71	264.35	2385.42	825.15	673.73	98.68	0.00%
1.5	$\pi_{DQN}^{Bayesian}$	2452.96	124.15	5811.49	266.28	116.10	85.97	409.61	33.74	**27.69%**
	$\pi_{DQN}^{Deterministic}$	2585.33	126.99	5740.58	270.62	502.52	121.69	440.45	39.46	23.79%
	$\pi_{heuristic}$	2586.62	123.22	6224.60	285.08	58.61	46.72	366.69	33.01	23.75%
	$\pi_{hospital}$	2584.80	123.19	6223.93	284.70	54.27	40.45	367.60	34.32	23.80%
	$\pi_{randomized}$	3392.28	379.03	5852.34	264.00	2291.87	802.58	672.98	100.53	0.00%
2	$\pi_{DQN}^{Bayesian}$	2479.18	124.69	5874.68	266.26	118.75	82.87	409.07	37.82	**28.78%**
	$\pi_{DQN}^{Deterministic}$	2591.59	131.06	5754.26	275.16	503.74	136.80	441.94	40.05	25.55%
	$\pi_{heuristic}$	2737.41	138.67	6570.00	302.54	87.78	62.58	371.47	35.73	21.36%
	$\pi_{hospital}$	2729.75	133.43	6566.13	301.15	71.73	52.82	373.01	34.30	21.58%
	$\pi_{randomized}$	3480.99	375.59	5972.75	273.03	2392.31	793.60	674.81	98.61	0.00%

6.3.5.2 Analysis on Speed Variations of the Service Robots

To further explore the impact of heterogeneity on the performance of different policies, we investigate their outcomes under another dimension of heterogeneity: the speed variations of the service robots compared to porters. We define speed variation as the percentage increase in travel times between two specific locations. For instance, if the service robots operate at 50% slower speed, their associated travel times are multiplied by a factor of 1.5. We call this factor the speed variation factor. We analyze scenarios with speed variation factors of 1, 1.25, 1.5, and 2. Table 6.13 and figures in Appendix E present the detailed performance statistics and visual summaries for the evaluated policies across these scenarios. Consistent with prior findings, $\pi_{DQN}^{Bayesian}$ sustains robust performance across all heterogeneity scenarios. As the speed variation factor increases, the performance of $\pi_{heuristic}$ and $\pi_{hospital}$ deteriorates significantly, with their objective value improvements declining from approximately 28% at a factor of 1 to around 21% at a factor of 2. Meanwhile, $\pi_{DQN}^{Bayesian}$ consistently maintains high performance, with objective value reductions stabilizing near 28% across all scenarios and narrow standard deviations (4.8-5% of the total mean), indicating robust and reliable outcomes across individual problem instances. Moreover, the performance gap between $\pi_{DQN}^{Bayesian}$ and $\pi_{heuristic}$ and $\pi_{hospital}$ policies widens as the speed variation factor increases. Specifically, while the differences in performance improvements over $\pi_{heuristic}$ and $\pi_{hospital}$ are -0.84% and -0.87% respectively in the homogeneous scenario, these values rise to 7.42% and 7.2% in the most heterogeneous scenario, consistently growing with higher speed variation factors. These results support the prior conclusion that $\pi_{DQN}^{Bayesian}$ is especially adept for environments marked by increased system heterogeneity, as it proficiently manages the dynamic interactions of variations in speed within the transporter fleet. The adaptability is evidenced in the box plots of figures in Appendix E, where the $\pi_{DQN}^{Bayesian}$ policy exhibits lower median values and narrower interquartile ranges in contrast to the broader dispersion of the other policies. Furthermore, the scatter plots illustrate that $\pi_{DQN}^{Bayesian}$ consistently attains superior performance across all 25 test instances, emphasizing its capacity to generalize and provide robust solutions

despite increasing system complexity.

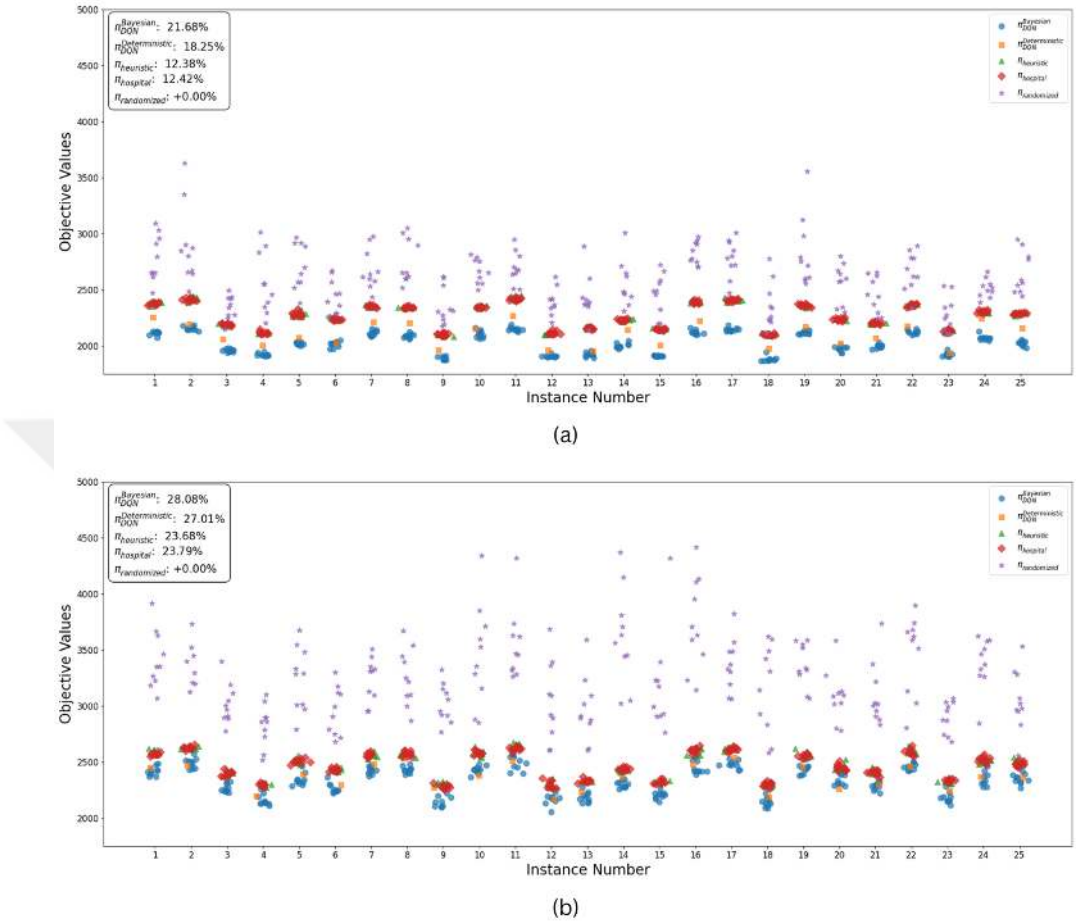


Figure 6.4: (a) Objective values across instances when porters are faster than service robots; (b) Objective values across instances with mixed speeds of porters and service robots.

Besides inter-type speed variation, we investigate a more nuanced form of heterogeneity wherein each transporter, either a porter or a service robot, functions at a distinct individual speed. This analysis examines two separate scenarios. First, porters are typically more expedient than service robots. Each porter is assigned a speed variation factor randomly selected from the interval [0.5, 1], whilst each service robot is allocated a factor from the interval [1, 2], indicating their comparatively slower performance. Second, we incorporate overlap between the speed ranges of porters and service robots to represent more realistic operational settings. Fif-

teen out of forty porters are randomly chosen and assigned a speed variation factor within the range of [1, 1.5], enabling certain porters to function at speeds similar to or slower than the service robots.

Figure 6.4 depicts the performance of various policies throughout 25 test instances, each assessed by 10 replications. Subfigure (a) illustrates the initial scenario, wherein porters typically exhibit greater speed than service robots, while subfigure (b) depicts the subsequent scenario, in which certain porters function at speeds that are either equivalent to or slower than those of robots. Since $\pi_{DQN}^{Deterministic}$ is a deterministic policy, it yields a unique result for each event, hence producing one data point per instance.

In every case, $\pi_{DQN}^{Bayesian}$ consistently achieves the lowest goal values, illustrating its strong flexibility to transporter-level diversity. In the original scenario, it exceeds $\pi_{heuristic}$ and $\pi_{hospital}$ by 9.3% and 9.26%, respectively. In the second case, the performance disparity diminishes marginally, exhibiting enhancements of 4.4% and 4.29% regarding the same policies. Nonetheless, $\pi_{DQN}^{Bayesian}$ retains its preeminence in both contexts. Furthermore, $\pi_{DQN}^{Deterministic}$, which had previously equaled or underperformed compared to rule-based policies in earlier tests, now exceeds both $\pi_{heuristic}$ and $\pi_{hospital}$ by 5.87% and 5.93% in the first and by 3.33% and 3.22% in the second scenario, respectively, underscoring its enhanced efficacy in more irregular heterogeneous conditions.

6.3.5.3 Overall Policy Comparison and Insights

The results across all evaluated conditions indicate that the BNN-based policy $\pi_{DQN}^{Bayesian}$ constantly surpasses competing methods as system heterogeneity escalates. In homogeneous settings, consisting solely of porters with identical speeds, rule-based policies such as $\pi_{heuristic}$ and $\pi_{hospital}$ exhibit comparable or even superior performance relative to learning-based methods. This is due to the uniform impact of activities in such environments, where the prevailing conditions and proximity are the primary determinants of results. These findings affirm the quality of our state definition, which encapsulates pertinent information to facilitate effective

decision-making, even through simplistic, heuristic-driven reasoning. Nonetheless, as heterogeneity is introduced, whether by incorporating service robots with reduced speeds, varying speed configurations, or assigning unique speeds to each transporter, the effectiveness of static policies diminishes, while learning-based methods demonstrate a clear advantage. The policy $\pi_{DQN}^{Bayesian}$ exhibits efficiency in these contexts by precisely modeling the probabilistic outcomes of various assignments and differentiating transporter capabilities, resulting in markedly improved objective values and diminished performance variability across replications. In the most intricate configuration, where each transporter operates at a distinct speed, both $\pi_{DQN}^{Bayesian}$ and $\pi_{DQN}^{Deterministic}$ significantly exceed rule-based baselines, with the Bayesian policy attaining the lowest objective values in the majority of cases. The findings indicate that in heterogeneous environments with high variability in action outcomes, learning-based strategies are not merely more successful but indispensable. Furthermore, $\pi_{DQN}^{Bayesian}$ provides the most consistent and dependable performance across various operational difficulties, underscoring its significance for real-world hospital logistics systems characterized by heterogeneous transporter attributes and system conditions.

6.4 Numerical Results: Assessing the Two-Phase Solution Approach

In this section, we evaluate our proposed two-phase approach against a conventional solution methodology frequently adopted in the literature. Specifically, we implement a reoptimization-based policy that leverages a metaheuristic, namely, Adaptive Tabu Search, not only to insert newly arrived requests but also to revise the entire assignment plan. This reoptimization capability enables the reassignment of previously assigned requests, potentially reducing the overall objective function value. The full implementation details of the Adaptive Tabu Search are presented in Benchmark Appendix C.

Inspired by the framework introduced by [Ton et al., 2024], which proposes four distinct reoptimization policies, we adopt a hybrid of two of these policies. In our model, reoptimization is triggered either when the number of requests in the queue

reaches a threshold N_{queue}, or when a time limit t_{reopt} has elapsed since the last optimization. This dual-condition trigger ensures timely updates to the solution without incurring excessive computational overhead. In particular, it avoids unnecessary reoptimization after every request arrival, which could otherwise lead to substantial increases in runtime, while also preventing long delays when the system is idle. We call this policy π_{reopt}. Additionally, to facilitate a more accurate assessment of system performance, we include the delay related to the elapsed time, represented by t_{elapse}. This parameter records the duration prior to the activation of any reoptimization conditions. Incorporating this delay allows us to consider the system's response time within practical operating limitations. For thoroughness, we also present the results obtained without accounting for the elapsed time delay, allowing us to measure the degree to which this factor affects overall performance. The results of this case also acts as baselines to compare.

Due to the difficult constraints and the real-time nature of our problem, applying the one-phase approach, which considers all transporters and all not initiated but assigned requests at each reoptimization point, requires a considerable amount of computational time. To make the problem more manageable for analysis, we design a medium-sized instance. In the first scenario, we use test instances from $\mathcal{I}^{HP}$, but limit the planning horizon to the first 50 requests, which approximately correspond to the initial two hours of a shift. Because the arrival times of requests are very close to each other, this creates a challenging case that demands more computational effort for reoptimization. In the second scenario, we use instances from $\mathcal{I}^{PP}_{\lambda=1/10}$, but this time consider the full planning horizon. In both scenarios, the transporter fleet consists of six porters and two service robots, with porters being faster than the service robots.

Tables 6.14 and 6.15 report the objective values, percentage difference between the objective values compare to the case without elapsed time delay, and total runtimes for each test instance under four different methods. The first two columns under the "One-phase approach" heading correspond to the approach $\pi_{reopt} + ATS$, evaluated in two modes: with elapsed time delay (t_{elapse}), and without ($\varnothing$). The next two columns under the "Two-phase approach" heading present results for our

Table 6.14: Comparison of one-phase and two-phase approaches in terms of objective and percentage difference with respect to $\pi_{reopt} + ATS\ (\varnothing)$.

Group	Instance	One-phase approach				Two-phase approach			
		$\pi_{reopt} + ATS\ (t_{elapse})$		$\pi_{reopt} + ATS\ (\varnothing)$		$\pi_{DQN}^{Bayesian} + BI$		$\pi_{DQN}^{Bayesian} + BI + AVNS$	
		Objective	**% Diff.**	**Objective**	**% Diff.**	**Objective**	**% Diff.**	**Objective**	**% Diff.**
$\mathcal{I}^{HP}$	1	590.93	18.26%	499.67	0.00%	529.72	6.01%	489.30	-2.08%
	2	618.44	14.50%	540.11	0.00%	609.62	12.87%	649.10	16.10%
	3	556.66	-5.83%	591.12	0.00%	678.57	14.79%	617.18	4.41%
	4	480.93	9.93%	437.49	0.00%	507.57	16.02%	481.49	10.06%
	5	508.58	37.65%	369.48	0.00%	457.41	23.80%	432.54	17.07%
	6	564.70	33.89%	421.79	0.00%	557.25	32.11%	509.74	20.86%
	7	811.92	-4.09%	846.57	0.00%	795.47	-1.26%	705.68	-16.63%
	8	911.48	2.86%	886.15	0.00%	1066.10	20.31%	1046.78	18.16%
	9	422.94	6.97%	395.43	0.00%	516.38	30.63%	457.60	15.72%
	10	466.99	-0.65%	470.03	0.00%	541.90	15.30%	529.72	12.71%
$\mathcal{I}^{PP}_{\lambda=1/10}$	1	615.91	39.22%	442.47	0.00%	477.46	7.91%	474.43	7.22%
	2	1394.25	300.31%	348.18	0.00%	359.69	3.31%	353.52	1.53%
	3	381.97	7.77%	354.45	0.00%	352.19	-0.64%	361.82	2.08%
	4	1285.89	348.18%	286.86	0.00%	294.55	2.68%	303.61	5.83%
	5	510.33	13.00%	451.63	0.00%	473.54	4.86%	471.59	4.42%
	6	1691.30	239.52%	498.06	0.00%	509.61	2.33%	526.00	5.61%
	7	724.69	86.74%	388.07	0.00%	414.76	6.89%	426.62	9.95%
	8	748.39	70.36%	439.38	0.00%	432.67	-1.53%	433.11	-1.43%
	9	2383.49	378.43%	498.31	0.00%	530.34	6.42%	518.87	4.13%
	10	1336.70	202.97%	441.12	0.00%	485.71	10.10%	474.33	7.53%
$\mathcal{I}^{HP}$ Avg.		593.66	11.83%	530.89	0.00%	576.30	17.42%	529.14	-0.33%
$\mathcal{I}^{PP}_{\lambda=1/10}$ Avg.		1037.89	168.34%	413.76	0.00%	433.15	4.68%	430.15	3.95%
Total Avg.		815.78	90.47%	472.33	0.00%	504.72	10.64%	479.65	1.55%

proposed method in two configurations. The first configuration, $\pi_{DQN}^{Bayesian} + BI$, uses a Bayesian Deep Q-Network with a greedy Best Insertion heuristic to construct the plan. The final configuration, $\pi_{DQN}^{Bayesian} + BI + AVNS$, augments the initial plan with an AVNS to further improve solution quality.

In $\mathcal{I}^{HP}$ test instances, the proposed two-phase approach demonstrates substantial advantages in terms of computational efficiency while often delivering comparable or improved solution quality. Across all ten instances, both configurations, $\pi_{DQN}^{Bayesian} + BI$ and its AVNS-enhanced version, consistently outperform the one-phase policy in runtime. For example, in Instance 1, the baseline method ($\pi_{reopt} + ATS\ (t_{elapse})$) requires 1438.09 seconds to achieve an objective of 590.93, whereas BI delivers a

Table 6.15: Comparison of one-phase and two-phase approaches in terms of runtime for each instance.

Group	Instance	One-phase approach		Two-phase approach	
		$\pi_{reopt} + ATS\ (t_{elapse})$	$\pi_{reopt} + ATS\ (\varnothing)$	$\pi_{DQN}^{Bayesian} + BI$	$\pi_{DQN}^{Bayesian} + BI + AVNS$
$\mathcal{I}^{HP}$	1	1438.09	874.45	1.65	222.80
	2	957.08	517.71	1.78	332.71
	3	595.40	1085.35	1.59	235.00
	4	933.26	554.33	1.37	399.89
	5	641.18	523.85	1.26	213.60
	6	1009.13	1156.58	1.50	253.62
	7	868.97	999.15	1.91	351.80
	8	1523.49	1293.82	4.72	395.79
	9	670.84	426.90	1.25	317.46
	10	903.42	634.96	1.63	275.92
$\mathcal{I}^{PP}_{\lambda=1/10}$	1	311.75	290.03	1.38	282.67
	2	454.07	628.54	1.28	179.78
	3	386.57	249.39	1.15	167.25
	4	208.80	221.66	0.89	103.68
	5	517.33	459.37	2.01	678.68
	6	920.79	770.32	2.81	1131.18
	7	337.84	351.26	1.33	276.54
	8	870.06	466.23	1.86	496.75
	9	803.83	691.54	2.09	756.23
	10	455.64	427.02	1.80	455.39
$\mathcal{I}^{HP}$ Avg.		954.97	806.71	1.67	299.56
$\mathcal{I}^{PP}_{\lambda=1/10}$ Avg.		526.67	457.64	1.70	454.82
Total Avg.		740.82	632.18	1.69	377.19

better solution of 529.72 in just 1.65 seconds, and BI+AVNS further improves the objective to 489.30 within 222.80 seconds. This trend continues in Instances 5 and 6, where BI+AVNS offers objective values lower than the one-phase policy (432.54 and 509.74 vs. 508.58 and 564.70, respectively), while reducing runtime by over 60%. In other cases, such as Instances 3 and 8, BI shows weaker solution quality (678.57 and 1066.10) compared to the one-phase values (556.66 and 911.48), but this is mitigated by the significantly lower runtimes (1.59 and 4.72 seconds vs. 595.40 and 1523.49 seconds). Importantly, the integration of AVNS helps recover or even improve quality in such cases: for instance, in Instance 6, the addition of AVNS

reduces the objective from 557.25 (BI) to 509.74 (BI+AVNS). Even if we do not consider the elapsed time in delay calculation and instead compare to the baseline run without elapsed time delay, the improvements of the two-phase method persist, confirming that the gains are not a result of adding the elapsed time delay to the objective, but of a more efficient and scalable planning structure.

On the other hand, in the $\mathcal{I}^{PP}_{\lambda=1/10}$ test instances, where requests arrive less frequently and the full planning horizon is considered, both BI and BI+AVNS provide clear advantages in solution quality and computational speed. For example, in Instance 2, the baseline solution achieves an objective of 1394.25 in 454.07 seconds, whereas BI and BI+AVNS reduce the objective to 359.69 and 353.52, respectively, while executing in under 2 minutes. Similar trends are observed in Instances 1 and 4, where the two-phase variants consistently yield better objective values (e.g., 474.43 and 303.61) compared to 615.91 and 1285.89 from the one-phase policy. Even in more challenging instances such as Instance 6, BI+AVNS maintains strong performance (526.00 in 1131.18 seconds), outperforming the one-phase solution both in objective (1691.30) and runtime (920.79). The only notable exception occurs in Instance 5, where AVNS slightly increases runtime (678.68 seconds) while providing a comparable objective (471.59 vs. 510.33), which is still a favorable trade-off given the overall system load. This substantial difference observed in certain instances, such as Instances 4, 6, 9, and 10, can be attributed to the one-phase approach waiting until a reoptimization condition is triggered. As a result, delays accumulate significantly, leading to a considerable increase in the objective function. However, in these instances, when the elapsed delay is not considered, the two-phase approach performs comparably to the one-phase approach, demonstrating its robustness. Nevertheless, it achieves these solutions in significantly less time, in some cases requiring only one-third of the runtime of the one-phase approach.

The results demonstrate that in more relaxed operational contexts, where the reoptimization window is wider, the two-phase approach is not only faster but also capable of producing superior solutions. Furthermore, the consistency of this performance across varying levels of request intensity and time horizon reinforces the robustness and scalability of the approach. Again, when compared to the no-delay

version of the one-phase approach, the results remain competitive, confirming that the improvements are not simply due to the inclusion of elapsed time in the baseline but reflect the structural advantage of the two-phase decomposition.

When averaged across both instance groups, the results clearly favor the two-phase architecture over the one-phase reoptimization baseline. In $\mathcal{I}^{HP}$, BI+AVNS achieves an average runtime of 299.56 seconds compared to 954.97 under $\pi_{reopt} + ATS$ (t_{elapse}), while preserving a comparable objective value, 529.14 with -0.33% difference and 593.66 with 11.83% difference, respectively, meaning our two-phase approach 12.16% performs better. In $\mathcal{I}^{PP}_{\lambda=1/10}$, the advantage becomes even more pronounced: BI+AVNS improves the average objective from 1037.89 (168.34%) to 430.15 (3.95%) and reduces runtime from 526.67 to 454.82 seconds. The BI variant also delivers strong performance. In $\mathcal{I}^{HP}$, it reduces runtime dramatically from 954.97 to just 1.67 seconds, with an average objective of 576.30 (17.43%). In $\mathcal{I}^{PP}_{\lambda=1/10}$, BI achieves an objective of 433.15 (4.68%) compared to 413.76 under the baseline without considering t_{elapse}, while reducing runtime from 457.64 to only 1.70 seconds. Similarly, BI+AVNS achieves objective values of 430.15 (3.95%) across the second groups, offering runtime reductions of over 65%.

The superior performance of the two-phase approach compared to the one-phase approach can be attributed to the BNN-based policy's ability to make decisions that account for long-term outcomes. This policy effectively captures the current system's state and uses that information to guide decisions that remain beneficial over time. In contrast, the one-phase reoptimization method focuses solely on minimizing the current objective function, without considering future implications. As a result, it often makes decisions that may be locally optimal but lead to suboptimal outcomes in the long run. Furthermore, optimizing the entire plan in the one-phase setting becomes increasingly difficult as the number of assigned requests grows. The effectiveness of Adaptive Tabu Search diminishes in such cases, even when strong local search operators are employed, since the potential improvement in the objective function declines. On the other hand, the two-phase approach only reoptimizes a single transporter's route, which is a much simpler task. As a result, the performance of the local search remains stable and effective, even as more requests are

added to the schedule.

Furthermore, within the one-phase approach, achieving a balance between runtime efficiency and solution quality necessitates stricter reoptimization conditions. For example, in the instance set $\mathcal{I}^{PP}_{\lambda=1/10}$, triggering reoptimization for every single request in the queue leads to excessive computation times, preliminary results show that runtime can increase by a factor of 4 to 5 in some cases. To mitigate this, it is necessary to increase the t_{reopt} threshold. However, this adjustment comes at the cost of higher delay penalties, as delaying reoptimization defers service. This trade-off is clearly observed when comparing the objective values of the cases with and without the elapsed time penalty.

Taken together, these results support that the two-phase approach offers a strong alternative to traditional reoptimization-based methods, particularly in settings where decision speed and solution quality must be jointly optimized. The BI variant serves well when computational resources or time are constrained, while the BI+AVNS variant is recommended when some post-decision improvement time is acceptable. In both cases, the methods maintain their advantage regardless of whether elapsed time is considered, underscoring their robustness in real-time transport planning applications.

6.5 Numerical Results: Impact of Utilizing Service Robots in Intra-Hospital Transportation

Table 6.16: Percentage distribution of request types for each test instance.

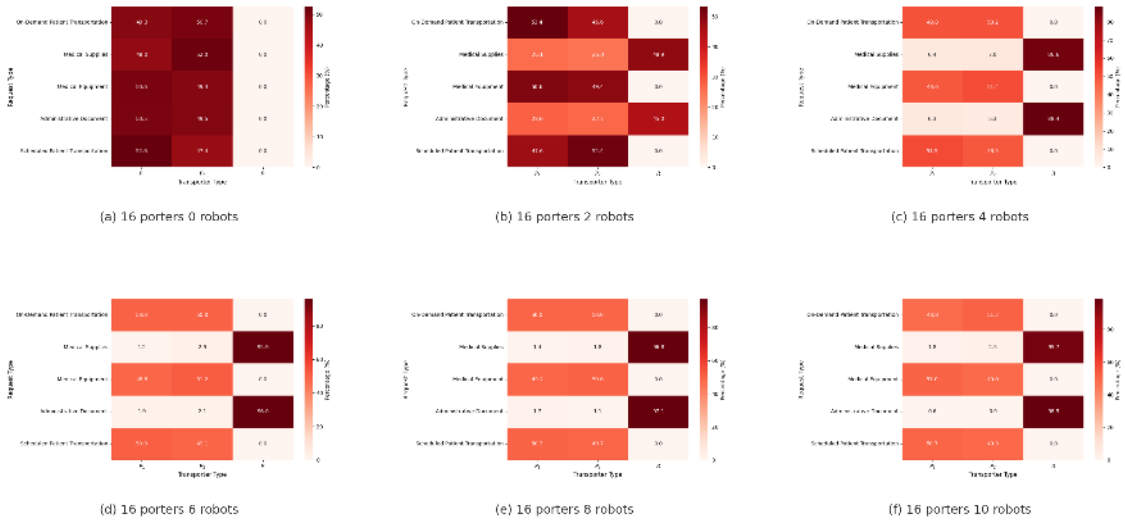
Instance Number	On-demand Patient Transportation (%)	Medical Supplies (%)	Medical Equipment (%)	Administrative Document (%)	Scheduled Patient Transportation (%)
1	15.96	23.40	17.38	19.86	23.40
2	21.55	18.73	18.73	19.08	21.91
3	20.73	22.91	18.55	20.73	17.09
4	21.25	21.25	15.83	15.83	25.83
5	25.18	19.50	18.79	19.86	16.67
6	24.12	18.40	17.18	19.38	20.92
7	18.41	21.34	15.06	24.69	20.49
8	18.47	23.57	18.47	16.24	23.24
9	18.06	19.10	19.44	20.49	22.92
10	20.00	22.31	16.92	19.23	21.54

Table 6.17: Percentage of requests that can and cannot be handled by service robots in each test instance.

Instance	**Handled by Both (%)**	**Handled by Porters Only (%)**	**Total Requests**
1	43.26	56.74	282
2	37.81	62.19	283
3	43.64	56.36	275
4	37.08	62.92	240
5	39.36	60.64	282
6	38.37	61.63	257
7	39.08	60.92	239
8	42.98	57.02	314
9	35.42	64.58	288
10	42.69	57.31	260

This section examines the impact of service robots, working alongside with porters, on overall system performance. Through the comparison of scenarios featuring robotic agents against those without of them, we seek to understand the overarching implications of their incorporation on request assignment, resource utilization, and system responsiveness. To investigate this, we perform experiments on 10 test instances, each containing 16 human porters. In each instance, we alter the number of service robots from 0 to 10 to assess the impact of varying levels of automation. In contrast to Section 6.3.5, where we consider 40 porters to increase the number of actions and create a more complex decision environment, this section focuses on a setting with 16 porters. This allows for a clearer observation of the role and impact of service porters on the intra-hospital transportation system. Tables 6.16 and 6.17 indicate the distribution and combination of request types employed in each test instance as well as the distribution of request that can be handled by service robots. Table 6.16 indicates that patient transportation tasks, both on-demand and scheduled, represent the main set of request types, together comprising over 40% of the workload in every instance. Table 6.17 indicates that only around 40% of requests eligible for assignment to service robots. On the other hand, human porters can accommodate all of request types without such constraints. We examine

whether the integration of service robots, notwithstanding their capacity to manage less than fifty percent of total requests, can yield significant enhancements in system performance by decreasing travel times, minimizing delay penalties, and achieving more equitable workloads.



(a) 16 porters 0 robots (b) 16 porters 2 robots (c) 16 porters 4 robots

(d) 16 porters 6 robots (e) 16 porters 8 robots (f) 16 porters 10 robots

Figure 6.5: Comparison of transport type usage across different service robot integration scenarios.

6.5.1 System-Wide Performance Implications of Service Robot Integration

Figure 6.5 presents the distribution of request assignments among transporter groups, group 1 porters (V_1), group 2 porters (V_2), and service robots (R), across varying levels of robot integration. The heatmaps demonstrate a clear and interpretable policy behavior: request types that are eligible for robot handling, specifically Medical Supplies and Administrative Documents, are gradually reassigned from porters to service robots as the number of available robots increases. When no robots are present (R0), all tasks are distributed between the two porter groups, with roughly even splitting across request types. However, starting with R2 (i.e., two service robots), a notable shift occurs, approximately 46.8% and 45.0% of Medical Supplies and Administrative Document requests, respectively, are reassigned to robots. This suggests that the dispatch policy begins to exploit robot capacity, but still main-

tains a conservative allocation in order to avoid performance degradation across other metrics. As the robot fleet increases to R4 and R6, the percentage of these request types assigned to robots jumps significantly, reaching over 95% by R6. This indicates that the system reaches a point of sufficient robotic coverage, allowing the policy to fully offload standardized and low-touch deliveries to service robots. Beyond R6, the assignment ratios plateau, confirming that the marginal benefit of additional robots diminishes once the demand for robot, eligible tasks is saturated. The results validate that the incorporation of service robots is advantageous, as it significantly decreases the workload of human porters by increasing the percentage of eligible requests assigned to robots, thereby facilitating a more equitable and efficient distribution of requests among porters.

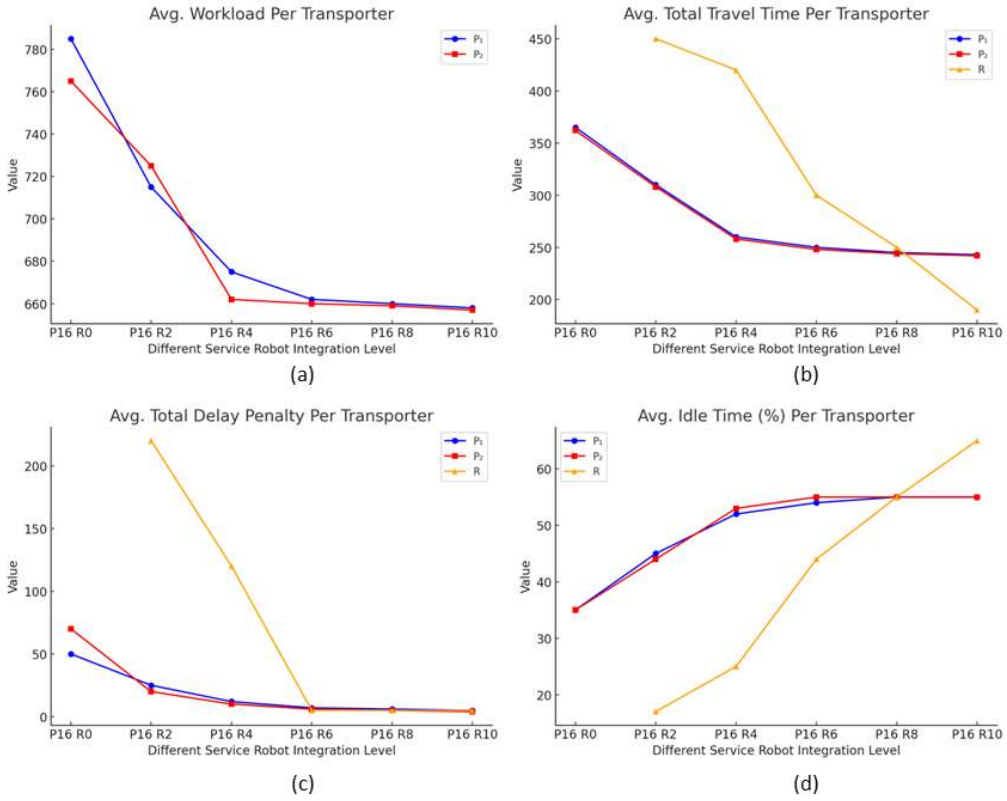


Figure 6.6: Average performance metrics of each transporter type (V_1, V_2, and R) across varying service robot integration levels.

This claim is supported by the results in Figure 6.6. As shown in the average workload plot, introducing service robots into the current system leads to a substantial decrease in the average workload, with a reduction of approximately 15% for both porter groups observed when six robots are deployed. Beyond this point, the reduction becomes negligible. This behavior is expected, as most requests eligible for robot handling are assigned accordingly by that point. This reduction is particularly beneficial in scenarios of sudden demand surges, where freed-up porters can be redirected toward more critical and human-dependent tasks, such as patient transportation.

A similar trend is observed in the average travel time plot. For both porter groups, average travel times decrease by roughly 31.3% when the number of robots increases to six. Interestingly, the average travel time for service robots starts off higher than that of porters when only two robots are in the system, but then decreases as more robots are added. This pattern reflects the system's tendency to assign tasks to robots without workload constraints up to an optimal point—consistent with the earlier observations in Figure 6.5.

The plot of average delay penalty further supports this trend. For the porters, delay penalties decrease significantly, by 86.4% for group one and 92.3% for group two, up to the inclusion of six robots. Initially, the delay penalty for robots is relatively high due to the small number of robots being overloaded with assigned tasks, which leads to higher penalties as delays accumulate over time. However, once the robot count reaches six, this metric stabilizes and remains low thereafter.

In contrast, the average idle time percentage increases for all transporters. For porters, it rises from approximately 34% to 55%, and for service robots, it is already around 44% at six robots and continues to increase. This is expected, given that only about 40% of requests (see Table 6.17) are eligible to be handled by robots; thus, adding more robots beyond a certain point merely inflates idle time without contributing to system performance. Nevertheless, when examining the other metrics, we observe that the lowest idle time values correspond to the highest values in workload and other performance penalties. Therefore, in a hospital environment—where responsiveness is more critical than pure utilization—this level of idle

time is considered acceptable.

Overall, across all four performance metrics, the configuration with 16 porters and 6 service robots yields the most significant improvements. Beyond this point, the marginal gains are minimal, suggesting that this level of robot integration strikes an effective balance between efficiency and resource allocation.

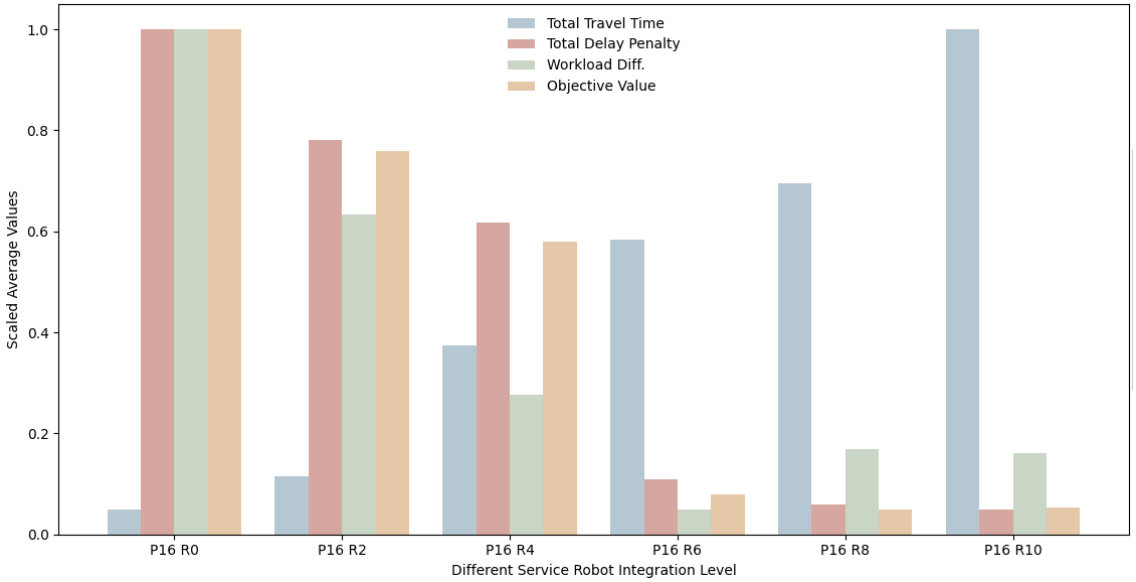


Figure 6.7: Comparison of normalized objective function terms across different service robot integration levels.

Table 6.18: Average objective function terms across different service robot integration levels.

Integration Level	**Travel Time**	**Delay Penalty**	**Workload Diff.**	**Objective**
P16 R0	5711.84	954.26	322.26	2730.89
P16 R2	5714.55	755.92	256.48	2639.49
P16 R4	5725.47	608.82	192.50	2572.22
P16 R6	5734.26	148.08	152.02	2383.34
P16 R8	5739.00	104.27	173.34	2371.98
P16 R10	5751.83	95.10	171.77	2373.13

As a final analysis, we examine the impact of service robot integration on the

components of the defined objective function. Unlike the previous section, which focused on performance metrics for each transporter type individually, this part evaluates overall system-level performance. The results are summarized in Figure 6.7 and Table 6.18. To enable a consistent comparison across different metrics, values in the figure were normalized using the formulation $\tilde{x}_i = \frac{x_i - \min(x)}{\max(x) - \min(x) + \varepsilon}$, where ε is a small constant added for numerical stability.

Across all configurations, total travel time remains relatively constant. The minor observed variations are primarily due to depot assignments and routing randomness. In contrast, delay penalties exhibit a substantial reduction: with the introduction of just two service robots, the system experiences a 21% decrease, and with ten robots, the delay penalty reduction reaches nearly 90%. This is particularly critical in hospital settings, where responsiveness and timeliness are paramount for operational success and patient safety. In addition to improving responsiveness, service robots also contribute to balancing the workload among porters. The workload difference drops from 322.26 to a minimum of 152.02 when six robots are employed. This not only alleviates pressure on the human workforce but also enhances the system's flexibility in accommodating unexpected demand surges. When evaluating the total objective function, which reflects the hospital's weighted prioritization of travel time, delay penalty, and workload balance, substantial improvements are again observed. The objective function value decreases by as much as 13.15% in the configuration with eight robots. However, consistent with earlier findings, the results suggest that deploying six service robots is sufficient to achieve a well-balanced, responsive, and effective intra-hospital transportation system. Additional robots beyond this threshold yield marginal benefits, highlighting the importance of targeted, rather than excessive, automation.

6.5.2 Overall Insights

The findings of this part yield several important managerial implications for hospital administrators and healthcare system designers considering the adoption of service robots in intra-hospital transportation. Notably, our analysis departs from

traditional cost-efficiency narratives and instead foregrounds responsiveness, task suitability, and policy-aware deployment as key levers in automation design.

First, one of the most compelling insights is that the integration of service robots can significantly enhance system responsiveness, even when they are only eligible to handle less than half of the total request types. In our setting, approximately 40% of requests can be served by robots, yet incorporating them led to notable reductions in travel time and delay penalties per transporter. These benefits were especially prominent when the robots operated at speeds comparable to human porters. This challenges the prevailing assumption that automation only becomes beneficial when it covers a majority of operational tasks. In time-sensitive healthcare environments, the performance impact of automation should therefore be evaluated not merely in terms of task coverage or utilization rates, but rather through its contribution to timely service delivery.

Second, the results show that performance improvements tend to saturate after a certain level of integration. Specifically, with six service robots, the system achieves substantial improvements across workload, delay penalty, and travel time, but further increases in the number of service robots result in diminishing returns. This plateau effect is due to the limited share of requests that are robot-eligible. Importantly, the rise in fleet size is accompanied by a growth in robot idle time; however, this idle capacity supports robustness under stochastic conditions, acting as a buffer against variability in request arrivals, corridor congestion, and elevator usage. Therefore, instead of indiscriminately adding more service robots, hospitals might achieve greater performance gains by redesigning workflows to make more requests compatible with robotic execution, or by fine-tuning scheduling algorithms to better exploit robot availability during low-load periods.

Another important insight is the benefit of role separation between porters and service robots. By reassigning repetitive and standardized delivery requests, such as those for medical supplies, linen, and administrative documents, to service robots, porters are freed to focus on more critical, human-dependent requests such as patient transportation. This hybrid division of labor enhances workforce flexibility and ensures that scarce human labor is allocated where it is most valuable. Im-

portantly, the study does not advocate for replacing porters but rather shows how robots can complement human workers in a way that preserves service quality while improving operational flow. In fact, our findings suggest that even a simple policy, assigning straightforward, robot-eligible tasks to service robots and reserving the remaining complex requests for human porters, can produce significant performance improvements without requiring intricate rule systems or full-scale automation.

Interestingly, the increase in robot idle time as the fleet grows is not accompanied by a decline in performance. On the contrary, the best-performing configuration includes six robots, even though idle times are relatively high at that level. This suggests that idle time in such systems should not be interpreted purely as inefficiency. In healthcare settings, excess capacity may function as a buffer against demand spikes or delays, contributing to overall system resilience. Managers should thus consider idle capacity as a form of strategic flexibility rather than a metric to be minimized.

Lastly, the integration of robots fundamentally reshapes the system's performance frontier. By significantly reducing delay penalties and workload imbalances without increasing total travel time, service robot integration enables a new set of operating points that were previously unattainable. This shifts the trade-off landscape between responsiveness, efficiency, and fairness in resource allocation. For managers, this opens new possibilities in setting key performance indicators (KPIs), defining service-level agreements, and aligning logistics strategy with clinical care goals.

In summary, this section highlights the value of service robots not simply as task executors, but as enablers of a more responsive, balanced, and robust hospital transportation system. These insights call for a more strategic and policy-aware view of automation, one that maximizes complementarity between human and robotic capabilities rather than treating them as substitutes.

Chapter 7

CONCLUSION

This study addresses the intra-hospital transportation problem with a focus on modeling and optimizing the assignment and routing of various transportation requests in healthcare environments where both human porters and autonomous service robots are utilized. The primary aim is to minimize total travel time and workload imbalance across available transporters, while effectively managing heterogeneous transportation tasks. The problem includes three distinct types of transportation requests: patient requests, laboratory and pharmaceutical requests, and administrative and supply-related requests. Each request type differs in terms of handling requirements, urgency, and eligibility for service by different transporter types. Human porters are allowed to serve all request types without restrictions, while service robots can only serve a subset of tasks, specifically laboratory and pharmaceutical as well as administrative and supply requests, due to technological and infrastructural limitations. The system incorporates multiple operational constraints, including transporter capacities, compatibility between transporter and request types, precedence between pickup and delivery, service times at pickup and delivery points, and the possibility of having multiple delivery points for a single request. Further constraints include the availability of robots, their need to recharge after a limited number of tasks, and the assumption that each transporter may only handle one request at a time. Additionally, the environment is subject to dynamic request arrivals over time, requiring real-time decision-making under uncertainty. The combination of multiple transporter types, diverse request categories, capacity limitations, and operational rules defines a complex logistics problem aimed at improving the efficiency and responsiveness of intra-hospital transportation systems.

This study considers both the static and dynamic versions of the intra-hospital transportation problem to reflect the varying levels of uncertainty and information

availability in real-world hospital operations. In the static version of the problem, all transportation requests are assumed to be known in advance, allowing for a deterministic and centralized optimization model that jointly determines the assignment of requests to transporters and their corresponding service routes. This formulation enables a clear evaluation of the problem's computational structure and provides a baseline for solution quality under ideal information conditions. In contrast, the dynamic version reflects realistic hospital settings where transportation requests arrive over time and decisions must be made sequentially with incomplete information about future events. The dynamic problem incorporates stochastic elements and requires real-time assignment of requests to available transporters upon their arrival, while also adapting routing plans to account for ongoing service operations, transporter availability, and request urgency. By addressing both static and dynamic versions, the study captures the full operational spectrum of intra-hospital logistics and facilitates a comprehensive evaluation of algorithmic performance under different informational assumptions.

In the static version of the problem, all transportation requests are assumed to be known before the planning horizon begins, allowing the system to make global assignment and routing decisions in a centralized manner. This setting is modeled as a multi-depot Dial-a-Ride Problem (DARP), where each transporter, whether a human porter or a service robot, operates from its own depot and serves a subset of requests subject to a set of feasibility and operational constraints. The objective is to determine a feasible set of routes and assignments that minimize the total travel time and workload imbalance across transporters. The formulation accounts for multiple critical features inherent to intra-hospital settings, including compatibility constraints between request types and transporter types, multiple delivery points per request, service times at both pickup and delivery locations, and transporter-specific eligibility restrictions. Each request must be served exactly once, with pickup operations preceding deliveries, and all service operations must occur within the time horizon. The model also ensures that each transporter can only serve one request at a time and that the cumulative load and number of stops on each route do not exceed the allowed capacity thresholds. This static DARP-based formulation

provides a mathematically rigorous foundation for analyzing the core combinatorial structure of the problem and serves as a benchmark for evaluating the performance of dynamic policies under full-information conditions.

In the dynamic version of the intra-hospital transportation problem, transportation requests arrive in real time, and the system must make sequential decisions without full knowledge of future events. This setting closely mirrors actual hospital operations, where tasks such as patient transfers and medical item deliveries are generated unpredictably and need to be handled promptly. The dynamic problem is significantly more challenging than the static case due to the uncertainty in request arrivals, variable transporter availability, and real-time operational constraints. To effectively manage these challenges, the study proposes a two-phase solution methodology that separates the problem into assignment and routing stages, enabling a more tractable and responsive decision-making process.

The first phase of the dynamic solution framework is the assignment phase, where newly arrived requests are assigned to available transporters upon triggering events. These triggers occur when either a predefined number of requests has accumulated in the queue or a specified time interval has passed since the last assignment decision. The assignment problem is modeled as a bipartite matching between active transporters and waiting requests. To handle the complexity and variability of the real-time environment, the assignment policy is learned using a Deep Q-Learning algorithm with a Bayesian Neural Network as the function approximator. This architecture enables the model to quantify uncertainty in Q-value estimates and thus support more robust decision-making in the presence of sparse or variable data. The Q-values represent the expected long-term cost of assigning a particular request to a transporter, and the action selection incorporates both mean and variance in the predictions to improve exploration and reliability. The output of this phase is a set of assigned request-transporter pairs to be processed in the routing phase.

The second phase is the routing phase, where the objective is to generate or update service routes for each transporter based on the current assignments. This phase addresses the inherent complexity of consolidating multiple requests on a single route while respecting precedence, service time, and feasibility constraints. For

this purpose, the study implements a heuristic-based approach combining Best Insertion and Adaptive Variable Neighborhood Search (AVNS). The Best Insertion procedure is used to efficiently construct initial feasible routes by inserting assigned requests into the current route of each transporter in a way that minimally increases the objective function. Following this, the AVNS method is applied to explore the local search space and improve route quality through a sequence of neighborhood operations. These operations are adaptively selected based on their historical performance, allowing the search process to dynamically focus on the most promising modifications. The use of AVNS also helps overcome local optima and enhances solution robustness under varying request characteristics.

Together, the two-phase framework allows the system to reactively assign and schedule transportation requests in a computationally efficient and scalable manner. By decoupling assignment from routing, the approach enables the system to make real-time decisions without needing to reoptimize the full solution space at each event, thereby reducing computational overhead while maintaining high-quality solutions. The integration of reinforcement learning for assignment and metaheuristics for routing leverages the strengths of both machine learning and combinatorial optimization, offering a hybrid framework that is well-suited to the demands of intra-hospital logistics operations under uncertainty.

Our computational experiments thoroughly examined and quantified the performance of the proposed Bayesian Deep Q-Learning (DQL) assignment policy across various operational scenarios involving different transporter fleet configurations and distinct speed variations among human porters and service robots. We explicitly compared this policy to several baseline methods, including the deterministic DQL policy, the heuristic assignment approach, the hospital's existing operational policy, and a randomized benchmark. The Bayesian DQL policy consistently demonstrated superior performance, particularly in complex, heterogeneous fleet environments characterized by mixed human-robot transporter combinations and varying transporter speeds. Specifically, under scenarios with 30 porters and 10 robots, the Bayesian DQL policy achieved an average objective value improvement of approximately 36.4% over the randomized benchmark, outperforming deterministic

DQL and heuristic-based methods by margins exceeding 10% and 4%, respectively. Furthermore, the robustness of the Bayesian DQL policy was underscored by its consistently low standard deviation in performance across all examined instances, signifying stable outcomes and adaptability to the dynamic conditions inherent in hospital operations. Notably, we assessed the performance under different speed variation scenarios, wherein robots operated at equal, slower, or combined speeds relative to human porters, necessitating the training of separate Bayesian Neural Networks (BNNs) tailored explicitly to each speed scenario. This training approach enhanced the adaptability and efficacy of the Bayesian DQL policy, which maintained a performance improvement of nearly 29% even in scenarios involving robots at significantly reduced speeds. Additionally, our analysis of scenarios in which transporters had individually assigned heterogeneous speeds revealed further advantages of our learning-based assignment method, especially when compared to static or rule-based baselines. Importantly, our evaluations indicated that, even under conditions where robots operated at faster speeds, the frequency of their visits to recharging stations remained manageable, ensuring operational continuity and minimal disruptions. Collectively, these rigorous computational insights underscore the substantial advantages of employing a Bayesian DQL approach in real-time intra-hospital transportation systems, particularly highlighting its capability to effectively navigate complexities and heterogeneity, thereby offering substantial improvements in operational efficiency and reliability compared to conventional assignment policies.

As an extension to the assignment optimization evaluated in Phase 1, we further investigated the routing optimization problem (Phase 2) through rigorous comparative analyses between our Best Insertion, coupled with Adaptive Variable Neighborhood Search (AVNS) heuristic, and exact solutions generated by a Mixed Integer Linear Programming (MILP) model. The MILP model served as a robust benchmark, solved to optimality or near-optimality within a one-hour time frame using Gurobi Optimizer. Computational experiments were conducted across systematically generated intra-hospital transportation instances, varying from smaller scenarios with four requests to more complex settings involving up to eight requests, effectively

simulating realistic operational conditions. Results revealed that for smaller-scale instances, our proposed heuristic consistently delivered solutions closely aligning with the optimal values obtained from the MILP model, typically exhibiting objective function deviations of less than 2%. This highlights the heuristic's capability to achieve near-optimal performance even in strictly constrained environments. In contrast, as problem complexity escalated to scenarios involving six and eight requests, the heuristic increasingly demonstrated significant computational advantages. Although the MILP model consistently struggled to find high-quality solutions within acceptable timeframes, often fully exhausting the allotted computational time, the heuristic maintained both solution quality and computational efficiency, frequently outperforming the MILP approach by substantial margins. Specifically, for eight-request instances, the heuristic solutions deviated from optimality by an average of only around 3.5%, yet achieved these outcomes with computational times several orders of magnitude lower than the MILP benchmarks. Thus, this analysis strongly supports the heuristic's practical suitability and scalability for real-time routing decisions in dynamic intra-hospital environments, where timely, near-optimal solutions are crucial.

Furthermore, the numerical results provide robust evidence that our proposed two-phase approach substantially outperforms the conventional one-phase reoptimization method in both computational efficiency and solution quality. Across the tested scenarios, the two-phase method consistently demonstrates significant runtime improvements, achieving up to 99.8% reductions compared to the one-phase baseline, reducing average computation times from approximately 954.97 seconds to merely 1.67 seconds in the intensive $\mathcal{I}^{HP}$ instances. Importantly, these runtime gains do not sacrifice solution quality; indeed, our method often yields superior objective values. For instance, in high-frequency request scenarios ($\mathcal{I}^{HP}$), the Bayesian neural network-based assignment followed by Best Insertion augmented with Adaptive Variable Neighborhood Search (BI+AVNS) provides average objective improvements of up to 12.16% compared to the one-phase policy that considers elapsed delays. In the less frequent request arrival scenario ($\mathcal{I}^{PP}_{\lambda=1/10}$), the advantages become even more pronounced: BI+AVNS reduces average objective values

dramatically from 1037.89 to 430.15, corresponding to an improvement exceeding 58%, while maintaining substantial runtime advantages. The clear separation of assignment and routing decisions simplifies each subproblem, enabling local search heuristics to more effectively optimize transporter routes and thus enhancing overall performance. Moreover, the two-phase approach's reliance on Bayesian neural network predictions allows for foresighted decision-making, contrasting sharply with the myopic reoptimization employed by the one-phase baseline. Consequently, our methodology not only significantly improves real-time responsiveness, crucial in dynamic intra-hospital environments, but also maintains robustness and scalability under varying operational conditions, making it particularly suitable for implementation in practice.

Our experiments investigating various operational and tactical decisions reveal significant operational advantages stemming from the incorporation of service robots alongside human porters within intra-hospital transportation systems. Notably, despite robots being restricted to approximately 40% of total request types, their integration substantially enhances overall system responsiveness and effectiveness. Specifically, introducing as few as six robots into a team of sixteen human porters yields notable reductions, including a decrease in the delay penalties by approximately 84%, a 15% reduction in porter workloads, and a significant improvement in workload balance among porters, effectively addressing critical healthcare demands for rapid, reliable, and equitable task fulfillment. Furthermore, when robots are operating at speeds comparable to human porters, their efficiency becomes particularly pronounced, underscoring their capability to effectively alleviate the burden on human resources. Importantly, these benefits plateau beyond a certain robot fleet size, with marginal gains observed when expanding the robot fleet beyond six units. This diminishing return highlights the necessity for thoughtful fleet-sizing and operational policies. Additionally, the clear separation of duties"assigning standardized requests, such as administrative documents and medical supplies, to robots, while reserving patient transport and other human-sensitive tasks exclusively for porters"emerges as a highly effective policy. This strategy capitalizes on the complementary strengths of automated and human resources, enabling hospital managers to achieve enhanced

responsiveness without sacrificing operational flexibility or increasing total travel time. Ultimately, these findings reinforce that effective robotic integration is less about maximizing automation levels and more about strategically matching robotic capabilities to task profiles, emphasizing a hybrid operational model designed to leverage robotic efficiency and human adaptability to elevate overall hospital transport performance.

The most salient feature of our study is the implementation of a comprehensive two-phase solution approach for dynamic intra-hospital transportation problems, integrating service robots alongside human porters. Our objective has been to enhance system responsiveness and operational efficiency by strategically assigning different request types"including on-demand and scheduled patient transportation, medical supplies, medical equipment, and administrative documents"to the appropriate transporter types based on eligibility and capacity constraints. We accounted for the heterogeneous capabilities of transporters, considering that human porters can handle all request types, whereas service robots have limitations in handling certain requests, primarily patient-related tasks. Our methodology involves employing a Bayesian Deep Q-Network (DQN) integrated with a Best Insertion heuristic for initial assignment, followed by Adaptive Variable Neighborhood Search (AVNS) to further optimize routes dynamically. This novel integration of machine learning and heuristic optimization strategies has demonstrated significant improvements in system performance metrics, including reduced travel times, lower delay penalties, and more equitable workload distributions.

Future research directions stemming from our study include:

- Extending the developed models to multi-period settings to account for longer planning horizons, where resource availability, transporter schedules, and demand patterns might vary significantly across days or weeks.

- Investigating advanced predictive analytics and real-time forecasting techniques to anticipate request arrivals, transporter availability, and congestion hotspots within the hospital, thereby improving proactive and dynamic allocation policies.

- Developing more sophisticated robot eligibility criteria, potentially expanding the types of requests service robots can handle through technological advancements or task simplification, to increase automation effectiveness.

- Exploring the incorporation of uncertainty and stochastic elements into the model, such as varying transporter speeds, elevator congestion, or patient readiness delays, utilizing robust or stochastic optimization frameworks.

- Considering integrated planning of service robot recharging and maintenance schedules within operational assignment and routing problems, which is crucial for ensuring continuous robot availability and operational reliability.

- Investigating human factors and ergonomics considerations by studying how service robot integration affects human porter satisfaction, task complexity perception, and overall workforce well-being, thus promoting better human-robot collaboration.

- Incorporating heterogeneous skill sets among human porters into the assignment and routing models, recognizing that not all porters may be equally qualified for all request types, thereby enabling more realistic and efficient task allocations.

- Evaluating policies for idle transporter behavior, such as whether porters and service robots should return to their depot or remain stationed near likely demand hotspots, to balance responsiveness with energy and time efficiency.

These future avenues promise further enhancements in both theoretical frameworks and practical applications, contributing significantly to the advancement of intra-hospital transportation logistics and operational management.

BIBLIOGRAPHY

[Abdulghani et al., 2022] Abdulghani, H. A., Nijdam, N. A., and Konstantas, D. (2022). Analysis on security and privacy guidelines: Rfid-based iot applications. *Ieee Access*, 10:131528–131554.

[Aldaihani and Dessouky, 2003] Aldaihani, M. and Dessouky, M. M. (2003). Hybrid scheduling methods for paratransit operations. *Computers & Industrial Engineering*, 45(1):75–96.

[Attanasio et al., 2004] Attanasio, A., Cordeau, J.-F., Ghiani, G., and Laporte, G. (2004). Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30(3):377–387.

[Awawdeh and Abu-Shanab,] Awawdeh, N. and Abu-Shanab, E. Rfid in healthcare industry: A strategic framework for implementation.

[Aziez et al., 2022] Aziez, I., Côté, J.-F., and Coelho, L. C. (2022). Fleet sizing and routing of healthcare automated guided vehicles. *Transportation Research Part E: Logistics and Transportation Review*, 161:102679.

[BačÍK et al., 2017] BačÍK, J., ĎUrovskỳ, F., Biroš, M., Kyslan, K., Perdukova, D., and Padmanaban, S. (2017). Pathfinder–development of automated guided vehicle for hospital logistics. *Ieee Access*, 5:26892–26900.

[Bard and Jarrah, 2013] Bard, J. F. and Jarrah, A. I. (2013). Integrating commercial and residential pickup and delivery networks: A case study. *Omega*, 41(4):706–720.

[Bärmann et al., 2024] Bärmann, A., Martin, A., Müller, A., and Weninger, D.

(2024). A column generation approach for the lexicographic optimization of intra-hospital transports. *OR Spectrum*, pages 1–50.

[Baugh Jr et al., 1998] Baugh Jr, J. W., Kakivaya, G. K. R., and Stone, J. R. (1998). Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. *Engineering Optimization*, 30(2):91–123.

[Beaudry et al., 2010] Beaudry, A., Laporte, G., Melo, T., and Nickel, S. (2010). Dynamic transportation of patients in hospitals. *OR spectrum*, 32:77–107.

[Berbeglia et al., 2012] Berbeglia, G., Cordeau, J.-F., and Laporte, G. (2012). A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS Journal on Computing*, 24(3):343–355.

[Bernhard et al., 2023] Bernhard, L., Amalanesan, A. F., Baumann, O., Rothmeyer, F., Hafner, Y., Berlet, M., Wilhelm, D., and Knoll, A. (2023). Mobile service robots for the operating room wing: balancing cost and performance by optimizing robotic fleet size and composition. *International Journal of Computer Assisted Radiology and Surgery*, 18(2):195–204.

[Bettinelli et al., 2011] Bettinelli, A., Ceselli, A., and Righini, G. (2011). A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 19(5):723–740.

[Blundell et al., 2015] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

[Bochem et al., 2022] Bochem, A., Abugabah, A., and Al Smadi, A. (2022). Smart rfid application in health care: Using rfid technology for smart inventory and logistic systems in hospitals. In *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 18–23. IEEE.

[Bozer and Eamrungroj, 2020] Bozer, Y. A. and Eamrungroj, C. (2020). Analysis of patient-mover dispatching and equipment-marshalling areas: a simulation study at the university of michigan hospital. *Health Systems*, 9(3):226–252.

[Braekers et al., 2014] Braekers, K., Caris, A., and Janssens, G. K. (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, 67:166–186.

[Camacho-Cogollo et al., 2020] Camacho-Cogollo, J. E., Bonet, I., and Iadanza, E. (2020). Rfid technology in health care. In *Clinical Engineering Handbook*, pages 33–41. Elsevier.

[Chane-Haï et al., 2020] Chane-Haï, T., Vercraene, S. V., and Monteiro, T. (2020). Sharing a ride time constraint in a multi-trip dial-a-ride problem. an application to the non-urgent patient transportation problem. In *MOSIM 2020-13th International Conference on Modeling, Optimization and Simulation*.

[Chen et al., 2005] Chen, L., Gerschman, M., Odegaard, F., Puterman, D. K., Puterman, M. L., and Quee, R. (2005). Designing an efficient hospital porter system. *Healthcare Quarterly*, pages 1–7.

[Cheng et al., 2023] Cheng, L., Zhao, N., Wu, K., and Chen, Z. (2023). The multi-trip autonomous mobile robot scheduling problem with time windows in a stochastic environment at smart hospitals. *Applied Sciences*, 13(17):9879.

[Chit et al., 2021] Chit, M. M. T., Srisiri, W., Siritantikorn, A., Kongruttanachok, N., and Benjapolakul, W. (2021). Application of rfid and iot technology into specimen logistic system in the healthcare sector. In *2021 3rd International Conference on Advancements in Computing (ICAC)*, pages 7–12. IEEE.

[Cordeau, 2006] Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations research*, 54(3):573–586.

[Cordeau and Laporte, 2003a] Cordeau, J.-F. and Laporte, G. (2003a). The dial-a-ride problem (darp): Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1:89–101.

[Cordeau and Laporte, 2003b] Cordeau, J.-F. and Laporte, G. (2003b). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594.

[Cordeau and Laporte, 2007] Cordeau, J.-F. and Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Annals of operations research*, 153:29–46.

[Coslovich et al., 2006] Coslovich, L., Pesenti, R., and Ukovich, W. (2006). A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research*, 175(3):1605–1615.

[COURIER, 2000] COURIER, S. O. R. (2000). Simulation of robotic courier deliveries in hospital distribution services.

[Cruz et al., 2024] Cruz, E. M., Oliveira, S., and Correia, A. (2024). Robotics application in the hospital domain: Operator-patient-robot interface.

[Dahle et al., 2019] Dahle, L., Andersson, H., Christiansen, M., and Speranza, M. G. (2019). The pickup and delivery problem with time windows and occasional drivers. *Computers & Operations Research*, 109:122–133.

[Dershin and Schaik, 1993] Dershin, H. and Schaik, M. S. (1993). Quality improvement for a hospital patient transportation system. *Journal of Healthcare Management*, 38(1):111.

[Desrosiers et al., 1986] Desrosiers, J., Dumas, Y., and Soumis, F. (1986). A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences*, 6(3-4):301–325.

[Detti et al., 2017] Detti, P., Papalini, F., and de Lara, G. Z. M. (2017). A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega*, 70:1–14.

[Elmbach et al., 2015] Elmbach, A. F. v., Boysen, N., Briskorn, D., and Mothes, S. (2015). Scheduling pick-up and delivery jobs in a hospital to level ergonomic stress. *IIE Transactions on Healthcare Systems Engineering*, 5(1):42–53.

[Ermagan et al., 2023] Ermagan, U., Yildiz, B., and Salman, S. (2023). Machine learning-enhanced column generation approach for express shipments with autonomous robots and public transportation. *Available at SSRN 4471292*.

[Fanti et al., 2020] Fanti, M. P., Mangini, A. M., Roccotelli, M., and Silvestri, B. (2020). Hospital drugs distribution with autonomous robot vehicles. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1025–1030. IEEE.

[Fdata Co., 2025] Fdata Co., L. (2025). A801 tug robot for hospital. https://www.fdatarobot.com/products/a801-hospital-delivery-robot/. Accessed: 2025-04-14.

[Fiegl and Pontow, 2009] Fiegl, C. and Pontow, C. (2009). Online scheduling of pick-up and delivery tasks in hospitals. *Journal of Biomedical Informatics*, 42(4):624–632.

[Fragapane et al., 2020] Fragapane, G., Hvolby, H.-H., Sgarbossa, F., and Strandhagen, J. O. (2020). Autonomous mobile robots in hospital logistics. In *IFIP International Conference on Advances in Production Management Systems*, pages 672–679. Springer.

[Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

[Geary et al., 2022] Geary, U., Ward, M. E., Callan, V., McDonald, N., and Corrigan, S. (2022). A socio-technical systems analysis of the application of rfid-enabled technology to the transport of precious laboratory samples in a large acute teaching hospital. *Applied ergonomics*, 102:103759.

[Ghiani et al., 2022] Ghiani, G., Manni, A., and Manni, E. (2022). A scalable anticipatory policy for the dynamic pickup and delivery problem. *Computers & Operations Research*, 147:105943.

[Gschwind and Irnich, 2015] Gschwind, T. and Irnich, S. (2015). Effective handling of dynamic time windows and its application to solving the dial-a-ride problem. *Transportation Science*, 49(2):335–354.

[Gurobi Optimization, LLC, 2024] Gurobi Optimization, LLC (2024). How do i use the no relaxation (norel) heuristic? https://support.gurobi.com/hc/en-us/articles/30992739009553. Accessed: 2025-05-08.

[Guzmán Ortiz et al., 2021] Guzmán Ortiz, E., Andres, B., Fraile, F., Poler, R., and Ortiz Bas, Á. (2021). Fleet management system for mobile robots in healthcare environments. *Journal of Industrial Engineering and Management (JIEM)*, 14(1):55–71.

[Haddara and Staaby, 2018] Haddara, M. and Staaby, A. (2018). Rfid applications and adoptions in healthcare: a review on patient safety. *Procedia computer science*, 138:80–88.

[Häll et al., 2015] Häll, C. H., Lundgren, J. T., and Voss, S. (2015). Evaluating the performance of a dial-a-ride service using simulation. *Public Transport*, 7:139–157.

[Häll and Peterson, 2013] Häll, C. H. and Peterson, A. (2013). Improving paratransit scheduling using ruin and recreate methods. *Transportation planning and technology*, 36(4):377–393.

[Hamdi et al., 2024] Hamdi, K., Megdiche, I., Lamine, E., and Duval, D. (2024). An optimization model for patient transportation problem within stretchers activity in hospitals. In *2024 IEEE/ACS 21st International Conference on Computer Systems and Applications (AICCSA)*, pages 1–6. IEEE.

[Hanne et al., 2009] Hanne, T., Melo, T., and Nickel, S. (2009). Bringing robustness to patient flow management through optimized patient transports in hospitals. *Interfaces*, 39(3):241–255.

[Ho et al., 2018] Ho, S. C., Szeto, W. Y., Kuo, Y.-H., Leung, J. M., Petering, M., and Tou, T. W. (2018). A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological*, 111:395–421.

[Holland et al., 2021] Holland, J., Kingston, L., McCarthy, C., Armstrong, E., O'Dwyer, P., Merz, F., and McConnell, M. (2021). Service robots in the healthcare sector. *Robotics*, 10(1):47.

[Jaw et al., 1986] Jaw, J.-J., Odoni, A. R., Psaraftis, H. N., and Wilson, N. H. (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological*, 20(3):243–257.

[Jeong and Moon, 2024] Jeong, J. and Moon, I. (2024). Dynamic pickup and delivery problem for autonomous delivery robots in an airport terminal. *Computers & Industrial Engineering*, 196:110476.

[Jin et al., 2021] Jin, H., He, Q., He, M., Lu, S., Hu, F., and Hao, D. (2021). Optimization for medical logistics robot based on model of traveling salesman problems and vehicle routing problems. *International Journal of Advanced Robotic Systems*, 18(3):17298814211022539.

[Jorgensen et al., 2007] Jorgensen, R. M., Larsen, J., and Bergvinsdottir, K. B.

(2007). Solving the dial-a-ride problem using genetic algorithms. *Journal of the operational research society*, 58(10):1321–1331.

[Kergosien et al., 2023] Kergosien, Y., Bélanger, V., and Ruiz, A. (2023). A capacity sharing approach to manage jointly transportation and emergency fleets at ems organisations. *International Journal of Production Research*, 61(3):880–897.

[Kergosien et al., 2011] Kergosien, Y., Lente, C., Piton, D., and Billaut, J.-C. (2011). A tabu search heuristic for the dynamic transportation of patients between care units. *European Journal of Operational Research*, 214(2):442–452.

[Khalid et al., 2021] Khalid, M., Awais, M., Singh, N., Khan, S., Raza, M., Malik, Q. B., and Imran, M. (2021). Autonomous transportation in emergency healthcare services: framework, challenges, and future work. *IEEE Internet of Things Magazine*, 4(1):28–33.

[Kiechle et al., 2007] Kiechle, G., Doerner, K., Gendreau, M., and Hartl, R. (2007). Patient transportation-dynamic dial-a-ride and emergency transportation problems. In *Preprints of Sixth Triennial Symposium on Transportation Analysis (TRISTAN)*.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kingma et al., 2013] Kingma, D. P., Welling, M., et al. (2013). Auto-encoding variational bayes.

[Kuchera and Rohleder, 2011] Kuchera, D. and Rohleder, T. R. (2011). Optimizing the patient transport function at mayo clinic. *Quality Management in Healthcare*, 20(4):334–342.

[Kumari et al., 2015] Kumari, L., Narsaiah, K., Grewal, M., and Anurag, R. (2015). Application of rfid in agri-food sector. *Trends in Food Science & Technology*, 43(2):144–161.

[Landry and Philippe, 2004] Landry, S. and Philippe, R. (2004). How logistics can service healthcare. In *Supply Chain Forum: An International Journal*, volume 5, pages 24–30. Taylor & Francis.

[Lee and He, 2019] Lee, D. and He, N. (2019). Target-based temporal-difference learning. In *International Conference on Machine Learning*, pages 3713–3722. PMLR.

[Lim et al., 2017] Lim, A., Zhang, Z., and Qin, H. (2017). Pickup and delivery service with manpower planning in hong kong public hospitals. *Transportation Science*, 51(2):688–705.

[Lin, 1992] Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8:293–321.

[Luo et al., 2019] Luo, Z., Liu, M., and Lim, A. (2019). A two-phase branch-and-price-and-cut for a dial-a-ride problem in patient transportation. *Transportation Science*, 53(1):113–130.

[Madsen et al., 1995] Madsen, O. B., Ravn, H. F., and Rygaard, J. M. (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of operations Research*, 60:193–208.

[Marmaglio et al., 2023] Marmaglio, P., Consolati, D., Amici, C., and Tiboni, M. (2023). Autonomous vehicles for healthcare applications: a review on mobile robotic systems and drones in hospital and clinical environments. *Electronics*, 12(23):4791.

[Martinez Perez et al., 2018] Martinez Perez, M., Dafonte, C., and Gómez, Á. (2018). Traceability in patient healthcare through the integration of rfid technology in an icu in a hospital. *Sensors*, 18(5):1627.

[Melachrinoudis et al., 2007] Melachrinoudis, E., Ilhan, A. B., and Min, H. (2007).

A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research*, 34(3):742–759.

[Mladenović and Hansen, 1997] Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100.

[Molenbruch et al., 2017a] Molenbruch, Y., Braekers, K., and Caris, A. (2017a). Typology and literature review for dial-a-ride problems. *Annals of Operations Research*, 259:295–325.

[Molenbruch et al., 2017b] Molenbruch, Y., Braekers, K., Caris, A., and Berghe, G. V. (2017b). Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. *Computers & Operations Research*, 77:58–71.

[Munoz-Ausecha et al., 2021] Munoz-Ausecha, C., Ruiz-Rosero, J., and Ramirez-Gonzalez, G. (2021). Rfid applications and security review. *Computation*, 9(6):69.

[Parragh et al., 2006] Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2006). A survey on pickup and delivery models part ii: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58:81–117.

[Profetto et al., 2022] Profetto, L., Gherardelli, M., and Iadanza, E. (2022). Radio frequency identification (rfid) in health care: where are we? a scoping review. *Health and technology*, 12(5):879–891.

[Psaraftis, 1980] Psaraftis, H. N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2):130–154.

[Psaraftis, 1983] Psaraftis, H. N. (1983). An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation science*, 17(3):351–357.

[Qu and Bard, 2015] Qu, Y. and Bard, J. F. (2015). A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. *Transportation Science*, 49(2):254–270.

[Quadrifoglio et al., 2008] Quadrifoglio, L., Dessouky, M. M., and Ordóñez, F. (2008). A simulation study of demand responsive transit system design. *Transportation Research Part A: Policy and Practice*, 42(4):718–737.

[Rondoni et al., 2024] Rondoni, C., Scotto di Luzio, F., Tamantini, C., Tagliamonte, N. L., Chiurazzi, M., Ciuti, G., and Zollo, L. (2024). Navigation benchmarking for autonomous mobile robots in hospital environment. *Scientific Reports*, 14(1):18334.

[Ropke et al., 2007] Ropke, S., Cordeau, J.-F., and Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks: An International Journal*, 49(4):258–272.

[Rossetti and Selandari, 2001] Rossetti, M. D. and Selandari, F. (2001). Multi-objective analysis of hospital delivery systems. *Computers & Industrial Engineering*, 41(3):309–333.

[Ruan et al., 2018] Ruan, W., Sheng, Q. Z., Yao, L., Li, X., Falkner, N. J., and Yang, L. (2018). Device-free human localization and tracking with uhf passive rfid tags: A data-driven approach. *Journal of Network and Computer Applications*, 104:78–96.

[Schall, 1988] Schall, R. T. (1988). Increased productivity through a central transportation system. *Hospital materiel management quarterly*, 9(4):77–81.

[Séguin et al., 2019] Séguin, S., Villeneuve, Y., and Blouin-Delisle, C.-H. (2019). Improving patient transportation in hospitals using a mixed-integer programming model. *Operations Research for Health Care*, 23:100202.

[Søraa and Fostervold, 2021] Søraa, R. A. and Fostervold, M. E. (2021). Social domestication of service robots: The secret lives of automated guided vehicles (agvs) at a norwegian hospital. *International journal of human-computer studies*, 152:102627.

[Sun et al., 2025] Sun, C. L., Copenhaver, M. S., Zenteno Langle, A. C., Viscomi, B., Raeke, E., Daily, B. J., Dunn, P. F., and Levi, R. (2025). Improved intrahospital transport time via proximity-based staff assignments. *Journal of the American Medical Informatics Association*, page ocaf081.

[Sutton, 2018] Sutton, R. S. (2018). Reinforcement learning: An introduction. *A Bradford Book*.

[Tempczyk et al., 2022] Tempczyk, P., Smoczyński, K., Smolenski-Jensen, P., and Cygan, M. (2022). One simple trick to fix your bayesian neural network. *arXiv preprint arXiv:2207.13167*.

[Ton et al., 2024] Ton, V. M., da Silva, N. C., Ruiz, A., Pécora, J. E., Scarpin, C. T., and Bélenger, V. (2024). Real-time management of intra-hospital patient transport requests. *Health Care Management Science*, pages 1–15.

[Tóth et al., 2024] Tóth, M., Hajba, T., and Horváth, A. (2024). Milp models of a patient transportation problem. *Central European Journal of Operations Research*, pages 1–20.

[Toth and Vigo, 1996] Toth, P. and Vigo, D. (1996). Fast local search algorithms for the handicapped persons transportation problem. *Meta-heuristics: theory and applications*, pages 677–690.

[Toth and Vigo, 1997] Toth, P. and Vigo, D. (1997). Heuristic algorithms for the handicapped persons transportation problem. *Transportation science*, 31(1):60–71.

[Tsai et al., 2019] Tsai, M.-H., Pan, C.-S., Wang, C.-W., Chen, J.-M., and Kuo, C.-B. (2019). Rfid medical equipment tracking system based on a location-based service technique. *Journal of Medical and Biological Engineering*, 39:163–169.

[Turan et al., 2011] Turan, B., Schmid, V., and Doerner, K. F. (2011). Models for intra-hospital patient routing. In *3rd IEEE International Symposium on Logistics and Industrial Informatics*, pages 51–60. IEEE.

[Uchimura et al., 2002] Uchimura, K., Takahashi, H., and Saitoh, T. (2002). Demand responsive services in hierarchical public transportation system. *IEEE Transactions on Vehicular Technology*, 51(4):760–766.

[von Elmbach et al., 2019] von Elmbach, A. F., Scholl, A., and Walter, R. (2019). Minimizing the maximal ergonomic burden in intra-hospital patient transportation. *European Journal of Operational Research*, 276(3):840–854.

[Wilson et al., 1971] Wilson, N. H., Sussman, J. M., Wong, H.-K., and Higonnet, T. (1971). *Scheduling algorithms for a dial-a-ride system.* Massachusetts Institute of Technology. Urban Systems Laboratory.

[Wilson and Colvin, 1977] Wilson, N. H. M. and Colvin, N. J. (1977). *Computer control of the Rochester dial-a-ride system.* Number 77. Massachusetts Institute of Technology, Center for Transportation Studies.

[Yao et al., 2010] Yao, W., Chu, C.-H., and Li, Z. (2010). The use of rfid in healthcare: Benefits and barriers. In *2010 IEEE International Conference on RFID-Technology and Applications*, pages 128–134. IEEE.

[Yu et al., 2020] Yu, S., Puchinger, J., and Sun, S. (2020). Two-echelon urban deliveries using autonomous vehicles. *Transportation Research Part E: Logistics and Transportation Review*, 141:102018.

[Zhang et al., 2023] Zhang, Y., Pan, J., Li, L. K., Liu, W., Chen, Z., Liu, X., and Wang, J. (2023). On the properties of kullback-leibler divergence between mul-

tivariate gaussian distributions. *Advances in Neural Information Processing Systems*, 36:58152–58165.

[Zhang et al., 2015] Zhang, Z., Liu, M., and Lim, A. (2015). A memetic algorithm for the patient transportation problem. *Omega*, 54:60–71.

Appendix A

SUMMARY TABLE FOR PREVIOUS STUDIES IN INTRA-HOSPITAL TRANSPORTATION

Table A.1: Previous survey comparison in intra-hospital transportation.

Attribute	[Hanne et al., 2009]	[Kuchera and Rohleder, 2011]	[Bärmann et al., 2024]	[Séguin et al., 2019]	[Beaudry et al., 2010]	[Fiegl and Pontow, 2009]
Dynamic/Static	Dynamic	Static	Dynamic	Static	Static	Dynamic
Assignment (Scheduling)	✓	✓	✓	✓	✓	✓
Routing	✓		✓	✓	✓	
Consolidation	✓				✓	✓
Operational Cost	✓	✓	✓	✓	✓	✓
Fair Request Distribution	✓		✓			
Delay Penalty	✓		✓		✓	✓
Porters (Humans)	✓	✓	✓	✓	✓	✓
Service Robots						
Different Requests (Patients and Items)						
Solution Approach	Load balancing heuristics + Insertion heuristics (FF, BF, BI) + EA	Forecast-based planning + M/M/s + MIP	Dynamic column generation with label-setting (Exact)	Mixed-Integer Programming (Exact)	Two-phase heuristic + Insertion + Tabu Search	Dynamic scheduling with Tabu + Insertion Heuristics

Table A.2: Previous survey comparison in intra-hospital transportation.-*Cont'd*

Attribute	[Sun et al., 2025]	[Kergosien et al., 2011]	[Bozer and Rabbani, 2020]	[Ton et al., 2024]	[Turan et al., 2011]	[Elmbach et al., 2015]
Dynamic/Static	Static	Dynamic	Static	Dynamic	Static	Static
Assignment (Scheduling)	✓	✓	✓	✓	✓	✓
Routing		✓		✓	✓	
Consolidation						
Operational Cost		✓	✓		✓	
Fair Request Distribution						
Delay Penalty	✓		✓	✓	✓	
Porters (Humans)	✓	✓	✓	✓	✓	✓
Service Robots						
Different Requests (Patients and Items)						
Solution Approach	Simulation-based scenario analysis (Heuristic)	Tabu Search with online rescheduling (Metaheuristic)	Simulation-based dispatch policy evaluation (Heuristic)	Constructive + Local Search	Dynamic column generation + label-setting (Exact)	MIP + Heuristics (Exact + Heuristic)

Table A.3: Previous survey comparison in intra-hospital transportation.-*Cont'd*

Attribute	[von Elmbach et al., 2019]	[Hamdi et al., 2024]	[Cheng et al., 2023]	[Fanti et al., 2020]	[Jin et al., 2021]	[Bernhard et al., 2023]
Dynamic/Static	Static	Static	Static	Static	Static	Dynamic
Assignment (Scheduling)	✓	✓	✓	✓	✓	✓
Routing	✓	✓		✓	✓	
Consolidation						
Operational Cost		✓	✓	✓	✓	
Fair Request Distribution	✓	✓				
Delay Penalty		✓	✓			
Porters (Humans)	✓	✓				
Service Robots			✓	✓	✓	✓
Different Requests (Patients and Items)	✓	✓	✓	✓	✓	✓
Solution Approach	Neighborhood Search + Tailored Tabu Search	MILP with CPLEX (Exact)	VNS + chance-constrained stochastic MIP	Integer Linear Programming (Exact)	Genetic Algorithm (Metaheuristic)	Discrete Event Simulation + Cost Analysis

Table A.4: Previous survey comparison in intra-hospital transportation.-*Cont'd*

Attribute	[Azaiez et al., 2022]	[Guzmán Ortiz et al., 2021]
Dynamic/Static	Dynamic	Dynamic
Assignment (Scheduling)	✓	✓
Routing	✓	✓
Consolidation		
Operational Cost	✓	✓
Fair Request Distribution		✓
Delay Penalty		✓
Porters (Humans)		
Service Robots	✓	✓
Different Requests (Patients and Items)		
Solution Approach	Matheuristic (MILP + Dynamic Re-optimization)	Rule-Based Scheduling + Simulation

Appendix B

BASELINE POLICIES

In this appendix, we present a detailed explanation of the baseline policies used for comparison with our proposed learning-based approach. These policies, namely $\pi_{randomized}$, $\pi_{heuristic}$, $\pi_{DQN}^{Deterministic}$, and $\pi_{hospital}$ represent a spectrum of decision-making strategies, ranging from uninformed to heuristically informed, and serve as important reference points in our experimental evaluation. Each policy is applied at every decision epoch t_k, when a new request $j_k \in J$ is revealed to the system and must be assigned to a transporter $v \in V$.

The randomized policy $\pi_{randomized}$ is the simplest among the all and serves as a naive benchmark. Upon the arrival of request j_k at decision epoch k, this policy selects a transporter uniformly at random from the set of available transporters V_{t_k}. This selection process is entirely independent of the system's state, including the current routes, workloads, or locations of the transporters. Formally, the policy samples $\pi_{randomized}(j) \sim U(V_{t_k})$, where U denotes the discrete uniform distribution. Since this policy does not exploit any operational information, it does not aim to optimize efficiency or service quality, but provides a lower-bound performance benchmark, useful for quantifying the value of incorporating even basic system awareness into dispatching decisions.

The heuristic policy $\pi_{heuristic}$ introduces a more sophisticated decision-making framework by incorporating key operational features into a cost-based evaluation. At each decision epoch k, for each available transporter $v \in V_{t_k}$, the policy computes a weighted cost function that captures three dimensions of performance: workload, travel time, and delay penalty as in the main objective function, the current workload $WL_v(t_k)$, the travel time component $TT_v(t_k)$, and the delay penalty $DP_v(t_k)$. In fact, this policy uses the feature set as we consider for training our agent (See Section 6.3.2). These components are then combined using scalar weights w_1, w_2, and w_3,

resulting in a total cost value $C_{v,k} = w_1 \cdot TT_v(t_k) + w_2 \cdot WL_v(t_k) + + w_3 \cdot DP_v(t_k)$. The request is then assigned to the transporter that has minimum total cost, such that $\pi_{heuristic}(j) = \arg\min_{v \in V_{t_k}} C_{v,k}$. This policy, while not adaptive or learning-based, effectively captures the primary objectives of the our problem, balancing efficiency, fairness, and punctuality. It thus offers a strong heuristic benchmark, representing the performance of a dispatcher that makes informed yet immediate decisions based on full knowledge of the current system state.

The Neural Network (NN)-based policy $\pi_{DQN}^{Deterministic}$ serves as a learning-based baseline and is constructed using the same feature set employed for training our proposed agent (see Section 6.3.2). This policy adopts a standard Deep Q-Network (DQN) framework, where the Q-function is approximated by a deterministic Neural Network trained via temporal-difference learning. At each decision epoch k, for each available transporter $v \in V_n$, where $V_n \subset V_{t_k}$ that contains N actions with minimum weighted sum as we have for BNN, the policy estimates the Q-value of assigning request j_k to transporter v using a forward pass through the trained Q-network. The assignment decision is then made by selecting the transporter with the highest predicted Q-value. Unlike our Bayesian approach, this policy does not model uncertainty and thus lacks the capacity to capture predictive variance across state-action pairs. Nevertheless, it provides a useful baseline for evaluating the contribution of uncertainty modeling in decision performance under dynamic and stochastic hospital conditions.

The hospital policy $\pi_{hospital}$ reflects the real-world strategy employed in the hospital's current transportation system and is used as a constructive heuristic in our first benchmark, as defined in Appendix C. However, we make an improvement in the current policy according to preliminary results. In the current system, at each decision epoch k, when a new request $j_k \in J$ arrives, the system identifies the distance of all the transporters to the pickup location s_{j_k}, based on RFID-derived travel time estimates. the system identifies a subset of transporters located within a predefined proximity threshold to the pickup location s_{j_k}, based on RFID-derived travel time estimates. Let $V_{t_k} \subseteq V$ denote the set of transporters who are within this acceptable proximity range to s_{j_k}. Among this set, the dispatcher selects the

transporter $v^* \in V_{t_k}$ with the minimum number of assigned requests. The delay thresholds are defined as 10 minutes for patient transport requests and 15 minutes for non-patient transport requests. However, according to our preliminary results, this strategy although may result in good solution in some case, mainly results in substantially worse solutions. Consequently, we modify the current strategy by considering the distance as penalty to the current strategy, meaning that if a transporter is outside the defined proximity, cost function defined before which is the weighted sum of $TT_v(t_k)$, $WL_v(t_k)$, and $DP_v(t_k)$ is increased by a percentage.

We introduce this modification because solely considering the number of requests assigned to a transporter, without accounting for workload, travel time, and delay penalty, may not fully reflect the operational state necessary for improved decision-making. Moreover, in certain scenarios, assigning a request to a transporter outside of the predefined distance threshold may actually be more beneficial. Our approach is inspired by the proximity-based assignment methodology in [Sun et al., 2025], where the hospital is divided into proximity zones and assignments are adjusted by a percentage if the transporter and request's pickup location are in different zones. In our adaptation, we define these "zones" in terms of delay thresholds and apply a percentage-based penalty to the assignment cost, which itself is the weighted sum of workload, travel time, and delay penalty, if the transporter is outside the acceptable proximity range. This improvement introduces a trade-off between the transporter's spatial distance and its current operational state, promoting a more balanced and context-aware assignment policy.

Appendix C

BENCHMARK

For our benchmark, we investigate methodologies that contrary to our solution approach, simultaneously tackles the assignment and routing problems in the hospital environment, which is the main solution approach in the studies for intra-hospital transportation [Ton et al., 2024, Bärmann et al., 2024, Fiegl and Pontow, 2009, Beaudry et al., 2010, Kergosien et al., 2011, Molenbruch et al., 2017b].

The Dial-A-Ride Problem (DARP) is a generalization of the Traveling Salesman Problem (TSP) and is classified as NP-hard. As a result, our problem, which is described within the DARP framework, possesses this complexity and is hence NP-hard. Considering this classification, it is essential to note that resolving such problems in polynomial time by traditional optimization approaches, including branch-and-bound techniques, is unachievable. Consequently, metaheuristic algorithms are crucial for identifying near-optimal solutions within a reasonable time. Therefore, to create a strong bencmark, we implement a Adaptive Tabu Search (ATS) algorithm, which has demonstrated its effectiveness in addressing DARPs, and for this reasons, many papers addressing the DARP uses TS [Ho et al., 2018].

The algorithm begins by the initial solution found by $\pi_{hospital}$. In the static version, an initial complete solution is first constructed using the policy, after which the Adaptive Tabu Search is applied to improve it once all requests have been scheduled. In contrast, in the dynamic version, the policy is used to insert each newly arriving request into the current solution, followed immediately by a Adaptive Tabu Search to refine the updated solution.

The improvement process begins with a shaking phase, where a 2-Exchange move is applied to perturb the current solution and help escape local minima. During the shaking phase, the algorithm performs a 2-Exchange move, where requests between transporters are exchanged to disrupt local optimality and encourage exploration

of new solutions. In order to position the exchanged request in the best position in the routes, we use Algorithm 1. Following the implementation of 2-Exchange move, a local search is triggered. The local search phase utilizes four operators adapted from [Braekers et al., 2014]: Relocate and Exchange inter-route operators, the Swap intra-route operator, and a route Eliminate operator. These operators are selected for their computational efficiency, enabling thorough exploration within limited computational timeframes. As explained by [Braekers et al., 2014], although several advanced operators have been introduced, these operators are computationally affordable, enabling a significant number of iterations within a restricted amount of computation time. As a result, given the need for rapid decision-making in a real-time setting of our problem, we employ such operators to enable extensive searching through a large number of iterations in a limited amount of computational time.

The Relocate operator selectively removes a request from one transporter's route and attempts to reinsert it into the most advantageous position within another transporter's route using Algorithm 1. Formally, let $F_{v,k} \subseteq J_{v,k}$ be the subset of requests that have already been initiated and thus are fixed at decision epoch k. The operator is applied on each node $j \in J_{v,k} \setminus F_{v,k}$ of a single randomly selected route $\Gamma_{v,k}$. For each such request, the algorithm evaluates all other transporters $v' \in V \setminus \{v\}$ and computes the cost of inserting j into $\Gamma_{v',k}$, denoted as $\Delta C_{jv'}$. The insertion is only accepted if it maintains feasibility. The request j is then reassigned to the transporter and insertion position that yields the minimum increase in cost:

$$v^* = \arg \min_{v' \in V \setminus \{v\}} \{\Delta C_{jv'} : \text{insertion of } j \text{ into } \Gamma_{v'} \text{ is feasible}\}.$$

The *Exchange* operator, on the other hand, alternates between exchanging two requests from different routes or exchanging the transporters of two routes. Each implementation begins by randomly selecting a transporter $v \in V$. In the first variant, the algorithm considers exchanging a request $j \in J_{v,k} \setminus F_{v,k}$ with a request $j' \in J_{v',k} \setminus F_{v',k}$ from another transporter $v' \in V \setminus \{v\}$. For each feasible pair (j, j'), the algorithm evaluates the cost of mutual exchange and selects the pair that

minimizes the total objective cost:

$$(j^*, j'^*, v') = \arg \min_{\substack{j \in J_{v,k} \setminus F_{v,k} \\ j' \in J_{v',k} \setminus F_{v',k} \\ v' \in V \setminus \{v\}}} \Big\{ C\big(\Gamma_{v,k} \setminus \{s_j, d_j^1, d_j^2\} \cup \{s_{j'}, d_{j'}^1, d_{j'}^2\}, \\ \Gamma_{v',k} \setminus \{s_{j'}, d_{j'}^1, d_{j'}^2\} \cup \{s_j, d_j^1, d_j^2\}\big) \Big\} \tag{C.1}$$

In the second variant, the entire route $\Gamma_{v,k}$ is exchanged with $\Gamma_{v',k}$, again provided that this exchange is feasible with respect to fixed requests, robot/patient compatibility, and other operational constraints.

Finally, the Eliminate operator aims to decrease the number of transporters in use. It operates by randomly selecting a transporter $v \in V$, removing all allowed requests from its route, and then reinserting these requests across other routes in random order to the most suitable positions in the best routes. Let $J_{elim} = J_{v,k} \setminus F_{v,k}$ denote the set of removable requests. For each $j \in J_{elim}$, the algorithm identifies a transporter $v' \in V \setminus \{v\}$ that can accommodate the insertion of j, and assigns:

$$v_j^* = \arg \min_{v' \in V \setminus \{v\}} \{\Delta C_{jv'} : \text{insertion of } j \text{ into } \Gamma_{v',k} \text{ is feasible}\}.$$

If all requests in J_{elim} are successfully reassigned without violating feasibility, then $\Gamma_{v,k}$ is eliminated, thus reducing fleet size.

The Swap operator is similar to the operator defined in Phase 2 of our solution approach. It performs intra-route refinement by selecting pairs of requests within a single route and evaluating whether exchanging their positions improves the solution. Specifically, for each transporter $v \in V$, and for every unordered pair $(n_1, n_2) \in \Gamma_{v,k} \setminus \bigcup_{j \in F_{v,k}} \{s_j, d_j^1, d_j^2\}$, the operator swaps their positions in $\Gamma_{v,k}$ if the modified route $\Gamma'_{v,k}$ is feasible and yields a lower cost:

$$\text{If } C(\Gamma'_{v,k}) < C\Gamma_{v,k}), \text{ then apply } (n_1 \leftrightarrow n_2).$$

During each iteration, the algorithm maintains a current solution and generates a candidate solution by first applying a shaking phase, followed by one of the three primary neighborhood operators, Relocate, Exchange, or Eliminate, selected stochastically. The resulting solution is further refined using the Swap operator,

which performs intra-route adjustments to improve local structure. Each candidate solution is evaluated based on the weighted objective function $C(S)$, composed of total travel time, workload imbalance, and delay penalties, as defined in Equation 4.50. If the candidate solution is either better than the best-known solution or not forbidden by the tabu list, it is accepted as the new current solution. The tabu list, which stores recently applied moves, is dynamically updated to avoid cycling and diversify the search.

If the new solution S' yields a lower cost than the best-known solution S^{best}, it is adopted as the new best: $S^{\text{best}} \leftarrow S'$. The tabu list size is also adaptively adjusted, shrinking when improvements occur to encourage intensification, and expanding when stagnation is detected to promote diversification. This iterative process continues until a maximum number of iterations L_{max} is reached or no improvements are observed over a predefined number of iterations $L_{\text{no\_impr}}$. The complete description of the Adaptive Tabu Search algorithm for the assignment and routing problem is presented in Algorithm 4.

Table C.1: Summary statistics (min, max, mean) for Objective Difference (%) and Runtime Ratio of ATS across different numbers of requests.

Requests	**Obj % Diff.**			**Runtime Ratio**		
	Min	**Max**	**Mean**	**Min**	**Max**	**Mean**
8	-6.30	12.88	5.58	24.85	136.36	86.95
10	-9.44	9.15	1.46	13.99	60.17	40.30
12	-29.42	0.49	-12.35	7.82	30.14	19.65

We evaluate the effectiveness of the Adaptive Tabu Search (ATS) against our MILP model with 1 hour on time limitation by performing tests on 10 distinct test cases for each request size using the instance set $\mathcal{I}^{HP}$. Each instance was outfitted with a varied fleet of three transporters, comprising two porters and one service robot. Table C.1 presents the summary statistics of this comparison, specifying the minimum, maximum, and average objective percentage differences as well as the runtime ratios between ATS and MILP-TL. A positive objective difference indicates

Algorithm 4 Adaptive Tabu Search - Assignment and Routing Problem

Input: $distance\_matrix$, $service\_times$, $robots\_list$, $fixed\_nodes$, $initial\_solution$, $initial\_cost$, $depot$, $transporter$, max_iter, $min\_tabu\_list\_size$, $max\_tabu\_list\_size$, $max\_non\_improvement$.

Output: $best\_solution, best\_cost$

Initialize $tabu\_list \leftarrow []$
$current\_solution \leftarrow initial\_solution$
$best\_solution \leftarrow initial\_solution$
$best\_cost \leftarrow initial\_cost$
$tabu\_size \leftarrow min\_tabu\_list\_size$
$iter \leftarrow 0$
$no\_improvement\_count \leftarrow 0$
while $iter \leq max\_iter$ **do**
    $iter \leftarrow iter + 1$
    $improvement\_found \leftarrow False$
    Randomly choose one of the local search operators (relocate, exchange, eliminate)
    $new\_solution \leftarrow ApplyOperator(current\_solution)$
    Perform *Swap* operator for all routes in $new\_solution$
    $new\_solution \leftarrow GetRoutesWithRechargingStations(new\_solution)$
    $current\_cost \leftarrow compute\_total\_cost(new\_solution)$
    $current\_solution \leftarrow new\_solution$
    $move \leftarrow (operator\_type, new\_solution)$
    if $current\_cost < best\_cost$ **or** $move \notin tabu\_list$ **then**
        $tabu\_list$.append($move$)
        if len($tabu\_list$) $> tabu\_size$ **then**
            Remove the oldest move from $tabu\_list$
        end if
        if $current\_cost < best\_cost$ **then**
            $best\_solution \leftarrow current\_solution$
            $best\_cost \leftarrow current\_cost$
            $tabu\_size \leftarrow \max(min\_tabu\_list\_size, tabu\_size - 1)$
            $improvement\_found \leftarrow True$
            $no\_improvement\_count \leftarrow 0$
        else
            $tabu\_size \leftarrow \min(max\_tabu\_list\_size, tabu\_size + 1)$
        end if
    end if
    if not $improvement\_found$ **then**
        $no\_improvement\_count \leftarrow no\_improvement\_count + 1$
    end if
    if $no\_improvement\_count \geq max\_non\_improvement$ **then**
        break
    end if
end while
return $best\_solution, best\_cost$

that ATS underperformed compared to MILP-TL, whilst a negative value signifies superior performance by ATS.

For the smaller 8-request instances, ATS demonstrates a mean objective increase of 5.58%, indicating marginally worse solutions compared to MILP-TL. Nonetheless, this disparity is mitigated by a significant decrease in runtime, with ATS operating on average 87 times more swiftly. In the cohort of 10 requests, ATS yields an average objective of +1.46% compare to MILP-TL, along with notable computational efficiencies, approximately 40 times faster.

Notably, when the problem size escalates to 12 requests, ATS surpasses the MILP-TL model, attaining an average objective reduction of –12.35%, with peak improvements reaching –29.42%. This transition results from MILP solvers' failure to obtain high-quality solutions within the given time frame, whereas ATS consistently navigates and exploits the solution space efficiently. The software preserves its computational efficiency, achieving an average runtime ratio of around 20, even under increasingly complex situations.

The results highlight the effectiveness and scalability of the suggested ATS metaheuristic. Although MILP-TL may produce slightly superior solutions for smaller problem sizes, its effectiveness significantly declines as the instance size escalates due to temporal constraints. Conversely, ATS consistently provides superior solutions across all evaluated instance sizes, frequently surpassing MILP-TL in bigger cases while necessitating considerably less computational time. This illustrates the resilience of ATS in managing heightened problem complexity and its appropriateness for operational contexts when computational resources are constrained or swift decision-making is critical.

Appendix D

SCATTER AND BOX PLOTS FOR DIFFERENT FLEET CONFIGURATIONS

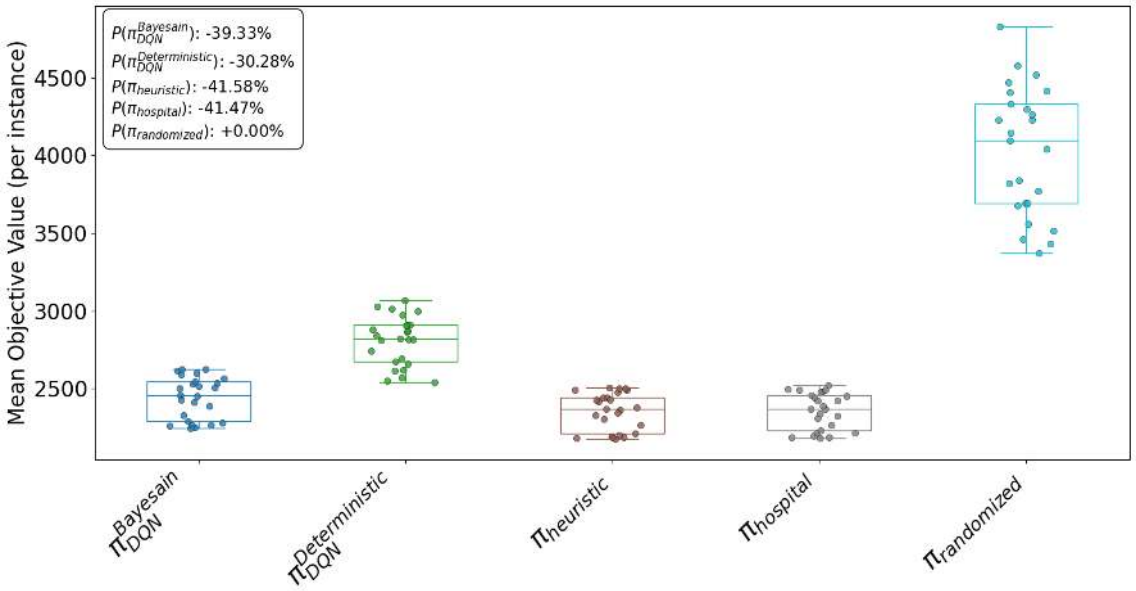


Figure D.1: Scatter and box plots for 25 test instances across various policies, and 30 porters and 0 service robots.

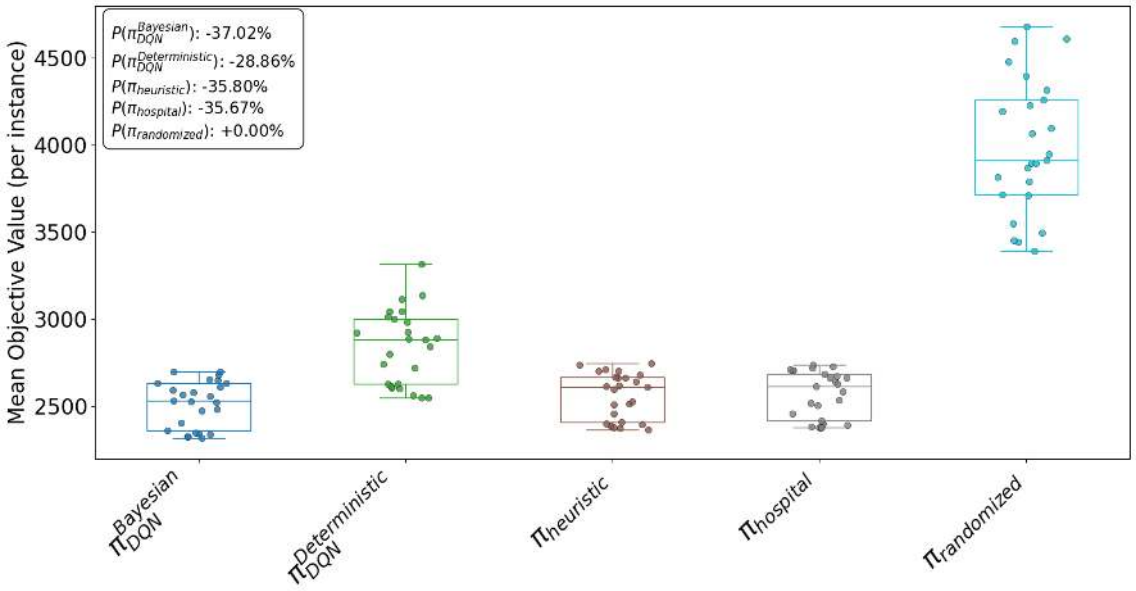


Figure D.2: Scatter and box plots for 25 test instances across various policies, and 30 porters and 5 service robots.

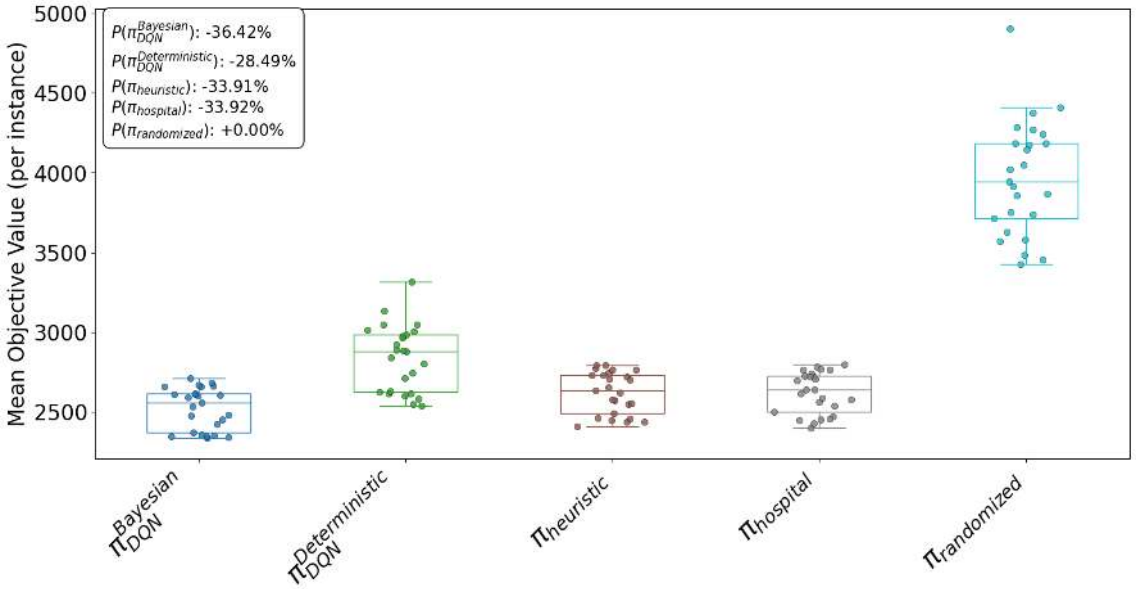


Figure D.3: Scatter and box plots for 25 test instances across various policies, and 30 porters and 10 service robots.

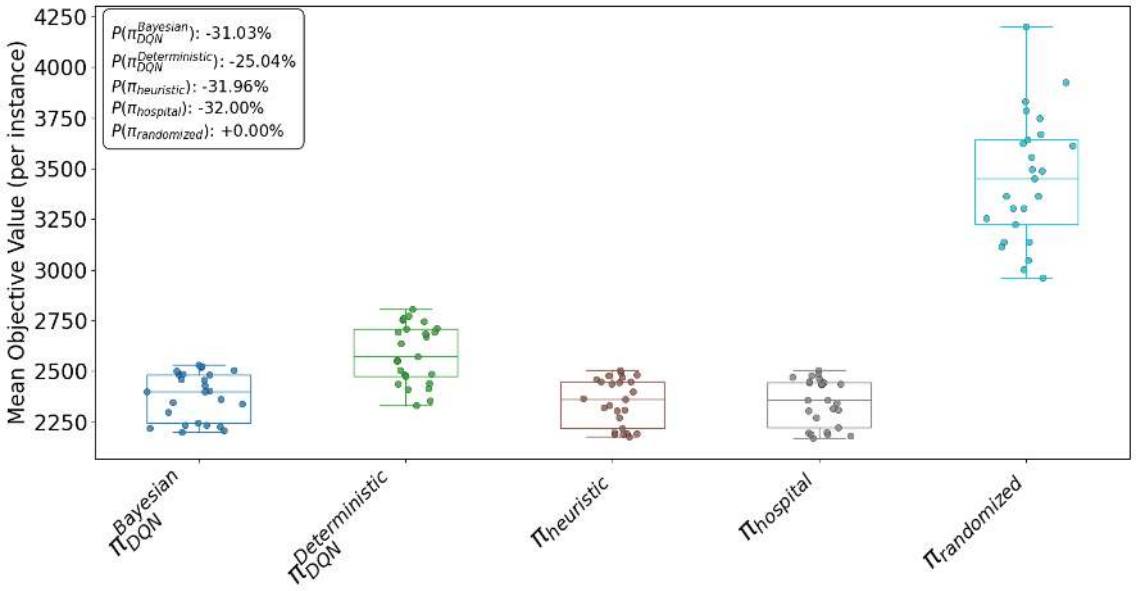


Figure D.4: Scatter and box plots for 25 test instances across various policies, and 40 porters and 0 service robots.

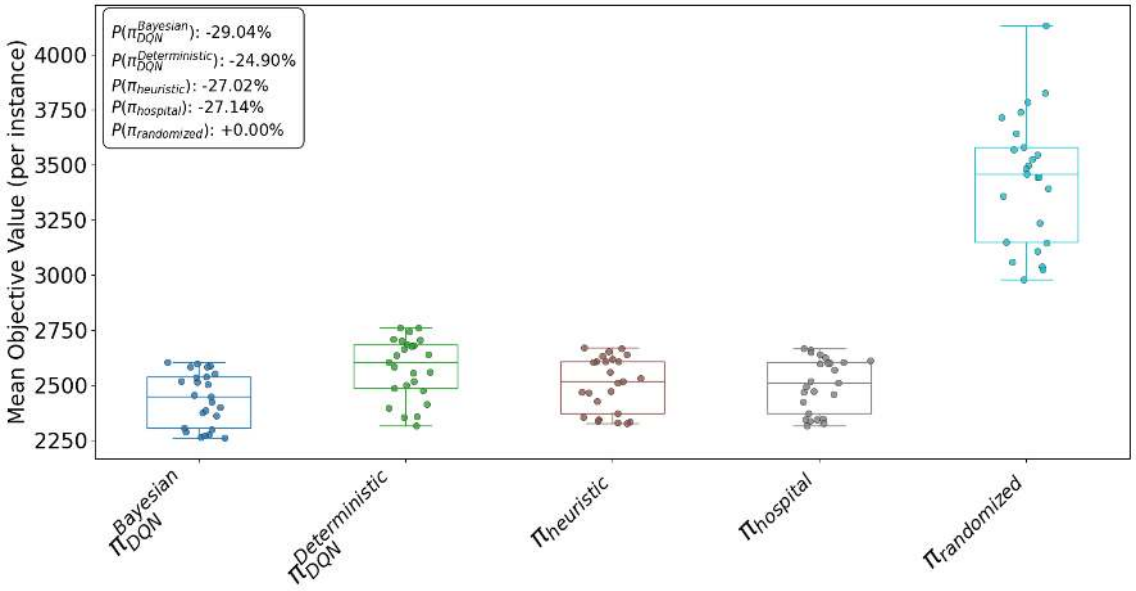


Figure D.5: Scatter and box plots for 25 test instances across various policies, and 40 porters and 5 service robots.

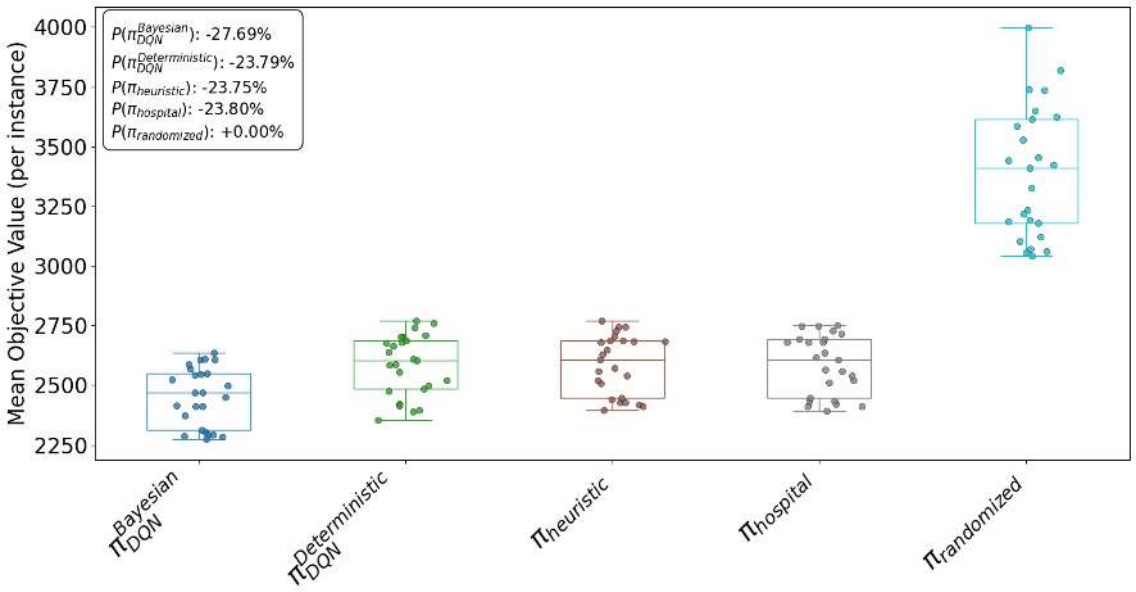


Figure D.6: Scatter and box plots for 25 test instances across various policies, and 40 porters and 10 service robots.

Appendix E

SCATTER AND BOX PLOTS FOR DIFFERENT SERVICE ROBOTS SPEED VARIATION FACTORS

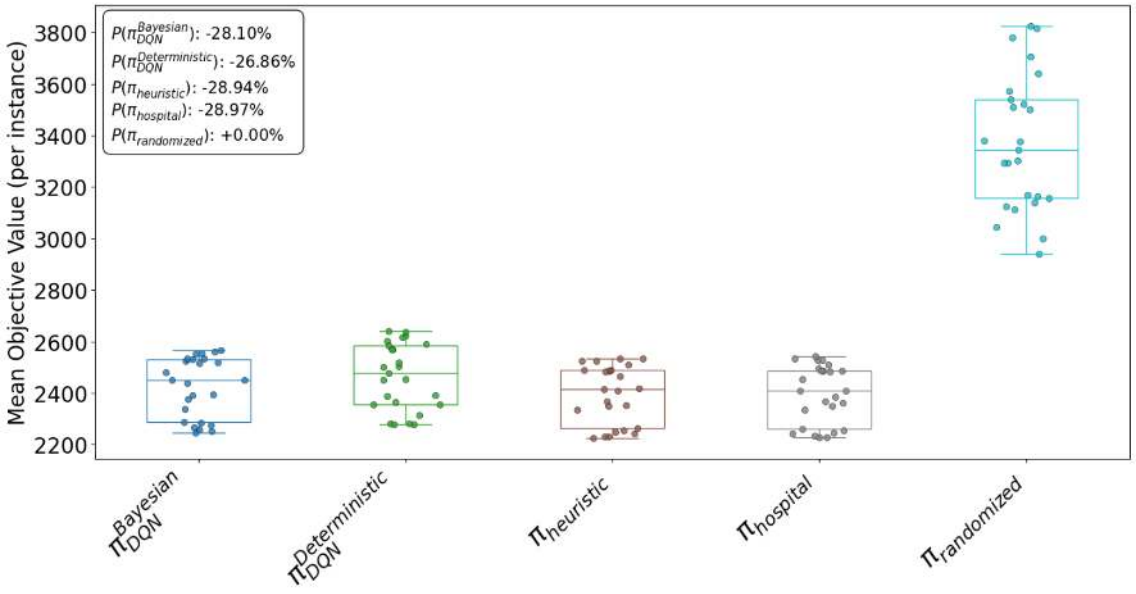


Figure E.1: Scatter and box plots for 25 test instances across various policies, and speed variation factor 1.

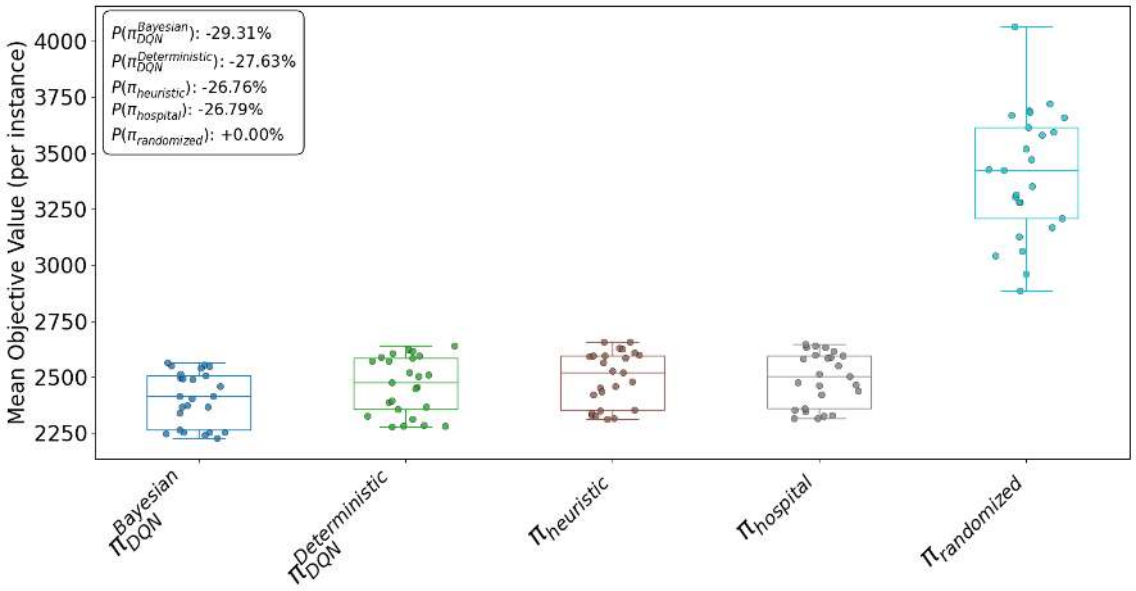


Figure E.2: Scatter and box plots for 25 test instances across various policies, and speed variation factor 1.25.

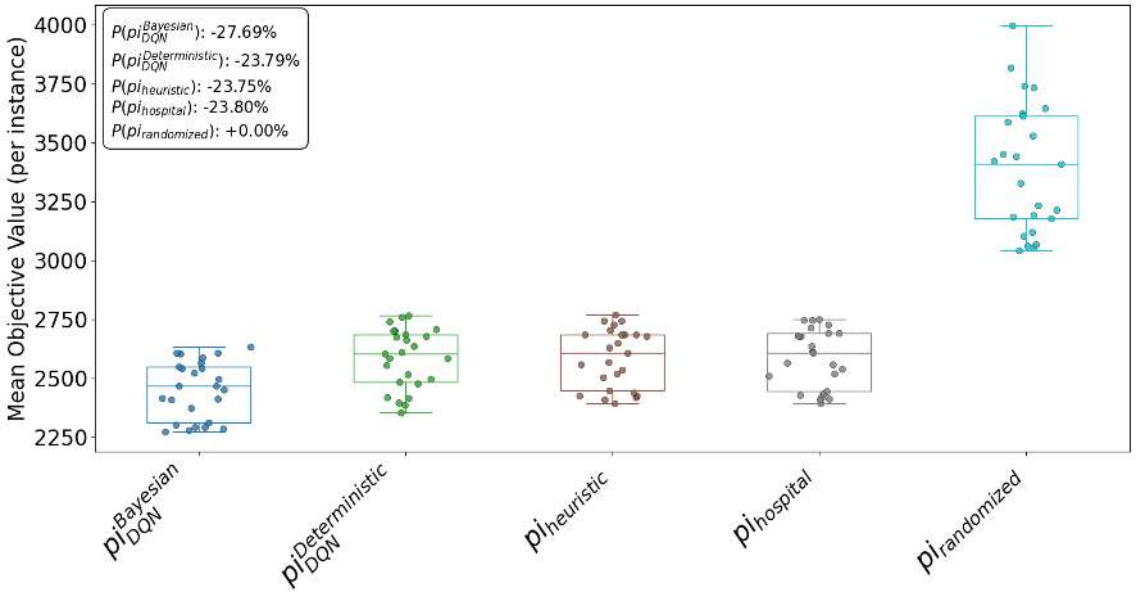


Figure E.3: Scatter and box plots for 25 test instances across various policies, and speed variation factor 1.5.

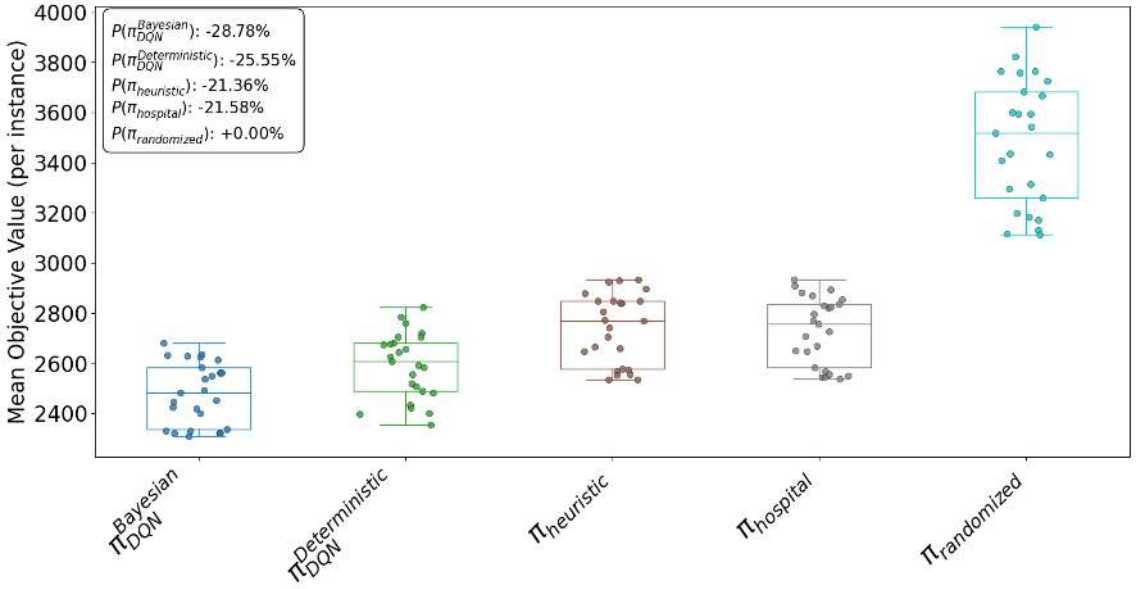


Figure E.4: Scatter and box plots for 25 test instances across various policies, and speed variation factor 2.

Appendix F

NOTATION SUMMARY

Table F.1: Table of notations for sets.

Notation	**Explanation**
J	Set of transportation requests.
P	Set of pickup points.
D	Set of delivery nodes.
D_1	Set of first delivery points.
D_2	Set of second delivery points.
N	Set of all nodes.
A	Set of all arcs.
Q	Set of all depots.
RS	Set of recharging stations.
V	Set of all transporters.
V_1	Set of all group one porters.
V_2	Set of all group two porters.
R	Set of all service robots.
R	Set of all service robots.
J_i	Set of requests type i.
K	Set of capacity types.

Table F.2: Table of notations for parameters.

Notation	**Explanation**
s_j	Pickup point of request j.
d_j^1	First delivery point point of request j.
d_j^2	Second delivery point of request j.
c_j	Type of request j.
i_j	Priority type of request j.
k_j	Capacity type of request j.
t_j	Arrival time of request j.
b_j	The duration of time for request j after which the penalty begins to be counted.
β_j	the penalty for each time unit of delay for request j.
α	Increase rate for slop of the penalty function.
τ	The break-point where the slope of the linear function increases.
BL	The base level for service robots recharging policy driven from (r, Q) inventory management policy.
RL	The recharge level for service robots recharging policy driven from (r, Q) inventory management policy.
T	Length of planning horizon.
d_{mn}	Travel time between nodes m and n.
q_{mk}	Weight type k associated with node m.
γ_m	Difficulty score of node m.
c_k	Capacity type k of transporters.
s_m	Service time associated with node m.
c_{mnkv}	Capacity-related parameter equal to $\min\{c_k, c_k + q_{mk}\}$, where c_k is the capacity of type k for transporter v.
a_{mj}^o	Binary parameter equal to 1 if node $m \in N$ is the pickup point ($o = 0$), first delivery point ($o = 1$), or second delivery point ($o = 2$) of request $j \in J$; 0 otherwise.
d_{mn}	Travel time between nodes $m \in N$ and $n \in N$.
q_{mk}	Required capacity of type $k \in K$ at node $m \in N$.
s_m	Service time at node $m \in N$.
w_1, w_2, w_3	Weights associated with each term in the objective function.
t_j	Arrival time of request $j \in J$.
b_j	Time after which the delay penalty begins for request $j \in J$.
T	End of the planning horizon.

Table F.3: Table of notations for parameters.-*Cont'd*

Notation	**Explanation**
β	Penalty coefficient for delay in completing request j.
γ_m	Score assigned to node $m \in N$.
M	A sufficiently large constant used in constraint formulations.
l	Maximum allowed ride time.
f_{jv}	Binary parameter equal to 1 if request $j \in J$ can be handled by transporter $v \in V$; 0 otherwise.
r	Discharging rate of battery per unit of time or distance.
ρ	Fixed recharging amount at each recharging visit.
λ	Recharging rate of the battery during charging.
h_{max}	Maximum battery capacity of each service robot.
π	Minimum allowed charge level (safety threshold).

Table F.4: Table of notations for variables.

Notation	**Explanation**
L_{jv}	Ride time of request $j \in J$ when handled by porter $v \in V$.
C_v	Equal to 1 if porter $v \in V$ is assigned to at least one request; 0 otherwise.
W_{nvk}	Load of type $k \in K$ carried by porter $v \in V$ upon arriving at node $n \in N$.
A_{nv}	Arrival time of porter $v \in V$ to node $n \in N$.
U_{nv}	MTZ (Miller–Tucker–Zemlin) subtour elimination variable for node $n \in N$ and porter $v \in V$.
E_j^1, E_j^2	Delay values associated with the start time of request $j \in J$.
B^1	Maximum total working time among all porters.
B^2	Minimum total working time among all porters.
F_{mnlv}	Binary parameter equal to 1 if service robot $v \in R$ visits recharging station $l \in RS$ between its path from node $m \in N$ to node $n \in N$.
X_{mnv}	Equal to 1 if node $n \in N$ is directly or indirectly visited after node $m \in N$ by transporter $v \in V$; 0 otherwise.
H_{nv}	Battery level of service robot $v \in R$ upon leaving node $n \in N$.
F_{mnlv}	Binary parameter equal to 1 if service robot $v \in R$ visits recharging station $l \in RS$ between its path from node $m \in N$ to node $n \in N$.
X_{mnv}	Equal to 1 if node $n \in N$ is directly or indirectly visited after node $m \in N$ by transporter $v \in V$; 0 otherwise.
H_{nv}	Battery level of service robot $v \in R$ upon leaving node $n \in N$.

Table F.5: Table of notations (others)

Notation	Explanation
T_{action}	Set of decision epoch.
$J_k = (J_{P,k}, J_{R,k})$	Set of assigned transportation requests at decision epoch k.
$\Gamma_k = (\Gamma_{R,k}, \Gamma_{R,k})$	Set of planned routes at decision epoch k.
$X(S_k)$	Set of possible actions at decision epoch k.
$\Gamma^x_{R,k} = (\Gamma^x_{P,k}, \Gamma^x_{R,k})$	Set of updated planned routes of service robots at decision epoch k.
Π	Set of all policies at all decision epochs (Strategy).
J_v	Set of requests assigned to transporter v.
S	Solution.
$\mathcal{I}^{PP}_{\lambda=1/10}$	Instance set generated using Poison Process with $\lambda = 1/10$.
$\mathcal{I}^{HP}$	Instance set generated using hospital data.
$P(\pi_a)$	Percentage improvement of policy π_a over policy $\pi_{randomized}$ outcomes.
$\mathbb{1}_{\{\neg \text{Done}_{k+n}\}}$	Indicator function that disables bootstrapping if the episode terminates before step $k+n$.
j_k	Newly arrived request at decision epoch k.
S_k	System state at decision epoch k.
$R(S_k, x_k)$	Reward of an action x_k at decision epoch k.
$W_{k+1} = \{j_{k+1}, t_{k+1}\}$	Transformation from the current state to the post-decision state.
S^x_k	Post-decision state.
π	Policy.
π^*	Optimal policy.
$C(S)$	The objective function value of solution S.
h	Neighborhoods size in AVNS.
x_k	Action taken at decision epoch k.
$X^\pi(S_k)$	Action selected by policy π at state S_K.
$V(S_k)$	State value function; expected cumulative reward starting from state S_k under policy π.
$Q_\theta(s_k, x_k)$	Parameterized action value function; estimates the expected cumulative reward starting from state s_k, taking action x_k, and following a policy represented by parameters θ thereafter.
N_{nodes}	Number of neurons in the Neural Network.
N_{layers}	Number of hidden layers in the Neural Network.
$N_{episodes}$	Number of episodes in the Neural Network.
N_{robots}	Number of service robots.
w	Weight in the Neural Network.
b	Bias in the Neural Network.

Table F.6: Table of notations (others).-*Cont'd*

Notation	**Explanation**
q	Posterior distribution in the Neural Network.
μ	Mean in the Neural Network.
σ	Standard deviation in the Neural Network.
ρ	Unfixed learnable parameter in the Neural Network.
$TT_v(t_k)$	The total travel time of transporter v at time t_k.
$WL_v(t_k)$	The workload of transporter v at time t_k.
$DP_v(t_k)$	The delay penalty of transporter v at time t_k.
B	Minimatch.
$\mathcal{L}(\theta)$	Kullback-Leibler (KL) divergence for parameters θ.
λ_k	KL weight.
y_i	Target Q-value with parameter θ^- for the i-th transition.
$\mathcal{M}$	Reply buffer.
N_{queue}	Number of requests in the queue.
t_{reopt}	Threshold for the time has elapsed since the last optimization.
t_{elapse}	The elapsed time between two consecutive reoptimization points.