FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Crane-operated warehouses and e-commerce deliveries: models and algorithms for theory and practice

Sam Heshmati

Programa Doutoral em Engenharia e Gestão Industrial

Supervisors:
Prof. Maria Antónia Carravilla
Prof. José Fernando Oliveira
Prof. Greet Vanden Berghe

September 6, 2020

Crane-operated warehouses and e-commerce deliveries: models and algorithms for theory and practice

Sam Heshmati

Programa Doutoral em Engenharia e Gestão Industrial

September 6, 2020

"Yesterday I was clever, so I wanted to change the world.
Today I am wise, so I am changing myself."

Jalāl ad-Dīn Muhammad Rūmī

Acknowledgments

First and foremost, I would like to express my deepest gratitude and warmest thanks to my advisers, Prof. Maria Antónia Carravilla, Prof. José Fernando Oliveira and Prof. Greet Vanden Berghe who have supported me throughout this journey with their patience and knowledge. It was an honor to have your support and guidance. I am deeply indebted to Maria Antónia and José Fernando who provided me with the opportunity to start my PhD at FEUP. I cannot begin to express my thanks to Greet, for believing in me and giving me the opportunity to work with her amazing research group at KU Leuven. Thank you so much, Maria Antónia, José Fernando and Greet! Indeed, without your valuable help and contributions, this journey would not have been completed.

I was blessed with a friendly and cheerful colleagues both at Porto (INESC TEC) and Gent (CODeS). I really enjoyed the time we spent together both in the office and outside. I wish to say thank you specially to: Teresa, Luis, Pedro, Gonçalo, Elsa, Dianna, Beatriz and Marçio at Porto. I am also grateful to my friends and colleagues at CODeS: Tulio, Luke, Wim, Tony, Catherine, Thomas, Evert-Jan, David, Thomas Sys, Eline, Jan, Jannes, Federico, Everton, Bert and Pieter. You all have made INESC TEC and CODeS a very enjoyable working place. I also would like to thank my friends outside the research area, specially: Filipe, Alex, Thais, Claudia, Joana, Reza and Zeinab.

I had the privilege of working on multiple problems, which provided me with the great pleasure of working or having enlightening discussions with different researchers and practitioners. Therefore, I would like to thank Rosaldo Rossetti, Zafeiris Kokkinogenis, Joris Kinable and Jurgen De Zaeytijd.

The completion of my dissertation would not have been possible without the support and nurturing of my beloved parents and brother. Thank you mom and dad for your unconditional support through out my life. I know you have given up many things for me to live my dreams. Thank you Reza for never letting me down.

My final words goes to love of my life, Hana, who stood up by me through out all my travails, impatience and my absences. Azizam, without your love and extreme support I could never reach to the end. I owe you everything!

Sam Heshmati

Abstract

Given the globalization trend and the continuous growth of international trade, logistics management has firmly established as a critical business concern. An efficient logistics management results in a significant reduction in operations cost and environmental impacts and eventually improvement of competitive advantage and companies' profitability. Efficient warehousing and transportation systems provide substantial economic and service benefits to both businesses and their end users. This dissertation discusses theory and practice of these logistic management challenges from two perspectives: introducing integrated optimization, and incorporating ecological concerns into problem formulation, thereby, making scientific and industrial contributions.

The state-of-the-arts in warehouse management traditionally focused on either location assignment or crane scheduling assuming the location assignments are given, while considering some simplified assumptions. Therefore, there is a gap in aspects of exploiting an integrated optimization approach. By studying crane scheduling operations in rail-rail transshipment terminals, an approach to integrate container location assignment and crane scheduling is established. The new insights obtained by such approach identified how integrated decision making may benefit theory and practice. Moreover, realistic constraints were incorporated in the scheduling crane operations problem in rail-rail transshipment terminals. The results show potential gains obtained by implementing the proposed algorithm compared to typical dispatching rules in rail-rail transshipment terminals.

The impacts of integrating the location assignment and crane scheduling is studied by introducing a new problem, crane-operated warehouse scheduling problem (CWSP) and comparing the proposed integrated approach to a sequential approach. First, a general model for the CWSP is introduced, which may be easily adapted to other warehouses, land-side container terminals or any other industry employing multiple gantry cranes for product handling. Second, an efficient algorithm is introduced to solve real-world sized instances. Computational experiments illustrate significant improvements are attainable when integrating location assignment and crane scheduling. The effectiveness and versatility of the proposed approaches are illustrated by extensive experiments upon sets of instances inspired by real-world platforms such as a container terminal and a crane-operated warehouse.

Finally, the scope of this dissertation is extended to include green operations in transportation and particularly in e-commerce delivery. The outcome of this research enables finding the most profitable mode of operation while simultaneously limiting greenhouse emissions on realistic problem instances. E-commerce delivery problem is introduced and solution approaches are proposed to handle different ranges of instance size. The contributions of this research enable practitioners to organize delivery from distribution center to the customer in an ecological manner. In addition, simulating alternative delivery policies provided new insights about different delivery time windows, collection point network and bundle deliveries. All instances and results are publicly available to enable validation and encourage future research.

Resumo

Dada a tendência da globalização e o crescimento contínuo do comércio internacional, a gestão da logística estabeleceu-se firmemente como uma preocupação crítica dos negócios. Uma gestão da logística eficiente resulta numa redução significativa do custo das operações, do impacto ambiental e, eventualmente, numa maior competitividade e lucratividade das empresas. Os sistemas eficientes de armazenamento e transporte fornecem benefícios econômicos e de serviço substanciais para as empresas e seus utilizadores finais. Esta dissertação discute a teoria e a prática destes desafios da gestão logística sobre duas perspetivas: introdução da otimização integrada e incorporação de preocupações ecológicas na formulação dos problemas, deixando assim, contribuições para a ciência e indústria.

O estado da arte sobre a gestão de armazéns foca tradicionalmente ou na atribuição de locais ou no agendamento de guindastes, assumindo que as atribuições de locais eram dadas, considerando algumas suposições simplificadas. Portanto, há uma lacuna na exploração de uma abordagem de otimização integrada. Ao estudar as operações de agendamento de guindastes nos terminais de transporte ferroviário, é estabelecida uma abordagem para integrar a atribuição da localização dos contentores e respetivo agendamento de guindastes. As novas ideias obtidas por essa abordagem identificaram como a tomada de decisão integrada pode beneficiar a teoria e a prática. Além disso, restrições realistas foram incorporadas no problema do agendamento na operação dos guindastes entre terminais de transbordo ferroviário. Os resultados mostram os potenciais ganhos obtidos com a implementação do algoritmo proposto em comparação com as regras típicas no despacho de terminais de transbordo ferroviário.

Os impactos da integração da atribuição de local e agendamento de guindaste são estudados através da introdução de um novo problema, o problema de agendamento em armazéns operados por guindaste (CWSP - crane-operated warehouse scheduling problem) e comparando a abordagem integrada proposta com uma abordagem sequencial. Primeiro, é introduzido um modelo geral para o CWSP, que pode ser facilmente adaptado a outros armazéns, terminais de contentores terrestres ou qualquer outra indústria que utilize vários guindastes de pórtico para a manipulação de produtos. Segundo, é introduzido um algoritmo eficiente para resolver instâncias de tamanho real. Simulações computacionais ilustram as melhorias significativas possíveis ao integrar a atribuição de local e o agendamento de guindastes. A eficácia e versatilidade das abordagens propostas são ilustradas por extensas simulações em conjuntos de instâncias inspiradas em dados reais de plataformas, como um terminal de contentores e um armazém operado por guindaste.

Finalmente, o objetivo desta dissertação é estendido para incluir operações ecológicas no transporte e, particularmente, na entrega de comércio eletrônico. O resultado desta investigação permite encontrar o modo de operação mais lucrativo e, ao mesmo tempo, limitar a emissão de gases com efeito de estufa em problemas reais. O problema de entrega de comércio eletrônico é introduzido e abordagens de solução são propostas para lidar com instâncias de diferentes tamanhos. As contribuições desta investigação permitem que os profissionais organizem a

entrega desde o centro de distribuição até ao cliente de uma maneira ecológica. Além disso, a simulação de políticas de entrega alternativas forneceu novas ideias sobre diferentes janelas de tempo de entrega, rede de pontos de coleta e entregas de pacotes. Todas as instâncias e resultados estão disponíveis ao público para permitir a validação e incentivar futuras investigações.

Contents

List of Figures

List of Tables

Abbreviations

1RPP	One Route Per Parcel
2NG	$2n$ Genes
ALNS	Adaptive Large Neighborhood Search
BOP	Best Order Permutation
BRKGA	Biased Random Key Genetic Algorithm
CD	Consumer Direct
CI	Cheapest Insertion
CNG	Compressed Natural Gas
CPU	Central Processing Unit
CR	Crane Re-assignment
CSP	Crane Scheduling Problem
CWSP	Crane-operated Warehouse Scheduling Problem
DR	Dispatching Rules
E-VRPTW	Electrical Vehicle Routing Problem with Time Window
E-VRPTWMF	Vehicle Routing Problem with Time Window Mixed-Fleet
ECV	Electric Commercial Vehicle
EDP	E-commerce Delivery Problem
GLA	Greedy Location Assignment
ICCV	Internal Combustion Commercial Vehicle
KPI	Key Performance Indicator
LAHC	Late Acceptance Hill Climbing
LAP	Location Assignment Problem
LR	Location Re-assignment
LS	Location Swap
MBOP	Moving Best Order Permutation
MDVRP	Multi-Depot Vehicle Routing Problem
MF	Mixed Fleet
MIP	Mixed Integer Programming
NGH	n Genes Heuristic
NLA	Nearest Location Assignment
OS	Order Swap
PRP	Pollution Routing Problem
R&R	Ruin and Recreate
RBI	Random Best Insertion
RI	Random Insertion
RMGC	Rail-Mounted Gantry Crane
RRTT	Rail-Rail Transshipment Terminal
RTGC	Rubber-Tired Gantry Cranes

SCOP	Scheduling Crane Operations Problem
SNC	Size of Non-allocated Containers
SNCF	Société Nationale des Chemins de fer Français (French national railway company)
SOA	Size of Overlap Area
STP	Scheduling the Train Process
TDVRP-PF	Time-Dependent Vehicle Routing Problem with Path Flexibility
TS	Transshipment Sequence
VRPTW	Vehicle Routing Problem with Time Window
VT	Vehicle Type
WWW	*World Wide Web*

Chapter 1

Motivation and overview

In the global economy, companies and organizations are seeking to develop innovative ways to enhance their competitiveness. Those who are pioneer in logistics management are able to optimize their operations and gain a significant competitive advantage, since logistic costs constitute an important part of the overall production costs.

Logistics management is the process of planning, implementing, and controlling the efficient flow and storage of goods, services, and related information from point of origin to the point of consumption for the purpose of conforming to customer requirements. Conventionally, each component of logistics management has been optimized separately and then connected sequentially in decision chain. However, in current competitive business environment the importance of coordination and integration of decisions within a system have become conspicuous. Sequential approach ignores the effect of decisions on one another and consequently results in sub-optimal solutions. Whereas, integrated approach considers all the related decisions simultaneously and optimizes jointly. Organizations are becoming increasingly aware of the impact of integrated optimization approach on the efficiency of logistics management.

The efficiency of such decisions and operations is measured by different metrics such as cost, response time, or by environmental footprint such as CO_2 emissions. Recently, there has been an increasing need for integrating environmentally sound choices into logistics management research and practice. This expands the focus of efficiency from cost and time to also more environmental friendly performances. It is generally perceived that concern for the environment in logistics management promotes efficiency and helps to enhance environmental performance, minimize waste and achieve cost savings. Eventually, environmental awareness is expected to enhance the corporate image, competitive advantage and marketing exposure (Rao and Holt, 2005). Therefore, it is important for companies and organizations to take advantage of operations research and optimization techniques to solve complex logistics problems to achieve higher competitive advantages and enhance environmental performance at the same time.

This thesis is the result of problem-driven research motivated by the operational problems arising in the e-commerce logistics. Increasing environmental awareness and the proliferation of

e-commerce worldwide has created extreme competitive environment with great pressure towards both operational and ecological efficiency on the players in this sector. An efficient warehouse management, as well as a sustainable e-commerce delivery can enhance both the operational and ecological efficiency significantly within e-commerce logistics management.

As shown in Figure 1.1, this thesis focuses on tackling some of the unresolved challenges in *warehouse scheduling* and *e-commerce delivery* at the core of e-commerce logistics management from two perspectives, namely, integrated decision making and sustainable decision making (incorporating environmental issues into the problem formulation).

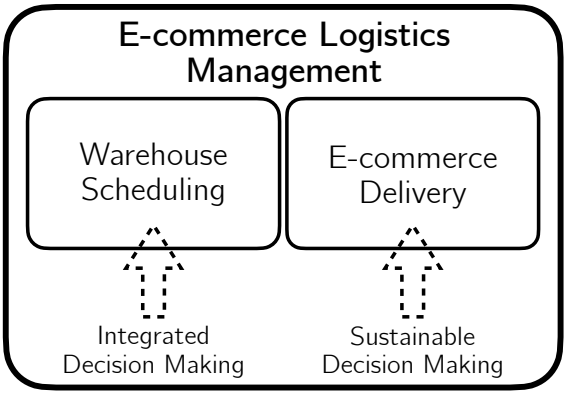


Figure 1.1: Dissertation focus and framework

1. **Warehouse scheduling** accounts for a large part of logistics cost in e-commerce and inefficiency in warehouse scheduling causes substantial waste of resources as well as increase in the number of damaged products.

   Warehouse scheduling concerns efficient allocation of resources to receive, store, sort and retrieve commodities and products to meet production planning and customer requirements. Resources in a typical warehouse include labor, storage space, machinery and time. The labor, storage space and machinery constitute a significant portion of logistics cost.

   This dissertation focuses on crane-operated warehouses and industrial yards which employ overhead cranes to move enormous products such as containers or steel coils. An integrated decision-making approach is proposed to constitute breakthroughs in scientific and practical grounds. Therefore, realistic constraints and features from real-world problems are considered when proposing new mathematical formulations and efficient solution approaches. Two main industrial settings are considered to study warehouse scheduling within e-commerce logistics:

   (a) Rail-rail transshipment terminals (RRTT) where containers are transshipped between multiple freight trains by gantry cranes. Despite the attention on operations in container terminals and inter-model terminals, only few studies tackled decision problems within RRTTs. Therefore, there remains several questions unexplored, specifically questions about integrating decisions related to crane operations in RRTTs. This work

aims to contribute to close this scientific gap, by considering the scheduling crane operations problem including three decision problems (i) determining the load plan, (ii) assigning container moves to cranes, and (iii) sequencing container moves per crane while considering more realistic constraints.

(b) Crane-operated warehouses which employ overhead cranes to move enormous and heavy products.

Given the two main decision making problems in warehouses; location assignment problems and crane scheduling problems, this thesis, therefore, focuses on proposing an efficient integrated approach addressing the two problems under one integrated optimization problem. Then, the proposed integrated approach will be compared by sequential decision making approach. The outcome of this study provides a general setting which may be easily adapted to other warehouses, land-side container terminals or any other industry employing multiple overhead cranes to handle products in the storage area.

All the formulations and algorithms are to be tested using the instances inspired by real-world warehouses or container terminals and should be extensible to other similar storage yards or other industries facing similar problems.

2. **E-commerce delivery** in e-commerce logistics concerns decisions to move products and inventory efficiently within the supply chain and eventually deliver to the end consumers. Conventionally, the goal is to manage the transportation cost by minimizing the number of employed vehicles, labor, as well as increasing the on-time delivery rate. There has been a growing ecological concerns in e-commerce logistics management to develop sustainable delivery policies as transportation emissions represent a significant portion of global emissions. The literature is however lacking studies exploring the most profitable transportation policies while simultaneously minimizing ecological impact on realistic problem instances. This thesis, therefore, attempts to address ecological impacts while maintaining cost efficiency within e-commerce logistics management. This is done by introducing the e-commerce delivery problem (EDP) which generalizes the vehicle routing problem. The objective is minimizing operational costs and emissions, while in addition to traditional vehicle routing objectives and constraints, it requires the selection of vehicle types (electrical or otherwise), the determination of delivery dates and/or the merging of delivery points.

The contributions of this thesis are aligned in two main axes: *i*) theoretical and *ii*) practical grounds which includes both mathematical formulations and solution approaches. Proposed mathematical formulations consider realistic features from real-world problems within industrial crane-operated warehouses. Solution approaches explore efficient algorithms solving integrated optimization problems to show the potential benefit of such algorithms in solving real-world problems. All the formulations and algorithms are tested against various datasets inspired by real-world cases and should be extensible to other industries facing similar problems.

The remainder of this chapter is structured as follows. Section 1.1 presents the research objectives, and thesis structure is detailed in Section 1.2.

1.1. Research Objectives

This thesis is motivated by a set of operations research problems within industrial settings to provide methods and solutions to support efficient and sustainable decision making in e-commerce logistics management. The focused areas are *i*. efficiency in “Crane-operated warehouses" where overhead cranes are employed to move products within the storage area and *ii*. sustainability in “E-commerce deliveries" wherein e-commerce online orders must be delivered to the customers. The proposed formulation and algorithms provide a general setting which may be easily adapted. Therefore, other industries which share similar features with either crane-operated warehouses or e-commerce deliveries may also take advantage of this thesis’s contributions.

The research objectives of this thesis are directly aligned with two challenges in e-commerce logistics management:

O.1 Integrating operational decisions in warehouses and storage areas.

O.2 Limiting ecological impact of e-commerce deliveries while maintaining cost efficiency.

O.1 - Integrating operational decisions in industrial warehouses and storage areas.

In crane-operated warehouses, cranes move the products from input points to the storage area and from storage area to output points. This includes a series of decisions: *i*. assigning the storage locations to incoming products and those which must be relocated within the storage area. *ii*. assigning cranes to move the products. *iii*. sequencing the moves for each crane’s schedule.

These operational problems conventionally are solved separately and sequentially. Due to inter-related nature of such decisions, considering them integrated and solving them simultaneously results in potential improvements. To date, there has been a lack of research which assesses the impact of integrating these sub-problems. This thesis pursues to investigate the impact of integrating these problems into one operational problem and develop efficient algorithms which are able to solve real-world size problems.

To this end, following research thrusts have been defined:

O.1-1 *Develop integrated mathematical modeling which considers realistic constraints*

Due to the fast growing rate of handling operations in the storage areas, many industries are employing multiple cranes operating simultaneously within the same storage area. Typically in crane-operated warehouses, cranes cannot overtake each other, thereby it is important to keep proper safety measures to avoid collisions. The goal of this research line is to develop an integrated mathematical model which considers realistic constraints such as employing multiple cranes while handling interference and keeping safety distances.

O.1-2 *Design and develop efficient algorithms*

Despite the attention given to the modeling of location assignment and crane scheduling problems, there is still work to be done in terms of solution procedures to a problem known to be NP-hard. Such solution procedure would include metaheuristics to integrate location assignment and crane scheduling decisions in storage areas. This would help companies employing warehouses with overhead cranes or container terminals to achieve higher efficiency for the available capacity and resources. The outcome related to this objective is efficient tools to solve this NP-hard problem for real-world size problems.

O.1-3 *Analyzing the impact of weight parameters in integrated optimization approach*

The objective function in integrated optimization approach is a weighted combination of objective functions of multiple sub-problems. Therefore, it is very important to analyze the impact of weight parameters on the solution space. By extensive simulations, this research line aims at obtaining insights into the effects of adjusting the weight parameters.

O.2 - Limiting ecological impact of e-commerce deliveries while maintaining cost efficiency.

Despite the increasing interest in green logistics, finding the most profitable mode of operation while simultaneously limiting greenhouse emissions on realistic problem instances remains unexplored. This research objective attempts to answer following research questions:

i) How may one organize delivery from distribution center to the customer in an ecological manner?

ii. Which operational shifts may reduce emission levels without incurring unacceptable costs?

iii) Which concessions related to delivery time windows or deviation from the delivery location are acceptable for environmentally-conscious consumers?

To address these questions, the e-commerce delivery problem (EDP) is introduced to generalize the vehicle routing problem. Its objective is to compose routes beginning and ending at the depot and visiting each delivery location while simultaneously minimizing operational costs and emissions. In addition to traditional vehicle routing objectives and constraints, the EDP requires the selection of vehicle types (electrical or otherwise), the determination of delivery dates, and/or the merging of delivery points.

To achieve this objective following steps are considered:

O.2-1 *Identify the crucial operational features of e-commerce delivery and propose a mathematical formulation for the e-commerce delivery problem*

The goal is to identify operational features to introduce an integer programming formulation which incorporates vehicle-dependent transportation costs and emissions, driving times and delivery times. By incorporating such operational features, the impact of alternative delivery

strategies — such as the introduction of a heterogeneous fleet including electric vehicles, relaxing delivery time windows, or aggregating delivery destinations — may be accurately investigated.

O.2-2 *Design and develop new solution approaches to tackle e-commerce delivery problem*

Efficient and fast algorithms are required to solve real-world size instances of the EDP within simulation environment. Therefore, this thesis aims at developing algorithms to tackle the EDP. Efficient algorithms including both population-based and local search based metaheuristics are introduced to enable simulations of various e-commerce delivery scenarios. The case study data provided by e-commerce delivery carriers constitutes an unique opportunity to explore the effects of the proposed policies.

1.2. Thesis structure

The main chapters of this thesis consist of a collection of papers that seek to answer the research objectives defined in the previous section. Chapters 2 and 3 are aligned with the objectives pursued in the O.1 - Integrating operational decisions in industrial warehouses and storage areas. Chapter 4 covers answers to the challenges identified for the O.2 - Limiting ecological impact of e-commerce deliveries while maintaining cost efficiency.

This section overviews the main aspects covered and the most substantial contributions associated with each of them.

Chapter 2 provides a broad literature review on rail-rail transshipment terminals (RRTT) and classifies the decisions in RRTTs into four problems. RRTTs represent an emerging technology in railway systems and freight transshipment terminals. The scheduling of crane operations problem (SCOP) in RRTTs is further investigated. This problem includes decisions concerning the positioning of containers on outbound trains (determining the load-plan), assigning container moves to cranes while considering crane overlap areas, and sequencing each crane's container moves. A mixed integer programing formulation is proposed which integrates the decisions in the SCOP and considers realistic constraints such as crane interference. In this problem cranes have their own working area and a common working area is considered between two neighboring cranes. The containers must be moved among the trains so that all containers with same destination be placed on one train. Algorithms based on the biased random-key genetic algorithm framework are detailed in this chapter. These algorithms are tested against on another and typical dispatching rules on a set of instances generated inspired by a real terminal. Extensive computational experiments performed to assess the performance of the proposed algorithms.

Following the work in Chapter 2, Chapter 3 considers integrating the location assignment problem and crane scheduling in crane-operated warehouses. A new mathematical formulations is proposed which considers location assignment and crane scheduling simultaneously. This

model implements various realistic operational constraints such as multiple cranes working simultaneously in the storage area and precedence constraints. Note that each crane can traverse the entire storage area provided safety distances between all cranes are respected. Efficient local search based metheuristics are developed and tested upon a set of instances inspired by insights extracted from a relevant industrial case.

Chapter 4 pursues the sustainable decision making in e-commerce logistics, thereby, proposes a general optimization approach to the e-commerce delivery problem, enabling the assessment of various operational delivery policies on both costs and emissions. A mixed integer programming formulation which considers both vehicle-dependent transportation costs and CO_2 emissions is proposed. Vehicle dependent cost includes labor costs, vehicle write off, fuel consumption, insurance, and maintenance. The objective function also considers total distance traveled by all vehicles and the number of vehicles required to satisfy all deliveries. Then three different heuristics are proposed, two local search based and one population based metaheuristics. These efficient heuristics enable simulation of various e-commerce delivery scenarios such as alternative parcel delivery vehicles, collection points, carrier bundling and regional monopolies and reducing the time required for delivering parcels.

Finally, Chapter 5 summarizes the work by presenting final conclusions and directions for future research.

Part of the observations and results of this thesis have been already published in the following peer reviewed journals and conferences (the contributions of this thesis author is described for each paper):

1. Sam Heshmati, Túlio A.M. Toffolo, Wim Vancroonenburg, and Greet Vanden Berghe. Crane-operated warehouses: Integrating location assignment and crane scheduling. *Computers & Industrial Engineering*, 129:274–295, 2019. ISSN 0360-8352.

   Contributions of thesis author: The author had the primary role in investigating the literature to form the study and research objectives. Mathematical formulations and heuristics were developed and implemented by the author and contribution of other co-authors. In addition, the author was responsible for generating the instances, running all the experiments and drafting the manuscript. The author enjoyed the consultation, advisory and editorial support of the co-authors.

2. Sam Heshmati, Jannes Verstichel, Eline Esprit, and Greet Vanden Berghe. Alternative e-commerce delivery policies. *EURO Journal on Transportation and Logistics*, 2018. ISSN 2192-4384.

   Contributions of thesis author: A literature review to identify the research questions was conducted by the author, followed by problem description and mathematical formulations. The heuristic based on Genetic Algorithm (GA) was developed and implemented by the

author. In addition, the experiments corresponding to the GA based heuristic, as well as the discussions and a comparative analysis of all proposed algorithms were conducted by the author. The author also participated in analyzing the alternative policies and drafting the manuscript along with other co-authors.

3. Sam Heshmati, Zafeiris Kokkinogenis, Rosaldo J. F. Rossetti, Maria Antónia Carravilla, and José Fernando Oliveira. An agent-based approach to schedule crane operations in rail-rail transshipment terminals. In Raquel J. Fonseca, Gerhard-Wilhelm Weber, and João Telhada, editors, *Computational Management Science*, pages 91–97, Cham, 2016. Springer International Publishing. ISBN 978-3-319-20430-7.

   Contributions of thesis author: The author had the leading role in reviewing the literature, forming the research objectives, developing the solution methods, running the experiments, and drafting the manuscript. The author enjoyed the consultation, advisory, editorial support of the co-authors.

* Sam Heshmati, Maria Antónia Carravilla, and José Fernando Oliveira. Scheduling crane operations in rail-rail transshipment terminals. To be submitted to Journal of Scheduling.

Contributions of thesis author: The author had the primary role in conducting this research and enjoyed the editorial and supervision of the co-authors.

In the spirit of reproducible science Kendall et al. (2016), whenever possible the source code associated with elements of this thesis was made available online, together with all instance and solution files.

Chapter 2

Scheduling Crane Operations in Rail-Rail Transshipment Terminals

Rail-rail transshipment terminals represent an emerging technology in railway systems and freight transshipment terminals. They aim to increase both transport frequency and the number of destinations that are reached via rail networks. Rail-rail transshipment terminals play the hub role in hub-and-spoke rail networks where containers are transshipped between multiple freight trains by gantry cranes. This study investigates the scheduling of gantry crane operations problem in rail-rail transshipment terminals. The scheduling of crane operations includes policies concerning the positioning of containers on outbound trains (determining the load-plan), assigning container moves to cranes while considering crane overlap areas, and sequencing each crane's container moves. The objective is to minimize total transshipment time and the number of containers left behind on the yard. Algorithms based on the biased random-key genetic algorithm framework are developed. The effectiveness and Robustness of the proposed algorithms are illustrated by extensive experiments upon sets of instances inspired by real-world rail-rail transshipment terminals.

2.1. Introduction

Freight transshipment terminals in rail networks are special nodes where train loads are collected, rearranged, intermediately stored, loaded, and picked up (Boysen et al., 2013). Despite rail systems have significant advantages over road network transportation, such as lower carbon footprints and the avoidance of road congestion, the share of freight traffic handled by rail systems has decreased during the past 35 years (of the European Commiunities, 2007). A major obstacle associated with freight terminals is the high fixed cost, meaning transportation is only profitable if on large scales over long distances. Rail-rail transshipment terminals (RRTTs) represent an emerging technology within freight transshipment terminals and railway systems (Macharis and Bontekoning, 2004; Bontekoning et al., 2004). They are employed to increase transport frequencies and number of destinations to be reached simultaneously, and offer more suitable operation times for

shippers and customers at terminals with higher reliability, when compared to the conventional shunting yards (Terminet-Consortium, 2001).

Maritime container terminals have received considerable attention in the academic literature, (see Steenken et al. (2004) and Stahlbock and Voß (2008)), however, there are few studies focused on RRTTs. The majority of the studies in RRTTs are focused on traditional classification yards, where wagons are rearranged instead of containers being exchanged by the use of cranes (Boysen et al., 2012a). Scheduling trains and rearranging them represent the primary problems associated with this type of terminal (Dahlhaus et al., 2000; Hansmann and Zimmermann, 2008; He et al., 2003).

The strategic level of decision-making in RRTTs primarily concerns policies about structural and technical aspects (Rotter, 2004) in addition to the design of the terminal layout (Wiegmans et al., 2007). In this study, we exclusively focus on RRTTs from an operational level. The operational decision problems in RRTTs may be categorized into four categories: (*i*) Scheduling the train process (STP), (*ii*) Scheduling crane operations (SCO), (*iii*) Positioning containers in the yard, and (*iv*) Scheduling the shuttle cars.

"STP" contains three sub-problems: (i) *Scheduling the service slots of trains*, wherein the objective is to minimize the linear combination of both the number of delayed containers and the number of split moves Boysen et al. (2011, 2012b). A split move occurs when a train's assigned container is in a different service slot. (ii) *The assignment of each train to a parking position* is explored by considering both the vertical (assignment of trains to tracks) and horizontal parking positions (parking position on the track) to minimize the makespan of train processing. A mathematical model has been also presented for the train location problem and different heuristic solution procedures are described and tested in Kellner et al. (2012). (iii) *Assign a destination to each train.*

"SCO" includes the following sub-problems: (i) *Determine container positions on outbound trains (determining the load plan).* Bostel and Dejax (1998) studied the sub-problem of determining the load plan by considering four scenarios, the most general objective being to minimize the total container processing cost. Weight and length constraints of the wagons are not considered and each container may be placed anywhere on the train. This study employed a real dataset from the French national railway company (SNCF). A similar problem concerns determining the load plan for rail-road terminals where, in addition to the transshipment between multiple trains, there is the possibility to transship containers between trains and trucks. Corry and Kozan (2008), addressed this problem with two weighted objectives: minimizing train length, and minimizing container processing time. (ii) *Container moves assignments to cranes.* Boysen et al. (2010) used a dynamic programming procedure to solve the problem. The solutions were tested against typical real-world policies with static assignment of equally sized working areas of cranes. Boysen and Fliedner (2010), tackled a similar problem in rail-road terminals. Since each truck can move to the working area of each crane, minimizing the number of indirect moves is an irrelevant objective for rail-road terminals. Thus, they aimed at evenly distributing the total workload among all cranes. (iii) *Determine the sequence of container moves per crane.* Martínez et al. (2004) studied the

sequencing of container moves at Port-Bou in the border of France and Spain. The study considered four different rules for sequencing the container moves and the rules were compared by a simulation model.

The problem of positioning containers in the yard is addressed with the objective of minimizing the crane rate, which is itself attained by minimizing the number of stacked containers and the total distance of crane moves Jaehn (2012). A weighted linear combination of the two objectives was used as objective function. The study assumes all the other sub-problems are already solved and the cranes working areas are disjoint.

Another set of studies investigates combinations of decision problems in RRTTs. Alicke (2005) formulated an optimization model composed of three sub-problems: assigning container moves to cranes, determining the sequence of container moves per crane, and scheduling the shuttle cars. The overall problem was modeled as a constraint satisfaction problem and tested on the German MegaHub in Hanover data set. Gonzalez et al. (2008) presented a mixed integer model for determining the load-plan and the sequence of container moves per crane. Several constraints related to container and wagon characteristics are considered. The objective is the same as in Bostel and Dejax (1998): the minimization of the total number of container movements. The approach is validated by using real-world data from a terminal in the Spanish–French border, the Port-Bou Terminal. In another study, Souffriau et al. (2009) used a decomposition approach to simultaneously assign the destinations to the trains, determine the load plan, and the sequence of the transshipments. Three different types of containers and length restrictions for the wagons are considered. The load plan was determined by minimizing the transportation cost of container moves within a mixed integer model. The transshipments sequence was modeled as a resource-constrained project scheduling problem and solved by a variable neighborhood search. Fedtke and Boysen (2017) proposed a mathematical formulation for sequencing container moves per crane and scheduling the shuttle cars simultaneously. They assumed container positions on outbound trains is determined, thus, each container move is assumed to have a given start and destination wagon. Their model did not consider overlapping working area for cranes, so that the yard assumed to be partitioned into disjunct crane areas each operated by a dedicated crane. Therefor, container moves assignment for cranes was not considered as each crane is operating in its own dedicated area. Nossack et al. (2018) proposed a branch and cut algorithm for dual crane scheduling in container terminals where cranes operating on different levels and may pass each other. Li et al. (2019), considered scheduling a single crane in rail-road transshipment yards while assume the destinations of the containers are given. The authors proposed a decomposition algorithm, which decomposes the initial problem for a smaller set of jobs.

Literature on RRTTs is scarce, when compared with the attention given to container processing (mainly in port terminals), inter-modal transportation, and railway systems. This work aims to contribute to close this scientific gap, by considering the SCOP, one of the main problems in RRTTs. In this paper the container transshipment process in RRTTs is analyzed from an operational point of view. Three decision problems (*i*) determining the load plan, (*ii*) assigning container moves to cranes, and (*iii*) sequencing container moves per crane are considered in a yard where

two neighboring cranes have overlapping working area. The objective is to minimize the total transshipment time while loading as many containers as possible on outbound trains. Efficient algorithms are proposed to solve the integrated problem, using genetic algorithms and customized heuristics to decode the solutions.

The remainder of the paper is organized as follows. Following, a detailed problem description and a mathematical formulation are presented in Sections 2.2 and 2.3, respectively. Genetic algorithms, with different encoding schemes, are proposed in Section 2.4. Section 2.5 contains the computational experiments, where the performance of the algorithms is tested. Finally, conclusions and future research directions are summarized in Section 2.6.

2.2. Problem definition

A typical RRTT consists of a number of parallel tracks, multiple semi-automated or automated rail mounted gantry cranes, and a buffer lane along the entire length of the terminal, with shuttle cars deployed to deliver containers along the terminal. A schematic view of a RRTT is depicted in Figure 2.1. Containers are transshipped between multiple trains by cranes without the need to move wagons (unlike shunting yards). Cranes have limits on their operation areas, i.e. generally a crane can not work on the whole terminal. Consequently, multiple cranes work in parallel to cover the entire yard. In this setting cranes have their own working area which overlap with the working area of neighboring cranes.

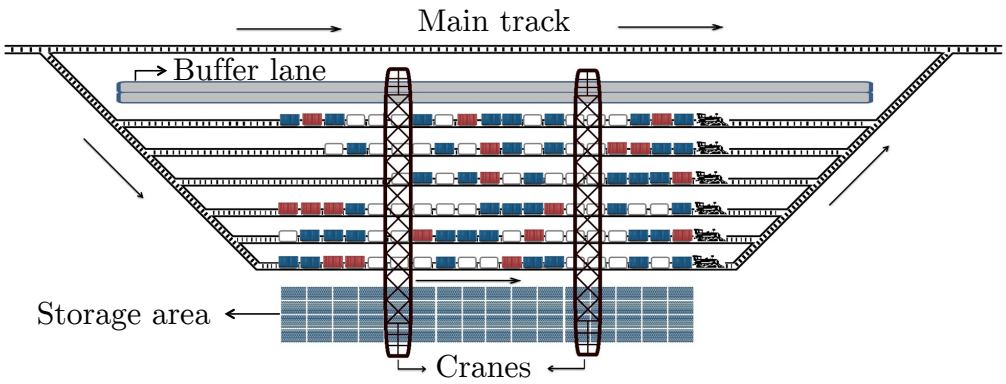


Figure 2.1: A schematic rail-rail transshipment terminal

The SCOP is an operational problem that must be solved for each train service slot in a RRTT. It determines position of each container on outbound trains (load plan), assigns container moves to cranes and finally sequences the transshipment of containers for each crane. The objective is to minimize the total container transshipment time while simultaneously loading as many containers as possible onto the trains. The quality of each solution is obtained by Equation 2.1, where, C_{max} is total transshipment time, and E_{nc} represents total size of the containers which were not able to fit on outbound trains.This is a weighted linear combination of the total transshipment time and

the cost of assigning a container to the yard, proportional to containers size, where ω_0 and ω_1 are weights defining the relative importance of the terms.

$$\mathscr{E} = \quad \omega_0 C_{max} + \omega_1 E_{nc} \tag{2.1}$$

Scheduling trains service slot is considered to be a previous planing step and therefore it is assumed that a list of trains, in a service slot, and both destination and parking track of each train are supplied as input data. Transshipment operations concern the movement of a container by a crane, beginning from the position where the container is currently located (either on a train or at the yard) and ending at the container's final position (either on a train or at the yard). The term "operation" is employed in this chapter to indicate transshipment operation for simplicity.

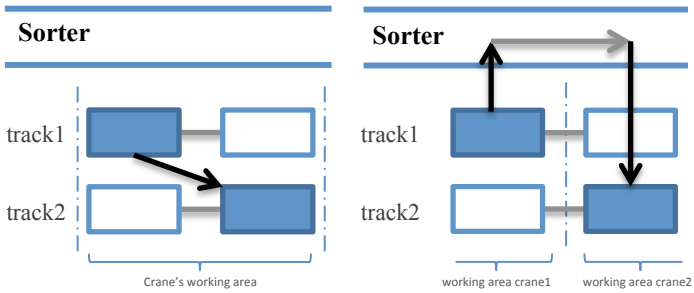
Yards in RRTTs consist of a number of parallel tracks and a storage area. The yard length is mapped to a horizontal coordinate axis which are ordered from left to right. Trains are composed of a set of wagons with varying capacities, which can hold more than one container. The wagon capacity is defined as a length. Each wagon is associated with a horizontal coordinate h_w which indicates the location of wagon w along the yard.

A set of cranes are employed to transship containers between trains or the yard. Cranes in the yards cannot overtake each other when operating, since they share tracks along the yard. The yard is divided into partitions, with each partition representing the working area of a single crane. The working area of two neighboring cranes may overlap. It is necessary to ensure that the cranes do not collide while working in such overlapping areas. This is achieved by blocking the overlapping area for one crane when another crane enters it. Therefore, when one crane is working inside the overlapping area, the other must either work outside of it or remain idle until the first one leaves the area. Therefore, large overlapping areas potentially result in long idle times. Crane movements are asymmetric in distance, since there is an unloaded crane movement connecting two loaded moves.

Containers are located on specific train wagons, with each container having a specific destination and size (length). The width is equal for all containers (and wagons) and may therefore be ignored for SCO planning purposes. A container may be placed on a wagon if the wagon's capacity is equal to or greater than the container's size.

Based on the origin and destination of a transshipment, a container may be transshipped from one train to another train by direct or indirect moves. Figure 2.2 illustrates these two types of container movements. When the origin and destination wagon of a transshipment are in the same crane's working area, the container is handled once and only by one crane. This type of movement is called a **direct** move. **Indirect** moves occur when the origin and destination wagons of a container are in the working area of different cranes.

To execute an indirect move, a crane in starting area puts the container in the buffer lane, where the container is transferred to the second crane's area by a ground vehicle. Then the second crane processes the container from the buffer lane to its final position. In an indirect move the container is handled more than once, implying an extra pick-up and drop and a dependency upon the buffer lane.



(a) Direct move (b) Indirect move

Figure 2.2: Direct move versus indirect move (adapted from Kellner et al. (2012)).

A transshipment may only be carried out when the container's final position on the outbound train is free. Whenever a container blocks the final position of another, precedence for container moves must be established.

2.3. Continuous-time model for the SCOP

An integrated continuous-time model for the SCOP was developed including location assignment for containers on outbound trains, crane assignment for each transshipment operation, and the sequencing of these operations on each crane. Crane activities and container handling are realistically modeled.

This model is based on the models proposed in Li et al. (2012) and Heshmati et al. (2019). The model proposed in Li et al. (2012) deals with cranes in container terminals and considers only the scheduling of crane operations. It does not include the decisions regarding where to put the containers on the yard. As there are trucks bring/taking the containers, right in front of the storage/retrieval row, the cranes are not moving during an operation and so there are not conflicting moves and the duration of the operations is fixed and equal. Another difference is that in this model there are no separate working zones for cranes, i.e. cranes can move along the yard, which is a different setting for the working zones of the cranes. The model proposed in Heshmati et al. (2019) concerns operations in crane-operated warehouses. The cranes, in this problem, can move along the yard at the same time by respecting a safety distance to avoid collisions.

The model now proposed is applied in a train terminal context, and the setting of the crane zones is different. In fact the cranes move during the operations and therefore conflicting moves exist and have to be modeled to be avoided. This model also decides where to put the containers on the outgoing trains. Capacity constraints are also considered for the wagons, but just with one level of container storage.

Table 2.1 summarizes the notation employed in the formulation.

Table 2.1: Notations

Indices	
w, v :	wagons;
i, j :	operations;
c, c' :	cranes;
Sets	
$\mathscr{O}$:	Set of transshipment operations;
$\mathscr{O}_w$:	Set of transshipment operations which have wagon w as origin;
$\mathscr{W}$:	Set of wagons;
$\mathscr{W}_i$:	Set of wagons on the destination train of operation i's container;
$\mathscr{C}$:	Set of cranes;
$\mathscr{C}_i$:	Set of cranes that can execute operation i, $\mathscr{C}_i \subset \mathscr{C}$;
Parameters	
NC :	number of cranes in the yard;
η_c :	order of crane c in the yard, $0 < \eta_c < NC$;
$H^-_{cc'}$:	left margin of common area between cranes c and c';
$H^+_{cc'}$:	right margin of common area between cranes c and c';
q_i :	size of the container in operation i;
h_w :	horizontal coordinate of wagon w
t_{wv} :	traveling time of a crane from wagon w to wagon v;
l_w :	initial load (occupied length) of wagon w;
f_w :	capacity (length) of wagon w;
b_i :	origin wagon of operation i;
e_i :	destination wagon of operation i;
M :	a large number;
Decision variables	
x_{iw} :	is equal to 1 if wagon w is assigned as destination for operation i, and zero otherwise;
y_{ic} :	is equal to 1 if operation i is handled by crane c, and zero otherwise;
s_i :	is the start time of the transshipment operation i, a non-negative variable;
$o_{ijcc'}$:	is equal to 1 if operations i and j are conflicting in the common working area of cranes c and c', and zero otherwise;
n_{ij} :	is equal to 1 if s_j is greater than the end time of operation i, and zero otherwise;
m_{ijw} :	is equal to 1 if wagon w is destination of operation i and operation i is carried out before operation j, and zero otherwise;
Auxiliary variables	
d_i :	duration of operation i;
$g_{ic jc'}$:	required waiting time between two conflicting operation i handled by crane c and operation j handled by crane c';
h_i^- :	horizontal coordinate of the left wagon of operation i's destination;
h_i^+ :	horizontal coordinate of the right wagon of operation i's destination;
$o^a_{icc'}$:	binary variable equal to 1 if right wagon of operation i is to the right of left margin of common area between cranes c and c' and 0 otherwise;
$o^b_{icc'}$:	binary variable equal to 1 if right wagon of operation i is to the left of right margin of common area betwenn cranes c and c' and 0 otherwise;
$o^c_{icc'}$:	binary variable equal to 1 if left wagon of operation i is to the right of left margin of common area between cranes c and c' and 0 otherwise;
$o^d_{icc'}$:	binary variable equal to 1 if left wagon of opeartion i is to the left of right margin of common area betwenn cranes c and c' and 0 otherwise;

2.3.1 Objective function

The objective is to transship containers between the train as fast as possible with minimum number containers left on the yard.

$$Minimize \quad \omega_0 C_{max} + \omega_1 \sum_{i \in \mathscr{O}} q_i x_{iyard} \tag{2.2}$$

Objective function (2.2) minimizes a linear combination of total transshipment time and the cost of assigning a container to the yard, proportional to its size, where ω_0 and ω_1 are weights indicating the relative importance of total transshipment time and total size of the containers assigned to yard, respectively.

2.3.2 Constraints

This section presents the SCOP's constraints, organized in four categories: *location assignment for containers*, *container move assignment to cranes*, *identifying conflicting operations*, and *start time of consecutive operations*.

2.3.2.1 Location assignment for containers

The following set of constraints concern location (wagon) assignment to a container in destination train.

$$\sum_{w \in \mathscr{W}_i} x_{iw} = 1 \qquad \forall i \in \mathscr{O} \tag{2.3}$$

$$m_{ijw} \geq x_{iw} + n_{ij} - 1 \qquad \forall w \in \mathscr{W}, i, j \in \mathscr{O} : i \neq j \tag{2.4}$$

$$m_{ijw} \leq x_{iw} \qquad \forall w \in \mathscr{W}, i, j \in \mathscr{O} : i \neq j \tag{2.5}$$

$$m_{ijw} \leq n_{ij} \qquad \forall w \in \mathscr{W}, i, j \in \mathscr{O} : i \neq j \tag{2.6}$$

$$l_w + q_i x_{iw} - \sum_{j \in \mathscr{O}_w} q_j n_{ji} + \sum_{j \in (\mathscr{O} - \mathscr{O}_w)} q_j m_{jiw} \leq f_w \qquad \forall w \in \mathscr{W}, i \in \mathscr{O} \tag{2.7}$$

Constraints 2.3 ensure that exactly one wagon from $\mathscr{W}_i$ is assigned to the container of operation i. Let $\mathscr{W}_i$ represent the set of wagons in train which has the same destination as operation i's container. The set $\mathscr{W}_i$ consists of wagons with the following conditions: the wagons are located in the destination train, and the wagons must be within the same crane working area of the origin wagon. If the origin wagon is in the common working area of two cranes then $\mathscr{W}_i$ consists of destination train wagons in working are of either cranes. Constraints (2.4) - (2.6) determine the value of variable m_{ijw} which is equal to 1 if wagon w is destination of operation i and operation i is carried out before operation j. Constraints (2.7) ensure that the capacity of the wagons is respected after each operation.

2.3.2.2 Container move assignment to cranes

$$\sum_{c \in \mathscr{C}_i} y_{ic} = 1 \qquad \forall i \in \mathscr{O} \tag{2.8}$$

$$s_i + d_i \geq s_j - n_{ij} M \qquad \forall i, j \in \mathscr{O} : i \neq j \tag{2.9}$$

$$s_i + d_i \leq s_j + (1 - n_{ij}) M \qquad \forall i, j \in \mathscr{O} : i \neq j \tag{2.10}$$

$$y_{jc} \leq 1 + n_{ij} + n_{ji} - y_{ic} \qquad \forall i, j \in \mathscr{O} : i \neq j, c \in \mathscr{C} \tag{2.11}$$

Constraints (2.8) assure that each operation is assigned to exactly one crane. Constraints (2.9) and (2.10) determine the value of variable n_{ij}, whereby if $n_{ij} + n_{ji} = 0$ operations i and j overlap in time. Constraints (2.11) state that a crane can handle one operation at a time, and if operations i and j are handled by the same crane, then their operation times must not overlap.

2.3.2.3 Identifying conflicting operations

Each operation has an origin and destination wagon, the wagon with greater h coordinate is called right wagon and the other one left wagon, regardless if it is origin or destination wagon.

$$h_i^- = \sum_{w \in \mathscr{W}_i : h_w < h_{b_i}} h_w x_{iw} + \sum_{w \in \mathscr{W}_i : h_w > h_{b_i}} h_{b_i} x_{iw} \qquad \forall i \in \mathscr{O} \tag{2.12}$$

$$h_i^+ = \sum_{w \in \mathscr{W}_i : h_w < h_{b_i}} h_{b_i} x_{iw} + \sum_{w \in \mathscr{W}_i : h_w > h_{b_i}} h_w x_{iw} \qquad \forall i \in \mathscr{O} \tag{2.13}$$

The horizontal coordinate of the left wagon of transshipment operation i (h_i^-) and the right wagon of transshipment operation i (h_i^+) are calculated by equations (2.12) and (2.13) respectively.

Given a set of operations $\mathscr{O}$, any two operations i and j occurring within the common working area of two cranes pose an additional challenge given that only one crane may exist in this area at any time. Such operations are considered conflicting with one another ($o_{ijcc'} = 1$).

$$h_i^+ \leq H_{cc'}^- + o_{icc'}^a M \qquad \forall i \in \mathscr{O}, c, c' \in \mathscr{C} \tag{2.14}$$

$$h_i^+ \geq H_{cc'}^+ - o_{icc'}^b M \qquad \forall i \in \mathscr{O}, c, c' \in \mathscr{C} \tag{2.15}$$

$$h_i^- \leq H_{cc'}^- + o_{icc'}^c M \qquad \forall i \in \mathscr{O}, c, c' \in \mathscr{C} \tag{2.16}$$

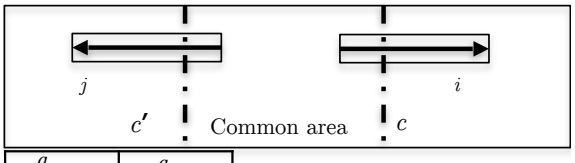
$$h_i^- \geq H_{cc'}^+ - o_{icc'}^d M \qquad \forall i \in \mathscr{O}, c, c' \in \mathscr{C} \tag{2.17}$$

$$o_{ijcc'} \geq o_{icc'}^a + o_{icc'}^b + o_{jcc'}^c + o_{jcc'}^d - 3 \qquad \forall i, j \in \mathscr{O} : i \neq j; c, c' \in \mathscr{C} : c \neq c' \tag{2.18}$$

$$o_{ijcc'} \geq o_{jcc'}^a + o_{jcc'}^b + o_{icc'}^c + o_{icc'}^d - 3 \qquad \forall i, j \in \mathscr{O} : i \neq j; c, c' \in \mathscr{C} : c \neq c' \tag{2.19}$$

Binary variables $o_{icc'}^a$, $o_{icc'}^b$, $o_{icc'}^c$, and $o_{icc'}^d$ are used to assess conflicting operations. Constraints (2.14) - (2.17) determine the value of these auxiliary variables. Constraints (2.18) and (2.19) guarantee that if the right wagon of one operation and the left wagon of another are simultaneously inside the common area, they are considered conflicting operations ($o_{ijcc'} = 1$).

Figure 2.3 presents an example where two operations i and j with origins in the common area of cranes c and c'. These two operations cannot be carried out at the same time since two cranes cannot be present in the common area. The correspondent values of variables $o_{icc'}^a$, $o_{icc'}^b$, $o_{icc'}^c$, $o_{icc'}^d$ and of $o_{ijcc'}$ are presented in the figure.



$o_{jcc'}^a = 1$	$o_{icc'}^a = 1$
$o_{jcc'}^b = 1$	$o_{icc'}^b = 0$
$o_{jcc'}^c = 0$	$o_{icc'}^c = 1$
$o_{jcc'}^d = 1$	$o_{icc'}^d = 1$

$$Oijcc' = 1$$

Figure 2.3: An example of two conflicting operations.

2.3.2.4 Start time of consecutive operations

Duration of operations depend on chosen destination wagon, hence constraints (2.20) are employed to set operations' duration.

$$d_i = \sum_{w \in \mathscr{W}_i} t_{b_i w} x_{iw} \qquad \forall i \in \mathscr{O} \tag{2.20}$$

Cranes are able to handle one container at a time, therefore to execute the operations one by one cranes require ample time to execute one operation and then move to the origin wagon of the next operation. If operation j is scheduled after i, and both operations are handled by same crane ($y_{ic} = 1$, $y_{jc} = 1$, and $n_{ij} = 1$), constraints (2.21) ensure that start time of operation j is after operation i is finished ($s_i + d_i$), plus the time crane c needs to travel from destination of i to j's origin (t_{ij}).

$$s_j \geq s_i + d_i + t_{e_i b_j} - (3 - y_{ic} - y_{jc} - n_{ij})M \qquad \forall i, j \in \mathscr{O} : i \neq j, c \in \mathscr{C} \tag{2.21}$$

$$s_j \geq s_i + g_{icjc'} - (3 - y_{ic} - y_{jc'} - o_{ijcc'} + n_{ji})M \qquad \forall i, j \in \mathscr{O} : i \neq j, c, c' \in \mathscr{C} : c \neq c' \tag{2.22}$$

Constraints (2.22) set the start time of two operations handled by different cranes. Start times of non-conflicting operations assigned to different cranes are independent from each other. However, start time of conflicting operations must be set in such a way that only one crane be present in common areas at any time. Assume two conflicting operations i and j are handled by neighboring cranes c and c' respectively, ($y_{ic} = y_{jc'} = o_{ijcc'} = 1$). If operation j is scheduled after i ($n_{ij} = 1$, $n_{ji} = 0$), then s_j, must be after start time of i, s_i, plus additional waiting time to ensure only one crane is present in the common area ($g_{icjc'}$).

There are four cases to consider when calculating gantry time, $g_{icjc'}$, depending on operation j's origin and cranes' location relative to each other. Assume crane c is located to the right of crane c' ($\eta_{c'} < \eta_c$) and operation i's destination, e_i, is located inside the common area between cranes c and c'. In this case gantry time, $g_{icjc'}$, must be greater than operation i's duration plus crane c's traveling time from operation i's destination, e_i, towards out of common area, $H^+_{cc'}$, plus the time that c' needs to travel from the left-hand border to b_j, $t_{b_j H^-_{cc'}}$.

If operation i's destination, e_i, is located out of common area, then $g_{icjc'}$, equals crane c's traveling time from origin of operation i, b_i, to common area's right border, $H^+_{cc'}$, plus traveling time of crane c' from left border to operation j's origin, $t_{b_j H^-_{cc'}}$. Constraints (2.23) are employed to find $g_{icjc'}$ when crane c is to the right of crane c' ($\eta_{c'} < \eta_c$) and operation j's origin is outside of common area ($h_{b_j} > H^-_{cc'}$).

$$g_{icjc'} \geq \sum_{w \in \mathscr{W}_i : h_w < H^+_{cc'}} (t_{b_i w} + t_{w H^+_{cc'}}) x_{iw} + \sum_{w \in \mathscr{W}_i : h_w > H^+_{cc'}} (t_{b_i H^+_{cc'}} x_{iw}) + t_{b_j H^-_{cc'}}$$
$$\forall i, j \in \mathscr{O} : i \neq j, h_{b_j} > H^-_{cc'} \quad c, c' \in \mathscr{C} : \eta_{c'} < \eta_c \tag{2.23}$$

Constraints (2.24) set gantry time, $g_{icjc'}$, for two conflicting operations i and j, when crane c is the right neighbor of crane c', operation j's origin locates out of common area ($b_j < H^-_{cc'}$) and operation i starts before j ($n_{ji} = 0$). If destination of operation i is located in the common area, $g_{icjc'}$ equals i's duration plus the time crane c' requires to leave the common area (traveling time of crane c from i's destination to right border of common area), plus traveling time of crane c' from left border of common area to j's origin, $g_{icjc'} \geq d_i + t_{e_i H^+_{cc'}} + t_{H^-_{cc'} b_j}$. If destination of operation i is located out of common area, $g_{icjc'}$ equals traveling time of crane c from i;'s origin to right border of common area plus traveling time of crane c' from left border of common area to j's origin, $g_{icjc'} \geq t_{b_i H^+_{cc'}} + t_{H^-_{cc'} b_j}$.

$$g_{icjc'} \geq \sum_{w \in \mathscr{W}_i : h_w < H^+_{cc'}} (t_{b_i w} + t_{w H^+_{cc'}}) x_{iw} + \sum_{w \in \mathscr{W}_i : h_w > H^+_{cc'}} (t_{b_i H^+_{cc'}} x_{iw}) - t_{b_j H^-_{cc'}}$$
$$\forall i, j \in \mathscr{O} : i \neq j, h_{b_j} < H^-_{cc'} \quad c, c' \in \mathscr{C} : \eta_{c'} < \eta_c \tag{2.24}$$

Constraints (2.25) are employed to calculate gantry time, $g_{icjc'}$, for two conflicting operations i and j, when crane c' is to the right of crane c', operation j's origin is inside of common area ($b_j < H^+_{cc'}$) and operation i starts before j ($n_{ji} = 0$).

$$g_{icjc'} \geq \sum_{w \in \mathscr{W}_i : h_w < H^-_{cc'}} t_{b_i H^-_{cc'}} x_{iw} + \sum_{w \in \mathscr{W}_i : h_w > H^-_{cc'}} (t_{b_i w} + t_{w H^-_{cc'}}) x_{iw} + t_{b_j H^+_{cc'}}$$
$$\forall i, j \in \mathscr{O} : i \neq j, h_{b_j} < H^+_{cc'} \quad c, c' \in \mathscr{C} : \eta_{c'} > \eta_c \tag{2.25}$$

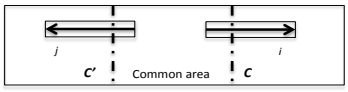
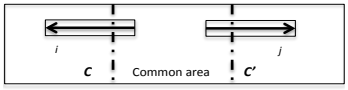
Constraints (2.26) set gantry time, $g_{icjc'}$, for two conflicting operations i and j, when crane c is the left neighbor of crane c', operation j's origin locates out of common area ($b_j > H^+_{cc'}$) and operation i starts before j ($n_{ji} = 0$).

$$g_{icjc'} \geq \sum_{w \in \mathscr{W}_i : h_w < H^-_{cc'}} t_{b_i H^-_{cc'}} x_{iw} + \sum_{w \in \mathscr{W}_i : h_w > H^-_{cc'}} (t_{b_i w} + t_{w H^-_{cc'}}) x_{iw} - t_{b_j H^+_{cc'}}$$
$$\forall i, j \in \mathscr{O} : i \neq j, h_{b_j} > H^+_{cc'} \quad c, c' \in \mathscr{C} : \eta_{c'} > \eta_c \tag{2.26}$$

To calculate $g_{icjc'}$ there are eight combinatorial scenarios based on the position of b_j and e_i, and cranes' position in relation to each other, which are addressed in eqs. (2.23) to (2.26).

Table 2.2 summarizes the eight cases that can arise when calculating $g_{icjc'}$ by depicting a schematic view of each situation.

Table 2.2: Eight scenarios to calculate $g_{icjc'}$.

Conditions	Schematic view of the case	Constraints
$\eta_{c'} < \eta_c$ $b_j > H^-_{cc'}$ $e_i < H^+_{cc'}$		(2.23)
$\eta_{c'} < \eta_c$ $b_j > H^-_{cc'}$ $e_i > H^+_{cc'}$		
$\eta_{c'} < \eta_c$ $b_j < H^-_{cc'}$ $e_i > H^+_{cc'}$		(2.24)
$\eta_{c'} < \eta_c$ $b_j < H^-_{cc'}$ $e_i < H^+_{cc'}$		
$\eta_{c'} > \eta_c$ $b_j < H^+_{cc'}$ $e_i > H^-_{cc'}$		(2.25)
$\eta_{c'} > \eta_c$ $b_j < H^+_{cc'}$ $e_i < H^-_{cc'}$		
$\eta_{c'} > \eta_c$ $b_j > H^+_{cc'}$ $e_i > H^-_{cc'}$		(2.26)
$\eta_{c'} > \eta_c$ $b_j > H^+_{cc'}$ $e_i < H^-_{cc'}$		

$$C_{max} \geq s_i + d_i \qquad \forall i \in \mathscr{O} \tag{2.27}$$

Constraints (2.27) guarantee that C_{max} is greater than the end time of all operations.

The SCOP may be reduced to the Sequential Ordering Problem (SOP) as a special case of the sequencing container moves with one crane and known operations destination. Hence, the SCOP is at least as hard as the SOP which is proven to be a $\mathscr{NP}$-hard (Montemanni et al., 2009). Therefore, the SCOP is also $\mathscr{NP}$-hard.

2.4. Algorithms

Efficient heuristic procedures are required for solving the SCOP instances of real-world size. This study presents efficient algorithms based on the biased random-key genetic algorithm (BRKGA), namely 2NG and NGH algorithms. Both the 2NG and NGH algorithms are based on the BRKGA, however they employ different solution representations. This enables further investigations of solution representation. Application of the BRKGA is inspired by successful approaches in the literature. Numerous studies employed the BRKGA for combinatorial problems including scheduling problems (Gonçalves and Resende, 2011).

The BRKGA is a variant of the random-key genetic algorithms. It builds on the concept of "general-purpose metaheuristic". A solution is represented by a *chromosome* which in the BRKGA a chromosome consists of a vector of real numbers in $[0,1)$ called *genes* Toso and Resende (2014). There is a clear division between the algorithm's problem-independent and problem-dependent parts in this framework. The problem-independent part includes the algorithm's evolutionary engine which includes population, selection, recombination and mutation. Whereas, algorithm's problem-dependent part includes chromosome representation, a decoder and evaluation function. It decodes a chromosome into a solution, and then evaluates the solution and returns its evaluation to the problem-independent part.

The remainder of this section details the problem-dependent part of the algorithms.

2.4.1 2NG algorithm

This section presents chromosome representation and decoder of 2NG algorithm.

Each chromosome of 2NG algorithm is composed of $2n$ genes, where n is the number of operations (containers that must be transshipped), i.e.,

$$Chromosome = \left(\underbrace{gene_1, ..., gene_n}_{\text{OS}}, \underbrace{gene_{n+1}, ..., gene_{2n}}_{\text{OD}} \right)$$

The first n genes are employed to obtain operations sequence (OS), whereas the next n genes are employed to generate the operations' destination (OD). Therefore, there are two genes associated to each operation i, $gene_i$ to indicate the OS and $gene_{i+n}$ to obtain the OD. A *decoder* is employed to generate a solution from a chromosome, Algorithm 1 details 2NG *decoder*. Before decoding a chromosome, a list χ of (operation, gene, gene)-tuples must be generated. Each tuple in this list consists of an operation and its associated OS and OD genes, respectively. The output of *decoder* is a list, $\mathscr{L}$, of (operation, wagon, crane)-tuples which for each operation, i, indicates its destination (wagon) and a crane to handle it.

First, 2NG *decoder* sorts the list χ by OS genes in ascending order to get the *operations sequence* (Line 3), so order of sorted tuples indicates the sequence of the operations. Then for each tuple, t, a list of feasible wagons of operation i, $\mathscr{W}_i$, is created. A feasible wagon is one with enough capacity to fit container of operation i and located in a train with container's destination

(Line 5). The assigned wagon to operation i is $(\lfloor m \times (gene_{i+n}) \rfloor + 1)th$ wagon in $\mathscr{W}_i$, where m is the size of $\mathscr{W}_i$ and $gene_{i+n}$ is OD gene of operation i. If there is no feasible wagon for operation i then its container will be placed on the yard (Lines 7 – 11). The yard is modeled as a wagon with infinite container capacity. Whenever a crane moves a container of an operation i to the yard does not move along the horizontal axis, therefore, horizontal coordinate of yard is set equal to horizontal coordinate of origin wagon of operation i, ($h_{yard} = h_{b_i}$).

Let set $\mathscr{C}_w$ indicate the cranes which can operate over wagon w. If w locates in a overlap area then $|\mathscr{C}_w| = 2$, otherwise $|\mathscr{C}_w| = 1$. Based on the origin and destination wagon of an operation there are three cases to consider when assigning a crane to handle it. First, when both the origin

Algorithm 1: 2NG *chromosome* decoder

Input: Ordered list χ of tuples {operation, gene, gene}

1 $\mathscr{L} \leftarrow emptylist$
2 $\mathscr{I} \leftarrow emptylist$
3 $\chi \leftarrow$ sorted list χ by OS genes ascending.
4 **foreach** tuple $(i, g_i, g_{i+n}) \in \chi$ **do**
5 $\quad \mathscr{W}_i \leftarrow$ list of feasible wagons of operation i
6 $\quad m \leftarrow |\mathscr{W}_i|$
7 $\quad$ **if** $m = 0$ **then**
8 $\quad\quad$ wagon $w_i \leftarrow yard$
9 $\quad$ **else**
10 $\quad\quad index \leftarrow \lfloor m(g_{i+n}) \rfloor + 1$
11 $\quad\quad$ wagon $w_i \leftarrow \mathscr{W}_i[index]$ // $e_i \leftarrow w_i$
12 $\quad$ set $C \leftarrow \mathscr{C}_{b_i} \cap \mathscr{C}_{e_i}$
13 $\quad$ **if** $|C| = 1$ **then**
14 $\quad\quad$ crane $c_i \leftarrow C[0]$
15 $\quad$ **else if** $|C| = 2$ **then**
$\quad\quad$ // $\mathscr{C}_{b_i} \cap \mathscr{C}_{e_i} = \{c, c'\}$ and $\eta_{c'} = \eta_c + 1$ **if** $h_{b_i} + h_{e_i} < H^-_{cc'} + H^+_{cc'}$ **then**
16 $\quad\quad\quad$ crane $c_i \leftarrow c$ // The crane on the left (crane c) is assigned to operation i
17 $\quad\quad$ **else**
18 $\quad\quad\quad$ crane $c_i \leftarrow c'$ // The crane on the right (crane c') is assigned to operation i
19 $\quad$ **else**
20 $\quad\quad$ New operation j
21 $\quad\quad$ wagon $w_j \leftarrow e_i$
22 $\quad\quad b_j \leftarrow yard$ // operation j's origin is yard and $h_{yard} = h_{w_j}$
23 $\quad\quad$ crane $c_j \leftarrow c \in \mathscr{C}_{w_j}$
24 $\quad\quad$ wagon $w_i \leftarrow yard$
25 $\quad\quad$ crane $c_i \leftarrow c \in \mathscr{C}_{b_i}$
26 $\quad\quad$ insert (j, w_j, c_j) to $\mathscr{I}$
27 $\quad$ insert (i, w_i, c_i) to $\mathscr{L}$
28 **return** $\mathscr{L}$, $\mathscr{I}$

and destination wagons are in the working area of one crane ($|\mathscr{C}_{b_i} \cap \mathscr{C}_{e_i}| = 1$), then the respective crane carries out the operation (Lines 13 – 14). The second case is when an operation's origin and destination wagons are both in the overlap area of two cranes c and c' ($\mathscr{C}_{b_i} \cap \mathscr{C}_{e_i} = \{c, c'\}$ and $\eta_{c'} = \eta_c + 1$), hence both cranes are able to handle the operation (Line 15). Operation i will be assigned to crane c on the left ($\eta_c < \eta_{c'}$) if the average of h coordinates of origin (h_{b_i}) and destination (h_{e_i}), is to the left of overlap area ($h_{b_i} + h_{e_i} < H^-_{cc'} + H^+_{cc'}$) (Line 16), otherwise crane c' on the right side of the overlap area is assigned to handle operation i (Line 18). The third case occurs when the operation is an *indirect move* ($|\mathscr{C}_{b_i} \cap \mathscr{C}_{e_i}| = 0$) (Line 19). In this situation two cranes and the buffer lane are necessarily involved in the operation. The crane in the origin area moves the container to the buffer lane where a ground vehicle is waiting to move the container to the destination area, and finally the crane in destination area moves the container from the ground vehicle to its final position.

Each *indirect move* imposes an additional operation to the crane schedules. 2NG *chromosome* decoder breaks the original operation i into two operations by generating a new operation j to represent the last segment of *indirect move* (Line 20) and modifying operation i to represent the first segment of *indirect move*. First, it sets the origin, destination and a crane to handle operation j (Lines 21–23), then destination of operation i is modified to *yard* (Line 24). After assigning a crane to handle operation i (Line 25), then the tuple comprised of operation j, its destination (w_j), and the crane to handle it (c_j) is inserted to list $\mathscr{I}$ to be inserted in the cranes schedule later (Line 26). Finally, operation i, its destination and crane are inserted in the list $\mathscr{L}$ (Line 27).

The extra operation j resulted by *indirect move* should be inserted into the schedule of the crane which executes the last segment of the *indirect move* (from buffer lane to the outbound train). The last segment of an indirect move may only be executed after a ground vehicle delivers the container to the second crane's working area. Operation j is inserted into the crane schedule by a local best fit search algorithm. The local best fit search, checks the certain range of transshipments after the container is available and inserts the movement into the best-fitting place in the crane schedule.

Through the dynamics of the genetic algorithm, the system learns the relationship between random-key vectors and solutions with good objective function values by feeding the quality of the solutions (chromosomes) back into the evolutionary process (problem independent part of the algorithm). The fitness function is computed by equation (2.1).

2.4.2 NGH algorithm

The NGH algorithm employs genetic algorithm to obtain operation sequence and a heuristic for assigning destination of operations. The chromosomes in the NGH algorithm have n genes, where n is the number of operation that must be carried out.

$$Chromosome = \left(\underbrace{gene_1, \ldots, gene_n}_{\text{OS}} \right)$$

Each gene is associated with an operation and the genes are employed to obtain sequence of operations. Algorithm 2 details the NGH *chromosome* decoder.

Determination of operations' sequence and crane assignment to execute the operations in the NGH algorithm are similar to the 2NG algorithm. Whereas, the NGH employs a greedy heuristic to assign destination wagon to operations. Equation (2.28) is employed as a greedy criterion to define an auxiliary cost, $D(i,w)$, for assigning wagons w to operations i.

$$D(i,w) = \omega_0\, t_{b_i w} + \omega_1\, R(i,w) \tag{2.28}$$

Algorithm 2: NGH *chromosome* decoder

Input: Ordered list χ of tuples {operation, gene, gene}

$\mathscr{L} \leftarrow emptylist$
$\mathscr{I} \leftarrow emptylist$
$\chi \leftarrow$ sorted list χ by OS genes ascending.
foreach tuple $(i, g_i, g_{i+n}) \in \chi$ **do**
  $\mathscr{W}_i \leftarrow$ list of feasible wagons of operation i
  $m \leftarrow |\mathscr{W}_i|$
  if $m = 0$ **then**
    wagon $w_i \leftarrow$ *yard*
  else
    wagon $w_i \leftarrow w \in \mathscr{W}_i$ which has minimum $D(i,w)$ // greedy criterion: $D(i,w)$
  set $C \leftarrow \mathscr{C}_{b_i} \cap \mathscr{C}_{e_i}$
  if $|C| = 1$ **then**
    crane $c_i \leftarrow C[0]$
  else if $|C| = 2$ **then**
    // $\mathscr{C}_{b_i} \cap \mathscr{C}_{e_i} = \{c, c'\}$ and $\eta_{c'} = \eta_c + 1$ **if** $h_{b_i} + h_{e_i} < H^{-}_{cc'} + H^{+}_{cc'}$ **then**
      crane $c_i \leftarrow c$ // The crane on the left (crane c) is assigned to operation i
    else
      crane $c_i \leftarrow c'$ // The crane on the right (crane c') is assigned to operation i
  else
    New operation j
    wagon $w_j \leftarrow e_i$
    $b_j \leftarrow$ *yard* // operation j's origin is yard and $h_{yard} = h_{w_j}$
    crane $c_j \leftarrow c \in \mathscr{C}_{w_j}$
    wagon $w_i \leftarrow$ *yard*
    crane $c_i \leftarrow c \in \mathscr{C}_{b_i}$
    insert (j, w_j, c_j) to $\mathscr{I}$
  insert (i, w_i, c_i) to $\mathscr{L}$
return $\mathscr{L}$, $\mathscr{I}$

ω_0 and ω_1 are the coefficients employed in objective function (2.2), $R(i,w)$ is wasted space of in wagon w when assigned to operation i. $R(i,w)$ equals zero if capacity of wagon w after moving operation's i container, l_{iw}, is greater than size of the smallest container remained to be scheduled. If l_{iw} is smaller than the size of all other containers, it is considered as wasted space.

$$R(i,w) = \begin{cases} 0 & \text{if } l_{iw} \geq \text{size of the smallest container} \\ l_{iw} & \text{if } l_{iw} < \text{size of the smallest container} \end{cases} \tag{2.29}$$

Similar to the 2NG, output of the NGH algorithm is list $\mathscr{L}$ which includes the sequence, crane and destination of operations and list $\mathscr{I}$ which contains extra operations resulted by *indirect move*. The operations listed in list $\mathscr{I}$ should be inserted into the cranes' schedules. A local best fit search algorithm is employed to insert these operations into the respected cranes. Note the last segment of an indirect move may only be executed after a ground vehicle delivers the container to the second crane's working area.

2.5. Computational experiments

The primary goal of the computational experiments is to validate the approach under various problem settings, thereby demonstrating the flexibility and robustness of the algorithms. To achieve this goal proposed algorithms are compared against a heuristic based on dispatching rules employed commonly in transshipment terminals. Secondly, the proposed algorithms are compared against one another to test the different solution representations.

This section is organized into three parts. First a set of benchmark instances inspired by a real-world terminal is introduced to enable validation and analysis of the algorithms and also encourage the other researchers in the field to compete with the proposed solution method (Section 2.5.1). The comparison of the algorithms 2NG, NGH and a heuristic based on dispatching rules is presented in Section 2.5.2, followed by a discussion on the weights of the objective function (see Equation (2.1)) in addition to an analysis of the instances' attributes in Section 2.5.3.

2.5.1 Instances

A set of instances is selected and modified from the literature (Souffriau et al., 2009) to be employed as SCOP instances. These instances are important and valid since they are inspired by a real-world terminal and by minor modifications are applicable to current research. The modified instances are employed to tune the algorithms. Then, a set of 45 instances is generated based on those from literature to carry out additional computational experiments. The new instances are generated so that with a certain number of operations there would be instances with various load factors which enable observing the impact of the load factor and number of operations on the performance of the algorithms.

The instances are generated inspired by a terminal with eight track, a yard and two cranes. The attributes of the generated instances are the following:

- Number of trains = 5;
- Number of tracks = 8;
- Number of cranes = 2 (working area of each crane is 60% of the yard, results in 20% overlap);
- Number of wagons in each train = 20;
- Wagon types: Two types of wagons are considered with sizes 60ft and 80ft long;
- Container sizes: Three sizes of containers are considered: small: 20ft, medium: 30ft, and large: 40ft. These are the most common sizes of containers in the industry.
- Train load factors: The load factor is defined as the total container length divided by the total train length. Nine levels of load factor are considered: $[0.1, 0.2, \ldots 0.9]$ with a step of 0.1, where 0.1 means trains are rather empty while 0.9 indicates trains are rather full.
- Number of operations: The benchmark set includes instances with nine levels of the number of operations from 17 to 255.

2.5.2 Comparison of proposed algorithms and the baseline

To validate the approach under various problem settings and test the performance of the proposed solution approach, both algorithms are compared against one another and also to a greedy constructive heuristic inspired by dispatching rules common in rail-rail transshipment terminals. Algorithm 3 details the greedy constructive SCOP (DR) algorithm. The DR algorithm starts by selecting a random operation i (Line 2), then assigns the nearest available wagon to operation i (Lines 4 – 5). The nearest crane to origin of operation i (b_i) is assigned to carried out operation i (Line 6). Then, (i, e_i, c)-tuple will be inserted to the list $\mathscr{L}$ (Line 7) which the sequence of tuples in it indicates the sequence of operations. Next operation will be the which its origin is nearest to destination of operation i (Line 10).

All algorithms presented in the previous section were implemented in C++, compiled using a gcc compiler and run on an Intel® processor with a clock speed of 2.40 GHz and 4.00 GB of RAM, under Linux operating system.

The *irace* package (López-Ibáñez et al., 2016) which implements the iterated racing for automatic algorithm configuration is employed to tune the algorithms. The configuration of tuned parameters is obtained by running the *irace* with the budget of 2000 runs. Table 2.3 details the parameters, their role, type, range and value per algorithm, obtained by *irace*.

The objective function (2.1) is a weighted sum of the total transshipment time and total size of non-allocated containers (E_{nc}), where ω_0 and ω_1 reflect the relative importance of C_{max} and E_{nc}, respectively. Given the importance of minimizing the number of containers remaining in the terminal due to limited capacity of the terminals and delivery time of the containers, minimizing the number of containers remaining in the terminal is more important than total transshipment

Algorithm 3: Greedy constructive SCOP (DR) algorithm

$\mathscr{L} \leftarrow$ empty list `// or empty ordered set`
operation $i \leftarrow$ random operation in $\mathscr{O}$
`// select a random operation from the list of operations` $\mathscr{O}$
while $\mathscr{O}$ is not empty **do**
  $W_i \leftarrow$ list of feasible wagons of operation i
  $e_i \leftarrow$ the nearest available wagon in W_i to the origin of operation i
  $c \leftarrow$ nearest crane to b_i `// greedy criterion: nearest crane to the origin of operation` i `is selected`
  insert (i, e_i, c) to $\mathscr{L}$
  remove o from $\mathscr{O}$
  if ($\mathscr{O}$ is not empty) **then**
    operation $i \leftarrow$ the operation which its origin is the nearest to the current location of the crane (e_i)
return list of tuples $\mathscr{L}$

Table 2.3: Tuning parameters by irace package

Parameter	Role	Type	Range	Value	
				2NG	NGH
$\gamma = \frac{p}{n}$	p is population size and n is chromosome length	integer	(1, 100)	20	54
max_g	Maximum number of generation	integer	(200, 3000)	464	2128
pe	Size of elite population	real	(0.10, 0.25)	0.124	0.127
pm	Size of mutant population	real	(0.02, 0.20)	0.088	0.134
ρ_e	Elite allele inheritance probability	real	(0.60, 0.90)	0.687	0.627

time. Hence, throughout this paper it is assumed that $\omega_0 = 1$ and $\omega_1 = 100$. However, it is also necessary to study the impact of weight parameters. A discussion on the impact of the weights follows in Section 2.5.3.

Results of the computational experiments are presented in Table 2.4, averaged over the values of ten runs with different random seeds, showing the total number of containers (#C), the number of operations (#O), the load factor (LF), the objective value (Obj), the total transshipment time (C_{max}), the total size of non-allocated containers (E_{nc}), and the cpu time for each instance and algorithm.

Table 2.4: Summary of the computational results for the three algorithms

Ins.	#C	# O	LF	DR algorithm			2NG algorithm				NGH algorithm			
				Obj	C_{max}	E_{nc}	Obj	C_{max}	E_{nc}	cpu	Obj	C_{max}	E_{nc}	cpu
1_17	29	17	0.1	806.0	806.0	0	**710.6**	710.6	0	8.5	771.0	771.0	0	11.3
2_17	58	17	0.2	880.0	880.0	0	**667.3**	667.3	0	8.7	675.0	675.0	0	10.9
3_17	82	17	0.3	837.0	837.0	0	**717.0**	717.0	0	8.6	787.0	787.0	0	10.4
4_17	116	17	0.4	1417.0	1417.0	0	**911.2**	911.2	0	8.3	927.0	927.0	0	9.6
5_17	138	17	0.5	2625.0	2625.0	0	**1275.0**	1275.0	0	8.7	1283.5	1283.5	0	9.4
6_17	170	17	0.6	1915.0	1915.0	0	**1166.8**	1166.8	0	8.0	1283.5	1283.5	0	8.9
7_17	199	17	0.7	2636.0	2636.0	0	**1300.5**	1300.5	0	6.9	1336.0	1336.0	0	8.2
8_17	214	17	0.8	10594.0	3594.0	70	**7653.3**	1653.3	60	5.5	7782.5	1782.5	60	5.8

(continued on next page)

Ins.	#C	# O	LF	DR algorithm			2NGalgorithm				NGH algorithm			
				Obj	C_{max}	E_{nc}	Obj	C_{max}	E_{nc}	cpu	Obj	C_{max}	E_{nc}	cpu
9_17	255	17	0.9	25942.0	2942.0	230	**19518.6**	2518.6	170	9.0	19595.0	2595.0	170	7.5
2_42	58	42	0.2	2912.1	2912.1	0	**1360.0**	1360.0	0	65.7	1398.0	1398.0	0	58.7
3_42	82	42	0.3	3730.1	3730.1	0	1878.8	1878.8	0	57.7	**1376.0**	1376.0	0	56.1
4_42	116	42	0.4	2713.1	2713.1	0	1843.1	1843.1	0	61.8	**1753.0**	1753.0	0	53.5
5_42	138	42	0.5	2515.0	2515.0	0	**1983.1**	1983.1	0	54.1	2014.0	2014.0	0	49.8
6_42	170	42	0.6	3693.0	3693.0	0	2743.2	2743.2	0	63.6	**2683.0**	2683.0	0	47.2
7_42	199	42	0.7	6581.1	6581.1	0	3488.4	3488.4	0	48.6	**2961.0**	2961.0	0	42.3
8_42	214	42	0.8	18044.2	8044.2	100	7209.2	5209.2	20	55.4	**6557.0**	4557.0	20	32.9
9_42	255	42	0.9	48382.0	7382.0	410	**22944.4**	6944.4	160	54.1	23133.3	7133.3	160	86.7
3_62	82	62	0.3	4925.1	4925.1	0	2356.0	2356.0	0	140.7	**1635.0**	1635.0	0	122.3
4_62	116	62	0.4	3731.1	3731.1	0	3096.0	3096.0	0	155.7	**1968.0**	1968.0	0	112.9
5_62	138	62	0.5	6325.1	6325.1	0	3464.0	3464.0	0	145.5	**2516.0**	2516.0	0	107.4
6_62	170	62	0.6	4879.2	4879.2	0	4156.0	4156.0	0	164.0	**3256.0**	3256.0	0	104.1
7_62	199	62	0.7	7209.2	7209.2	0	4472.0	4472.0	0	127.4	**3466.0**	3466.0	0	94.6
8_62	214	62	0.8	23776.2	16776.2	70	9896.0	7896.0	20	170.3	**7826.0**	5826.0	20	92.4
9_62	255	62	0.9	56926.1	7926.1	490	28390.0	11390.0	170	147.7	**27427.4**	10427.4	170	155.8
4_85	116	85	0.4	3938.1	3938.1	0	3262.7	3262.7	0	281.0	**2410.0**	2410.0	0	217.0
5_85	138	85	0.5	5073.2	5073.2	0	4338.0	4338.0	0	355.8	**2922.0**	2922.0	0	227.9
6_85	170	85	0.6	9013.2	9013.2	0	5482.0	5482.0	0	327.7	**3548.0**	3548.0	0	247.6
7_85	199	85	0.7	11279.1	11279.1	0	7434.0	7434.0	0	302.2	**3889.8**	3889.8	0	239.0
8_85	214	85	0.8	27029.5	13029.5	140	9241.2	9241.2	0	314.3	**6697.0**	6697.0	0	174.8
9_85	255	85	0.9	83638.4	17638.4	660	35269.6	22169.6	131	368.9	**33285.3**	16285.3	170	253.0
5_97	138	97	0.5	6042.2	6042.2	0	5046.1	5046.1	0	390.4	**3011.0**	3011.0	0	275.1
6_97	170	97	0.6	10324.2	10324.2	0	7434.3	7434.3	0	339.4	**4511.0**	4511.0	0	321.2
7_97	199	97	0.7	15641.4	15641.4	0	8017.0	8017.0	0	320.4	**4866.0**	4866.0	0	322.8
8_97	214	97	0.8	34039.5	16039.5	180	13530.6	13530.6	0	370.8	**8072.0**	8072.0	0	291.3
9_97	255	97	0.9	100170.5	32170.5	680	39603.3	22503.3	171	373.9	**36977.0**	19977.0	170	287.2
6_127	170	127	0.6	12569.1	12569.1	0	11320.1	11320.1	0	407.3	**5894.0**	5894.0	0	375.0
7_127	199	127	0.7	18066.7	15066.7	30	11925.3	11925.3	0	420.2	**6321.0**	6321.0	0	381.0
8_127	214	127	0.8	60197.5	19197.5	410	31570.3	31570.3	0	486.5	**12149.0**	12149.0	0	411.9
9_127	255	127	0.9	108976.1	22976.1	860	49632.2	22632.2	270	513.8	**45822.2**	17822.2	280	386.6
7_148	199	148	0.7	54232.5	39232.5	150	11609.0	11609.0	0	551.1	**6794.0**	6794.0	0	434.8
8_148	214	148	0.8	53374.5	29374.5	240	23674.2	23674.2	0	554.6	**10933.0**	10933.0	0	401.2
9_148	255	148	0.9	115920.4	25920.4	900	57329.3	34329.3	230	667.4	**43261.8**	20261.8	230	492.9
8_166	214	166	0.8	56558.2	25558.2	310	43324.9	40324.9	30	873.3	**19496.0**	16496.0	30	594.5
9_166	255	166	0.9	113711.1	31711.1	820	52064.8	27964.8	241	896.7	**42945.8**	19945.8	230	579.7
9_202	255	202	0.9	138805.1	32805.1	1060	62184.2	26184.2	360	1336.3	**59329.0**	23329.0	360	650.4

The results show the 2NG algorithm outperforms the other algorithms when the number of operations is low ($\#O = 17$), as the number of operations increases the 2NG algorithm no longer outperforms the other algorithms and it competes with the NGH when the $\#O = 42$. However for larger instances ($\#O \geq 62$) the NGH outperform the other algorithms. Although the 2NG performs best for small instances, considering the size of real-world instances the NGH represent the best option for solving real-world size instances.

Figure 2.4 illustrates how both proposed algorithms improve the performance of the DR heuristic. The horizontal axis details the number of operations. The improvements over the DR are, represented on the vertical axis. The improvements are averaged over all instances with equal

number of operations. For small instances (#$O = 17$), the 2NG algorithm yields highest improve-

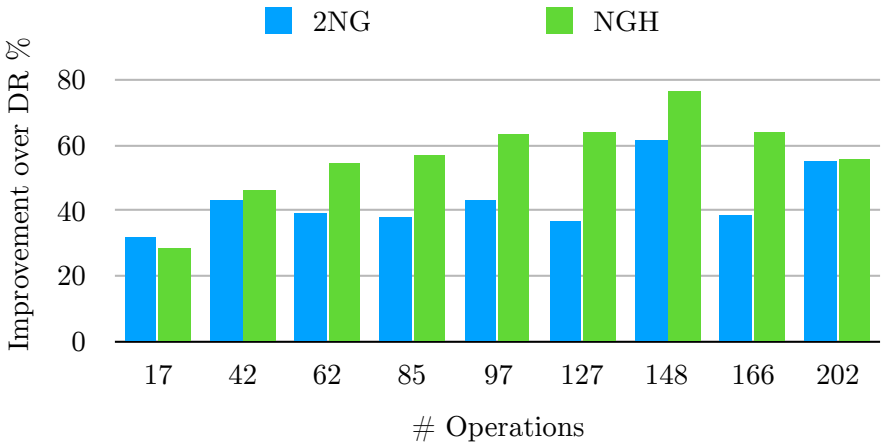


Figure 2.4: Average improvement of proposed algorithms over greedy heuristic based on dispatching rules (DR).

ment over the DR. By growing the number of operations the NGH results in higher improvement (#$O \geq 42$), it reaches to the highest average improvement, 76%, for instances sized #$O = 148$. The results show algorithms 2NG and NGH improved the DR in average 40% and 52%, respectively.

2.5.3 Insights and discussion

The impact of weight parameters ω_0 and ω_1 on the objective function is investigated by testing three algorithms (2NG, NGH and DR) over instance "9_166". The selected instance has trains 90% full of containers (load factor = 0.9) and high number of operations (#$O = 166$) which is able to represent the contrast of the both parts of the objective function C_{max} and E_{nc} (for instances with lower load factor E_{nc} is not significant as there will be ample space to put all the containers on the outbound train). Following experiments illustrate the impact of adjusting ω_1 in the range of $(0, 1, ..., 1E{+}5)$ incrementing exponentially while ω_0 is set to 1.

For $\omega_1 = 0$ the algorithms only optimize the transshipment time as the objective function equals to C_{max} and for $\omega_1 = 1E{+}5$ the algorithms are optimizing the size of non-allocated containers as $\omega_1 E_{nc}$ is relatively much larger than $\omega_0 C_{max}$.

Figures 2.5, 2.6, and 2.7 illustrate the behavior of the 2NG, NGH and DR algorithms with respect to C_{max}, E_{nc}, and total cost.

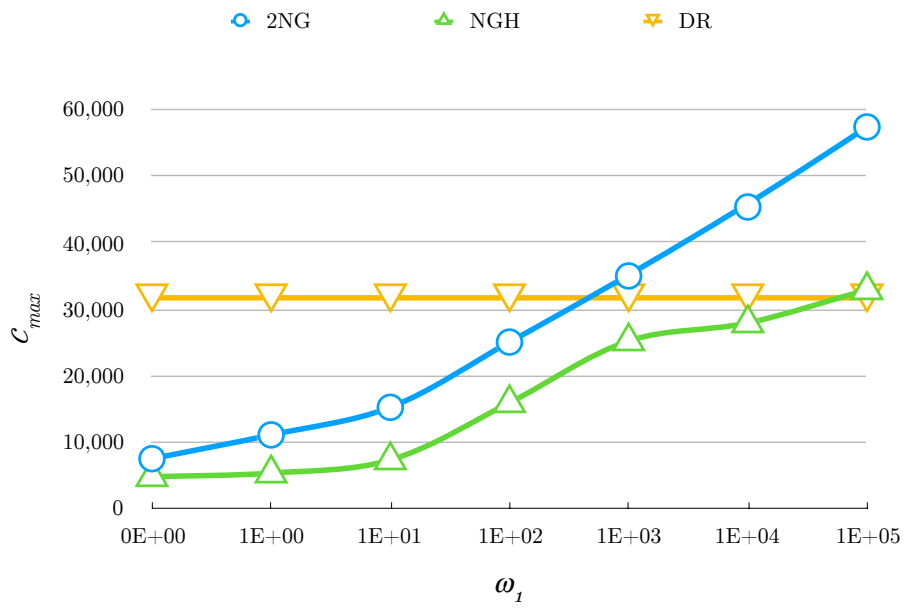


Figure 2.5: Impact of adjusting ω_1 on C_{max}.

The C_{max} value grows by increasing ω_1 (Figure 2.5) for both proposed algorithms, when $\omega_1 = 1000$ the 2NG goes above the DR (yields higher C_{max}) while the NGH remains below the DR for all values of ω_1.

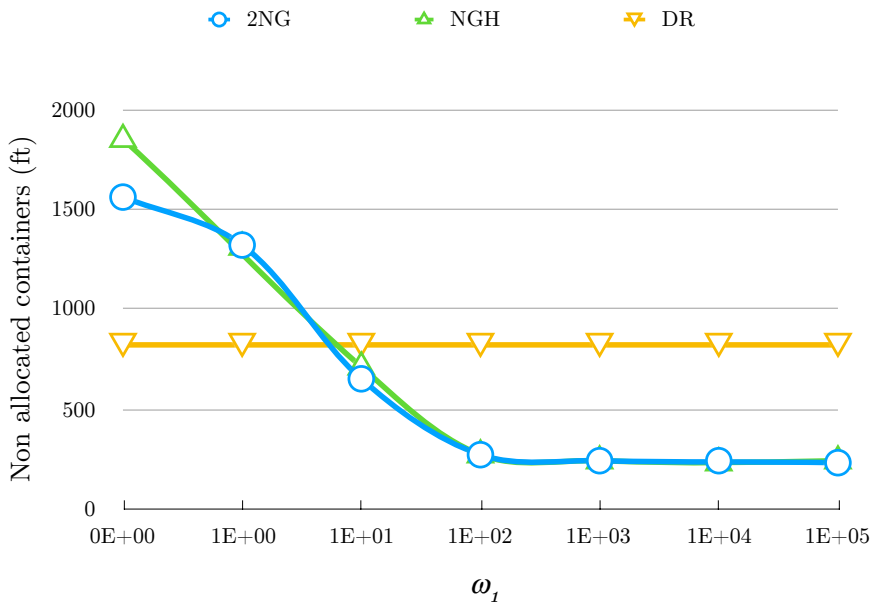


Figure 2.6: Impact of adjusting ω_1 on E_{nc}.

Figure 2.6 shows how E_{nc} decreases by increasing ω_1. Note that for $\omega_1 \geq 100$, E_{nc} remains constant for both proposed algorithms.

Figure 2.7 illustrates the changes of objective function by increasing the value of ω_1. It is evident that by increasing ω_1 total cost increases and for large values of ω_1 ($\omega_1 \geq 1000$), the 2NG

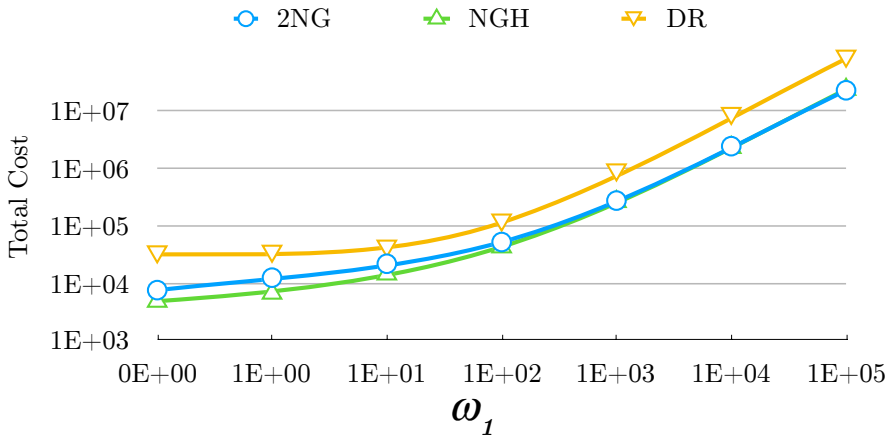


Figure 2.7: Impact of adjusting ω_1 on total cost.

and NGH yield similar total cost, since both have equal E_{nc} (see Figure 2.7). It is worth mentioning that NGH shows the lowest total cost for all values of ω_1.

To better illustrate the trade off between C_{max} and E_{nc}, parameter α is defined. Equation (2.30) defines the value of α, where $minC$ is the minimum value of C_{max} obtained by all algorithms when $\omega_1 = 0$, and $minE_{nc}$ is the minimum value of E_{nc} when $\omega_1 = 1E{+}5$.

$$\alpha = \frac{C_{max}}{minC} + \frac{E_{nc}}{minE_{nc}} \tag{2.30}$$

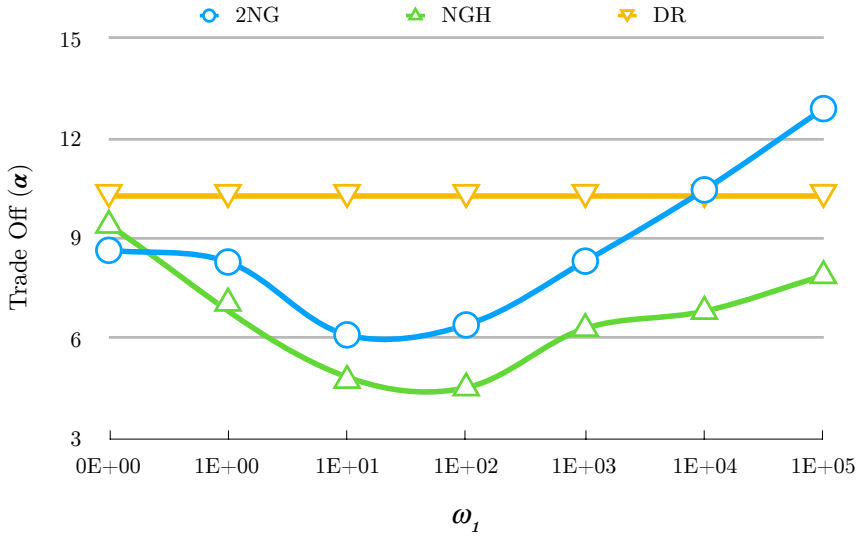


Figure 2.8: Impact of adjusting ω_1 on total cost.

Figure 2.8 shows how α varies for the algorithms $2NG$, NGH, and DR for different values of ω_1. The values used for parameters $minC$ and $minE_{nc}$ are 4837.6 (obtained by $NGH||\omega_1 = 0$) and 230 (obtained by $NGH||\omega_1 = 1E{+}4$ and $2NG||\omega_1 = 1E{+}5$) respectively. The minimum value of $\alpha = 4.52$ is obtained when $\omega_1 = 100$ by the NGH algorithm. Theoretically, the minimum value that α can get equals 2 (when $c_{max} = minC$ and $E_{nc} = minE_{nc}$), however in practice it is not

achievable since having one objective minimized (either c_{max} or E_{nc}) requires compromise on the other objective.

Figure 2.9 illustrates the average *cpu* time of the both algorithms when varying the number of operations. It highlights how, for the NGH algorithm, the computational time increases linearly when increasing the number of operations.

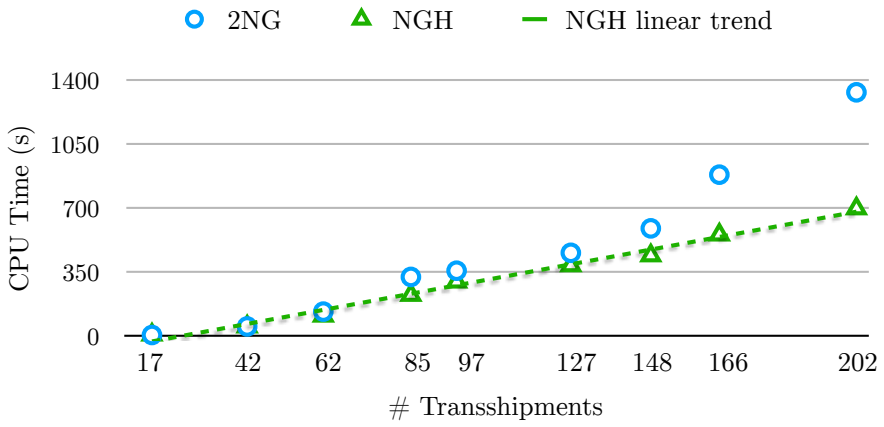


Figure 2.9: Average solution time of algorithms.

Finally, impact of the overlap area's size upon total transshipment time (C_{max}) is examined. The size of overlap area (SOA) is the percentage of the yard shared between two cranes. Three sets of instances are employed: *low*, *medium*, and *high* load factor, each set includes three instances. Figures 2.10, 2.11, and 2.12 depict the relative C_{max} value of NGH for SOA varying from 0% to 50% with increasing step of 10% for instances with low, medium, and high load factor, respectively. The relative C_{max} value is defined as the ratio of C_{max} for different SOA to the minimum C_{max} of all SOAs.

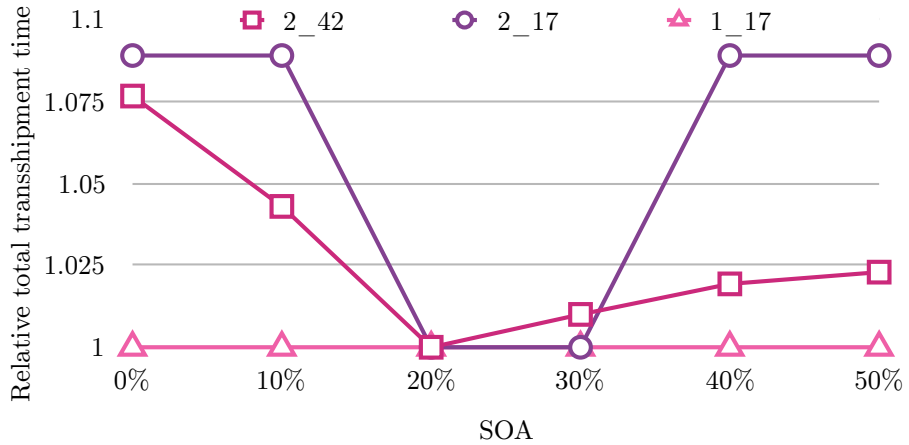


Figure 2.10: Impact of SOA on low load factor instances.

The results indicate that when the number of transshipments is low (#T = 17), SOA does not have a significant impact on the performance of the cranes. However when the yard is heavily loaded (load factor = 90%), SOA $\geq$ 40% results in higher C_{max}. In this situation (high load

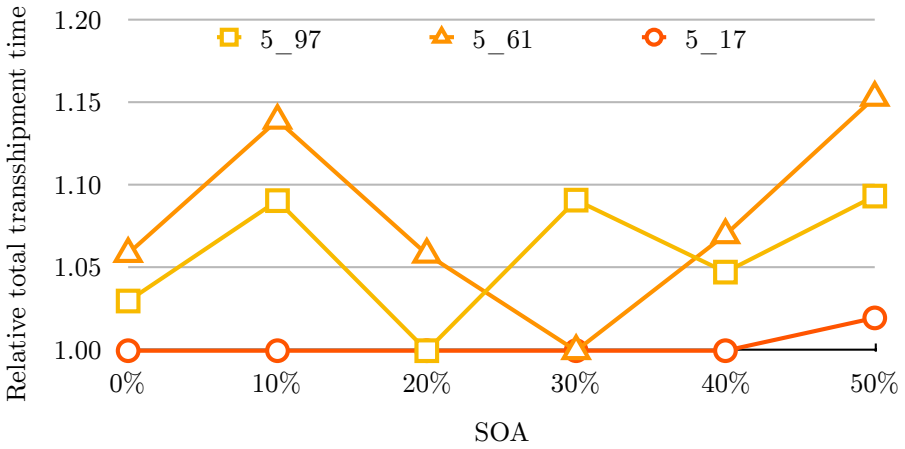


Figure 2.11: Impact of SOA on medium load factor instances.

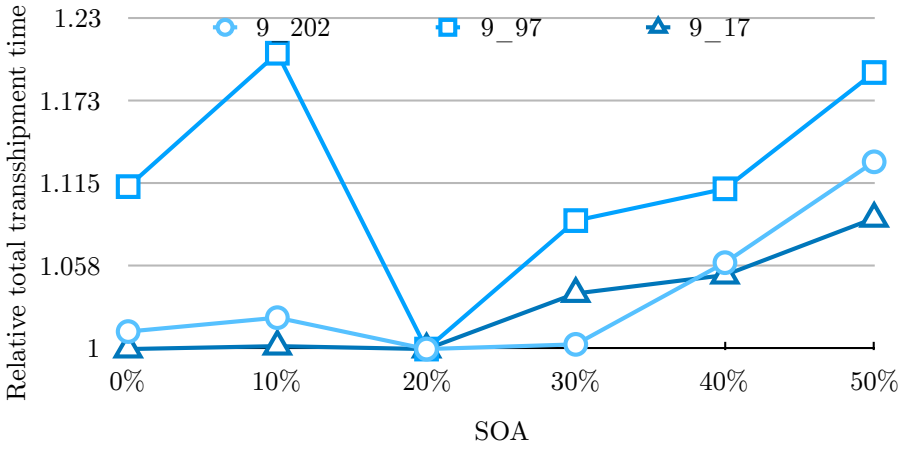


Figure 2.12: Impact of SOA on high load factor instances.

factor and low number of transshipments), reducing the SOA reduces the idle time of cranes, and eventually improves the C_{max}. For instances with low and medium load factor (Figures 2.10 and 2.11), there is not a significant difference in the C_{max}. In cases of instances with high load factor (Figure 2.12), when the SOA is very small (SOA = 10%) or very large (SOA = 50%), C_{max} increases significantly by almost 20%. By contrast, one can conclude that SCA = 20% results in a better trade-off between the idle time of cranes and the number of indirect moves for instances with high load factor and large number of transshipments.

2.6. Conclusions

This section proposed a new approach to solve the scheduling crane operations problem in rail-rail transshipment terminals. In this approach wagon assignment for containers, assignment of container move operations to cranes and sequencing operations per crane were considered simultaniously and addressed jointly.

Two different algorithms, namely the 2NG and NGH based on the biased random key genetic algorithm (BRKGA) were proposed. Each algorithm builds on a different chromosome (solution

representation). The 2NG with $2n$ genes and the NGH with n genes for each chromosome where n is the number of operations.

The algorithms' performance and robustness were tested by comparing them against an algorithm based on dispatching rules and also against one another. A set of 45 instances generated inspired by real terminal. The results show that proposed algorithms 2NG and NGH improved dispatching rules by 43% and 57% in average respectively, which indicates that there is a significant room to improve the common performance in the terminals. Comparing the proposed algorithms against one another show the NGH algorithm outperforms the 2NG algorithm for medium and large instances, while the 2NG algorithm yields a better solutions for small instances ($\#O = 17$). The experiments also indicate that for instances with low and medium load factor, the size of overlap area has no significant impact on total transshipment time, whereas, to achieve better transshipment times, the size of overlap area should be around 20%.

Future research directions that may be explored include addressing related crane operation scheduling problems that exist within rail-road transshipment terminals which serve as the interface between two separate modes of transportation. In these terminals gantry cranes span all tracks, storage areas and truck lanes, thereby enabling the transshipment of containers between multiple trains, storage areas and trucks. Effectively scheduling crane operations, as demonstrated in this paper, has an important impact on the day-to-day operation of such terminals.

Chapter 3

Crane-operated warehouse scheduling: Integrating location assignment and crane scheduling

Crane-operated warehouses constitute an essential asset for the many industries which must temporarily store products on their way from manufacturers to consumers. Such warehouses are a practical necessity rather than an explicitly desired service and they introduce significant operational costs which should be minimized. The problem addressed by the current paper, the Crane-operated Warehouse Scheduling Problem (CWSP), concerns the location assignment of input products and the scheduling of cranes for product movement in such warehouses. Several constraints are associated with the problem, for example certain products should not be stored close to each other (due perhaps to a difference in temperature or aroma) and cranes must respect operational safety distances between each other in order to prevent dangerous collisions. The present paper explores a novel methodology which combines these two decisions – location assignment and crane scheduling - instead of solving them sequentially. In addition to mathematical formulations for location assignment and crane scheduling, both an integrated mathematical formulation and a fast heuristic are presented for the CWSP. The quality of the mathematical formulation and the heuristic are compared against the conventional sequential approaches. Experimentation upon an extensive range of instances show significantly improved results are attainable when integrating location assignment and crane scheduling, despite some (expected) increase in computational time.

3.1. Introduction

Warehouses constitute a form of infrastructure commonly employed by manufacturers, wholesalers and retailers to store goods not only during the production process but also during their

distribution. Warehouse efficiency therefore plays a crucial role in global economy. Their efficiency enhances the capacity of supply chains, providing significant economic and service benefits to both businesses and end users. Reducing storage and handling costs, increasing warehouse capacity and improving the timeliness of deliveries are essential to further sustaining and strengthening supply chains. The present study focuses on the first aspect, in particular in *Crane-operated warehouses*.

The term 'Craned-operated warehouse' refers to a type of warehouse or storage area which employs any type of overhead crane such as rubber-tired gantry cranes (RTGCs) or rail mounted gantry cranes (RMGCs). Overhead cranes are commonly used in industrial warehouses where the stored products are rather heavy and large-sized in nature (examples of which include steel coils, large pallets of goods, ...) or in container terminals where containers are temporarily stored in stacks before being transferred to their next destinations. The Crane-operated Warehouse Scheduling Problem (CWSP) studied throughout this paper concerns the optimization of both where to store products and the crane operations which are necessary to do so in warehouses which employ such overhead cranes. A warehouse typically consists of a set of input and output points which are located in the periphery of a storage area. Products are stored subject to a range of operational constraints and a set of cranes are employed for handling operations. In many cases, cranes cannot overtake each other (such as when they are operating on the same pair of rails), thereby necessitating proper safety measures to avoid collisions. The CWSP as such is composed of two constituent optimization problems, namely:

i) The Location Assignment Problem (LAP): assigning the storage locations to incoming products and those which must be relocated within the storage area.

ii) The Crane Scheduling Problem (CSP): scheduling the cranes' operations.

The objective is to minimize both total storage cost and tardiness of crane operations.

The CWSP is conventionally split into the two aforementioned sub-problems – the LAP and CSP - which are solved sequentially. First, the LAP is solved and the resulting storage locations for incoming/relocated products are fixed. Next, the CSP is solved to determine the best schedule for the handling operations. To date, there has been a considerable lack of research which assesses the impact of integrating these two sub-problems.

Container terminals represent one specific real-world application where the CWSP is encountered. Given the continuously increasing volume of containers being handled in terminals worldwide, which places significant pressure on terminals' infrastructure and operations, it is unsurprising that there exists a vast body of container terminal literature relevant to the problem.

The majority of studies related to the LAP involve optimization problems in container terminals such as the re-handling problem (Jovanovic and Voß, 2014; Ku and Arthanari, 2016) and the container stacking problem (Zhang et al., 2014; Gharehgozli et al., 2014). The re-handling problem concerns removing containers from stacks to enable a given set of container retrievals where the objective is to minimize the number of moves. Studies addressing the container stacking problem mostly focus on minimizing reshuffling, namely those unproductive moves required to gain

access to a desired container which is blocked (Chen and Lu, 2012; Boysen and Emde, 2016). Other objectives include minimizing travelling distance, wasted space, or estimated retrieval cost (Park et al., 2011).

The objective function of the LAP in the present study derives itself directly from operational practices found in production industries and differs from objective functions found in references related to container terminals. It includes cost terms related to storing a product in a specific location and others related to storing certain products adjacent to one another. The former cost terms are used to model the retrieval costs (specified as a distance from an output point), while the latter model operational constraints of production industries which seek to avoid storing certain products in close proximity. For instance, companies may wish to avoid storing aromatic products next to each other or may require hot products to be stored away from those which have already cooled down. While companies may often disallow such neighbouring location assignments altogether, in situations of high storage occupancy it may not always be feasible to do so. Addressing such situations as soft constraints, penalized as costs in the objective function, enables the necessary modelling flexibility and avoids infeasibility.

Many studies address the CSP independent from the LAP, considering the LAP's solution as a fixed input. The most relevant references to the present study are those focused on scheduling multiple cranes. For single crane scheduling, interested readers are referred to the survey by Boysen and Stephan (2016). Due to the increasing necessity to accelerate handling operations in warehouses, many recent papers have focused on scheduling multiple cranes operating simultaneously within the same storage area. Dorndorf and Schneider (2010) studied a container yard in which a pair of cranes operates on the same rails with another larger crane operating above them (cross-over crane) on its own pair of rails. Each of the two smaller cranes has its own distinct working area to avoid collisions. Given independent and mutually-exclusive working areas and the presence of a separate cross-over crane, crane interference does not pose a problem in such yards. By contrast, Li et al. (2009) considered a container terminal which employs multiple cranes that may interfere with one another. They proposed a discrete-time MIP model for the problem and a heuristic to solve it. Li et al. (2012) extended Li et al. (2009)'s work by proposing a continuous time MIP model capable of handling instances with a higher number of storage and retrieval requests. Wu et al. (2015) also considered a container terminal with multiple cranes, as Li et al. (2009, 2012), and proposed a polynomial time heuristic to solve their optimization problem. These studies are particularly interesting with regard to how they model the scheduling of multiple cranes operating in storage areas with inter-crane interference. However a noteworthy and significant difference with respect to the present work lies in how within all the aforementioned studies containers are delivered. This means cranes remain static at the stacking piles and do not move during handling operations. Consequently, the duration of all operations can be assumed to be equal. This simplifies modeling the problem by enforcing equal time durations for all operations. In a general setting, however, input and output may occur anywhere around the storage area and cranes move over that storage area while handling products. Gharehgozli et al. (2017) investigated a set of rules and their influence on the effect of temporary locations in a so-called *handshake*

area which facilitates container handover between cranes. The paper presented some managerial insights on the size, location, and number of such handshake areas.

Gharehgozli et al. (2015) attempted to integrate location assignment and crane scheduling problems in a container terminal, wherein the water-side crane performs all requests which must be stacked or retrieved from the water-side, and land-side operations are carried out similarly by a land-side crane. However a significant limitation to their model is that storage and retrieval requests are already assigned to cranes in advance. Moreover, the model was designed for only one land-side and one water-side crane, and thus cannot accommodate cases with more than two cranes, or cases where both cranes may handle requests from anywhere throughout the storage area.

In practice, warehouse managers are becoming increasingly aware that warehouse efficiency may be bolstered by exploiting an integrated optimization approach, where location assignment and crane scheduling decisions are simultaneously taken into consideration and jointly optimized (Darvish and Coelho, 2018). The literature is however lacking studies that investigate this. The present research, therefore, focuses on this integrated approach of handling the CWSP. It provides a general setting which may be easily adapted to other warehouses, land-side container terminals or any other industry employing multiple gantry cranes for product handling.

Mathematical formulations and heuristics are developed and tested upon a set of instances which are randomly-generated using probability distributions and insights extracted from a relevant industrial case. Results are compared against those obtained with a heuristic based on the dispatching rules and manual strategies employed in practice. The findings from this computational study reveal the significant benefits of combining the LAP and CSP when solving the CWSP.

The remainder of the paper is structured as follows. Section 3.2 provides a detailed problem definition of the CWSP. Section 3.3 presents mathematical formulations for the LAP and the CSP, and also formulates the CWSP by means of a continuous-time mixed integer programming model which considers realistic constraints. Section 3.4 presents a heuristic algorithm for solving the LAP, CSP and CWSP. Computational experiments and a comparative algorithmic performance analysis are detailed throughout Section 3.5. Finally, Section 3.6 summarizes the paper's primary findings and discusses possible future research directions.

3.2. Problem definition

Throughout this study, a crane-operated warehouse is considered which consists of a storage area within which products are placed. The storage area is composed of locations, with each location storing at most one product. A set of special locations representing input/output (I/O) points around the storage area is defined where input requests originate and output requests must be delivered. Each I/O point either originates input or collects output requests which must be processed by their due time.

Each request consists of a product that must be moved. Requests are divided into two sets:

- *Input requests* ($\mathscr{R}^I$): requests which require location assignment. $\mathscr{R}^I$ consists of requests for products at an input point requiring transfer to the yard or products that must be moved within the yard to enable cranes to access locations, located beneath them, associated with *output requests*;
- *Output requests* ($\mathscr{R}^O$): requests consisting of products within the yard requiring transfer to an output point.

Set $\mathscr{R}$ represents the union of the two sets: $\mathscr{R} = \mathscr{R}^I \cup \mathscr{R}^O$. A release time and due time are associated with each request, defining when the product is available for transfer and when it is due to be transferred.

The set of available locations L consists of locations that are already free or will become free during the scheduling horizon when their stored product has been moved. This includes the origin location of the *output* requests and those requests which move products inside the yard. L excludes locations which store products that will not be moved during the scheduling horizon. Following convention, the storage area length is mapped to a horizontal coordinate axis. A horizontal coordinate h_l is associated with each individual location $l \in L$. The horizontal coordinates in the yard are ordered from left to right. A product may be stored in a location above ground level, stacked on another product. Therefore, in addition to its horizontal and lateral coordinates, a location l is also defined by its level above the ground. To be able to store a product in a location above ground level, all locations beneath the product must be occupied by other products. Therefore, in addition to its horizontal and lateral coordinates, a location l is also defined by its level above the ground. To be able to store a product in a location above ground level, all locations beneath the product must be occupied by other products. Cranes are employed to execute input and output requests. A set of available, identical cranes $\mathscr{C}$ is defined, each being capable of handling one request at a time. Cranes are mounted on a pair of rails along the horizontal axis, and are ordered and indexed from left to right in the storage area. Additionally, cranes have no predefined working areas, the only restriction being that they cannot cross and that a safety distance must be respected between neighbouring cranes while moving throughout the storage area. This study assumes that cranes can reach all locations.

Figure 3.1 illustrates a top-view of a crane-operated storage area in which the gray border represents the input/output points. Three cranes are ordered from left to right and operate across the storage area. Note that the safety distance must be respected and therefore, cranes cannot pass over each other.

The CWSP consists of two optimzation problems, the LAP and the CSP. The LAP's objective is to minimize the total storage cost of *input* requests. The storage cost for a product is defined in terms of an assignment in the neighbourhood of other products in the storage area. The total storage cost includes the cost, summed over all input requests, of assigning an *input* request to a location in the storage area (pre-calculated and corresponding with the distance to neighbouring products which will not move during the scheduling horizon) and the cost of assigning two *input* requests in neighbouring locations.

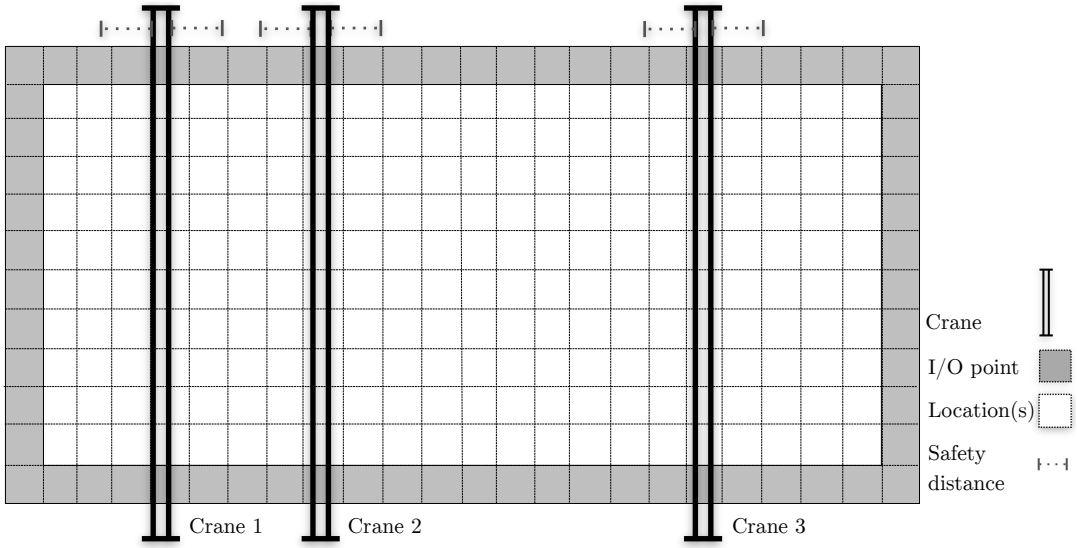


Figure 3.1: Top view of a warehouse employing three cranes.

The CSP consists of deciding when and by which crane each request will be executed, while respecting precedence constraints and constraints concerning safety distances. Precedence constraints may be predetermined or introduced during location assignment. Predetermined precedence constraints follow from when a product is stacked on top of a product associated with an *output* request, the top product must be removed first after which the output request may be executed. Precedence constraints introduced during location assignment follow from assigning an input request to the location of a request originated within the yard (either input or output) or assigning two input requests on top of each other. The objective is to minimize total tardiness of all requests. A request's tardiness equals the difference between its completion time and due time if positive, or zero otherwise.

The combined problem of solving both the LAP and CSP simultaneously is referred to as the CWSP. The objective of the CWSP is to minimize the weighted linear expression presented in Equation (3.1), where α and β are weights defining the relative importance of the terms, while E_{LA} and E_{CS} correspond to total storage cost and total tardiness, respectively.

$$\text{Total cost} = \alpha \cdot E_{LA} + \beta \cdot E_{CS} \tag{3.1}$$

3.3. Mathematical formulation

This section presents mathematical formulations for the LAP (Section 3.3.1) and CSP (Section 3.3.2), followed by a formulation for the CWSP (Section 3.3.3) which considers the LAP and CSP simultaneously.

Table 3.1: Notations for the LAP formulation

Sets:

$\mathscr{R}$:	set of all requests
$\mathscr{R}^I$:	set of input requests, $\mathscr{R}^I \subseteq \mathscr{R}$
L :	set of available locations, those locations that are currently empty or will become empty due to product movements.
U_l :	set of all available locations underneath location l, $U_l \subset L$
N_l :	set of all neighbouring locations of location l, $N_l \subset L$

Parameters:

ω_{ij} :	storage cost of assigning request i to a neighbouring location of product associated with request j
γ_{il} :	storage cost of assigning request i to location l
b_i :	origin location of request i

Decision variables:

x_{il} :	binary variable equal to 1 if request i is assigned to location l and 0 otherwise
z_{ij} :	binary variable equal to 1 if request i is assigned to a neighbouring location of request j's destination and 0 otherwise

3.3.1 Location assignment problem

Formulation $\mathscr{F}_{LA}$ concerns the assignment of destination locations to input requests. Table 3.1 summarizes the notation employed for the LAP formulation.

$$
\mathscr{F}_{LA}\begin{cases}
min \sum_{i\in\mathscr{R}^I}\sum_{l\in L}\gamma_{il}x_{il} + \sum_{i\in\mathscr{R}^I}\sum_{j\in\mathscr{R}^I}\omega_{ij}z_{ij} & & (3.2)\\
s.t. \quad \sum_{l\in L: l\neq b_i} x_{il} = 1 & \forall\, i\in\mathscr{R}^I & (3.3)\\
\sum_{i\in\mathscr{R}^I} x_{il} \leq 1 & \forall\, l\in L & (3.4)\\
x_{il} \leq \sum_{j\in\mathscr{R}^I} x_{jk} & \forall\, i\in\mathscr{R}^I,\ l\in L,\ k\in U_l & (3.5)\\
x_{il} + x_{jk} \leq 1 + z_{ij} & \forall\, i,j\in\mathscr{R}^I,\ l\in L,\ k\in N_l & (3.6)\\
x_{il}\in\{0,1\} & \forall\, i\in\mathscr{R}^I,\ l\in L & (3.7)\\
z_{ij}\in\{0,1\} & \forall\, i,j\in\mathscr{R}^I, & (3.8)
\end{cases}
$$

Objective function (3.2) minimizes the total storage cost. The storage cost is divided into two parts: the cost of assigning request i to location l and the cost of assigning requests i and $j \in \mathscr{R}^I$ in each other's neighbourhood, denoted by ω_{ij} and γ_{il} respectively. Constraints (3.3) and (3.4) are classic assignment constraints ensuring exactly one location is assigned to each input request and that each location receives, at most, a single request, respectively. Constraints (3.3) also prevent assigning *input* requests to their origin locations. If request i represents a product that must be moved within the yard to access a product below, other input requests may use b_i as their destination, after the product below b_i has been moved. Constraints (5) force all *available* locations underneath location $l \in L$ to have an *input* request assigned, thus ensuring that no product is stacked atop an empty location. It is sufficient to assure there is an *input* request assigned to

each *available* location $k \in U_l$, since the products not associated with any requests will not be moved during the scheduling horizon. Constraints (3.6) set the value of z_{ij} to 1 if requests i and j are placed in neighbouring locations and 0 otherwise. Constraints (3.7) and (3.8) state variables x_{il} and z_{ij} are binary.

Following the LAP, precedence constraints may be implied when input requests are assigned to the origin locations of output requests or when two input requests are assigned to locations where one is on top of the other. The outcome of the LAP (destination location for input requests) along with the set of precedence requests is the input for the CSP.

3.3.2 Crane scheduling problem

Formulation $\mathscr{F}_{CS}$ models the CSP which considers the crane assignment for requests and the sequencing of requests per crane. $\mathscr{F}_{CS}$ implements various realistic operational constraints such as multiple cranes working simultaneously in the storage area and precedence constraints. Note that each crane can traverse the entire storage area provided safety distances between all cranes are respected. This means that a crane may move beyond the storage area boundary to provide space for another crane to access storage location at or close to the area's perimeter.

The continuous-time formulation for the CSP presented in this paper was inspired by Li et al. (2012), who showed that for the CSP with multiple cranes this formulation significantly reduced the model's size and enabled larger instances to be solved compared to a discrete-time formulation for the same problem. Recall from Section 3.1 that Li et al. (2012) considered the CSP in a container terminal where containers were brought directly in front of the stacking pile. As a consequence, cranes do not move along the rails when moving a product. They instead move products laterally (along the crane beam). However, in a general setting of crane-operated warehouses the cranes move along the storage area to reach the respective input/output point during their operations. Conflicting requests and variable operation durations are consequently inevitable. The model presented in the following section accounts for this additional complexity. Table 3.2 summarizes the notation employed to formulate the CSP.

The constraints of $\mathscr{F}_{CS}$ are organised into three categories: (i) *request assignments for cranes*, (ii) *handling conflicting requests* and, finally, (iii) *setting the requests' starting times*.

(i) Request assignments for cranes

Constraints (3.9) ensure that exactly one crane is assigned to each request. Constraints (3.10) and (3.11) determine the value of variable n_{ij} which must be 1 if request i finishes before the starting time of request j and 0 otherwise.

Table 3.2: Notations for the CSP formulation

Sets:

$\mathscr{R}^O$: set of output requests, $\mathscr{R}^O \subseteq \mathscr{R}$
Γ_i : set of requests that must be executed before request i, $\Gamma_i \subseteq \mathscr{R}$
$\mathscr{C}$: set of cranes

Parameters:

$sd_{cc'}$: safety distance required between cranes c and c'
$st_{cc'}$: time required by a crane to travel the safety distance between cranes c and c'
M_L : yard length + $\sum_{c \in \mathscr{C}} sd_{c(c+1)}$
M_T : large number, 2 $\times$ required time for a crane to travel the yard length $\times$ number of requests
η_c : order of crane c in the storage yard, $0 \leq \eta_c < NC$
d_i : duration of request i
b_i : origin location of request i
e_i : destination location of request i
r_i : release time of request i
τ_i : due time of request i
h_l : horizontal coordinate of location l, denoting the coordinate along the rails
h_i^- : horizontal coordinate of the leftmost location of request i's trajectory
h_i^+ : horizontal coordinate of the rightmost location of request i's trajectory
$g_{icjc'}$: required waiting time between the start time of request i by crane c and the start time of request j by crane c' (due to possible conflicts)
t_{lk} : time required by a crane to travel from location l to location k
o_{ij}^a : equal to 1 if trajectory of request i is to the left of request j's trajectory and 0 otherwise

Decision variables:

y_{ic} : binary variable equal to 1 if request i is handled by crane c and 0 otherwise
s_i : continuous variable indicating the start time of request i
δ_i : continuous variable indicating the tardiness of request i

Auxiliary variables:

n_{ij} : binary variable equal to 1 if request i finishes before the start time of request j and 0 otherwise
q_{ij} : binary variable equal to 1 if request i begins before the start time of request j and 0 otherwise
o_{ij} : binary variable equal to 1 if requests i and j are conflicting and 0 otherwise
o_{ij}^b : binary variable equal to 1 if the crane assigned to request i is to the right of the crane assigned to request j and 0 otherwise.
o_{ij}^c : binary variable equal to 1 if the distance between h_i^- and h_j^+ is less than the safety distance required between cranes handling them and 0 otherwise

$$\sum_{c \in \mathscr{C}} y_{ic} = 1 \qquad \forall i \in \mathscr{R} \tag{3.9}$$

$$s_i + d_i \geq s_j - M_T n_{ij} \qquad \forall i, j \in \mathscr{R} : i \neq j \tag{3.10}$$

$$s_i + d_i \leq s_j + (1 - n_{ij}) M_T \qquad \forall i, j \in \mathscr{R} : i \neq j \tag{3.11}$$

Each crane may move only one product at a time. When two requests are scheduled within overlapping times ($n_{ij} = n_{ji} = 0$), they must be assigned to different cranes. Figure 3.2 presents two cases involving requests i and j where the horizontal axis represents time (t). In the first case, s_j (starting time of j) is larger than s_i and smaller than $s_i + d_i$ (finishing time of i), and thus $n_{ij} = n_{ji} = 0$ (time overlapping requests). In the second case, request j is executed after request i is finished and, therefore, $n_{ij} = 1$ and $n_{ji} = 0$ (non-overlapping moves).

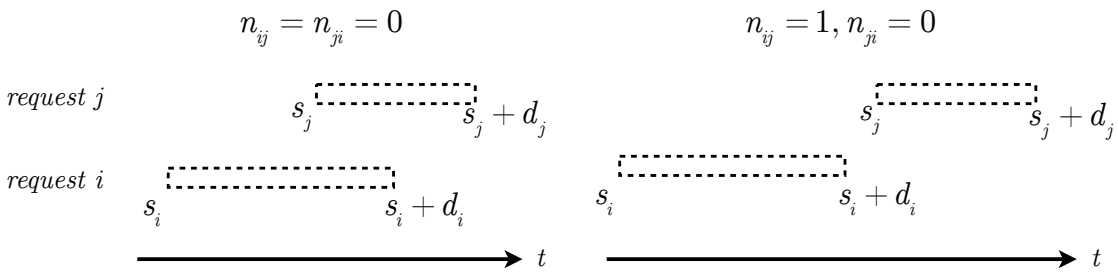


Figure 3.2: An example of overlapping and non-overlapping requests with respect to time.

Constraints (3.12) prevent the assignment of time-overlapping requests to the same crane.

$$y_{ic} + y_{jc} \leq 1 + n_{ij} + n_{ji} \qquad \forall i, j \in \mathscr{R} : i \neq j, \forall c \in \mathscr{C} \tag{3.12}$$

(ii) Handling conflicting requests

Cranes cannot pass each other and must respect a safety distance to avoid collision. The physical constraints due to non-crossing and safety requirements of cranes pose a significant challenge. If simultaneously executing requests i and j violates the safety distance, then these requests are conflicting and must be scheduled at different times. When this situation occurs, binary auxiliary variable o_{ij} is set to 1, indicating requests i and j are conflicting.

To assist in identifying conflicting requests, the *minimum horizontal coordinate* (h_i^-) and *maximum horizontal coordinate* (h_i^+) of a request i are employed and are independent of the requests movement direction. Since the origin and destination locations of requests are given by the location assignment, values $h_i^- = \min(h_{b_i}, h_{e_i})$ and $h_i^+ = \max(h_{b_i}, h_{e_i})$ are easy to pre-compute.

Requests assigned to different cranes may be conflicting depending on their minimum and maximum horizontal coordinates and on the position of the assigned cranes. Given two requests i and j handled by cranes c and c' respectively and a required safety distance $sd_{cc'}$, two different

situations are possible. The first situation arises when the cranes must pass each other to handle the requests whereas the second situation occurs when there is insufficient space for them to respect the safety distance and handle the requests.

Figure 3.3 illustrates trajectories of two requests i and j as well as of cranes c and c'. Two cases may be considered for these two requests. In the first case where crane c is assigned to request i and crane c' to request j, ($y_{ic} = y_{jc'} = 1$), the allocation is such that no conflict occurs. In the second scenario, however, inverting the crane assignments ($y_{ic'} = y_{jc} = 1$) renders the simultaneous handling of requests impossible, given that cranes cannot pass each other therefore, requests i and j are conflicting, and starting times s_i and s_j must be different.

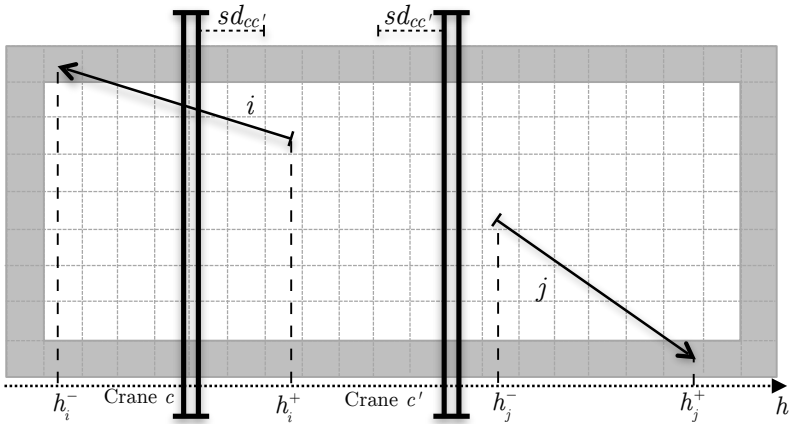


Figure 3.3: Conflicting requests due to crane assignments.

Parameter o^a_{ij} and auxiliary variable o^b_{ij} identify conflicting requests due to cranes requiring to pass each other. o^a_{ij} indicates whether the trajectory of request i is completely to the left of j's trajectory, such that $h_i^+ < h_j^- \Rightarrow o^a_{ij} = 1$. Constraints (3.13) are employed to define the values of o^b_{ij}, which equals 1 if the crane assigned to request i is to the right of the crane assigned to request j, $o^b_{ij} = 1$.

$$\sum_{c \in \mathscr{C}} \eta_c \, y_{jc} \geq \sum_{c \in \mathscr{C}} \eta_c \, y_{ic} - |\mathscr{C}| \cdot o^b_{ij} \qquad \forall i, j \in \mathscr{R} : i \neq j \tag{3.13}$$

Whenever both o^a_{ij} and o^b_{ij} equal one, i's trajectory is to the left of j's trajectory while i's crane is to the right of j's crane, resulting in a conflict. Constraints (3.14) force o_{ij} to take a value of 1 whenever $o^a_{ij} = o^b_{ij} = 1$. Likewise, Constraints (3.15) force o_{ij} to take value 1 whenever $o^a_{ji} = o^b_{ji} = 1$. When request i is conflicting with request j, then request j is conflicting with request i, implying $o_{ij} = o_{ji}$.

$$o_{ij} \geq o^a_{ij} + o^b_{ij} - 1 \qquad \forall i, j \in \mathscr{R} \tag{3.14}$$

$$o_{ij} \geq o^a_{ji} + o^b_{ji} - 1 \qquad \forall i, j \in \mathscr{R} \tag{3.15}$$

Another cause of request conflict concerns the safety distance between cranes. Figure 3.4 illustrates the trajectory of two requests i and j and cranes c and c'. Requests i and j are conflicting as there is insufficient space for the cranes to begin handling the requests while respecting the safety distance ($h_i^- - h_j^+ < sd_{cc'}$). Given the position of the requests' minimum and maximum horizontal coordinates, they are conflicting and the cranes cannot begin executing them simultaneously.

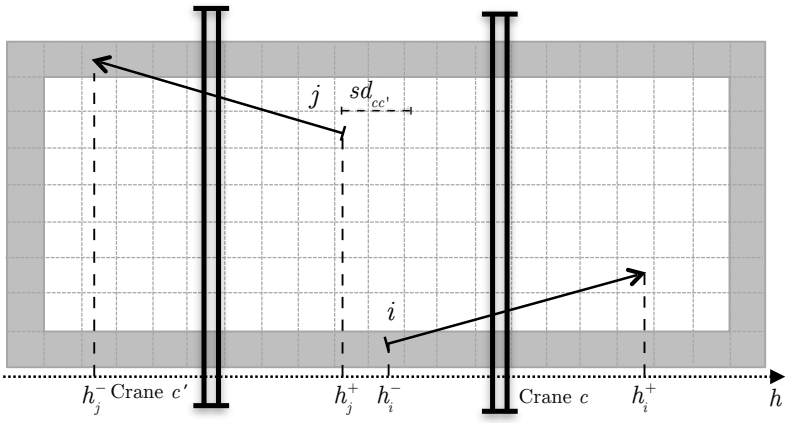


Figure 3.4: Conflicting requests with overlapping trajectories.

Auxiliary binary variables o_{ij}^c are introduced to identify such conflict. Constraints (3.16) determine the value of o_{ij}^c. Whenever i and j are handled by cranes c and c' ($y_{ic} = y_{jc'} = 1$ or $y_{ic'} = y_{jc} = 1$), the maximum value for $y_{ic} + y_{ic'} + y_{jc} + y_{jc'}$ equals two and therefore, Constraints (3.16) ensure o_{ij}^c equals one whenever the *minimum horizontal coordinate* of i, h_i^-, conflicts with the *maximum horizontal coordinate* of j, h_j^+, and 0 otherwise ($h_i^- - h_j^+ < sd_{cc'} \Rightarrow o_{ij}^c = 1$).

$$h_i^- - h_j^+ \geq sd_{cc'} - (2 - y_{ic} - y_{ic'} - y_{jc} - y_{jc'} + o_{ij}^c)M_L \qquad \forall i, j \in \mathscr{R} : i \neq j, c, c' \in \mathscr{C} : c \neq c' \tag{3.16}$$

Whenever both o_{ij}^c and o_{ji}^c equal one, a conflict is detected. Constraints (3.17) force o_{ij} (and o_{ji}) to take a value of 1 whenever $o_{ij}^c = o_{ji}^c = 1$.

$$o_{ij} \geq o_{ij}^c + o_{ji}^c - 1 \qquad \forall i, j \in \mathscr{R} : i \neq j \tag{3.17}$$

(iii) Setting the requests' starting times

A crane can handle one request at a time, therefore, to execute two consecutive requests, a crane requires sufficient time to finish executing the first request and then travel from its destination to the second request's origin.

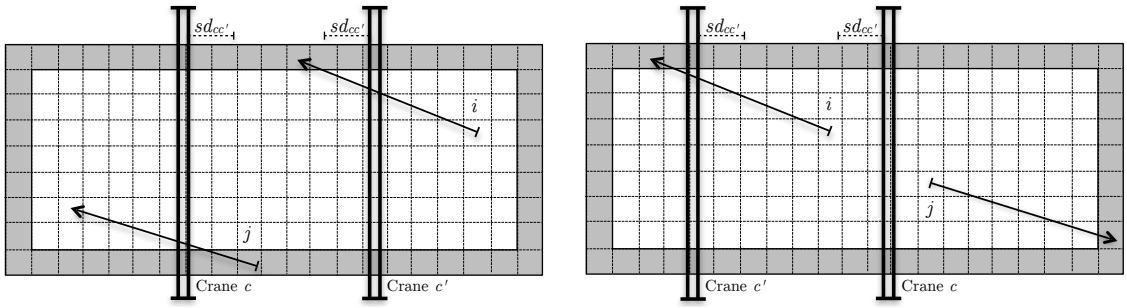
If two requests i and j are handled by a single crane c, ($y_{ic} = y_{jc} = 1$), i and j must be handled one at a time ($n_{ij} = 1$ or $n_{ji} = 1$). Assume request j is executed after finishing i ($n_{ij} = 1$). Constraints (3.18) ensure starting time s_j is greater than or equal to the sum of i's finishing time

$(s_i + d_i)$ and the time required by the crane to travel from i's destination to j's origin ($t_{e_i b_j}$). Constraints (3.18) hold when $y_{ic} + y_{jc} + n_{ij} = 3$ (requests are handled by a single crane and j is executed after finishing i), $\Rightarrow s_j \geq s_i + d_i + t_{e_i b_j}$.

$$s_j \geq s_i + d_i + t_{e_i b_j} - (3 - y_{ic} - y_{jc} - n_{ij})M_T \quad \forall i, j \in \mathscr{R}, i \neq j, \forall c \in \mathscr{C} \tag{3.18}$$

The starting time of two requests handled by different cranes depends on whether they are conflicting or not. When the cranes' trajectories while handling two requests are not conflicting, then the requests' starting times do not influence each other. However, if the two requests are conflicting, a waiting time must be applied between starting the requests. $g_{ic jc'}$ denotes the required waiting time between request i handled by crane c and request j handled by crane c'. The value of $g_{ic jc'}$ depends upon the position of the cranes involved and on their movement direction resulting in three different cases reflected by Equations (3.19), (3.20) and (3.21).

The first case (Figure 3.5(a)) occurs when $y_{ic} = y_{jc'} = 1$, crane c is to the left of c', $\eta_c < \eta_{c'}$ and i's destination is to the right of j's origin, $h_{e_i} > h_{b_j}$. In such situation crane c' cannot immediately begin executing j after the starting time of i, and must instead wait for c to finish i, d_i, plus the time c requires to travel from i's destination to j's origin, $t_{e_i b_j}$. Therefore the waiting time of crane c' to execute request j is $d_i + t_{e_i b_j}$. Another situation which results in the same value for $g_{ic jc'}$ occurs when crane c is to the right of c', $\eta_c > \eta_{c'}$ and i's destination is to the left of j's origin, $h_{e_i} < h_{b_j}$ (Figure 3.5(b)). Figure 3.5 presents these situations both resulting in $g_{ic jc'} = d_i + t_{e_i b_j}$. The value



(a) $h_{e_i} > h_{b_j}, \eta_c < \eta_{c'}$ (b) $h_{e_i} < h_{b_j}, \eta_c > \eta_{c'}$

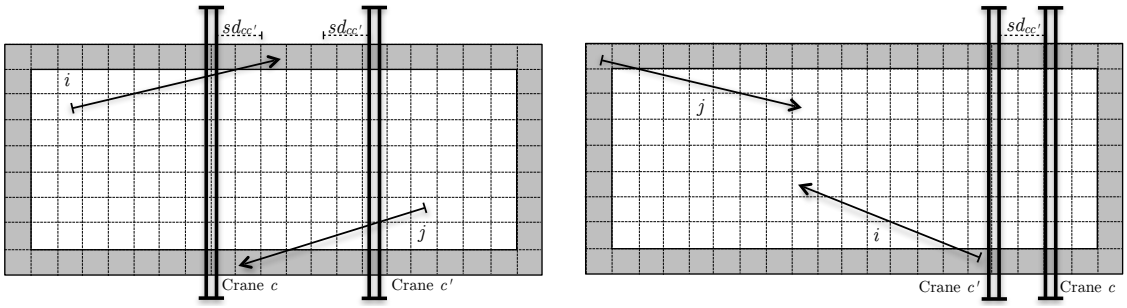
Figure 3.5: Two conditions which result in $g_{ic jc'} = d_i + t_{e_i b_j}$.

of $g_{ic jc'}$ is determined by Equations (3.19).

$$g_{ic jc'} = d_i + t_{e_i b_j} \qquad \forall i \in \mathscr{R}, \begin{cases} j \in \mathscr{R} : h_{e_i} > h_{b_j}, \forall c, c' \in \mathscr{C} : \eta_c < \eta_{c'} \\ j \in \mathscr{R} : h_{e_i} < h_{b_j}, \forall c, c' \in \mathscr{C} : \eta_c > \eta_{c'} \end{cases} \tag{3.19}$$

Conflicts may also occur when crane c' does not necessarily need to wait until the end of request i. Figure 3.6 illustrates examples where crane c' may begin executing request j, $t_{e_i b_j}$ time

units before concluding request i. Such value for $g_{icjc'}$ guarantees that by the time crane c finishes executing request i, crane c''s distance to crane c exceeds $sd_{cc'}$.



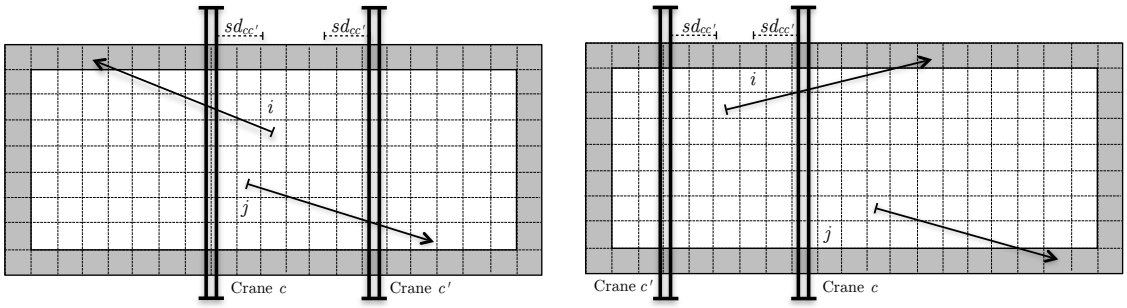
(a) $h_{b_i} < h_{e_i} < h_{b_j}$, $\eta_c < \eta_{c'}$ (b) $h_{b_j} < h_{e_i} < h_{b_i}$, $\eta_c > \eta_{c'}$

Figure 3.6: Two conditions which result in $g_{icjc'} = d_i - t_{e_i b_j}$.

Equations 3.20 define the conditions for $g_{icjc'} = d_i - t_{e_i b_j}$.

$$g_{icjc'} = d_i - t_{e_i b_j} \qquad \forall i \in \mathscr{R}, \begin{cases} j \in \mathscr{R} : h_{b_i} < h_{e_i} < h_{b_j}, \forall c, c' \in \mathscr{C} : \eta_c < \eta_{c'} \\ j \in \mathscr{R} : h_{b_j} < h_{e_i} < h_{b_i}, \forall c, c' \in \mathscr{C} : \eta_c > \eta_{c'} \end{cases} \tag{3.20}$$

Figure 3.7 presents two other scenarios which influence the value of $g_{icjc'}$. The first scenario occurs when crane c is to the left and c' is to the right, $\eta_c < \eta_{c'}$, and the destination of request i is to the left of j's origin, $h_{e_i} < h_{b_j}$ and $h_{e_i} < h_{b_i}$, or when crane c' is to the left of c, $\eta_c > \eta_{c'}$, and the destination of i is to the right of j's origin, $h_{e_i} > h_{b_j}$ and $h_{e_i} > h_{b_i}$. In these cases $g_{icjc'}$ is equal to the travel time from i's to j's origin, $t_{b_i b_j}$.



(a) $h_{e_i} < h_{b_j}$, $h_{e_i} < h_{b_i}$, $\eta_c < \eta_{c'}$ (b) $h_{e_i} > h_{b_j}$, $h_{e_i} > h_{b_i}$, $\eta_c > \eta_{c'}$

Figure 3.7: Two conditions which result in $g_{icjc'} = t_{b_i b_j}$.

When i is an output request and the conditions presented in Figure 3.7 are satisfied, Equations (3.21) determine the value of $g_{icjc'}$.

$$g_{icjc'} = t_{b_i b_j} \qquad \forall i \in \mathscr{R}, \begin{cases} j \in \mathscr{R} : h_{e_i} < h_{b_j}, h_{e_i} < h_{b_i}, \forall c, c' \in \mathscr{C} : \eta_c < \eta_{c'} \\ j \in \mathscr{R} : h_{e_i} > h_{b_j}, h_{e_i} > h_{b_i}, \forall c, c' \in \mathscr{C} : \eta_c > \eta_{c'} \end{cases} \tag{3.21}$$

Constraints (3.22) and (3.23) define the value of variable q_{ij} which must equal 1 if request i begins before j's starting time and 0 otherwise.

$$s_i \geq s_j - M_T q_{ij} \qquad \forall i, j \in \mathscr{R} : i \neq j \tag{3.22}$$

$$s_i \leq s_j + (1 - q_{ij}) M_T \qquad \forall i, j \in \mathscr{R} : i \neq j \tag{3.23}$$

Assume requests i and j are handled by two separate cranes, c and c'. If the requests are not conflicting, i's starting time does not influence j's, since the requests are assigned to different cranes. If, however, the requests are conflicting, then the starting time of j must consider i's starting time coupled with the position and direction of both cranes. Constraints (3.24) coordinate the starting times of consecutive requests if the following conditions are satisfied: (*i*) two different cranes (c and c') are assigned to execute requests i and j, $y_{ic} = y_{jc'} = 1$, (*ii*) the requests are conflicting, $o_{ij} = 1$, and (*iii*) request j begins after request i, $q_{ij} = 1$.

$$s_j \geq s_i + g_{icjc'} + st_{cc'} - (4 - y_{ic} - y_{jc'} - o_{ij} - q_{ij}) M_T \qquad \forall i, j \in \mathscr{R}, c, c' \in \mathscr{C}, i \neq j, c \neq c' \tag{3.24}$$

When the three conditions are satisfied, then request j must begin after the starting time of i plus the waiting time required between requests i and j ($g_{icjc'}$) and the time required for a crane to travel the safety distance between c and c' ($st_{cc'}$).

Constraints (3.25) ensure the starting time of request i occurs after its release time r_i.

$$s_i \geq r_i \qquad \forall i \in \mathscr{R} \tag{3.25}$$

When products are stacked on top of each other and the bottom one should be moved, the requests associated with the top products must precede those associated with products situated on the lower levels. A set of precedence constraints, Γ_i, indicates the set of requests which must precede i. Constraints (26) specify these precedence relations and state that request i may only begin after all its preceding requests are finished.

$$s_i \geq s_j + d_j \qquad \forall i \in \mathscr{R}, j \in \Gamma_i \tag{3.26}$$

The tardiness associated with request i is denoted by $\delta_i \geq 0$. Constraints (3.27) set the delay of each request to be at least the request's finishing time ($s_i + d_i$) minus its due time, τ_i.

$$\delta_i \geq s_i + d_i - \tau_i \qquad \forall i \in \mathscr{R} \tag{3.27}$$

All variables, except s_i (starting time of request i) and δ_i (tardiness of request i), are binary:

$$y_{ic} \in \{0,1\} \qquad \forall i \in \mathscr{R}, c \in \mathscr{C} \tag{3.28}$$

$$n_{ij}, q_{ij}, o_{ij}, o^a_{ij}, o^b_{ij}, o^c_{ij} \in \{0,1\} \qquad \forall i, j \in \mathscr{R} : i \neq j \tag{3.29}$$

$$s_i, \delta_i \geq 0 \qquad \forall i \in \mathscr{R} \tag{3.30}$$

The CSP formulation is given by $\mathscr{F}_{CS}$, where the objective function (3.31) minimises the tardiness of all requests.

$$\mathscr{F}_{CS} \begin{cases} min. & \sum_{i \in \mathscr{R}} \delta_i \\ s.t. & (3.9) - (3.30) \end{cases} \tag{3.31}$$

3.3.3 Integrated formulation for crane-operated warehouse scheduling problems

This section introduces an integrated continuous-time formulation for the CWSP ($\mathscr{F}_{CWS}$) which considers the location assignment for input requests, the crane assignment for all requests, and the sequencing of the requests per crane. $\mathscr{F}_{CWS}$ includes all constraints associated with the LAP and CSP plus some additional constraints. Since the *input* requests' destinations are undefined (whereas for the CSP, they are determined by first solving the LAP), the following parameters in $\mathscr{F}_{CS}$ become variables in $\mathscr{F}_{CWS}$: h^-_i, h^+_i, d_i and e_i for all requests $i \in \mathscr{R}^I$ and o^a_{ij} when at least one of requests i and j is an *input request*. Note that since these variables are defined only for *input* requests, the definitions in Table 3.2 remain valid for output requests.

Additional constraints are required to assist in identifying the conflicting requests. Equations (3.32) and (3.33) are employed to obtain h^-_i and h^+_i for request $i \in \mathscr{R}^I$, respectively. Constraints (3.34) are employed to define the values of o^a_{ij}.

$$h^-_i = \sum_{l \in L : h_l \leq h_{b_i}} h_l \, x_{il} + \sum_{l \in L : h_l > h_{b_i}} h_{b_i} \, x_{il} \qquad \forall i \in \mathscr{R}^I \tag{3.32}$$

$$h^+_i = \sum_{l \in L : h_l \leq h_{b_i}} h_{b_i} \, x_{il} + \sum_{l \in L : h_l > h_{b_i}} h_l \, x_{il} \qquad \forall i \in \mathscr{R}^I \tag{3.33}$$

$$h^+_i \geq h^-_j - M_L \cdot o^a_{ij} \qquad \forall i, j \in \mathscr{R} : i \neq j \tag{3.34}$$

Output requests have a fixed duration, while *input* requests' durations depend on the chosen destination. Constraints (3.35) are employed to compute the duration of *input* requests.

$$d_i = \sum_{l \in L} t_{b_i l} \, x_{il} \quad \forall i \in \mathscr{R}^I \tag{3.35}$$

As the destinations of *input* requests are decision variables, setting the value of $g_{ic\,jc'}$ when request i is an *input* request requires additional constraints. Equations (3.36) define the value of

$g_{icjc'}$ when crane c handling request i is to the left of crane c' which handles request j. When $h_{e_i} < h_{b_j}$ and $h_{e_i} < h_{b_i}$ then $g_{icjc'} = t_{b_ib_j}$ as in Constraints (3.21), if $h_{e_i} < h_{b_j}$ and $h_{e_i} > h_{b_i}$ then $g_{icjc'} = d_i - t_{e_ib_j}$ as in Constraints (3.20), and finally $h_{e_i} > h_{b_j}$ then $g_{icjc'} = d_i + t_{e_ib_j}$ as in Constraints (3.19).

$$g_{icjc'} = \sum_{l \in L: h_l < h_{b_j}, h_l < h_{b_i}} t_{b_ib_j} x_{il} + \sum_{l \in L: h_l < h_{b_j}, h_l > h_{b_i}} (d_i - t_{e_ib_j})\, x_{il} + \sum_{l \in L: h_l > h_{b_j}} (d_i + t_{e_ib_j})\, x_{il}$$
$$\forall i \in \mathscr{R}^I, j \in \mathscr{R}, \forall c, c' \in \mathscr{C} : \eta_c < \eta_{c'} \tag{3.36}$$

Similarly, Equations (3.37) set the value of $g_{icjc'}$ for *input* requests in case crane c handling request i is to the right of crane c' which handles request j.

$$g_{icjc'} = \sum_{l \in L: h_l < h_{b_j}} (d_i + t_{e_ib_j})\, x_{il} + \sum_{l \in L: h_l > h_{b_j}, h_l < h_{b_i}} (d_i - t_{e_ib_j})\, x_{il} + \sum_{l \in L: h_l > h_{b_j}, h_l > h_{b_i}} t_{b_ib_j} x_{il}$$
$$\forall i \in \mathscr{R}^I, j \in \mathscr{R}, \forall c, c' \in \mathscr{C} : \eta_c > \eta_{c'} \tag{3.37}$$

In case of *output* requests, as their destinations are given, the value of $g_{icjc'}$ is obtained in a similar way as in $\mathscr{F}_{\mathscr{CS}}$. Constraints (3.38), (3.39) and (3.40) are modified based on Constraints (3.19), (3.20) and (3.21) respectively to set $g_{icjc'}$ for request $i \in \mathscr{R}^O$ and request $j \in \mathscr{R}$.

$$g_{icjc'} = t_{b_ib_j} \qquad \forall i \in \mathscr{R}^O, \begin{cases} j \in \mathscr{R} : h_{e_i} < h_{b_i}, h_{e_i} < h_{b_j}, \forall c, c' \in \mathscr{C} : \eta_c < \eta_{c'} \\ j \in \mathscr{R} : h_{b_i} < h_{e_i}, h_{b_j} < h_{e_i}, \forall c, c' \in \mathscr{C} : \eta_c > \eta_{c'} \end{cases} \tag{3.38}$$

$$g_{icjc'} = d_i - t_{e_ib_j} \qquad \forall i \in \mathscr{R}^O, \begin{cases} j \in \mathscr{R} : h_{b_i} < h_{e_i} < h_{b_j}, \forall c, c' \in \mathscr{C} : \eta_c < \eta_{c'} \\ j \in \mathscr{R} : h_{b_i} > h_{e_i} > h_{b_j}, \forall c, c' \in \mathscr{C} : \eta_c > \eta_{c'} \end{cases} \tag{3.39}$$

$$g_{icjc'} = d_i + t_{e_ib_j} \qquad \forall i \in \mathscr{R}^O, \begin{cases} j \in \mathscr{R} : h_{e_i} > h_{b_j}, \forall c, c' \in \mathscr{C} : \eta_c < \eta_{c'} \\ j \in \mathscr{R} : h_{e_i} < h_{b_j}, \forall c, c' \in \mathscr{C} : \eta_c > \eta_{c'} \end{cases} \tag{3.40}$$

Another set of additional constraints is required due to implied precedence constraints during the location assignment. Constraints (3.41) assert that if an input request is assigned atop another input request, the bottom request is placed first. Constraints (3.42) ensure j is moved after i if it has been assigned to i's origin location.

$$n_{ij} \geq x_{jl} + x_{ik} - 1 \qquad \forall i, j \in \mathscr{R}^I : i \neq j, l \in L, k \in U_l \tag{3.41}$$
$$n_{ij} \geq x_{jb_i} \qquad \forall i \in \mathscr{R}, j \in \mathscr{R}^I \tag{3.42}$$

The objective function is a weighted linear combination of the LAP's objectives (Equation (3.2)) and those of the CSP (Equation (3.31)). The CWSP formulation is given by $\mathscr{F}_{CWS}$:

$$\mathscr{F}_{CWS}\begin{cases} min. \quad \alpha\left(\sum_{i\in\mathscr{R}^I}\sum_{l\in L}\gamma_{il}x_{il}+\sum_{i\in\mathscr{R}^I}\sum_{j\in\mathscr{R}^I}\omega_{ij}z_{ij}\right)+\beta\sum_{i\in\mathscr{R}}\delta_i \\ s.t. \quad (3.3)-(3.12),\ (3.16)-(3.18),\ (3.22)-(3.30),\ (3.32)-(3.42) \end{cases} \tag{3.43}$$

The CWSP is a complex problem that generalizes the Sequential Ordering Problem (SOP) (Escudero, 1988), which represents a special case of the CWSP scheduling component with a single crane and fixed request destinations. Consequently, the CWSP is considered at least as hard as the SOP. Given the SOP is known to be $\mathscr{NP}$-hard (Montemanni et al., 2009), by consequence, CWSP is an $\mathscr{NP}$-hard problem.

3.4. Heuristic approach

A local search based algorithm is proposed consisting of constructive and improvement phases for solving the CWSP. During both phases, an *indirect* solution representation capable of reducing the search space is employed (Section 3.4.1). All the algorithm's components are explained throughout the following sections. The constructive phase (3.4.2) consists of a greedy constructive heuristic inspired by a set of dispatching rules. During the improvement phase (3.4.3) a Late Acceptance Hill Climbing (LAHC) meta-heuristic (Burke and Bykov, 2017) is employed which considers several neighbourhood structures (3.4.4). Figure 3.8 presents a general overview of the proposed algorithm.

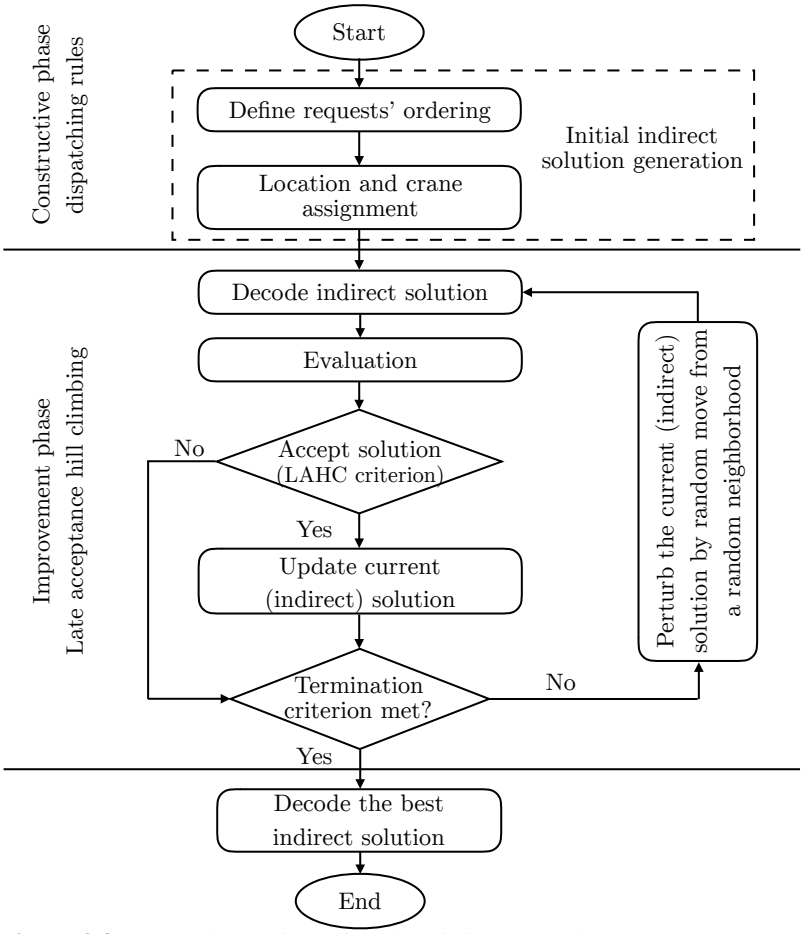


Figure 3.8: General overview of the heuristic approach

3.4.1 Solution representation

The local search-based algorithm considers an *indirect* solution representation. Each solution is represented by a list $\mathscr{L}$ of (request, location, crane)-tuples. The actual schedule is produced by a *decoder* which utilizes both the ordering of requests, the locations and crane assignments included in list $\mathscr{L}$.

The decoder is also employed to evaluate solutions by computing their storage cost and total tardiness. Whereas the total tardiness can be obtained by applying Equation (3.2), computing the total tardiness is less straightforward. It requires all requests' starting times, which determine on the requests' execution order in $\mathscr{L}$ in such a way that all conflict-related constraints are satisfied.

The decoder is presented by Algorithm 4. For each tuple (i, e_i, c) in $\mathscr{L}$, request i has its starting time s_i initialized as its release time (lines 1-2). If crane c was previously assigned to another request, the starting time s_i is set to be after the execution of the crane's previous request plus the time required for the crane to begin executing request i (lines 3-5). To handle conflicts with requests assigned to other cranes, the algorithm loops over the schedule of all other cranes from end to beginning (lines 6-8). In case request i conflicts with a request k of crane c''s schedule, s_i is set to be at least $s_k + g_{kc'ic} + st_{cc'}$ (lines 9-10), where $g_{kc'ic}$ is the waiting time defined in Section 3.3.

Once the algorithm finds a conflicting request k in crane c''s schedule, it is unnecessary to proceed checking, as all other requests assigned to c' are scheduled earlier than k. The algorithm then moves on to the next crane, if any remain. Therefore, the decoder will execute only $O(|\mathscr{R}| \times |\mathscr{C}|)$ operations in the best case. Since in most applications the number of cranes $|C|$ may be fixed as a constant, in the best case the decoder runs in linear time. In the worst case, however, $O(|\mathscr{R}|^2)$ operations may be required by the decoder. After the schedules of all cranes are checked, the minimum value for s_i is calculated and the algorithm includes request i into crane c's schedule (lines 12). Once all starting times are computed, the tardiness can be easily calculated employing Equations (3.27) and (3.31).

One of the advantages of the indirect solution representation proposed is that it prevents the generation of a class of unattractive solutions which include avoidable idle times between operations. Indeed, given that the indirect representation is no more than a sequence, the decoder will produce a left-active schedule which is always better than alternative schedules based on this sequence, but which include idle times.

Moreover, the indirect solution is simple to modify, requiring only a few verifications to guarantee feasibility. Its main disadvantage lies in the additional $O(|R|^2)$ operations required for decoding and evaluating solutions.

3.4.2 Constructive heuristic

An initial solution is constructed by a greedy algorithm which applies a set of dispatching rules. Algorithm 5 details this procedure. A directed acyclic graph is considered where nodes represent requests and arcs directing from r_i to r_j indicate precedence constraints (r_i must precede r_j). In this graph, requests are first sorted topologically. By sorting the directed acyclic graph topologically,

Algorithm 4: Decoding an *indirect* solution

Input: Ordered list $\mathscr{L}$ of tuples {request, location, crane}

1 **foreach** tuple $(i, e_i, c) \in \mathscr{L}$ **do**
2 $\quad s_i \leftarrow r_i$ `// minimum starting time` s_i `is the release time` r_i
3 $\quad$ **if** crane c's schedule is not empty **then**
4 $\quad\quad j \leftarrow$ last request in crane c's schedule
5 $\quad\quad s_i \leftarrow max(s_i\ ,\ s_j + d_j + t_{e_j b_i})$ `//` s_i `must consider when crane` c `is available at position` b_i
6 $\quad$ **foreach** crane $c' \in \mathscr{C}, c' \neq c$ **do**
7 $\quad\quad$ let schedule S^{-1} be the reverse of crane c''s schedule
8 $\quad\quad$ **foreach** request $k \in S^{-1}$ **do**
9 $\quad\quad\quad$ **if** request i conflicts with request k **then**
10 $\quad\quad\quad\quad s_i \leftarrow max(s_i\ ,\ s_k + g_{kc'ic} + st_{cc'})$ `//` s_i `must also take conflicting moves into account`
11 $\quad\quad\quad\quad$ **break** foreach-loop
12 $\quad$ add request i (ending at location e_i) into crane c's schedule

for every directed arc (r_i, r_j), r_i precedes r_j in the ordering ensuring that the precedence constraints are respected. Requests within the same topological level are sorted by their release times. This strategy defines the initial ordering of the requests (line 1). The solution is initialized as an empty list (line 2) and, afterwards, two steps are applied for each request (lines 3-6). First, the available location which has the lowest storage cost is assigned to *input* requests (lines 4-5). Note that *output* requests have preassigned locations and therefore do not require such an assignment. In the second step, requests are assigned to the nearest available crane (line 6). Both the requests' starting times and the cost of the solution corresponding with $\mathscr{L}$ are computed employing the decoder presented in Algorithm 4.

Algorithm 5: Constructive CWSP algorithm

```
1 R ← list of requests sorted topologically and chronologically
2 𝓛 ← empty list // or empty ordered set
3 foreach request i ∈ R do
4     if i ∈ 𝓡^I then // checking whether request i is an input request
        (these requests require assigning a location)
5         e_i ← the nearest available location to the origin of input request i
6     c ← nearest crane to b_i // greedy criterion: nearest crane to the
        origin of request i is selected
7     insert (i, e_i, c) to 𝓛
8 return list of tuples 𝓛
```

3.4.3 Late Acceptance Hill Climbing

The LAHC meta-heuristic represents an extension of the greedy hill-climbing algorithm which compares the candidate solution against the solution which was 'current' l iterations before. Consequently, the meta-heuristic permits the acceptance of worsening solutions, thus avoiding local optima. This study employs the LAHC meta-heuristic presented in Algorithm 6 which requires the following arguments: (*i*) initial solution s_0, (*ii*) parameter l, (*iii*) set of neighbourhoods $\mathscr{N}$, (*iv*) maximum number of iterations without improvement it^{max} and (*v*) $timeout$, which indicates the runtime of LAHC.

The LAHC meta-heuristic maintains a fixed-length list v containing objective function values of the solutions visited during the last l iterations. Initially, all v elements are set to the initial solution's objective value, given by $f(s_0)$ (line 1). Next, current solution s, best solution s^* and index i are initialized (lines 2-3). At each iteration a new candidate solution s' is generated by applying a randomly generated move from a randomly selected neighbourhood to the current solution s (lines 5-6). The candidate solution's objective value, $f(s')$, is compared against v_i and the current solution's value $f(s)$ (line 7). s' is accepted (line 12) to replace the current solution if its objective value is less than or equal to v_i or $f(s)$. If the candidate solution s' has a better objective value than the best solution s^* generated thus far, it replaces s^* (lines 13-14). If the objective value of s' is greater than v_i or $f(s)$, the number of iterations without improvement, it, increments by

Algorithm 6: Late acceptance hill climbing algorithm

Input: Initial solution s_0, list size l, set of neighbourhoods $\mathscr{N}$, maximum number of consecutive iterations without improvement it^{max}, and runtime limit $timeout$

1 $v_i \leftarrow f(s_0) \ \forall\, i \in \{0,...,l-1\}$
2 $s^* \leftarrow s \leftarrow s_0$
3 $i \leftarrow it \leftarrow 0$
4 **while** $it < it^{max}$ **and** $elapsed\ time < timeout$ **do**
5 Select a random neighbourhood $N \in \mathscr{N}$
6 $s' \leftarrow$ random neighbour solution $m \in N(s)$
7 **if** $f(s') \leq v_i$ **or** $f(s') \leq f(s)$ **then**
8 **if** $f(s') < f(s)$ **then**
9 $it \leftarrow 0$
10 **else**
11 $it \leftarrow it + 1$
12 $s \leftarrow s'$
13 **if** $f(s') < f(s^*)$ **then**
14 $s^* \leftarrow s$
15 **else**
16 $it \leftarrow it + 1$
17 $v_i \leftarrow f(s)$
18 $i \leftarrow (i+1)\ mod\ \ l$
19 **return** s^*

one (lines 15-16). Finally, v_i and i are updated: $v_i \leftarrow f(s)$ (replacing the oldest value), and i is set to point to the next position of list l (lines 17-18). Note that i acts as a cyclic pointer. This iterative process repeats until the elapsed time reaches $timeout$ or the number of iterations without improvement reaches it^{max}. The latter criterion prevents the heuristic from continuing the search in situations whre no more improvements are generated during a long period of time. Finally, the best solution obtained is then returned (line 19).

3.4.4 Neighbourhood structures

Ten neighbourhoods were developed to explore the CWSP's solution space. The neighbourhoods are grouped into two categories: (*i*) Location assignment and (*ii*) Crane scheduling neighbourhoods. All ten neighbourhoods operate over the list of tuples $\mathscr{L}$ and therefore modify only the indirect solution. Infeasible solutions may be obtained by applying some of these neighbourhood operators to a solution. For instance, both assigning a product atop an empty location and ignoring the precedence constraints when changing the requests' order result in infeasibility. Such infeasible solutions are discarded during the search.

The ten neighbourhoods are detailed as follows.

Location assignment neighbourhoods

– Location Re-assignment (LR): a random *input* request is selected and its destination location is replaced with another randomly selected available location.

– Location Swap (LS): two random input requests are selected and their destination locations are swapped.

– Greedy Location Assignment (GLA): a set of input requests is randomly selected and their location assignments are removed. Then, all possible locations are considered and the requests are greedily assigned to the **lowest-cost** location.

Crane scheduling neighbourhoods

– Crane Re-assignment (CR): a set of three requests is selected randomly and all crane assignment combinations are enumerated. The resulting solution is the one with the best quality among the enumerated solutions.

– Order Swap (OS): two requests are randomly selected and their tuples are swapped, changing their execution order.

– Random Insertion (RI): a random tuple is removed and re-inserted into a random position in the list.

– Random Best Insertion (RBI): a request is randomly selected and its tuple is inserted into the lowest-cost position within the assigned crane's schedule. Note that this neighbourhood requires $O(|R|)$ operations to identify the lowest-cost position.

– Nearest Location Assignment (NLA): a set of input requests is randomly selected and their location assignments are removed. Then, all possible locations are considered and each request is greedily assigned to the location nearest to its origin.

  This neighbourhood is employed as a crane scheduling neighborhood since it reduces the duration of requests and, as a consequence, reduces the potential risk of generating conflicting requests.

– Best Order Permutation (BOP): a range of requests in a crane's schedule is randomly selected; their tuples are subsequently removed and the best permutation of the selected tuples, determined by enumeration, is inserted into the list.

  This neighbourhood has exponential complexity and therefore the range must be limited to prevent prohibitive runtimes.

– Moving Best Order Permutation (MBOP): This neighbourhood begins from the first tuple in the list $\mathscr{L}$ and executes the BOP move within a range of three tuples. The procedure moves forward in $\mathscr{L}$ by one tuple and executes the BOP move for next three tuples. It ends after $|\mathscr{L}| - 2$ iterations by executing the BOP move for the last three tuples in the list.

3.5. Computational study

This section investigates the impact of integrating the location assignment and crane scheduling problems. The performance of both the formulations and heuristic are assessed across different scenarios: sequential and integrated.

The sequential approach, on the one hand, solves two problems. It begins by solving the LAP, after which the assignments obtained are fixed when solving the CSP. The integrated approach, on the other hand, solves only one large problem: the CWSP.

Table 3.3: Summary of sequential and integrated approaches

	Formulations		
		LAHC	
		Objective Function	Neighbourhoods
Sequential approach	$\mathscr{F}_{LA}$	Equation (2)	LR, LS, GLA
	$\mathscr{F}_{CS}$	Equation (33)	CR, OS, RI, RBI, BOP, MBOP
Integrated approach	$\mathscr{F}_{CWS}$	Equation (44)	LR, LS, GLA, CR, OS, RI, RBI, NLA, BOP, MBOP

Table 3.3 summarizes the formulations, objective functions and neighbourhood structures employed by each approach. The sequential approach employs formulations $\mathscr{F}_{LA}$ and $\mathscr{F}_{CS}$ to solve the LAP and CSP, respectively. It also utilizes Equation (3.2) and *location assignment neighbourhoods* to address the LAP heuristically. To solve the CSP by LAHC, the sequential approach employs Equation (3.31) and *crane scheduling neighbourhoods*. Whereas, the integrated approach solves formulation $\mathscr{F}_{CWS}$ and considers Equation (3.43) and all the neighbourhoods when employing LAHC. Table 3.3 summarizes the objective functions and neighbourhoods employed in the heuristic approaches to both the sequential and integrated problems.

The remainder of this section is organized as follows. First Section 3.5.1 presents the set of benchmark instances considered throughout the experiments. The comparison of the sequential and integrated approaches by employing the MIP formulations is presented in Section 3.5.2, while Section 3.5.3 presents the results of the sequential and integrated approaches employing the heuristic. Finally, Section 3.5.4 presents a discussion on the weights α and β employed within the objective function (see Equation (3.1)) in addition to an analysis of the instances.

3.5.1 Instances

A set of benchmark instances was generated in correspondance with the data obtained from a real-world warehouse. These instances are available online[1] to enable transparent comparison of the proposed formulations and algorithms. Four characteristics were considered for the instance generation:

[1] CWSP data instances, solutions & validator available at *https://bitbucket.org/Sam-Hes/cwsp/*.

Number of requests: the benchmark set includes instances with *five* levels denoting the number of requests: 10, 20, 30, 50 and 70.

Storage size: the dataset considers three storage sizes. *Small* size storage areas have 250 (10×25) locations per level. *Medium* size storage areas have 525 (15×35) locations per level and *large* storage areas 1000 (20×50) locations per level. A large storage area may have up to 5000 locations. Based on the data obtained from the real-world problem, a realistic storage area contains approximately 3000 locations.

Maximum stacking level: this limit is imposed for each storage area depending on various criteria such as the product type and safety requirements. Three stacking levels are considered: *one*, *three* and *five* levels, where *one* denotes the ground level.

Storage load: this value is defined as a percentage indicating the initial occupancy level of the storage area. Three *load factor* values are considered: 30%, 50% and 70%.

The value of each attribute is reflected in the instance name, which indicates, from left to right: (i) number of requests, (ii) storage size, represented by the letters *s*, *m* and *l* signifying *small*, *medium* and *large*, respectively, (iii) maximum stacking level, and (iv) storage load. Instance name `50s_1_30`, for example, indicates an instance with 50 requests, a small yard, maximum stacking level 1, and storage load equal to 30%.

Release times were generated according to a Poisson distribution with a fixed "*average frequency*" within the scheduling horizon. The scheduling horizon is computed by multiplying "*Number of requests*" by "*average frequency*". Due times were calculated as the sum of a request's "*release time*" and its "*time window*", wherein the "*time window*" was generated according to a log-normal distribution.

The number of yard products is calculated based on the *yard size* and *yard load*. For each stored product a location in the yard is randomly selected.

The storage cost of assigning an input request to a neighbouring location of any product is generated according to a uniform distribution. The cranes' travel times are measured by the Chebychev distance (the maximum of the lateral and longitudinal distance). It is assumed that the lateral and longitudinal speed of the cranes is identical, and thus the travel time may be substituted by the Chebychev distance. Table 4 summarizes the parameters of the distributions used in the instance generator.

Table 3.4: Instance generator distributions and parameters

Instance Parameter	Distribution	Distribution Parameter
Release time	Poisson	mean = 26.5
Time windows	Log normal	mean = 0.0666 * storage area's length standard deviation = 0.1 * storage area's width
Stored product location	Uniform	range = locations in the storage area
Storage cost	Uniform	range = [0, 50)

The cranes are initially lined-up according to the required safety distance. The crane's travel speed is set to one time unit per horizontal coordinate.

3.5.2 Comparison of sequential and integrated approaches by MIP formulations

In the sequential approach, the LAP completely disregards the possible impact of attaining good solutions for the CSP. The integrated approach, however, enables a trade-off between the LAP and the CSP by setting different values for their corresponding weights α and β in the objective function, Equation (3.1). Two settings are considered: (*i*) $\alpha >> \beta$, making the integrated approach put the highest priority to the LAP. This enables determining the benefit of an integrated approach over the sequential approach where both primarily focus on optimizing the storage cost. (*ii*) $\alpha = \beta$, which considers equal importance for storage cost and delays.

The mathematical formulations were solved by Gurobi Optimizer 7.5 and run on an Intel® Xeon® CPU E5-2650 v2 @ 2.6GHz with one hour runtime limit. Table 3.5 reports the results obtained by solving the sequential ($\mathscr{F}_{LA} + \mathscr{F}_{CS}$) and integrated ($\mathscr{F}_{CWS}$) formulations. The results of unsolved instances (no integer solution found within the runtime limit) are excluded from the table for brevity. Location assignment evaluation (E_{LA}), crane scheduling evaluation (E_{CS}) and total weighted cost (E_τ), both when $\alpha = \beta$ and $\alpha >> \beta$, are compared. Emboldened numbers indicate that optimal solutions were obtained. When both the LAP and CSP are solved to optimality only E_τ is emboldened.

Table 3.5: Computational results obtained by solving the mathematical formulations (considered scenarios: $\alpha = 100000$, $\beta = 1$ ($\alpha >> \beta$) and $\alpha = \beta = 0.50$).

Instance	$\mathscr{F}_{LA} + \mathscr{F}_{CS}$				$\mathscr{F}_{AWS}$ ($\alpha >> \beta$)			$\mathscr{F}_{AWS}$ ($\alpha = \beta$)		
	E_{LA}	E_{CS}	$E_\tau(\alpha >> \beta)$	$E_\tau(\alpha = \beta)$	E_{LA}	E_{CS}	E_τ	E_{LA}	E_{CS}	E_τ
10s_1_30	0	267.03	**2.670E+02**	**133.51**	0	55.13	**5.513E+01**	1	51.53	**26.26**
10m_1_30	0	478.41	**4.784E+02**	**239.20**	0	37.62	**3.762E+01**	0	37.62	**18.81**
10l_1_30	0	515.69	**5.157E+02**	**257.84**	0	76.82	**7.682E+01**	0	76.82	**38.41**
10s_1_50	0	229.25	**2.293E+02**	**114.63**	0	123.93	**1.239E+02**	11	78.92	**44.96**
10m_1_50	0	219.18	**2.192E+02**	**109.59**	0	56.45	**5.645E+01**	7	40.07	**23.53**
10l_1_50	0	447.85	**4.479E+02**	**223.92**	0	44.49	**4.449E+01**	4	32.49	**18.24**
10s_1_70	132	186.97	**1.320E+07**	**159.49**	132	135.97	**1.320E+07**	156	59.25	**107.62**
10m_1_70	48	348.86	**4.800E+06**	**198.43**	48	449.55	**4.800E+06**	86	185.92	**135.96**
10l_1_70	11	271.44	**1.100E+06**	**141.22**	11	148.17	**1.100E+06**	25	73.78	**49.39**
10s_3_30	13	352.10	**1.300E+06**	**182.55**	13	179.72	**1.300E+06**	49	76.07	62.53
10m_3_30	1	354.66	**1.004E+05**	**177.83**	1	107.51	**1.001E+05**	9	70.37	39.68
10l_3_30	0	295.28	**2.953E+02**	**147.64**	0	79.51	**7.951E+01**	-	-	-
10s_3_50	53	317.77	**5.300E+06**	**185.39**	53	201.77	**5.300E+06**	64	109.83	86.91
10m_3_50	31	409.77	**3.100E+06**	**220.39**	31	332.35	**3.100E+06**	41	126.77	83.88
10l_3_50	9	454.43	**9.005E+05**	**231.72**	9	145.71	**9.001E+05**	12	85.91	**48.95**
10s_3_70	130	392.07	**1.300E+07**	**187.73**	130	279.13	**1.300E+07**	190	102.40	146.20
10m_3_70	64	433.66	**6.400E+06**	**248.83**	64	492.57	**6.400E+06**	93	77.90	85.45
10l_3_70	98	441.02	**9.800E+06**	**269.51**	98	566.37	**9.800E+06**	134	118.11	126.05
10s_5_30	18	322.97	**1.800E+06**	**170.48**	18	182.53	**1.800E+06**	35	93.23	64.11
10m_5_30	653	**477.18**	**6.530E+07**	565.09	9	275.46	9.002E+05	24	76.36	50.18
10l_5_30	464	**457.64**	**4.640E+07**	460.82	6	244.62	**6.002E+05**	16	206.88	111.44
10s_5_50	66	435.57	**6.600E+06**	**250.79**	66	292.55	6.600E+06	-	-	-
10m_5_50	**25**	-	-	-	25	294.46	2.500E+06	42	196.59	119.30

(continued on next page)

Computational results obtained by solving the mathematical formulations (continued).

Instance	$\mathscr{F}_{LA}+\mathscr{F}_{CS}$				$\mathscr{F}_{AWS}$ $(\alpha >> \beta)$			$\mathscr{F}_{AWS}$ $(\alpha = \beta)$		
	E_{LA}	E_{CS}	$E_\tau(\alpha >> \beta)$	$E_\tau(\alpha = \beta)$	E_{LA}	E_{CS}	E_τ	E_{LA}	E_{CS}	E_τ
10l_5_50	13	602.07	**1.301E+06**	**307.54**	13	159.11	**1.300E+06**	21	82.93	51.96
10s_5_70	83	247.93	**8.300E+06**	**165.64**	83	247.93	**8.300E+06**	100	91.85	95.92
10m_5_70	59	419.26	**5.900E+06**	**239.13**	59	263.06	**5.900E+06**	75	211.27	143.13
10l_5_70	43	541.38	**4.301E+06**	**292.19**	43	418.37	**4.300E+06**	51	87.02	69.01
20s_1_30	0	1221.04	**1.221E+03**	**610.52**	0	160.38	1.603E+02	-	-	-
20m_1_30	0	1821.80	**1.822E+03**	**910.90**	0	225.29	2.252E+02	1	235.90	118.45
20l_1_30	0	2171.42	**2.171E+03**	**1085.71**	0	198.40	**1.984E+02**	0	261.18	130.59
20s_1_50	4	1412.59	**4.014E+05**	**708.30**	4	425.09	4.004E+02	-	-	-
20m_1_50	0	1373.24	**1.373E+03**	**686.21**	0	445.37	4.453E+02	9	351.42	180.21
20l_1_50	0	1851.12	**1.851E+03**	**925.56**	0	155.85	1.558E+02	8	118.23	63.11
20s_1_70	448	801.34	**4.480E+07**	**624.67**	448	692.26	4.480E+07	480	196.16	338.08
20m_1_70	241	2396.80	**2.410E+07**	**1318.90**	241	1287.71	2.410E+07	333	640.11	486.55
20l_1_70	77	1210.96	**7.701E+06**	**643.98**	77	758.37	7.700E+06	88	337.18	212.59
20s_3_30	54	1328.04	5.401E+06	691.02	-	-	-	-	-	-
20m_3_30	995	2031.35	9.950E+07	1513.18	15	740.32	1.500E+06	64	676.63	370.31
20l_3_30	648	-	-	-	0	277.50	2.775E+02	37	417.42	227.21
20s_3_50	168	1434.26	1.680E+07	799.13	-	-	-	-	-	-
20m_3_50	99	2203.80	9.902E+06	1151.40	-	-	-	181	366.17	273.58
20l_3_50	1344	1852.22	1.344E+08	1598.11	45	1123.48	4.501E+06	49	578.48	313.74
20s_3_70	423	1552.35	4.230E+07	987.67	424	2902.26	4.240E+07	528	390.82	459.41
20m_3_70	**168**	1491.69	1.680E+07	829.84	168	1407.24	1.680E+07	-	-	-
20l_3_70	274	2989.96	2.740E+07	1631.98	-	-	-	-	-	-
20s_5_30	923	939.96	9.230E+07	931.48	68	1630.26	6.801E+06	-	-	-
20m_5_30	886	-	-	-	43	2017.09	4.302E+06	-	-	-
20s_5_50	1527	1051.42	1.527E+08	1289.21	-	-	-	-	-	-
20m_5_50	1450	-	-	-	87	2062.80	8.702E+06	-	-	-
20s_5_70	330	1499.62	3.300E+07	914.81	-	-	-	-	-	-
20m_5_70	249	1435.31	2.490E+07	842.15	385	1738.69	3.850E+07	-	-	-
20l_5_70	129	-	-	-	-	-	-	158	812.23	485.11
30s_1_30	0	3495.86	**3.496E+03**	**1747.93**	0	668.80	6.688E+02	-	-	-
30m_1_30	0	4501.77	**4.502E+03**	**2250.88**	0	921.33	9.213E+02	-	-	-
30l_1_30	**0**	4911.02	4.911E+03	2455.51	0	634.49	6.344E+02	-	-	-
30s_1_50	98	3299.06	**9.803E+06**	**1698.53**	98	1303.66	9.801E+06	210	729.63	469.81
30m_1_50	19	3742.17	1.904E+06	1880.58	19	2616.36	1.902E+06	75	1880.72	977.86
30l_1_50	0	4683.93	**4.684E+03**	**2737.80**	0	724.51	7.245E+02	-	-	-
30s_1_70	934	2416.98	9.340E+07	1675.49	937	2667.16	9.370E+07	-	-	-
30m_1_70	544	5919.16	5.441E+07	3231.58	554	3078.50	5.540E+07	631	2046.16	1338.58
30l_1_70	249	3310.89	**2.490E+07**	**1779.94**	-	-	-	268	1623.10	945.55
30s_3_30	1585	2858.33	1.585E+08	2221.66	-	-	-	-	-	-
30s_3_50	330	3643.15	3.300E+07	1986.57	-	-	-	-	-	-
30s_3_70	821	3058.59	8.210E+07	1939.79	840	3613.93	8.400E+07	-	-	-
30m_3_70	371	3394.37	3.710E+07	1882.68	-	-	-	-	-	-
30l_3_70	2589	-	-	-	-	-	-	684	9339.08	5019.04
30s_5_70	756	4492.94	7.560E+07	2624.47	-	-	-	-	-	-
50s_1_30	0	16686.92	**1.669E+04**	**8343.46**	-	-	-	-	-	-
50m_1_30	0	17645.35	1.765E+04	8822.67	0	8264.20	8.264E+03	-	-	-
50s_1_50	401	12290.55	4.011E+07	6345.77	-	-	-	-	-	-
50m_1_50	250	13869.28	2.501E+07	7059.64	-	-	-	-	-	-
50l_1_50	11	15907.70	1.116E+06	7959.35	-	-	-	146	9867.24	5006.62
50s_1_70	1743	11544.01	1.743E+08	6643.50	1772	6898.15	1.772E+08	2370	4006.59	3188.29

(continued on next page)

Computational results obtained by solving the mathematical formulations (continued).

Instance	$\mathscr{F}_{LA}+\mathscr{F}_{CS}$				$\mathscr{F}_{AWS}\ (\alpha >> \beta)$			$\mathscr{F}_{AWS}\ (\alpha = \beta)$		
	E_{LA}	E_{CS}	$E_\tau(\alpha >> \beta)$	$E_\tau(\alpha = \beta)$	E_{LA}	E_{CS}	E_τ	E_{LA}	E_{CS}	E_τ
50m_1_70	1017	19637.64	1.017E+08	10327.32	-	-	-	-	-	-
50l_1_70	599	20988.08	5.992E+07	10793.54	-	-	-	-	-	-
50s_3_70	3904	12233.50	3.904E+08	8068.75	-	-	-	-	-	-
70s_1_30	67	19600.96	6.720E+06	9833.98	-	-	-	-	-	-
70s_1_50	1436	20060.26	1.436E+08	10748.13	-	-	-	-	-	-
70s_1_70	3225	16797.21	3.225E+08	10011.10	-	-	-	-	-	-
70m_1_70	2023	29211.91	2.023E+08	15617.45	-	-	-	-	-	-
70l_1_70	1332	41511.74	1.332E+08	21421.87	-	-	-	-	-	-

The results indicate that the sequential approach yields better LAP solutions for only five instances, 20m_5_70, 30s_1_70, 30m_1_70, 30s_3_70 and 50s_1_70. This can be explained by the observation that in these two cases, none of the alternative approaches finds an optimal solution. For the remaining instances, the value of E_{LA} obtained by $\mathscr{F}_{CWS}(\alpha >> \beta)$ is better than or equal to the value of E_{LA} obtained by $\mathscr{F}_{LA}+\mathscr{F}_{CS}$, while $\mathscr{F}_{CWS}(\alpha >> \beta)$ always finds a better crane scheduling solution. However, $\mathscr{F}_{CWS}(\alpha >> \beta)$ was only able to find feasible solutions for two large instances (number of requests greater than 50). $\mathscr{F}_{CWS}(\alpha >> \beta)$ may change location assignments and is therefore able to find better crane scheduling while still achieving the same quality of location assignments. This shows how by integrating the LAP and the CSP, better location assignments and crane schedules are achievable. Generating better crane schedules by $\mathscr{F}_{CWS}(\alpha >> \beta)$ may require compromises with respect to computational runtimes. The results show that $\mathscr{F}_{CWS}(\alpha >> \beta)$ tends to take more time to achieve optima for small instances. However for larger instances both approaches take the entire runtime.

When α is equal to β, it is evident that the E_{LA} values obtained by $\mathscr{F}_{LA}+\mathscr{F}_{CS}$ and $\mathscr{F}_{CWS}(\alpha >> \beta)$ are lower than those obtained by $\mathscr{F}_{CWS}(\alpha = \beta)$. However, E_τ is significantly lower in $\mathscr{F}_{CWS}(\alpha = \beta)$ than in $\mathscr{F}_{LA}+\mathscr{F}_{CS}(\alpha = \beta)$. Although $\mathscr{F}_{CWS}(\alpha >> \beta)$ is capable of finding good crane schedules while achieving high quality location assignments, the E_{CS} obtained by $\mathscr{F}_{CWS}(\alpha = \beta)$ is lower. There are only two instances, 20l_3_30 and 20l_1_30, where $\mathscr{F}_{CWS}(\alpha >> \beta)$ achieved both better location assignments and crane scheduling within the available runtime. For the remaining instances, $\mathscr{F}_{CS}(\alpha = \beta)$ compromises as regards location assignment to achieve better crane schedules. The number of instances solved to optimality decreases as the number of requests grows.

The detailed table of results including the optimality gap and computation time (cpu) of both approaches is presented in the Appendix (Table 3.A.1).

3.5.3 Comparison of sequential and integrated approaches by the proposed heuristics

The heuristic was implemented in C++11. All experiments were executed for maximum 300 seconds or maximum 10,000 consecutive iterations without improvement for all 135 instances. The

heuristic's parameters were tuned using the irace R-package which implements the iterated racing procedure for automatic algorithm configuration (López-Ibáñez et al., 2016) with a budget of 4,000 runs. The purpose of irace is to automate the task of configuring an optimization algorithm's parameters. It generates and tests a sample of parameter configurations for a given optimization algorithm on a set of instances. When sufficient statistical evidence is collected (by means of a Friedman test) that a certain parameter configuration is outperformed by others, it is discarded so as to focus on the remaining configurations. The best-performing configurations are reported.

Table 3.6 summarizes the parameters by their role, type, range and tuned values for both approaches (sequential and integrated).

Table 3.6: Tuning parameters by irace package

Parameter	Role	Type	Range	irace results	
				integrated	*sequential*
l_{cws}	$\frac{\text{Size of the late acceptance list for CWSP LAHC}}{\text{number of requests}}$	integer	(10, 250)	30	-
l_{lap}	$\frac{\text{Size of the late acceptance list for LAP LAHC}}{\text{number of requests}}$	integer	(10, 250)	-	234
l_{csp}	$\frac{\text{Size of the late acceptance list for CSP LAHC}}{\text{number of requests}}$	integer	(10, 250)	-	79
p_{lr}	Weight of using *LR* neighbourhood	real	(0.0, 1.0)	0.63	0.75
p_{ls}	Weight of using *LS* neighbourhood	real	(0.0, 1.0)	0.45	0.52
p_{gla}	Weight of using *GLA* neighbourhood	real	(0.0, 1.0)	0.59	0.14
p_{cr}	Weight of using *CR* neighbourhood	real	(0.0, 1.0)	0.99	0.83
p_{os}	Weight of using *OS* neighbourhood	real	(0.0, 1.0)	0.02	0.08
p_{ri}	Weight of using *RI* neighbourhood	real	(0.0, 1.0)	0.11	0.30
p_{rbi}	Weight of using *RBI* neighbourhood	real	(0.0, 1.0)	0.75	0.84
p_{nla}	Weight of using *NLA* neighbourhood	real	(0.0, 1.0)	0.73	-
p_{bop}	Weight of using *BOP* neighbourhood	real	(0.0, 1.0)	0.14	0.31
p_{mbo}	Weight of using *MBOP* neighbourhood	real	(0.0, 1.0)	0.06	0.29

Table 3.7 reports the results of the three heuristics when the values of α and β are equal to 0.5 ($\alpha = \beta = 0.5$). The location assignment evaluation (E_{LA}), crane scheduling evaluation (E_{CS}) and total cost (E_τ) obtained by the three heuristics are compared. The heuristic was run five times with different random seeds, and the average of these results is reported. The improvement of the integrated and sequential heuristics over the constructive heuristic (G), the upper bound (UB) defined as the best feasible solution obtained by either the sequential or integrated formulation, and the relative optimality gap (gap) and the computational times (cpu) are reported. The optimality gap is measured by comparing E_τ obtained by the integrated heuristic and the lower bound generated by the mathematical formulations. The gaps may be relatively large as the lower bounds are weak due to employing large numbers (M_L and M_T) in the mathematical formulations. When $\mathscr{F}_{CWS}(\alpha = \beta)$ was unable to find a lower bound due to an out-of-memory error, OOM is reported. Emboldened numbers indicate that optimum solutions were obtained.

The results show how the integrated heuristics find high quality solutions and are able to find optimum solutions for the small instances in a short amount of time. There are only two instances

`10s_1_70` and `10m_1_70` where the integrated heuristic was unable to find the optimum solution. Nevertheless, the optimality gaps are only 1.1% and 5.5%, respectively. There are instances for which the sequential heuristic finds better solutions than the optimum obtained by $\mathscr{F}_{LA} + \mathscr{F}_{CS}$. The explanation for these interesting results lies in how the sequential heuristic finds location assignments which enable better crane scheduling.

Table 3.7: Computational results of Constructive, Sequential and Integrated heuristics ($\alpha = \beta = 0.5$)

Instance	Dispatching rules			Sequential					Integrated						Best MIP bounds	
	E_{LA}	E_{CS}	E_{τ}	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	*gap*	UB	LB
10s_1_30	0	108.0	54.0	0	59.9	30.0	3.1	44.5	1	51.5	**26.3**	2.8	51.4	0.0	26.3	26.3
10m_1_30	0	234.1	117.1	0	37.6	**18.8**	5.4	83.9	0	37.6	**18.8**	3.6	83.9	0.0	18.8	18.8
10l_1_30	0	384.0	192.0	0	79.8	39.9	10.5	79.2	0	76.8	**38.4**	14.2	80.0	0.0	38.4	38.4
10s_1_50	0	235.8	117.9	0	137.9	69.0	3.6	41.5	11	78.9	**45.0**	2.9	61.9	0.0	45.0	45.0
10m_1_50	0	366.8	183.4	0	96.3	48.1	6.9	73.8	7	40.1	**23.5**	5.2	87.2	0.0	23.5	23.5
10l_1_50	0	350.4	175.2	0	45.4	22.7	12.8	87.0	1	35.5	**18.2**	5.0	89.6	0.0	18.2	18.2
10s_1_70	180	767.9	474.0	132	158.4	145.2	4.8	69.4	150	67.7	108.8	3.9	77.0	1.1	107.6	107.6
10m_1_70	48	1031.6	539.8	48	405.8	226.9	9.6	58.0	80	207.7	143.9	5.1	73.3	5.5	136.0	136.0
10l_1_70	11	843.8	427.4	11	166.1	88.5	12.8	79.3	25	73.8	**49.4**	12.2	88.4	0.0	49.4	49.4
10s_3_30	13	324.8	168.9	13	230.8	121.9	6.4	27.8	29	108.9	68.9	4.2	59.2	41.5	62.5	40.3
10m_3_30	1	400.8	200.9	1	178.2	89.6	31.3	55.4	6	76.4	41.2	8.3	79.5	81.1	39.7	7.8
10l_3_30	0	700.8	350.4	0	98.5	49.3	24.6	85.9	3	60.3	31.7	14.3	91.0	78.2	147.6	6.9
10s_3_50	58	538.0	298.0	55	238.0	146.5	16.4	50.8	73	98.0	85.5	5.2	71.3	48.6	86.9	43.9
10m_3_50	68	686.9	377.4	31	237.5	134.2	29.7	64.4	36	88.5	62.2	17.6	83.5	73.9	83.9	16.2
10l_3_50	9	1181.3	595.2	9	258.0	133.5	38.0	77.6	12	85.9	**49.0**	23.9	91.8	0.0	49.0	49.0
10s_3_70	150	1309.0	729.5	130	363.3	246.7	21.1	66.2	182	72.5	127.2	22.7	82.6	29.2	146.2	90.0
10m_3_70	64	1210.5	637.2	64	273.8	168.9	20.1	73.5	79	104.8	91.9	11.1	85.6	26.3	85.5	67.7
10l_3_70	106	697.3	401.7	106	175.5	140.7	67.3	65.0	115	83.2	99.1	34.7	75.3	42.9	126.1	56.5
10s_5_30	18	509.4	263.7	18	195.5	106.8	16.7	59.5	36	94.2	65.1	16.4	75.3	66.8	64.1	21.6
10m_5_30	9	972.4	490.7	9	372.8	190.9	23.1	61.1	29	66.8	47.9	11.1	90.2	77.5	50.2	10.8
10l_5_30	6	988.0	497.0	6	417.5	211.7	43.7	57.4	18	95.1	56.6	30.0	88.6	70.6	111.4	16.6
10s_5_50	66	717.0	391.5	66	288.8	177.4	16.2	54.7	74	112.2	93.1	19.5	76.2	54.5	250.8	42.3
10m_5_50	91	619.3	355.1	25	420.3	222.6	35.2	37.3	36	134.7	85.4	37.5	76.0	81.7	119.3	15.6
10l_5_50	13	772.0	392.5	13	199.3	106.1	52.2	73.0	19	85.1	52.1	47.6	86.7	56.3	52.0	22.8
10s_5_70	89	374.7	231.9	89	171.9	130.5	37.2	43.7	100	91.9	95.9	16.4	58.6	32.4	95.9	64.9
10m_5_70	59	1270.5	664.7	59	307.0	183.0	80.4	72.5	76	141.2	108.6	51.1	83.7	72.2	143.1	30.2
10l_5_70	44	1426.0	735.0	44	304.8	174.4	91.1	76.3	47	78.2	62.6	64.4	91.5	22.1	69.0	48.8
20s_1_30	0	358.9	179.4	0	190.5	95.2	8.3	46.9	1	152.7	76.8	5.3	57.2	63.0	610.5	28.4
20m_1_30	0	854.5	427.2	0	243.9	122.0	150.1	71.5	10	202.0	106.0	8.8	75.2	83.1	118.5	17.9
20l_1_30	0	2741.4	1370.7	0	205.4	102.7	21.3	92.5	0	195.2	97.6	19.0	92.9	67.5	130.6	31.7
20s_1_50	4	1652.1	828.1	4	540.7	272.3	50.3	67.1	31	249.5	140.3	13.8	83.1	74.9	708.3	35.2

(continued on next page)

Computational results of Dispatching rules, Sequential and Integrated heuristics (continued)

Instance	Dispatching rules			Sequential					Integrated						Best MIP bounds	
	E_{LA}	E_{CS}	E_{τ}	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	*gap*	UB	LB
20m_1_50	0	1556.1	778.1	0	467.9	233.9	71.9	69.9	23	272.5	147.7	13.4	81.0	88.4	180.2	17.2
20l_1_50	0	904.6	452.3	0	185.7	92.9	22.8	79.5	5	117.6	61.3	17.6	86.4	93.4	63.1	4.0
20s_1_70	550	2862.2	1706.1	448	629.7	538.8	17.7	68.4	473	202.7	337.8	10.5	80.2	30.0	338.1	236.6
20m_1_70	258	4543.9	2401.0	244	1345.9	794.9	87.5	66.9	341	466.8	403.9	20.0	83.2	63.6	486.6	147.1
20l_1_70	77	3110.3	1593.6	77	726.1	401.5	150.4	74.8	99	233.7	166.4	25.0	89.6	61.1	212.6	64.7
20s_3_30	100	2680.3	1390.1	54	957.0	505.5	21.1	63.6	92	203.3	147.7	20.4	89.4	79.6	691.0	30.1
20m_3_30	15	2364.1	1189.6	15	818.8	416.9	45.4	65.0	66	327.2	196.6	38.5	83.5	90.5	370.3	18.7
20l_3_30	0	2033.1	1016.6	0	307.8	153.9	50.5	84.9	34	153.0	93.5	145.5	90.8	92.2	227.2	7.3
20s_3_50	225	1467.3	846.1	159	1485.1	822.0	37.6	2.8	198	362.3	280.1	20.6	66.9	79.0	799.1	58.8
20m_3_50	130	3485.8	1807.9	97	1523.4	810.2	54.5	55.2	103	522.0	312.5	73.7	82.7	90.2	273.6	30.7
20l_3_50	45	4481.3	2263.1	45	840.6	442.8	150.2	80.4	74	156.2	115.1	79.0	94.9	73.7	313.7	30.3
20s_3_70	543	5436.3	2989.6	432	815.0	623.5	27.2	79.1	476	268.0	372.0	28.5	87.6	45.5	459.4	202.8
20m_3_70	197	4097.8	2147.4	181	1341.3	761.1	150.2	64.6	187	392.5	289.8	39.2	86.5	72.9	829.8	78.5
20l_3_70	307	3384.8	1845.9	274	1583.9	929.0	152.5	49.7	323	259.8	291.4	110.3	84.2	65.1	1632.0	101.8
20s_5_30	80	2458.7	1269.4	64	731.3	397.6	24.6	68.7	86	329.6	207.8	42.5	83.6	88.4	931.5	24.0
20m_5_30	37	2977.7	1507.3	37	1041.4	539.2	150.1	64.2	63	263.9	163.5	41.3	89.2	88.0	-	19.5
20l_5_30	30	3035.3	1532.6	30	814.3	422.1	151.7	72.5	56	275.9	165.9	70.6	89.2	-	-	OOM
20s_5_50	210	2552.3	1381.1	194	1164.5	679.2	47.0	50.8	220	273.9	246.9	36.0	82.1	80.4	1289.2	48.4
20m_5_50	192	2472.6	1332.3	82	1343.0	712.5	94.2	46.5	113	549.4	331.2	151.6	75.1	-	-	OOM
20l_5_50	108	2848.3	1478.1	65	1441.4	753.2	153.9	49.0	90	249.4	169.7	158.6	88.5	-	-	OOM
20s_5_70	390	3077.3	1733.6	310	1637.3	973.6	47.7	43.8	395	428.7	411.9	73.4	76.2	75.5	914.8	100.8
20m_5_70	325	4904.8	2614.9	248	1530.8	889.4	127.0	66.0	282	655.0	468.5	116.7	82.1	87.1	842.2	60.6
20l_5_70	130	5138.0	2634.0	123	1317.5	720.3	173.6	72.7	157	300.0	228.5	150.8	91.3	77.4	485.1	51.7
30s_1_30	0	1777.2	888.6	0	447.0	223.5	28.0	74.8	19	309.4	164.2	29.8	81.5	88.0	1747.9	19.7
30m_1_30	0	2025.8	1012.9	0	617.2	308.6	38.0	69.5	10	448.1	229.1	36.5	77.4	88.3	2250.9	26.8
30l_1_30	0	6808.6	3404.3	0	348.5	174.3	47.3	94.9	4	334.3	169.1	46.1	95.0	85.7	2455.5	24.2
30s_1_50	113	4217.1	2165.0	104	1659.9	882.0	38.6	59.3	124	515.4	319.7	33.2	85.2	79.0	469.8	67.2
30m_1_50	52	4486.5	2269.2	19	2101.4	1060.2	59.4	53.3	122	576.2	349.1	62.5	84.6	93.8	977.9	21.6
30l_1_50	7	2601.7	1304.3	2	532.6	267.3	150.0	79.5	20	243.7	131.8	57.0	89.9	89.4	2737.8	13.9
30s_1_70	1121	6534.2	3827.6	974	1801.1	1387.6	130.9	63.7	979	465.7	722.3	24.3	81.1	93.9	1675.5	43.8
30m_1_70	616	9826.3	5221.1	575	2593.5	1584.3	150.6	69.7	752	717.7	734.8	52.9	85.9	60.0	1338.6	294.2

(continued on next page)

Computational results of Dispatching rules, Sequential and Integrated heuristics (continued)

Instance	Dispatching rules			Sequential					Integrated						Best MIP bounds	
	E_{LA}	E_{CS}	E_{τ}	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	*gap*	UB	LB
30l_1_70	287	7409.5	3848.2	249	1790.4	1019.7	93.5	73.5	280	316.8	298.4	73.8	92.2	53.7	945.6	138.1
30s_3_30	173	5440.2	2806.6	132	2694.2	1413.1	39.7	49.7	182	531.6	356.8	93.9	87.3	89.6	2221.7	37.0
30m_3_30	59	5332.9	2695.9	59	2340.1	1199.5	87.3	55.5	196	665.9	431.0	100.6	84.0	-	-	OOM
30l_3_30	35	3494.1	1764.5	12	1365.2	688.6	219.5	61.0	47	282.5	164.7	148.0	90.7	-	-	OOM
30s_3_50	414	4759.2	2586.6	313	3123.0	1718.0	39.8	33.6	351	630.9	490.9	76.1	81.0	-	1986.6	OOM
30m_3_50	234	8504.2	4369.1	215	4333.2	2274.1	150.8	48.0	415	949.6	682.3	110.4	84.4	-	-	OOM
30l_3_50	115	9105.0	4610.0	115	3623.0	1869.0	150.3	59.5	152	429.5	290.7	300.0	93.7	-	-	OOM
30s_3_70	1075	10665.2	5870.1	835	3733.2	2284.1	65.0	61.1	935	797.1	866.0	79.8	85.2	65.2	1939.8	301.1
30m_3_70	408	10669.2	5538.6	355	3102.6	1728.8	160.8	68.8	449	897.8	673.4	126.0	87.8	80.6	1882.7	130.8
30l_3_70	630	11675.3	6152.7	526	4479.5	2502.8	177.6	59.3	599	657.2	628.1	226.3	89.8	84.5	5019.0	97.4
30s_5_30	193	5631.7	2912.3	170	1765.5	967.7	150.7	66.8	199	788.4	493.7	73.4	83.0	-	-	OOM
30m_5_30	92	8734.2	4413.1	92	2131.7	1111.8	162.5	74.8	110	616.2	363.1	133.4	91.8	-	-	OOM
30l_5_30	61	6883.5	3472.3	61	1500.8	780.9	217.6	77.5	101	545.0	323.0	180.1	90.7	-	-	OOM
30s_5_50	391	7790.2	4090.6	340	4467.2	2403.6	83.4	41.2	448	846.6	647.3	135.9	84.2	90.6	-	61.0
30m_5_50	275	6811.0	3543.0	164	5079.2	2621.6	154.4	26.0	220	1121.1	670.5	300.0	81.1	-	-	OOM
30l_5_50	246	10671.6	5458.8	158	3113.9	1635.9	246.4	70.0	212	464.1	338.1	300.0	93.8	-	-	OOM
30s_5_70	925	8077.2	4501.1	605	3442.2	2023.6	88.6	55.0	805	1182.9	993.9	135.2	77.9	84.0	2624.5	158.7
30m_5_70	654	9369.2	5011.6	528	4248.2	2388.1	190.3	52.3	596	1537.4	1066.7	259.9	78.7	-	-	OOM
30l_5_70	262	10267.7	5264.8	262	4647.1	2454.5	150.5	53.4	302	660.7	481.3	300.0	90.9	-	-	OOM
50s_1_30	2	8556.2	4279.1	0	3348.4	1674.2	113.0	60.9	93	1188.1	640.5	170.4	85.0	89.6	8343.5	66.5
50m_1_30	0	9082.2	4541.1	0	2185.7	1092.9	150.2	75.9	83	1502.5	792.7	130.0	82.5	92.8	8822.7	56.7
50l_1_30	0	22949.0	11474.5	0	1881.2	940.6	150.5	91.8	5	1478.0	741.5	191.0	93.5	-	-	OOM
50s_1_50	601	17726.0	9163.5	394	4512.3	2453.2	159.5	73.2	744	1162.0	953.0	249.5	89.6	79.1	6345.8	198.9
50m_1_50	161	15745.8	7953.4	116	7850.4	3983.2	159.6	49.9	422	2410.4	1416.2	272.4	82.2	94.1	7059.6	83.0
50l_1_50	39	14297.6	7168.3	13	6219.1	3116.1	183.4	56.5	85	1107.8	596.4	270.4	91.7	91.4	5006.6	51.1
50s_1_70	1954	17341.6	9647.8	1821	9377.2	5599.1	152.3	42.0	1928	1237.8	1582.9	189.4	83.6	49.7	3188.3	796.8
50m_1_70	1179	24878.6	13028.8	1002	14598.5	7800.3	151.8	40.1	1258	3164.2	2211.1	300.0	83.0	76.8	10327.3	513.7
50l_1_70	711	25748.3	13229.7	589	7576.5	4082.7	165.6	69.1	678	1234.7	956.4	284.9	92.8	67.5	10793.5	311.1
50s_3_30	394	17075.2	8734.6	276	13931.2	7103.6	209.8	18.7	421	2773.2	1597.1	226.4	81.7	-	-	OOM
50m_3_30	266	22043.1	11154.6	127	14950.5	7538.8	168.8	32.4	349	3295.0	1822.0	300.0	83.7	-	-	OOM
50l_3_30	131	18198.4	9164.7	53	10866.7	5459.9	234.4	40.4	162	1585.0	873.5	300.0	90.5	-	-	OOM

(continued on next page)

Computational results of Dispatching rules, Sequential and Integrated heuristics (continued)

Instance	Dispatching rules			Sequential					Integrated						Best MIP bounds	
	E_{LA}	E_{CS}	E_{τ}	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	*gap*	UB	LB
50s_3_50	858	18143.2	9500.6	657	11256.2	5956.6	168.5	37.3	836	3095.3	1965.7	256.7	79.3	93.2	-	133.4
50m_3_50	533	27830.5	14181.8	512	18231.5	9371.8	153.7	33.9	1146	4392.9	2769.4	300.0	80.5	-	-	OOM
50l_3_50	286	30411.5	15348.8	261	13002.6	6631.8	300.0	56.8	409	2629.2	1519.1	300.0	90.1	-	-	OOM
50s_3_70	2107	29275.2	15691.1	1702	12482.2	7092.1	156.3	54.8	2075	2676.2	2375.6	300.0	84.9	76.2	8068.8	564.4
50m_3_70	1111	27175.5	14143.3	894	17357.5	9125.8	154.9	35.5	1536	5780.3	3658.2	300.0	74.1	-	-	OOM
50l_3_70	1276	43016.4	22146.2	1175	19921.1	10548.1	157.8	52.4	1588	6316.4	3952.2	300.0	82.2	-	-	OOM
50s_5_30	575	18402.2	9488.6	375	11178.2	5776.6	258.3	39.1	606	3060.0	1833.0	300.0	80.7	-	-	OOM
50m_5_30	346	36127.5	18236.8	265	17397.6	8831.3	154.2	51.6	590	6258.6	3424.3	300.0	81.2	-	-	OOM
50l_5_30	162	25689.1	12925.5	162	10541.8	5351.9	151.5	58.6	373	2566.4	1469.7	300.0	88.6	-	-	OOM
50s_5_50	1370	19335.2	10352.6	1107	15368.2	8237.6	172.5	20.4	1565	4222.2	2893.6	300.0	72.0	-	-	OOM
50m_5_50	462	26708.5	13585.3	352	16759.5	8555.8	254.6	37.0	789	5450.7	3119.9	300.0	77.0	-	-	OOM
50l_5_50	469	37730.4	19099.7	343	19026.3	9684.7	155.5	49.3	481	4269.1	2375.1	300.0	87.6	-	-	OOM
50s_5_70	2139	23310.9	12725.0	1771	13313.9	7542.5	190.0	40.7	2467	4499.9	3483.5	300.0	72.6	-	-	OOM
50m_5_70	1813	29448.5	15630.8	1450	21269.5	11359.8	200.0	27.3	2203	10097.7	6150.4	300.0	60.7	-	-	OOM
50l_5_70	683	35730.4	18206.7	683	21014.4	10848.7	154.5	40.4	965	8154.3	4559.7	300.0	75.0	-	-	OOM
70s_1_30	219	20333.6	10276.3	71	20307.0	10189.0	158.7	0.8	612	3768.6	2190.3	300.0	78.7	96.3	9834.0	80.5
70m_1_30	0	20330.8	10165.4	0	5340.7	2670.3	150.4	73.7	94	3047.8	1570.9	300.0	84.5	-	-	OOM
70l_1_30	0	40020.2	20010.1	0	364.9	3182.4	151.4	84.1	13	2863.9	1438.4	300.0	92.8	-	-	OOM
70s_1_50	1554	36856.8	19205.4	1157	11944.0	6550.5	178.3	65.9	2077	3952.7	3014.8	300.0	84.3	85.2	10748.1	447.7
70m_1_50	559	34213.5	17386.2	466	25316.2	12891.1	155.1	25.9	1491	8227.0	4859.0	300.0	72.1	96.2	-	185.9
70l_1_50	277	40254.4	20265.7	123	18342.6	9232.8	230.5	54.4	397	2683.7	1540.4	300.0	92.4	-	-	OOM
70s_1_70	3760	37919.9	20840.0	3154	15801.4	9477.7	168.5	54.5	3627	3968.3	3797.7	300.0	81.8	64.0	10011.1	1366.7
70m_1_70	2132	50231.3	26181.6	1926	33423.5	17674.7	157.0	32.5	3026	8293.4	5659.7	300.0	78.4	84.1	15617.5	898.4
70l_1_70	1427	55031.2	28229.1	1231	21957.1	11594.1	198.1	58.9	2378	6132.0	4255.0	300.0	84.9	86.5	21421.9	575.7
70s_3_30	884	33551.0	17217.5	587	21845.0	11216.0	285.6	34.9	1050	6344.0	3697.0	300.0	78.5	-	-	OOM
70m_3_30	472	38405.1	19438.6	374	33120.1	16747.1	300.0	13.8	843	11464.7	6153.9	300.0	68.3	-	-	OOM
70l_3_30	263	38437.7	19350.3	141	36134.7	18137.8	251.9	6.3	357	7241.7	3799.4	300.0	80.4	-	-	OOM
70s_3_50	1511	42315.0	21913.0	1186	19875.0	10530.5	216.1	51.9	1717	7041.0	4379.0	300.0	80.0	-	-	OOM
70m_3_50	988	46579.1	23783.6	820	37096.1	18958.1	204.6	20.3	1699	12155.5	6927.3	300.0	70.9	-	-	OOM
70l_3_50	656	64629.5	32642.8	582	41251.4	20916.7	160.7	35.9	1257	12773.1	7015.0	300.0	78.5	-	-	OOM
70s_3_70	3647	47892.0	25769.5	3127	24973.0	14050.0	163.2	45.5	4032	6907.0	5469.5	300.0	78.8	-	-	OOM

(continued on next page)

Computational results of Dispatching rules, Sequential and Integrated heuristics (continued)

Instance	Dispatching rules			Sequential					Integrated						Best MIP bounds	
	E_{LA}	E_{CS}	E_{τ}	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	E_{LA}	E_{CS}	E_{τ}	*cpu*	G%	*gap*	UB	LB
70m_3_70	2089	56892.1	29490.6	1570	32673.1	17121.6	276.9	41.9	3061	13535.0	8298.0	300.0	71.9	-	-	OOM
70l_3_70	2057	70640.2	36348.6	1873	51255.7	26564.4	300.0	26.9	2278	18493.8	10385.9	300.0	71.4	-	-	OOM
70s_5_30	1162	34009.0	17585.5	816	22656.0	11736.0	300.0	33.3	1411	7709.0	4560.0	300.0	74.1	-	-	OOM
70m_5_30	624	73034.1	36829.1	532	37281.1	18906.6	300.0	48.7	931	12688.8	6809.9	300.0	81.5	-	-	OOM
70l_5_30	414	65472.1	32943.0	322	40266.3	20294.2	160.1	38.4	641	14891.1	7766.0	300.0	76.4	-	-	OOM
70s_5_50	2265	41174.0	21719.5	1840	25715.0	13777.5	169.6	36.6	2536	9337.0	5936.5	300.0	72.7	-	-	OOM
70m_5_50	850	53458.1	27154.1	751	30716.1	15733.6	160.5	42.1	986	14888.3	7937.2	300.0	70.8	-	-	OOM
70l_5_50	739	58429.7	29584.4	626	38672.4	19649.2	300.0	33.6	1089	16691.6	8890.3	300.0	69.9	-	-	OOM
70s_5_70	3568	39905.7	21736.9	2765	26061.7	14413.4	178.4	33.7	3506	10957.7	7231.9	300.0	66.7	-	-	OOM
70m_5_70	2882	62550.1	32716.1	2473	41072.1	21772.6	241.2	33.4	3281	19893.1	11587.1	300.0	64.6	-	-	OOM
70l_5_70	1332	72718.7	37025.4	1162	54057.5	27609.8	300.0	25.4	1753	23893.8	12823.4	300.0	65.4	-	-	OOM

Figure 3.9 illustrates how the integrated heuristic may improve over the constructive heuristic. The instances are grouped by their number of requests and storage sizes. The horizontal axis represents the instance categories while the vertical axis corresponds to the improvement over constructive heuristics. The results are averaged over each category and demonstrate that the improvement over the constructive heuristic is, on average, 82% over all instances. Given that the constructive heuristic reflects common operational dispatching rules, the potential performance increase is large.

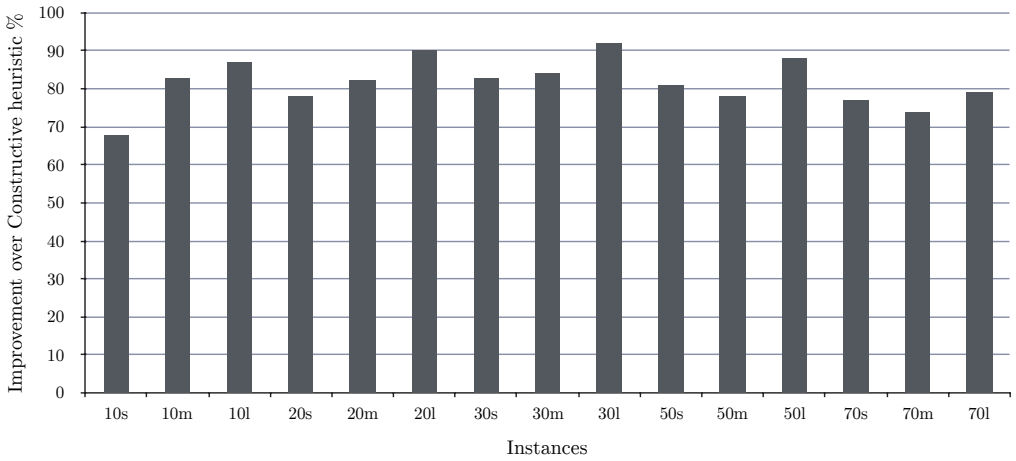


Figure 3.9: Comparison of the constructive and integrated heuristic's performance

3.5.4 Weight parameters and instance analysis

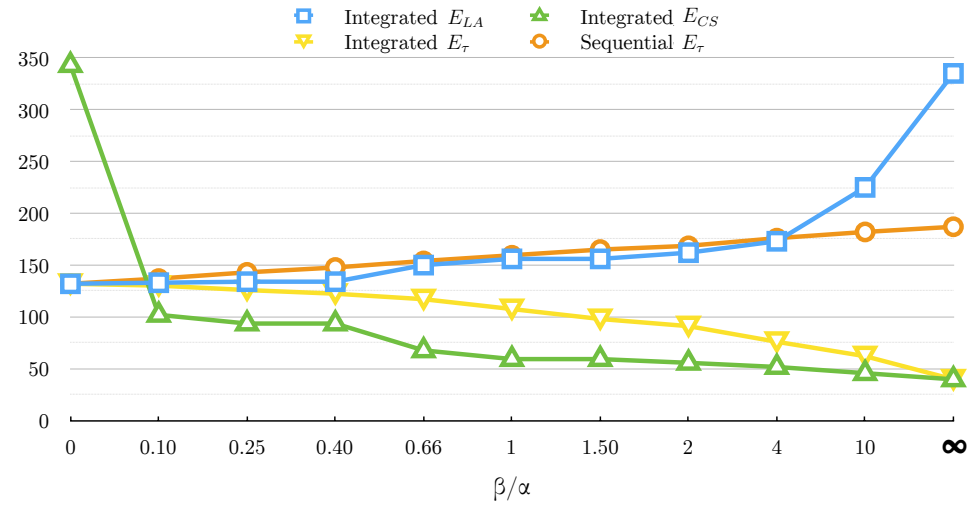
Section 3.5.2 already briefly touched on the impact of the weight parameters α and β. In this section, the effect of adjusting these weight parameters is further investigated. Both a small (`10s_1_70`) and large (`50m_1_50`) instance were selected to illustrate the weight parameters' effect. Figure 3.10 presents the impact of adjusting α and β for both the small (Figure 3.10(a), solved by $\mathscr{F}_{LA} + \mathscr{F}_{CS}$ and $\mathscr{F}_{CWS}$), and the large instance (Figure 3.10(b)), solved by the sequential and integrated heuristic. The figure illustrates the LAP (E_{LA}), CSP (E_{CS}) and weighted objective function (E_τ) values obtained by the integrated approach as well as the E_τ obtained by the sequential approach.

The weight parameters in the sequential approach do not impact upon the optimization process as the two sub-problems are solved separately. Sequential E_{LA} and E_{CS} are constant for all values of α and β. Therefore $E_\tau(\alpha, \beta)$ is a linear function.

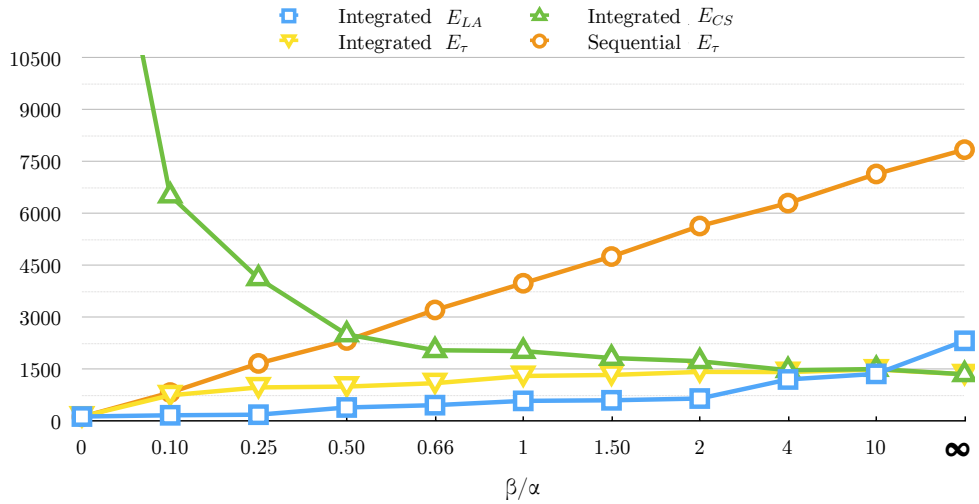
As illustrated for $\frac{\beta}{\alpha} = 0$:

$$\text{integrated } E_\tau = \text{sequential } E_\tau = \text{integrated } E_{LA} = \text{sequential } E_{LA}$$

By increasing $\frac{\beta}{\alpha}$, the integrated E_{LA} increases while the integrated E_{CS} decreases. Interestingly the integrated E_τ is always lower than the sequential E_τ and by increasing $\frac{\beta}{\alpha}$ the gap between the integrated and sequential E_τ increases.



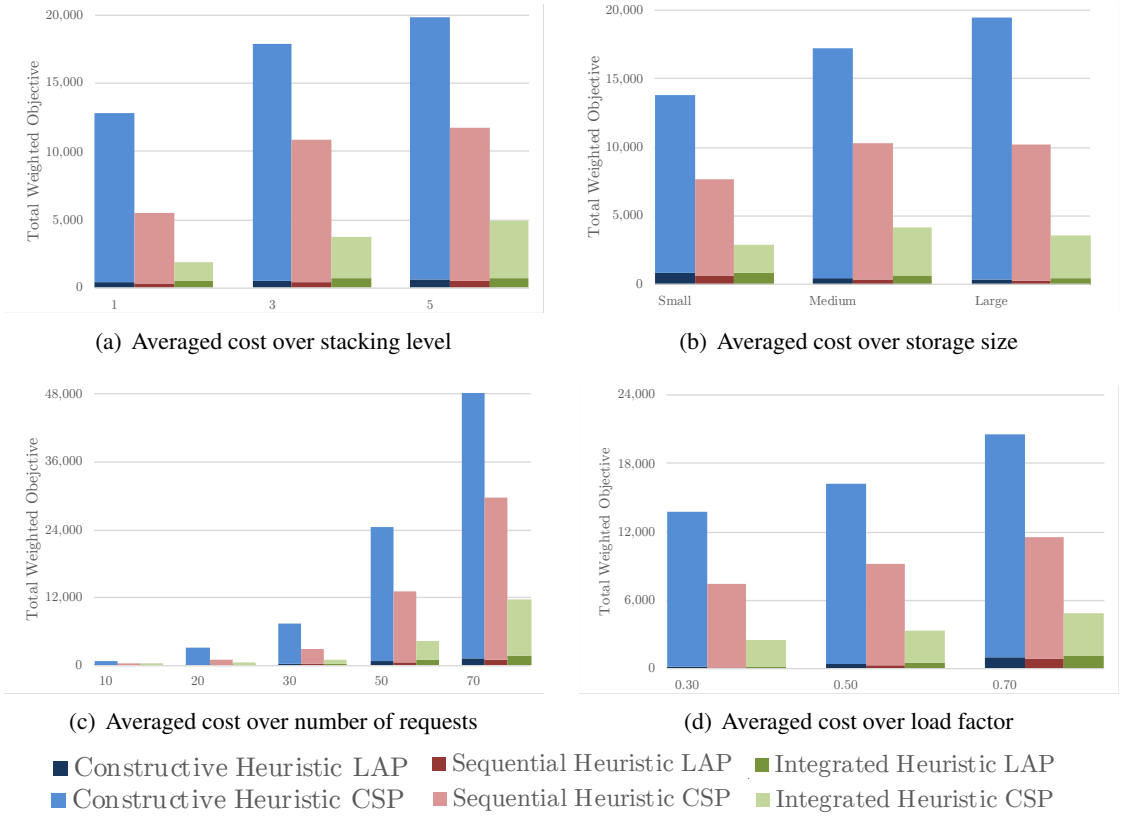
(a) Instance `10s_1_70` solved by the mathematical formulation



(b) Instance `50m_1_50` solved by the heuristic

Figure 3.10: Impact of the weight parameters' ratio.

Figure 3.11 demonstrates how different attributes of instances may impact the improvement of the sequential and integrated heuristic over the constructive heuristic. Figure 3.11(a) represents the average E_τ obtained by the constructive, sequential and integrated heuristic with respect to the stacking level. By increasing the maximum stacking level, the relative improvement over the solution obtained by the constructive heuristic decreases. By increasing the storage size from small to large (Figure 3.11(b)), the sequential and integrated heuristics improve the initial solution from 44% to 47% and from 78% to 81% respectively. Figure 3.11(d), meanwhile, illustrates how the load factor does not significantly influence the algorithms' improvement over the initial solution.



(a) Averaged cost over stacking level

(b) Averaged cost over storage size

(c) Averaged cost over number of requests

(d) Averaged cost over load factor

Figure 3.11: Instance attributes analysis by the constructive, sequential and integrated heuristic when $\alpha = \beta$.

3.6. Conclusion

This paper investigated the impact of integrating a location assignment problem (LAP) and a crane scheduling problem (CSP) in crane-operated warehouses by introducing the integrated Crane-operated Warehouse Scheduling Problem (CWSP). The CWSP assigns a storage location to input requests, assigns a crane to execute each request and decides how the set of requests must be sequenced per crane in such a way that the total storage cost and tardiness is minimized. Mixed Integer Programming (MIP) formulations for the LAP and CSP were presented in addition to a continuous-time MIP formulation which integrates both the LAP and CSP. This model considers realistic crane interactions in the storage area where cranes cannot pass each other and must keep a safety distance. Furthermore, a meta-heuristic based on Late Acceptance Hill Climbing (LAHC) was developed to overcome the limited scaling ability of solving the mathematical formulations by MIP solvers. In addition, 135 instances with various problem specifications were generated to enable validation and encourage future research.

A comprehensive computational study revealed how integrating the LAP and the CSP may lead to 48% improvement for the CSP while keeping the same level of quality for the LAP solutions. Subsequently, the results showed that by integrating the LAP and CSP into one problem, there is a significant reduction in the total weighted objective of 62%. Furthermore, the benefit of the

integrated heuristic over the MIP formulation was shown, where better solutions for both medium and large size instances were obtained compared to solving the MIP formulation on the respective instances. Additionally, a simulation of a real-world automated warehouse shows a significant potential for minimizing the storage cost and tardiness of the requests, when comparing the new procedures with typical dispatching rules.

In conclusion, integrating location assignment and crane scheduling coordinates the resources in automated warehouses or container terminals more effectively and eventually leads to efficiently storing the products or containers in the storage areas as well as minimizing the tardiness of the input and output requests.

Several interesting avenues exist to build upon this study in future research. Further solution approaches may be considered and investigated. Particularly, exact solution approaches would be a valuable contribution which may include proposing efficient lower bounding methods. Due to the nature of operations in automated warehouses, other research directions may focus on exploring the development of robust scheduling models which consider uncertain requests' arrival times.

Appendix

3.A. Detailed computational results

Table 3.A.1 presents detailed computational results obtained by the mathematical formulations. $gap_{la}\%$ (location assignment gap) and $gap_{cs}\%$ (crane scheduling gap) indicate the gap obtained by $\mathscr{F}_{LA}$ and $\mathscr{F}_{CS}$, respectively, when solving the problem sequentially. $gap\%$ denotes the gap obtained by $\mathscr{F}_{AWS}$.

Table 3.A.1: Detailed computational results obtained by mathematical formulations

Ins.	$\mathscr{F}_{LA}+\mathscr{F}_{CS}$							$\mathscr{F}_{AWS}$ $(\alpha >> \beta)$					$\mathscr{F}_{AWS}$ $(\alpha = \beta)$				
	E_{LA}	E_{CS}	$E_\tau(\alpha >> \beta)$	$E_\tau(\alpha = \beta)$	$gap_{la}\%$	$gap_{cs}\%$	cpu	E_{LA}	E_{CS}	E_τ	$gap\%$	cpu	E_{LA}	E_{CS}	E_τ	$gap\%$	cpu
10s_1_30	0	267.03	2.670E+02	133.51	0.00	0.00	107.28	0	55.13	5.513E+01	0.00	49.79	1	51.53	26.26	0.00	170.92
10m_1_30	0	478.41	4.784E+02	239.20	0.00	0.00	604.05	0	37.62	3.762E+01	0.00	60.78	0	37.62	18.81	0.00	513.46
10l_1_30	0	515.69	5.157E+02	257.84	0.00	0.00	302.12	0	76.82	7.682E+01	0.00	195.32	0	76.82	38.41	0.00	2044.59
10s_1_50	0	229.25	2.293E+02	114.63	0.00	0.00	67.69	0	123.93	1.239E+02	0.00	311.87	11	78.92	44.96	0.00	1009.35
10m_1_50	0	219.18	2.192E+02	109.59	0.00	0.00	360.00	0	56.45	5.645E+01	0.00	91.66	7	40.07	23.53	0.00	475.39
10l_1_50	0	447.85	4.479E+02	223.92	0.00	0.00	164.24	0	44.49	4.449E+01	0.00	97.42	4	32.49	18.24	0.00	187.29
10s_1_70	132	186.97	1.320E+07	159.49	0.00	0.00	115.83	132	135.97	1.320E+07	0.00	11.06	156	59.25	107.62	0.00	1227.04
10m_1_70	48	348.86	4.800E+06	198.43	0.00	0.00	121.24	48	449.55	4.800E+06	0.00	11.09	86	185.92	135.96	0.00	3290.77
10l_1_70	11	271.44	1.100E+06	141.22	0.00	0.00	155.79	11	148.17	1.100E+06	0.00	27.31	25	73.78	49.39	0.00	613.51
10s_3_30	13	352.10	1.300E+06	182.55	0.00	0.00	442.41	13	179.72	1.300E+06	0.00	119.08	49	76.07	62.53	35.51	3600.00
10m_3_30	1	354.66	1.004E+05	177.83	0.00	0.00	1254.72	1	107.51	1.001E+05	0.00	3600.00	9	70.37	39.68	80.35	3600.00
10l_3_30	0	295.28	2.953E+02	147.64	0.00	0.00	746.32	0	79.51	7.951E+01	0.00	2387.71	-	-	-	-	3600.00
10s_3_50	53	317.77	5.300E+06	185.39	0.00	0.00	879.75	53	201.77	5.300E+06	0.00	3535.55	64	109.83	86.91	49.46	3600.00
10m_3_50	31	409.77	3.100E+06	220.39	0.00	0.00	911.50	31	332.35	3.100E+06	0.00	303.92	41	126.77	83.88	80.67	3600.00
10l_3_50	9	454.43	9.005E+05	231.72	0.00	0.00	1056.12	9	145.71	9.001E+05	0.00	467.58	12	85.91	48.95	0.00	2366.16
10s_3_70	130	392.07	1.300E+07	187.73	0.00	0.00	187.72	130	279.13	1.300E+07	0.00	217.53	190	102.40	146.20	38.42	3600.00
10m_3_70	64	433.66	6.400E+06	248.83	0.00	0.00	212.35	64	492.57	6.400E+06	0.00	92.46	93	77.90	85.45	20.80	3600.00
10l_3_70	98	441.02	9.800E+06	269.51	0.00	0.00	574.97	98	566.37	9.800E+06	0.00	502.12	134	118.11	126.05	55.16	3600.00
10s_5_30	18	322.97	1.800E+06	170.48	0.00	0.00	2714.13	18	182.53	1.800E+06	0.00	3600.00	35	93.23	64.11	66.30	3600.00
10m_5_30	653	477.18	6.530E+07	565.09	100.00	0.00	3224.10	9	275.46	9.002E+05	0.01	3600.00	24	76.36	50.18	78.51	3600.00
10l_5_30	464	457.64	4.640E+07	460.82	100.00	0.00	2845.26	6	244.62	6.002E+05	0.00	3600.00	16	206.88	111.44	85.09	3600.00
10s_5_50	66	435.57	6.600E+06	250.79	0.00	0.00	2300.19	66	292.55	6.600E+06	6.82	3600.00	-	-	-	-	3600.00
10m_5_50	25	-	-	-	0.00	-	3600.00	25	294.46	2.500E+06	42.16	3600.00	42	196.59	119.30	86.90	3600.00
10l_5_50	13	602.07	1.301E+06	307.54	0.00	0.00	2306.55	13	159.11	1.300E+06	0.00	3391.92	21	82.93	51.96	56.24	3600.00
10s_5_70	83	247.93	8.300E+06	165.64	0.00	0.00	197.83	83	247.93	8.300E+06	0.00	176.25	100	91.85	95.92	32.34	3600.00
10m_5_70	59	419.26	5.900E+06	239.13	0.00	0.00	1313.89	59	263.06	5.900E+06	0.00	3600.00	75	211.27	143.13	78.93	3600.00
10l_5_70	43	541.38	4.301E+06	292.19	0.00	0.00	1755.06	43	418.37	4.300E+06	0.00	1223.19	51	87.02	69.01	29.27	3600.00
20s_1_30	0	1221.04	1.221E+03	610.52	0.00	0.00	1970.28	0	160.38	1.603E+02	35.40	3600.00	-	-	-	-	3600.00
20m_1_30	0	1821.80	1.822E+03	910.90	0.00	0.00	2098.25	0	225.29	2.252E+02	37.60	3600.00	1	235.90	118.45	84.91	3600.00
20l_1_30	0	2171.42	2.171E+03	1085.71	0.00	0.00	2425.32	0	198.40	1.984E+02	0.00	3420.09	0	261.18	130.59	75.71	3600.00

(continued on next page)

Detailed computational results obtained by mathematical formulations (continued)

Ins.	$\mathscr{F}_{LA}+\mathscr{F}_{CS}$							$\mathscr{F}_{AWS}$ $(\alpha >> \beta)$					$\mathscr{F}_{AWS}$ $(\alpha = \beta)$				
	E_{LA}	E_{CS}	$E_\tau(\alpha >> \beta)$	$E_\tau(\alpha = \beta)$	$gap_{la}\%$	$gap_{cs}\%$	cpu	E_{LA}	E_{CS}	E_τ	$gap\%$	cpu	E_{LA}	E_{CS}	E_τ	$gap\%$	cpu
20s_1_50	4	1412.59	4.014E+05	708.30	0.00	0.00	1898.00	4	425.09	4.004E+02	0.09	3600.00	-	-	-	-	3600.00
20m_1_50	0	1373.24	1.373E+03	686.21	0.00	0.00	1967.59	0	445.37	4.453E+02	85.75	3600.00	9	351.42	180.21	90.47	3600.00
20l_1_50	0	1851.12	1.851E+03	925.56	0.00	0.00	2205.82	0	155.85	1.558E+02	0.00	3600.00	8	118.23	63.11	93.63	3600.00
20s_1_70	448	801.34	4.480E+07	624.67	0.00	0.00	2062.93	448	692.26	4.480E+07	0.00	2151.22	480	196.16	338.08	30.03	3600.00
20m_1_70	241	2396.80	2.410E+07	1318.90	0.00	0.00	1948.97	241	1287.71	2.410E+07	0.00	1295.57	333	640.11	486.55	69.77	3600.00
20l_1_70	77	1210.96	7.701E+06	643.98	0.00	0.00	2020.08	77	758.37	7.700E+06	0.00	358.51	88	337.18	212.59	69.57	3600.00
20s_3_30	54	1328.04	5.401E+06	691.02	77.77	80.41	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
20m_3_30	995	2031.35	9.950E+07	1513.18	100.00	60.03	3600.00	15	740.32	1.500E+06	0.04	3600.00	64	676.63	370.31	94.94	3600.00
20l_3_30	648	-	-	-	100.00	-	3600.00	0	277.50	2.775E+02	47.91	3600.00	37	417.42	227.21	96.77	3600.00
20s_3_50	168	1434.26	1.680E+07	799.13	58.53	73.01	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
20m_3_50	99	2203.80	9.902E+06	1151.40	82.82	87.74	3600.00	-	-	-	-	3600.00	181	366.17	273.58	88.77	3600.00
20l_3_50	1344	1852.22	1.344E+08	1598.11	99.62	49.66	3600.00	45	1123.48	4.501E+06	66.67	3600.00	49	578.48	313.74	90.35	3600.00
20s_3_70	423	1552.35	4.230E+07	987.67	14.42	59.01	3600.00	424	2902.26	4.240E+07	14.62	3600.00	528	390.82	459.41	55.85	3600.00
20m_3_70	168	1491.69	1.680E+07	829.84	0.00	76.80	3600.00	168	1407.24	1.680E+07	11.31	3600.00	-	-	-	-	3600.00
20l_3_70	274	2989.96	2.740E+07	1631.98	44.16	81.51	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
20s_5_30	923	939.96	9.230E+07	931.48	99.89	42.39	3600.00	68	1630.26	6.801E+06	95.45	3600.00	-	-	-	-	3600.00
20m_5_30	886	-	-	-	100.00	-	3600.00	43	2017.09	4.302E+06	98.66	3600.00	-	-	-	-	3600.00
20s_5_50	1527	1051.42	1.527E+08	1289.21	99.21	34.04	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
20m_5_50	1450	-	-	-	99.72	-	3600.00	87	2062.80	8.702E+06	88.24	3600.00	-	-	-	-	3600.00
20s_5_70	330	1499.62	3.300E+07	914.81	59.09	70.46	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
20m_5_70	249	1435.31	2.490E+07	842.15	83.93	76.39	3600.00	385	1738.69	3.850E+07	78.95	3600.00	-	-	-	-	3600.00
20l_5_70	129	-	-	-	82.17	-	3600.00	-	-	-	-	3600.00	158	812.23	485.11	89.34	3600.00
30s_1_30	0	3495.86	3.496E+03	1747.93	0.00	0.00	2214.11	0	668.80	6.688E+02	91.89	3600.00	-	-	-	-	3600.00
30m_1_30	0	4501.77	4.502E+03	2250.88	0.00	0.00	2554.77	0	921.33	9.213E+02	90.25	3600.00	-	-	-	-	3600.00
30l_1_30	0	4911.02	4.911E+03	2455.51	0.00	94.73	3600.00	0	634.49	6.344E+02	80.19	3600.00	-	-	-	-	3600.00
30s_1_50	98	3299.06	9.803E+06	1698.53	0.00	0.00	3401.45	98	1303.66	9.801E+06	11.74	3600.00	210	729.63	469.81	85.70	3600.00
30m_1_50	19	3742.17	1.904E+06	1880.58	100.00	95.28	3600.00	19	2616.36	1.902E+06	99.99	3600.00	75	1880.72	977.86	97.79	3600.00
30l_1_50	0	4683.93	4.684E+03	2737.80	0.00	0.00	2737.80	0	724.51	7.245E+02	94.12	3600.00	-	-	-	-	3600.00
30s_1_70	934	2416.98	9.340E+07	1675.49	10.38	66.61	3600.00	937	2667.16	9.370E+07	11.20	3600.00	-	-	-	-	3600.00
30m_1_70	544	5919.16	5.441E+07	3231.58	8.63	87.56	3600.00	554	3078.50	5.540E+07	10.29	3600.00	631	2046.16	1338.58	78.02	3600.00
30l_1_70	249	3310.89	2.490E+07	1779.94	0.00	0.00	2641.01	-	-	-	-	3600.00	268	1623.10	945.55	85.39	3600.00

(continued on next page)

Detailed computational results obtained by mathematical formulations (continued)

Ins.	$\mathscr{F}_{LA}+\mathscr{F}_{CS}$							$\mathscr{F}_{AWS}$ ($\alpha >> \beta$)					$\mathscr{F}_{AWS}$ ($\alpha = \beta$)				
	E_{LA}	E_{CS}	$E_\tau(\alpha >> \beta)$	$E_\tau(\alpha = \beta)$	$gap_{la}\%$	$gap_{cs}\%$	cpu	E_{LA}	E_{CS}	E_τ	$gap\%$	cpu	E_{LA}	E_{CS}	E_τ	$gap\%$	cpu
30s_3_30	1585	2858.33	1.585E+08	2221.66	99.74	59.80	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
30s_3_50	330	3643.15	3.300E+07	1986.57	70.00	84.45	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
30s_3_70	821	3058.59	8.210E+07	1939.79	18.27	69.32	3600.00	840	3613.93	8.400E+07	19.35	3600.00	-	-	-	-	3600.00
30m_3_70	371	3394.37	3.710E+07	1882.68	39.35	84.73	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
30l_3_70	2589	-	-	-	97.29	-	3600.00	-	-	-	-	3600.00	684	9339.08	5019.04	98.06	3600.00
30s_5_70	756	4492.94	7.560E+07	2624.47	85.58	80.18	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
50s_1_30	0	16686.92	1.669E+04	8343.46	0.00	0.00	2892.64	-	-	-	-	3600.00	-	-	-	-	3600.00
50m_1_30	0	17645.35	1.765E+04	8822.67	0.00	97.98	3600.00	0	8264.20	8.264E+03	98.38	3600.00	-	-	-	-	3600.00
50s_1_50	401	12290.55	4.011E+07	6345.77	34.41	93.74	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
50m_1_50	250	13869.28	2.501E+07	7059.64	84.40	95.04	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
50l_1_50	11	15907.70	1.116E+06	7959.35	90.90	97.80	3600.00	-	-	-	-	3600.00	146	9867.24	5006.62	98.98	3600.00
50s_1_70	1743	11544.01	1.743E+08	6643.50	16.40	83.19	3600.00	1772	6898.15	1.772E+08	17.77	3600.00	2370	4006.59	3188.29	75.01	3600.00
50m_1_70	1017	19637.64	1.017E+08	10327.32	14.06	92.16	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
50l_1_70	599	20988.08	5.992E+07	10793.54	12.18	95.38	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
50s_3_70	3904	12233.50	3.904E+08	8068.75	78.32	72.48	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
70s_1_30	67	19600.96	6.720E+06	9833.98	100.00	96.20	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
70s_1_50	1436	20060.26	1.436E+08	10748.13	48.11	90.22	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
70s_1_70	3225	16797.21	3.225E+08	10011.10	18.63	80.97	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
70m_1_70	2023	29211.91	2.023E+08	15617.45	19.37	90.92	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00
70l_1_70	1332	41511.74	1.332E+08	21421.87	20.79	95.56	3600.00	-	-	-	-	3600.00	-	-	-	-	3600.00

Chapter 4

Alternative e-commerce delivery policies
A case study concerning the effects on carbon emissions

Contemporary shopping habits are undergoing rapid change, with more and more consumers purchasing goods online. The rapid growth of the online retail sector provides great opportunities for both wholesalers and transporters in servicing this newly-emergent type of customer. With both consumers and corporations acutely aware of the environmental impact of business activities, one of the most relevant research questions is how to organize the operations of an e-commerce delivery business while simultaneously minimizing its environmental impact.

The present paper addresses the *e-commerce delivery problem*, a mathematical formulation and fast heuristics which enable the simulation of various e-commerce delivery scenarios. The effects of the scenarios regarding more environmentally friendly e-commerce concerns are tested upon real-world data. In particular, the impact of new green(er) technology (such as electric bicycles and cars), aggregated collection points, carrier bundling and changing delivery times is investigated.

The obtained results are suitable for implementation at an organizational or operational level within both e-commerce delivery companies and transporters.

4.1. Introduction

There is an ongoing shift from the traditional, physical shopping environment towards online shopping. With annual online purchasing growth estimated at roughly 10% within Europe, it is not difficult to imagine a near future where online shopping overtakes in-store sales and becomes the new normal. Wholesalers promise the fast delivery of products to customers, however the logistical process itself is organized by road couriers and transporters. While offering a high level

of comfort to individual customers, e-commerce delivery practices represent a considerable source of greenhouse emissions. Additionally, urban areas are particularly vulnerable to complicating factors such as traffic congestion due to their high density of both residential and commercial destinations. The rapid growth of online shopping, or e-commerce, correlates with important environmental and societal challenges imposed by governments and environmentally-conscious consumers, namely: how one may operationally organize e-commerce deliveries such that their environmental impact is minimal; and how to simultaneously satisfy customer demands while minimizing transportation costs. Indeed, online wholesalers and transporters are eager to discover viable answers to these questions.

Many businesses have adopted a consumer direct (CD) model, which permits customers to purchase goods online and have them delivered directly to their addresses. Duin et al. (2007) provide an auctioning model which assigns orders to providers in a cost and punctuality-driven environment. Orders arrive dynamically, requiring flexibility concerning their insertion into routes. Campbell and Savelsbergh (2006) detailed various incentive policies, encouraging companies to pursue home deliveries. They presented the home delivery problem with time slot incentives, where customers select a delivery time slot based upon the associated monetary benefits being offered by the company. However, the environmental impact of such policies was not studied.

Bektaş and Laporte (2011) introduced an extension of the vehicle routing problem with time windows (VRPTW), the pollution routing problem (PRP), the objective of which is to minimize a cost function composed of emission costs, fuel costs and driver costs. The study concluded that the cost of working hours dominates the emission cost. Demir et al. (2012) also studied the PRP and proposed an ALNS algorithm with a specialized speed optimization component capable of computing optimal speeds on a given path. Çağrı Koç et al. (2014) extended the PRP by considering a heterogeneous fleet which only contained internal combustion commercial vehicles (ICCVs). They conducted experiments on instances from the literature and discovered that using a heterogeneous fleet without speed optimization reduces operational costs by a greater degree than when employing a homogeneous fleet with speed optimization.

Employing a homogeneous fleet of electrical vehicles was considered by Schneider et al. (2014) during their E-VRPTW study (the addition of 'E' signifying 'Electrical'). The objective of the E-VRPTW is to minimize the distances traveled by electric commercial vehicles (ECVs). Schneider et al. (2014) presented a mixed integer programming (MIP) model and metaheuristics to solve their generated instances. Goeke and Schneider (2015) studied E-VRPTWMF, an extension of E-VRPTW where two vehicle types were considered as a mixed-fleet (MF): ECVs and conventional ICCVs. This study proposed a more realistic energy consumption model for ECVs based on individual vehicle mass, speed and the terrain's gradient. Three different minimization objectives were considered: distance traveled; the sum of vehicle propulsion and labor cost; and battery replacement cost. All experiments were conducted upon randomly generated instances. Desaulniers et al. (2016) study the E-VRPTW with time windows, given an unlimited fleet of identical ECVs. They particularly focus on the electric vehicles' limited autonomy and investigate variants where vehicles may recharge once, several times, partially or fully during the course of a

route.

In recent years, green logistics has received increasing interest due to growing environmental concerns by citizens and governments. Roberti and Wen (2016) address the Electric Traveling Salesman Problem with Time Windows (ET-SPTW) which considers the limited capacity of existing electric vehicle batteries, thereby acknowledging the need for intermediate stops at recharging stations. Heuristics were developed and tested on generated instances. El-Berishy and Scholz-Reiter (2016) considered a homogeneous fleet with a single depot VRP. They proposed a two stage stochastic model where the delivery speed and emissions are uncertain. The first stage generates optimum routes; the second stage minimizes emissions by regulated vehicles' speed. Koç and Karaoglan (2016) addressed the green-VRP by considering a homogeneous fleet, limited driving range and refueling infrastructure. They developed a branch and cut algorithm and tested it upon a benchmark set with twenty customers. Leggieri and Haouari (2017) solved the green-VRP by integer programming and outperform Koç and Karaoglan (2016)'s algorithm. Shao et al. (2016) focus on avoiding congestion while investigating distribution strategies. Ehmke et al. (2016) modeled expected emissions costs as a function of the time at which the vehicle begins traveling an arc and its load while traversing some arc. They did not factor in waiting times. Huang et al. (2017) considered path selection in time-dependent vehicle routing problems (TDVRP-PF) where any arc between two customer nodes represents multiple paths. A homogeneous fleet of vehicles was considered to minimize fuel consumption and vehicle depreciation cost. Huang et al. (2017) modelled the TDVRP–PF both under deterministic and stochastic traffic conditions and generated instances based on Beijing's road network. Muñoz-Villamizar et al. (2017) investigated, similar to the present paper, both the delivery cost and environmental impact of employing a homogeneous ECV fleet in an urban environment. They modeled a multi-depot vehicle routing problem (MD-VRP) and experimented with real-world data obtained from a transportation network in Bogotá, Colombia.

Several successful approaches within the VRP literature proposed various types of population based evolutionary algorithms to address different variants of the problem. Nagata et al. (2010) and Koç et al. (2015) presented a hybrid genetic algorithm for the VRP with time windows. Tasan and Gen (2012) employed a genetic algorithm for the VRP with simultaneous pick-up and deliveries. Meanwhile, Vidal et al. (2012) developed an evolutionary algorithm which combines population-based evolutionary search and neighborhood-based metaheuristics to address three VRPs: multi-depot, periodic and the multi-depot periodic with capacitated vehicles. Vidal et al. (2013), presented a hybrid genetic search with diversity control for a large class of time-constrained vehicle routing problems. All referenced studies conducted extensive computational experiments on academic benchmark instances. The experimental results showed that the proposed algorithms achieved high quality solutions which encourage the usage of population-based evolutionary algorithms to address variants of the VRP.

Furthermore, there exist several survey papers within the field which highlight various aspects of green logistics and the incorporation of environmental issues into combinatorial optimization problems. Such papers most commonly focus on the vehicle routing problem (Dekker et al., 2012;

Demir et al., 2014; Lin et al., 2014).

Despite the valuable contributions made by the aforementioned studies, finding the most profitable mode of operation while simultaneously limiting greenhouse emissions on realistic problem instances remains unexplored. This subject may be addressed using three fundamental research questions: *i*) How may one organize delivery from distribution center to the customer in an ecological manner? *ii*) Which operational shifts may reduce emission levels without incurring unacceptable costs? *iii*) Which concessions related to delivery time windows or deviation from the delivery location are acceptable for environmentally-conscious consumers?

The present paper investigates the e-commerce delivery problem (EDP) from an operational perspective. The EDP generalizes the vehicle routing problem, wherein orders must be delivered to customers. Its objective is to compose routes beginning and ending at the depot and visiting each delivery location while simultaneously minimizing operational costs and emissions. In addition to traditional vehicle routing objectives and constraints, the EDP requires the selection of vehicle types (electrical or otherwise), the determination of delivery dates and/or the merging of delivery points.

An integer programming formulation which incorporates vehicle-dependent transportation costs and greenhouse emissions, driving times and delivery times is introduced in Section 4.2. By incorporating such factors, the impact of alternative delivery strategies - such as the introduction of a heterogeneous fleet including electric vehicles, relaxing delivery time windows, or aggregating delivery destinations - may be accurately investigated. New delivery strategies are subsequently presented during Section 4.3 and applied to real-world data provided by e-commerce delivery transporters. The availability of real-world data provides a unique opportunity to explore the effects of the proposed policies. A set of fast heuristics enabling the simulation of various e-commerce delivery scenarios is presented in Section 4.4. The performance of the proposed algorithms are analyzed followed by extensive simulations which enable the assessment of both the environmental and financial impact of such strategies. Experimental results (Section 4.5) reveal the interesting and often counter-intuitive effects of adjusting operations. The lack of research in mixed-fleet EDP means these results will likely have an immediate impact upon online businesses and will potentially contribute to more sustainable e-commerce delivery practices.

4.2. Problem statement

The EDP consists of three major components: depots, vehicles and parcels. A depot is a location from which parcels must be delivered to their associated destinations. Each depot has a set of vehicle types P where each individual vehicle type p is associated with the following parameters: *i*) number of available vehicles, m_p, *ii*) maximum traveling duration, T_p, *iii*) energy capacity, R_p, detailing the fuel, battery, or energy capacity of the vehicle type.

A parcel has three properties: a delivery window, origin depot and destination location. The delivery window consists of the earliest and latest possible delivery dates. The overall problem may be decomposed into a separate subproblem for each depot. Parcels with the same origin depot

and destination location may be bundled into one *multi-drop parcel provided their time windows are identical*. Each sub-problem is thus represented by an ordered list of multi-drop parcels, a depot and a mixed vehicle fleet of fixed size which delivers the parcels.

The EDP is often defined as an open vehicle routing problem (Li et al., 2007). By contrast, this paper assumes the transporter owns the vehicles employed for delivery - be they bicycles, cars, vans or trucks. Drivers are required to return their vehicle to the depot after concluding their deliveries, a requirement contributing to the driver's total working time.

The EDP is formulated as a mixed integer program ($\mathscr{F}_{EDP}$) inspired by the research of Goeke and Schneider (2015) for E-VRPTWMF where two vehicles types were considered with the possibility of recharging at certain stations. While $\mathscr{F}_{EDP}$ incorporates multiple vehicle types, it ignores both vehicle recharging and time windows. The EDP considers a single autonomous trip per vehicle. Additionally, assuming high level decision-making, the time windows are as long as the scheduling horizon. Consequently, time windows do not impact upon a manager's decision. Time windows for e-commerce deliveries are generally one week long, while in the EDP the decision horizon is one day. Therefore, the $\mathscr{F}_{EDP}$ formulation does not explicitly take time windows into consideration but rather aggregates all deliveries to the same destination into a single multi-drop parcel. Whereas regular single-drop deliveries take t_d time, multi-drop deliveries take t_b time for each additional parcel. Furthermore, vehicle capacity is considered sufficiently large to ignore capacity constraints.

The EDP is defined on a complete directed graph, $G = (V,A)$ where vertices 0 and $N+1$ correspond to the depot, while $V_{0,N+1} = \{v_1, ..., v_N\}$ represents the set of delivery points. The model proposed by Goeke and Schneider (2015) considers two vehicle types: ECVs and ICCVs. Given EDPs require the possibility of choosing between various vehicle types (electric vehicles, bikes, vans, trucks), a decision variable x_{ij}^p indicates whether or not arc ij is traveled by a vehicle of type p. Given the generally light weight parcels in e-commerce, the cost of traveling an arc is assumed independent of the vehicle's load.

The proposed model aggregates all parcels requiring delivery to the same address during preprocessing into a single delivery. The demand of vertex i, q_i, represents the number of parcels requiring delivery to vertex i. Service time at vertex $i \in V_{0,N+1}$, s_i, may be obtained by Equation (4.1), where t_d and t_b represent single and multi drop time, respectively.

$$s_i = t_d + (q_i - 1)\, t_b \tag{4.1}$$

Table 4.2.1 summarizes the notation employed throughout the paper.

Table 4.2.1: Notations for the EDP formulations

$0, N+1$:	Depot vertices;
V :	Set of all vertices;
A :	Set of arcs = $\{(i,j) \mid i,j \in V, i \neq j\}$;
V_0 :	Set of vertices excluding 0, $V_0 = \{v_1, ..., v_{N+1}\}$;
V_{N+1} :	Set of vertices excluding N+1, $V_{N+1} = \{v_0, ..., v_N\}$;
$V_{0,N+1}$:	Set of delivery vertices, $V_{0,N+1} = \{v_1, ..., v_N\}$;
P :	Set of vehicle types;
M :	A big number;
d_{ij} :	Distance between vertices i and j;
t_d :	Single drop time;
t_b :	Multi drop time;
t_{ij}^p :	Travel time between vertices i and j by vehicle type $p \in P$;
m_p :	Number of available vehicles of type $p \in P$;
R_p :	Fuel, battery or energy capacity of vehicle type $p \in P$;
T_p :	Maximum tour duration of vehicle type $p \in P$;
g_{ij}^p :	Fuel, battery or energy consumption of vehicle type $p \in P$ between vertices i and j;
h^p :	Fuel, battery or energy consumption per time unit while vehicle type $p \in P$ is idle;
q_i :	Demand of vertex i;
s_i :	Service time at vertex i ($s_0, s_{N+1} = 0$);
r_i^p :	Auxiliary variable indicating energy level of vehicle type p at vertex i;
τ_i :	Auxiliary variable for arrival time at vertex i;
x_{ij}^p :	Binary decision variable indicating if arc $(i,j) \in A$ is traveled by vehicle type p.

$$\sum_{j \in V_0} \sum_{p \in P} x_{ij}^p = 1 \qquad \forall\, i \in V_{0,N+1} \tag{4.2}$$

$$\sum_{j \in V_{N+1}} x_{ij}^p - \sum_{j \in V_0} x_{ji}^p = 0 \qquad \forall\, i \in V,\ p \in P \tag{4.3}$$

$$\sum_{j \in V_{0,N+1}} x_{0j}^p \leq m_p \qquad \forall\, p \in P \tag{4.4}$$

$$\tau_i + \sum_{p \in P} (s_i + t_{ij}^p)\, x_{ij}^p - M\,(1 - \sum_{p \in P} x_{ij}^p) \leq \tau_j \qquad \forall\, i \in V_{N+1},\ j \in V_0 \tag{4.5}$$

$$\tau_i + s_i + t_{i,N+1}^p\, x_{i,N+1}^p \leq T_p \qquad \forall\, p \in P,\ i \in V_0 \tag{4.6}$$

$$r_i^p - h^p.s_i.x_{ij}^p - g_{ij}^p.x_{ij}^p + R_p\,(1 - x_{ij}^p) \geq r_j^p \qquad \forall\, i \in V,\ j \in V_{N+1},\ p \in P \tag{4.7}$$

$$0 \leq r_i^p \leq R_p \qquad \forall\, i \in V_{0,N+1},\ p \in P \tag{4.8}$$

$$\tau_i \geq 0 \qquad \forall\, i \in V \tag{4.9}$$

$$x_{ij}^p \in \{0,1\} \qquad \forall\, (i,j) \in A,\ p \in P \tag{4.10}$$

Constraints (4.2) ensure each customer is followed by exactly one other customer in the route, except for the depot vertices. Constraints (4.3) are flow conservation constraints guaranteeing an equal number of incoming and outgoing arcs for each vertex. Constraints (4.4) express an upper limit for the number of employed vehicles of each type. Auxiliary variable τ_i represents

the vehicle's arrival time at vertex i and Constraints (4.5) link the arrival times at vertices i and j. Constraints (4.6) prevent each tour's total time from exceeding the maximum traveling duration of its vehicle type. Each tour's total time is obtained by summing the arrival time to the last vertex (l), τ_l, the service time of l, s_l and the travel time from l to the depot, t^p_{lN+1}. Auxiliary variable r^p_i represents the energy level of vehicle type p at vertex i. Constraints (4.7) set the energy level of vehicle type p at vertex j considering the energy level at its previous vertex i. Constraints (4.8) restrict the fuel, battery, or energy level of vehicle type p to be between zero and the maximum vehicle capacity at each vertex i. Constraints (4.9) state variables x^p_{ij} as binary.

The following Key Performance Indicators (KPIs) are considered during this study:

- CO_2 emissions associated with a parcel's delivery.
- Delivery cost.
- Total distance traveled by all vehicles.
- Number of vehicles required to satisfy all deliveries.

Operators wish to minimize the linear combination of these KPIs which results in the formation of the objective function, denoted as F in Equation (4.11). The term f_e within this equation denotes the total emission cost for parcel delivery, f_l the routing cost of delivery, f_d the total distance traveled by all vehicles and f_v the total number of vehicles required to satisfy all deliveries. f_l, f_d and f_v denote internal KPIs whereas f_e is called an external KPI. α, β, γ and δ are positive coefficients weighting the objectives.

$$F = \alpha f_e + \beta f_l + \gamma f_d + \delta f_v, \tag{4.11}$$

CO_2 emissions are derived by summing both a vehicle's travel and stationary (when ran idle) fuel consumption and then multiplying this total by the amount of CO_2 per fuel unit, c^p_e (the fuel unit may correspond to liter for diesel/gasoline, kilogram for CNG and kWh for electricity). By always considering the average consumption of vehicles when they are either traveling or stationary, their actual load at every stage of the route route may be ignored. g^p_{ij} indicates the fuel consumption of vehicle type p between vertices i and j, whereas h^p denotes vehicle type p's fuel consumption per time unit while it is idle at service points.

$$f_e = \sum_{i \in V_0} \sum_{j \in V_{N+1}} \sum_{p \in P} c^p_e \,.\, (g^p_{ij} + h^p s_i) \,.\, x^p_{ij} \tag{4.12}$$

The delivery cost consists of labor costs and vehicle costs which include vehicle write off, fuel consumption, insurance and maintenance. Parameters c_d and c^p_v denote driver wages per time unit

and vehicle costs per kilometer for vehicle type p, respectively.

$$f_l = \sum_{i \in V_{N+1}} \sum_{p \in P} c_d \left(\tau_i + s_i + t^p_{i,N+1}\right) x^p_{i,N+1} + \sum_{i \in V_{N+1}} \sum_{j \in V_0} \sum_{p \in P} c^p_v \,.\, d_{ij} \,.\, x^p_{ij} \tag{4.13}$$

The total distance is the sum of the length of all routes. The distance between vertices i and j is denoted by d_{ij}.

$$f_d = \sum_{i \in V_{N+1}} \sum_{j \in V_0} \sum_{p \in P} d_{ij} \,.\, x^p_{ij} \tag{4.14}$$

The number of vehicles required:

$$f_v = \sum_{p \in P} \sum_{j \in V_{0,N+1}} x^p_{0j} \tag{4.15}$$

A VRP instance may be considered an instance of the EDP with only one vehicle type and hence be solved by any EDP algorithm. Thus, the EDP is at least as hard as the VRP which is proven $\mathscr{NP}$-hard (Lenstra and Kan, 1981). Therefore, exact approaches to the EDP, based on a commercial integer programming solver, are unlikely to be applicable in real-life e-commerce delivery routing due to their size, both from a financial and performance perspective.

4.3. E-commerce delivery policies

4.3.1 Alternative parcel delivery vehicles

The vast majority of current e-commerce parcels are delivered by diesel vans (Dekker et al., 2012). While these vans have a low internal cost - they are cheap to buy, run and maintain - their external cost is much greater in urban environments. Given the inefficiency of diesel engines for short trips and slow start-stop traffic, CO_2 and NO_x emissions are considerable during city tours. Furthermore, van engines are often left idle (rather than being shut off) during delivery, exacerbating emission levels.

Despite alternative delivery technologies being widely available, the potential decrease in external cost is often outweighed by the perceived increase in internal cost. Higher purchase prices and minimal difference in CO_2 emissions often discourage companies to make the transition from low-cost diesel vans to Compressed Natural Gas (CNG) vehicles. Similarly, the high purchase price and smaller capacity of electric vans are considered economic deal-breakers for companies considering the use of environmentally-friendly vehicles. In (sub)urban environments, however, smaller vehicles are rarely an issue when the depot is located close to (or within) the city and only minor changes must be made to the delivery company's operations.

The impact of switching from diesel to CNG or electric vehicles is analyzed in greater detail during Section 4.5.2.1.

4.3.2 Collection points

Customers making purchases via e-commerce channels may have their goods delivered directly to their home, to a chosen collection point, or they may even choose to retrieve goods themselves at the company's physical store. When a customer's order exceeds a certain value, delivery fees are often waived. While it is cheaper for both e-commerce stores and carriers to deliver goods to a collection point or have them picked-up directly from the store, most online stores select home delivery by default - rarely offering any financial motivation for customers to alter their delivery method. Including more options or steps to the checkout process of online shopping significantly decreases the conversion rate (item views to sales), which is the primary reason for not offering alternative delivery options. An important downside associated with home delivery is that most people are away from home during daytime hours, which is precisely when carriers also work. This results in high delivery failure rates whereby couriers must attempt to deliver parcels two or more times before the customer is at home to accept delivery. Not only is this very expensive for the carrier, it is also very inconvenient for the customer since goods are delivered far later than anticipated.

It is highly plausible that, given some small financial incentive, many customers would choose to have their goods delivered to a collection point rather than at home. Customers are consequently able to collect their purchases at the time most convenient for them while simultaneously being financially rewarded for this decision. From the web shop and carrier's perspectives, this decision is also beneficial since they are able to ship and deliver parcels at a lower cost and no longer run the risk and cost of having multiple failed deliveries.

The impact of relocating a portion of all parcels to collection points on both internal and external costs is discussed in Section 4.5.2.2.

4.3.3 Carrier bundling and regional monopolies

In practice, different carriers often deliver parcels to the same street on the same day. Instinctively, one would perhaps consider this situation inefficient and conclude that it would be better to offer a monopoly to a single carrier on such streets in urban centers.

Rural regions, on the other hand, often have so few parcels to be delivered that it becomes very expensive for carriers to deploy or maintain activities in such areas. In this situation, regional monopolies could likely increase the density of parcels for a single carrier and thereby engender a profit margin as opposed to several different carriers generating losses in the region. This higher parcel density also translates into decreased emissions per parcel, thus decreasing the external cost.

The effects of enforcing regional monopolies is investigated within several scenarios in Section 4.5.2.3.

4.3.4 Delivery times

The vast majority of e-commerce parcels are delivered directly to the customer's door. While the time spent traveling from the van to the customer's door is rather small, the time spent waiting at the customer's door after ringing the door bell easily increases average parcel delivery times to 3-4 minutes. This waiting time represents a significant proportion of the total delivery cost. Furthermore, many e-commerce products are packaged in boxes much larger than the product itself with the result being that carriers are mostly storing and delivering empty space to customers throughout the country. There are two logical ways of reducing long waiting times for carriers:

i. Transitioning from home delivery to collection points (see Section 4.3.2), or

ii. Reducing the amount of empty space in e-commerce parcels, thus enabling more parcels to be delivered via mailboxes.

The impact of reducing the time required for delivering parcels is investigated in Section 4.5.2.4.

4.4. Algorithms for policy simulation

Efficient algorithms are required when solving EDPs in practice. Therefore, designing easy-to-implement fast heuristics within a simulation environment is essential.

Two constructive algorithms which generate initial solutions for the EDP are presented alongside a ruin and recreate (R&R) local search which is to be employed in combination with these constructive algorithms. Given the academic merit of genetic algorithms for VRP problems, one is proposed based on the biased random key genetic algorithm (BRKGA). Descriptions of the solution encoding and decoding heuristics, evolutionary process and fitness function are provided.

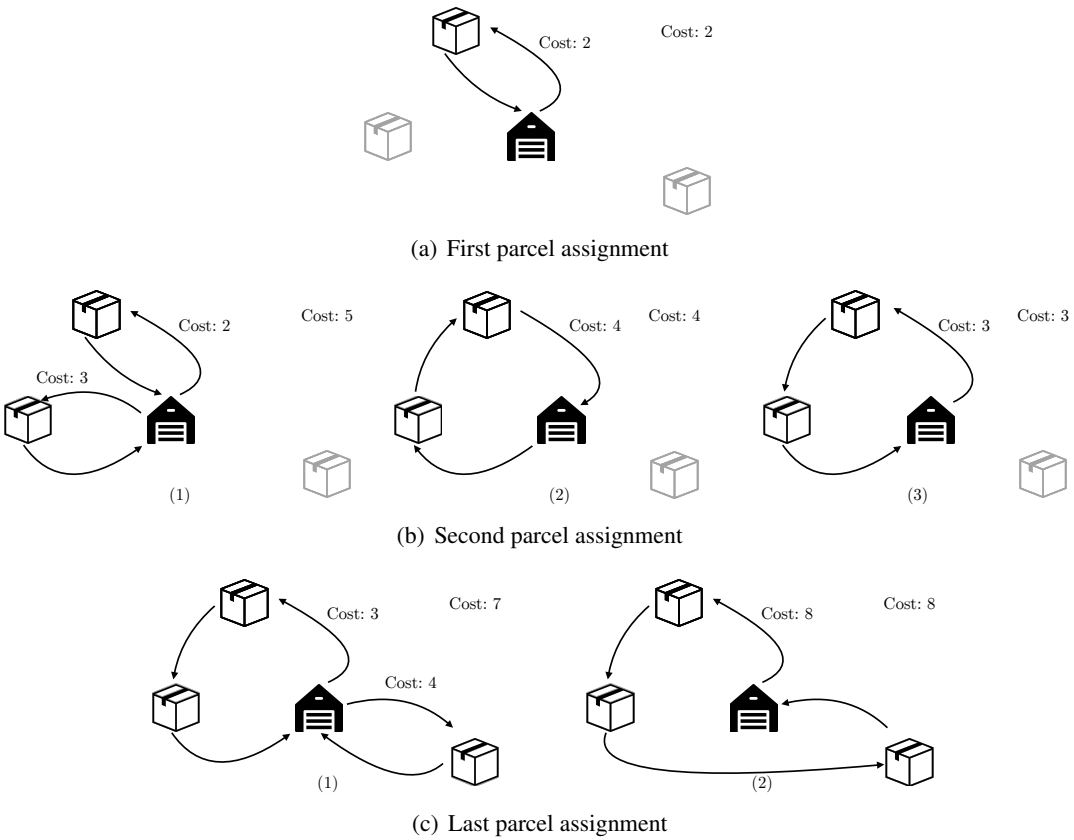
4.4.1 Constructive heuristics

A Cheapest Insertion (CI) heuristic is applied to generate an initial EDP solution.

The heuristic creates a feasible solution for each depot by iterating over its parcel list and assigning each parcel to a delivery route. The cost of inserting each parcel at all possible positions in existing routes and of creating a new route is computed. The latter cost is obtained by inserting an available vehicle with lowest possible emissions. Subsequently each parcel is inserted at the lowest cost position in the route.

Once all parcels have been included in this schedule, the CI heuristic is complete and a feasible solution for the EDP is available.

Figure 4.4.1 visualizes the cheapest insertion procedure. The first parcel in the depot's list is assigned to a new route in Figure 4.1(a). Figure 4.1(b) illustrates how the second parcel is assigned. The cost of creating a new route for the second parcel alone (Cost 3) is compared against the cost increase when inserting it at all possible positions in the existing route. Thus the lowest cost option (b3) is selected. The last parcel's assignment is depicted in Figure 4.1(c) where,



(a) First parcel assignment

(b) Second parcel assignment

(c) Last parcel assignment

Figure 4.4.1: CI constructive heuristic example for three parcels and one depot.

similarly, the cost increase of creating a new route (Cost 4) is compared against the best possible insertion in the existing route. Note that only two possibilities are represented in the figure. The solution constructed in this example thus employs two vehicles and its cost amounts to 7.

The primary advantage of this heuristic is its capability of quickly generating solutions for even the largest instances. One noteworthy disadvantage is, however, that the algorithm produces solutions located in local optima which may prove very difficult to escape from. The combination of standard diesel vans and electric vehicles proved challenging for the heuristic since electric vehicles are slightly cheaper to operate and have much lower emissions than diesel vans, whereas their range is very limited. This results in solutions where the first few parcels are loaded into the available electric vehicles, while all those remaining are loaded into diesel vans. Such solutions are far from efficient and therefore an alternative algorithm is proposed.

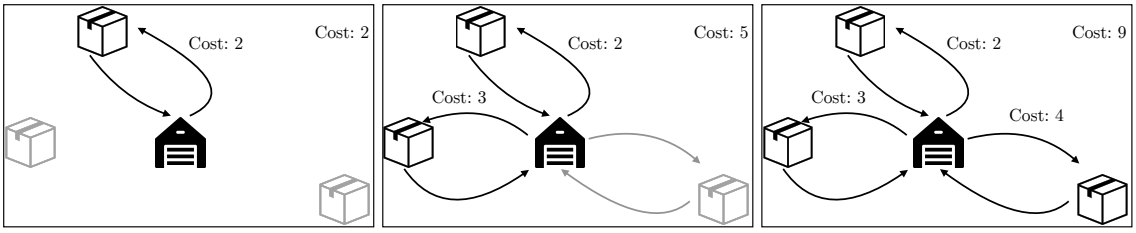


Figure 4.4.2: One Route Per Parcel constructive heuristic example.

The second constructive algorithm is the One Route Per Parcel heuristic (1RPP). This heuristic creates a solution by simply assigning each parcel to an individual delivery route. Figure 4.4.2 depicts 1RPP and clearly, although theoretically possible, its solutions incur extremely high costs. The advantage of this constructive heuristic, however, is that its resulting solution is not biased in any way and thus enables local search algorithms to efficiently alter the solution and find good quality solutions without prematurely converging to local optima.

4.4.2 Ruin and recreate local search

A ruin and recreate local search heuristic (Schrimpf et al., 2000; Pisinger and Ropke, 2007) is employed in combination with the aforementioned constructive heuristics. The R&R heuristic iteratively increases solution quality by randomly selecting a route and removing it from the solution. The removed route's parcels are re-added to the solution via CI, possibly after ordering these parcels in some way (such as decreasing distance from the depot) (Christiaens and Vanden Berghe, 2016). When a new solution is associated with a lower cost than the current best solution it replaces the latter, otherwise a different route is selected to be ruined.

After a specified number of iterations without any cost reduction, the number of routes to ruin is increased. The R&R heuristic ends after a predefined number of non-improving iterations, either when the number of ruined routes reaches the total number of routes in the solution or when a global iteration limit is reached.

4.4.3 Genetic algorithm

An evolutionary algorithm, based upon the BRKGA, is proposed to identify the number of routes, types of vehicles and parcel delivery order. BRKGA represents a variant of random-key genetic algorithms in which the initial population is composed of random-key vectors, with each vector's key being a real number sampled uniformly from the interval [0, 1). More information concerning the evolutionary process of BRKGA is provided by Gonçalves and Resende (2011).

A solution consists of a number of routes, each of which is associated with a vehicle type and parcel delivery order. This approach employs an indirect solution representation since direct solutions would be complex to represent and manipulate during the evolutionary process. A chromosome encodes a solution as a vector of random keys. Each chromosome is composed of $n+1$ genes, where n equals the number of parcels requiring delivery.

$$Chromosome = \left(\underbrace{gene_0}_{\text{\#routes}}, \underbrace{gene_1, \ldots, gene_n}_{\text{Parcels}} \right).$$

The first gene ($gene_0$) identifies the number of routes per depot while the remaining genes are *parcel* genes, each corresponding to a single parcel. Parcel genes are employed to assign their associated parcel to a route. Chromosomes are employed by the decoder when building a CFO

solution. The number of routes required per depot is denoted as r and decoded via:

$$r = \left\lfloor \frac{1}{gene_0} \right\rfloor \tag{4.16}$$

The decoding, or mapping, of each chromosome's last n genes into r routes is achieved by first dividing the $[0,1)$ interval into r sections where each section constitutes a route. Parcel p_i is assigned to route r_j if the value of $gene_i$ lies within section j. The route index for parcels $i = 1,...,n$ is generated using the following expression:

$$\text{parcel's route index } i = \lfloor gene_i \times r \rfloor \tag{4.17}$$

Once the parcels are assigned to the routes, the sequence of parcel deliveries per route is obtained by the greedy nearest neighbor (NN) algorithm which selects the nearest undelivered parcel as the next delivery parcel. This quickly results in a sub-optimal route, followed by a local search employing the 2-opt move (Croes, 1958). Each chromosome's quality is measured by the fitness function described in Equation (4.11), which feeds back into the evolutionary process.

4.5. Experiments and discussion

The experimental section is subdivided into three parts. The first compares the proposed algorithms, the second analyzes several existing and alternative real-world data policies and the third highlights the insights gained via experimentation.

All experiments were performed on instances based on real-world data from e-commerce delivery carriers which details their activities for between two and six months and their regional size which ranges from a single (sub)urban environment to an area of over 30,000 km^2. The data contains over one million parcels, specifies each parcel's earliest and latest delivery date, the parcel's origin depot and finally its delivery address. Furthermore, real-world vehicle characteristics (cost per kilometer, fuel consumption, range, driver wages) for different vehicle types were also provided by the e-commerce delivery carriers, thereby enabling a highly accurate simulation of the EDP's KPIs. Due to confidentiality and privacy issues, publishing the data in its original form is prohibited. A selection of real-world data was anonymized and posted online[1], enabling a transparent comparison of the proposed algorithms and also encouraging other researchers to compete with the proposed solution methods. All the destination of parcels and collection points are randomized within a radius of approximately 7 meters and depot locations are randomized within a radius of approximately 1 kilometer. Vehicle characteristics were also randomized via a confidential conversion-factor. Three sets of instances were thus generated: *rural*, *suburban* and *urban*.

[1] https://benchmark.gent.cs.kuleuven.be/hdp/

Real-world data was applied for the alternative policies since it was deemed important to provide results on unaltered data from e-commerce delivery carriers. Consequently, only the disclosure of aggregated information is possible for these experiments. However, this information certainly suffices for the purpose of the experiments. In addition to the aggregated results, instance size and primary vehicle characteristics are also supplied for each simulation thereby providing further insight regarding each scenario.

4.5.1 Algorithm analysis

The CI and R&R algorithms were implemented in Java, while the genetic algorithm was implemented via the BRKGA library in C++ 11. All experiments were executed on an Intel® Xeon® CPU E5-2640 v3 @ 2.6GHz processor.

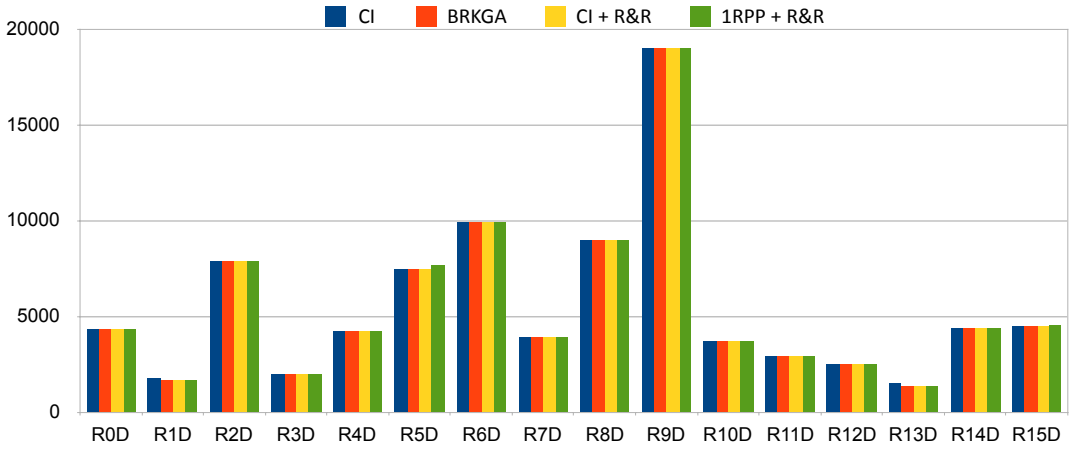
CI, BRKGA, CI + R&R and 1RPP + R&R were tested on three sets of instances: *rural* (R), *urban* (U) and *suburban* (S) instances. Experiments based on 1RPP without any improvement phase would certainly be uncompetitive and are therefore not considered. The 16 small instances ranging from 3 to 76 parcels correspond to the rural instances. The urban instances, meanwhile, correspond to five medium-sized instances ranging from 594 to 886 parcels. Finally, suburban instances are represented by five large instances ranging from 3743 to 5529 parcels.

Three fleet types are employed for rural instances: *diesel* (D), *CNG* (C) and *mixed* (M). Electric vehicles (E) are not employed given that some delivery points lie beyond their range. Unlike rural areas, delivery points in suburban and urban areas are close to the depot, thus enabling the utilization of electric vehicles in such regions.

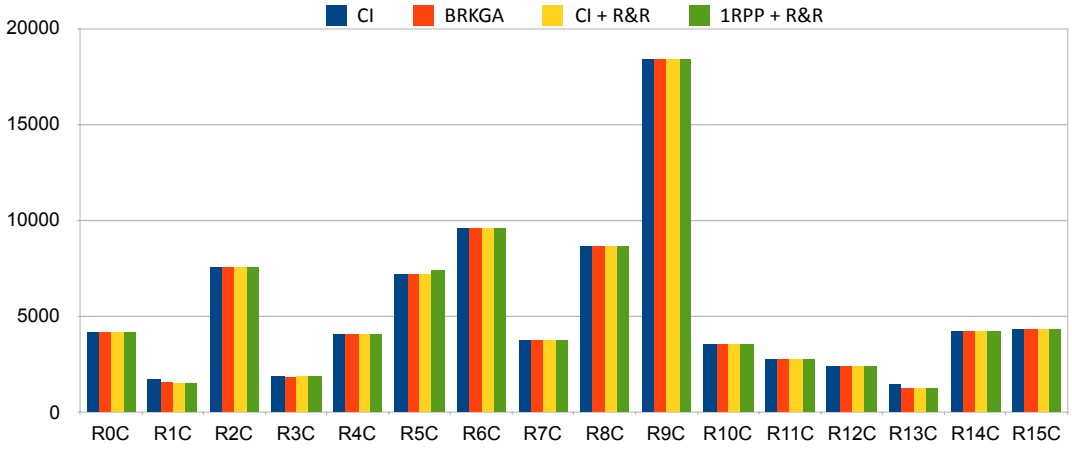
Equation (4.11) is employed as evaluation function in all experiments. For simplicity purposes α, β, γ and δ are set to 1.

Given all algorithms' stochastic behavior, all reported computational results presented in this paper are based on five runs per instance. Figure 4.5.1 compares the algorithms' results obtained for rural instances employing diesel, CNG and mixed vehicles, respectively.

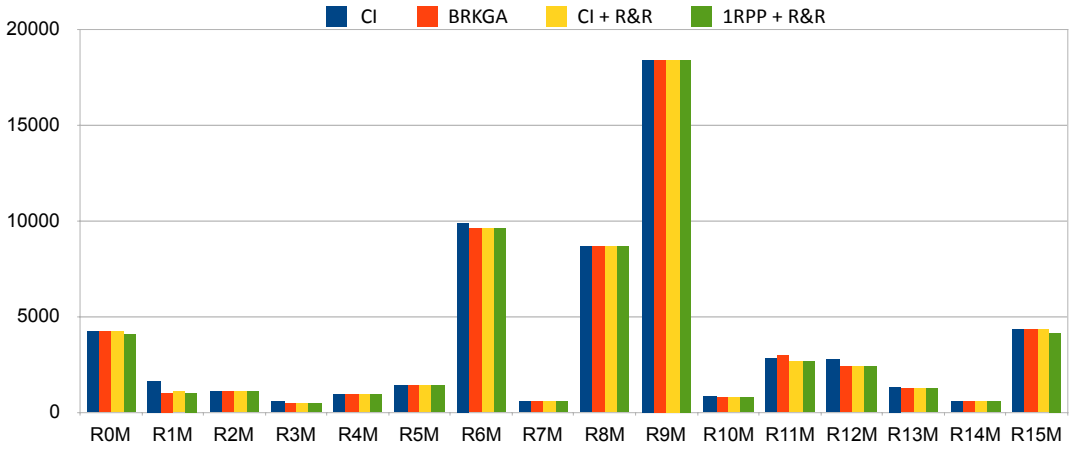
The horizontal axes represent the instances and the employed fleet (R0D, for example, represents rural instance number 0 addressed with diesel vehicles). Emissions, represented on the vertical axis, are the most relevant KPI for this particular study, while also being demonstrative of the trend among all other KPIs and therefore constitutes an accurate indicator of solution quality.



(a) Algorithmic performance on rural instances with diesel vehicles.



(b) Algorithmic performance on rural instances with CNG vehicles.



(c) Algorithmic performance on rural instances with mixed vehicles.

Figure 4.5.1: Algorithmic performance with respect to the emissions KPI on the rural (R) instances, assuming (a) diesel (D), (b) CNG (C) and (c) mixed (M) vehicles.

Table 4.5.1 compares the proposed algorithms in terms of number of best and worst solutions attained. The four algorithms yield solutions of identical quality for 24 out of the 48 instances. No significant performance difference between the four algorithms is noticeable with respect to the rural instances. BRKGA does however provide the most frequent occurrence of best solutions and least frequent occurrence of worst solutions with respect to this emissions KPI.

Table 4.5.1: Comparison of algorithm performance with respect to emissions - rural instances

	CI	BRKGA	CI + R&R	1RPP + R&R
Number of best solutions	4	11	9	10
Number of worst solutions	14	3	3	9

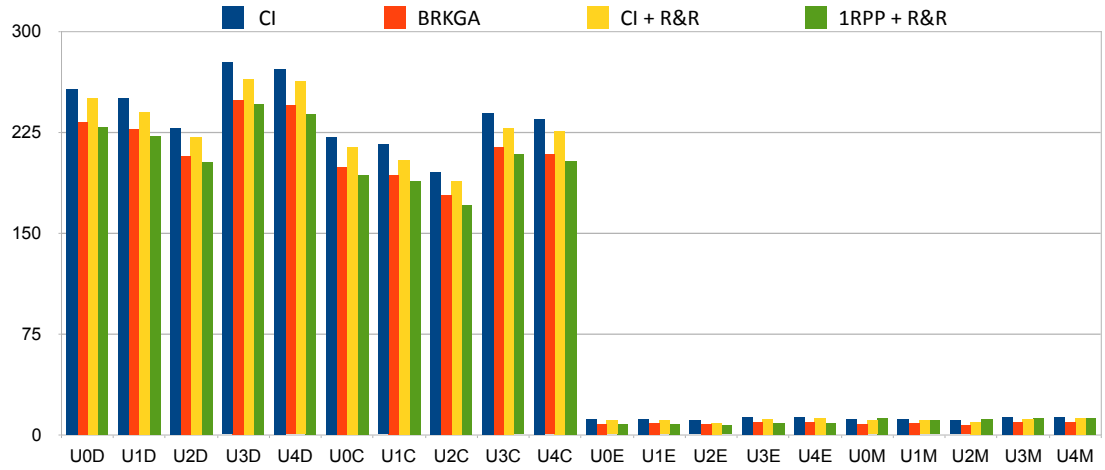


Figure 4.5.2: Algorithmic performance with respect to the emissions KPI on the urban instances.

Figure 4.5.2 illustrates the algorithmic performance on urban instances. As the number of parcels is higher for urban than for rural instances, the BRKGA algorithm no longer clearly outperforms the other algorithms and instead competes with 1RPP + R&R for the first place.

Figures 4.5.3 and 4.5.4 detail suburban instance results in terms of emissions and computational time, respectively.

Figure 4.5.3 is useful insofar as illustrating how the straightforward CI, as expected, exhibits the highest emission levels for almost all instances. BRKGA achieves better results, particularly for rural (academic-sized) instances. Finally, R&R achieves the lowest emission levels for all instances. Figure 4.5.4, meanwhile, presents computation time on a logarithmic scale, clearly illustrating how CI is (by far) the fastest-performing algorithm, with R&R taking longer and BRKGA coming in third.

In essence, while BRKGA performs best for academic-sized instances, it requires considerable computational time to reach high quality solutions for real-world instances. Due to this scaling issue, CI represents the only viable option when tackling larger real-world instances containing up to one million parcels. Indeed, while CI's emission levels are the highest among these three algorithms, its results are most comparable to those implied by historical data sourced from the companies.

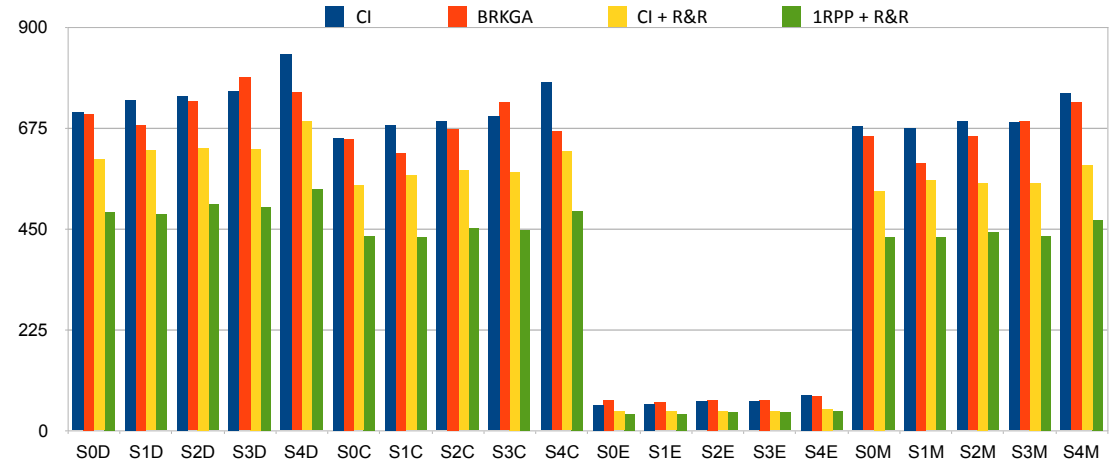


Figure 4.5.3: Algorithmic performance with respect to the emissions KPI on the suburban instances.

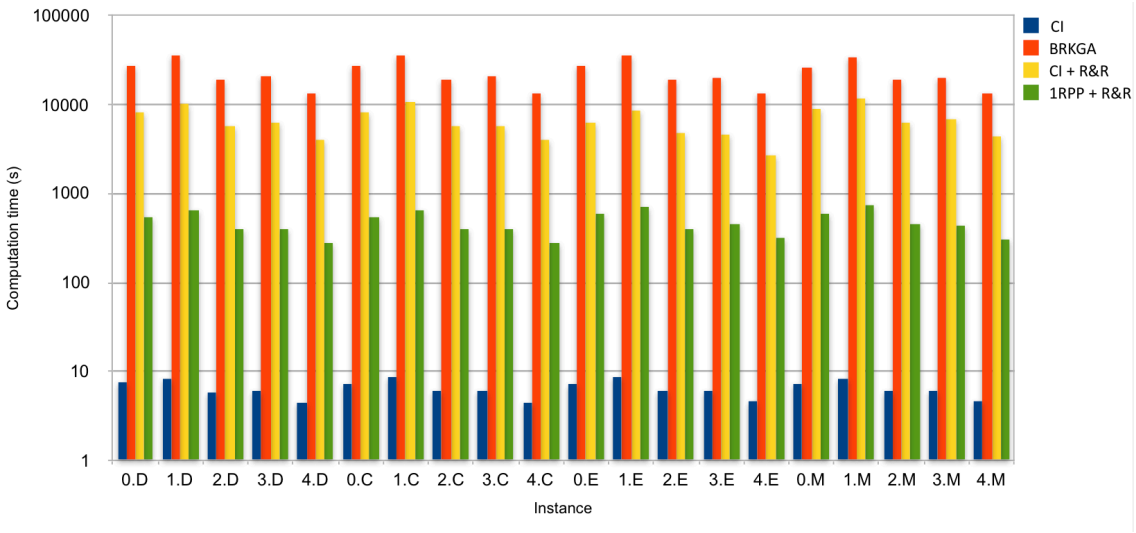


Figure 4.5.4: Algorithm performance with respect to computation time

Tables 4.A.1, 4.A.2 and 4.A.3 detail the results of the four algorithms – CI, BRKGA, CI + R&R and 1RPP + R&R – with respect to vehicle type (VT), number of parcels (#P), distance per parcel (Dis), emissions per parcel (Em), cost (€), number of routes (#R) and execution time (seconds).

4.5.2 Alternative policies analysis

This section simulates the proposed alternative parcel delivery policies on real-world e-commerce delivery data. Due to confidentiality issues, only aggregated results are presented. The size of the real-world instances makes CI the most appropriate algorithm for simulating scenarios enabling to investigate the impact of ECO vehicles on parcel delivery (Section 4.5.2.1), the impact of a shift from home deliveries to collection point retrievals (Section 4.5.2.2) and alternative delivery policies (Section 4.5.2.4). The instances employed to study the *regional carrier monopolies* (Section

4.5.2.3) are much smaller, which enables the application of the best performing algorithm in terms of quality, namely 1RPP + R&R.

4.5.2.1 Impact of ECO vehicles on parcel delivery

Three types of fuel were considered for these experiments: Diesel, CNG and Electricity (E). To enable an interesting comparison between the different fuel types and corresponding vehicles, drivers are assumed to have three primary tasks: load the van at the depot, deliver the parcels and debrief at the depot (such as reporting undelivered or refused parcels). Vehicle characteristics are defined for each fuel type in Table 4.5.2 under the assumptions that the maximum working time for a parcel delivery driver is ten hours per day and the process of both loading and debriefing takes approximately one hour in total.

Two electric vehicle types are defined: E-single which has a maximum tour duration of 9 hours, allowing for a single route per vehicle per day and an alternative E-double vehicle with a maximum tour duration of 4 hours, resulting in two routes per vehicle per day. E-double vehicles employ quick chargers at the depot capable of charging the battery up to 80% in one hour, thereby enabling the execution of up to two routes per day. The quick-charge duration is equivalent to the time required to debrief for the first route and load parcels for the second. Both CNG and E-vehicles employ start-stop technology, meaning their engines are switched off and zero emissions temporarily occur while the driver is outside the vehicle. Start-stop technology is optional for diesel vehicles and therefore a comparison is made between scenarios with and without start-stop technology.

Table 4.5.2: Vehicle properties for the ECO vehicle experiments.

	Diesel	Diesel S/S	CNG S/S	E-single	E-double
Speed	30 km/h	30 km/h	30 km/h	30 km/h	30 km/h
Range [a]	300 km	300 km	300 km	140 km	112 km
Max route time	9 h	9 h	9 h	9 h	4 h
Routes per day	1	1	1	1	2
Start/Stop	no	yes	yes	yes	yes
CO_2 emission [b]	3140 g/l	3140 g/l	2532 g/kg	278 g/kWh	278 g/kWh

[a] Ranges for the Diesel, Diesel S/S, and CNG vehicles are adjusted to reflect the limits imposed by vehicle speed and maximum route time.

[b] Electric vehicles do not produce emissions while driving, but the reported emissions are those corresponding to the production of the required electricity, assuming the average values reported by the EU (European Commission, 2011).

The effect of employing ECO vehicles for parcel delivery is simulated for a large (sub)urban area, with the depot located at the edge of the city center. 47,000 parcels require delivery over a period of two months and the maximum distance between depot and parcel destination is 18 km.

All parcels are delivered by a standard diesel van unequipped with start-stop technology in the reference scenario and this is compared against scenarios employing ECO vehicles.

Figure 4.5.5 summarizes the results for ECO-vehicle simulations. The chart demonstrates how applying start-stop technology during parcel delivery results in considerable emission decreases.

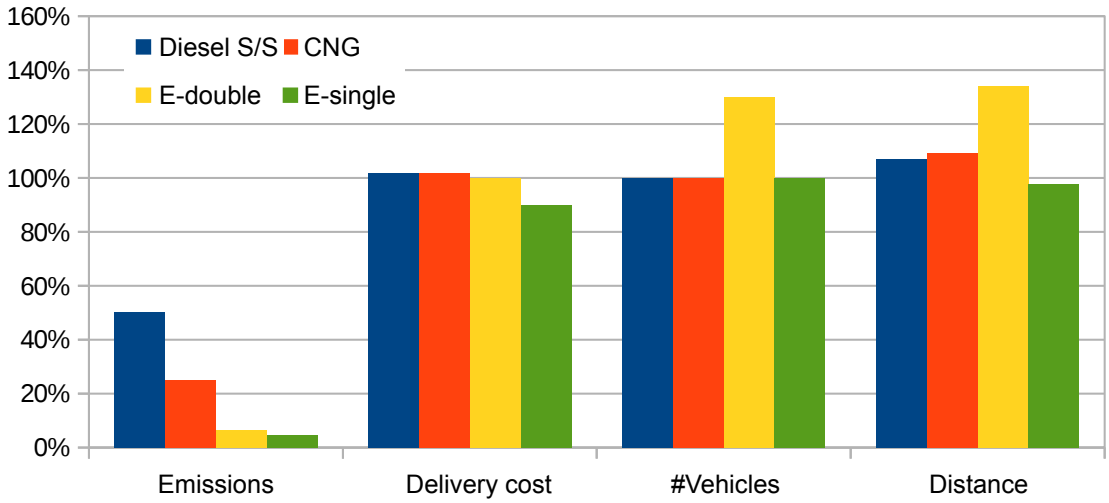


Figure 4.5.5: Percentual gain (vertical) of employing ECO vehicles on the different KPIs (horizontal).

Applying start-stop technology has no significant influence on other KPIs. Fuel costs, for example, represent such a small cost share that the decrease in fuel consumption results in no significant total cost reduction. When switching to CNG vehicles, CO_2 emissions decrease even further, while total costs remain stable. Furthermore, NO_x emissions are practically reduced to zero. Switching to CNG vehicles may therefore prove ecologically worthwhile for carriers. When employing E-double vehicles performing two routes per day with quick-charge technology, small decreases in the cost KPI are noticeable. There are, however, 25% more vehicles required to deliver the parcels, rendering them undesirable from a practical and investment perspective. When considering the E-single vehicles, however, a completely different result is obtained. The cost KPI is significantly reduced compared to the diesel van scenario and all parcels are delivered using an equal number of vehicles. Therefore, employing electric vehicles is both economically and ecologically worthwhile for carriers operating within urban environments.

4.5.2.2 Impact of a shift from home deliveries to collection point retrievals

Data concerning delivery parcels in an urban area over a period of two months was employed to accurately investigate the effects of transitioning from home deliveries to collection point retrievals. Approximately 28,000 parcels were delivered over this two month period with between 60 and 1,000 delivered on any single day. Characteristics of the vehicles employed are those of a standard diesel van running at 30 km/h, as detailed in Table 4.5.2.

An existing network of 24 collection points was utilized with all original delivery addresses within 3 kilometers of their collection point for the simulation. By contrast, all parcels are delivered to their original destination in the reference scenario. 10 - 100% of the parcels are randomly moved from their original destination to the nearest collection point.

Figure 4.5.6 illustrates the decrease in emissions when delivering varying percentages of parcels to collection points. Emission costs decrease by 0.3 to 78% when utilizing a dense collection point network. Internal KPIs (f_l, f_d and f_v) demonstrate a similar reduction, as indicated by

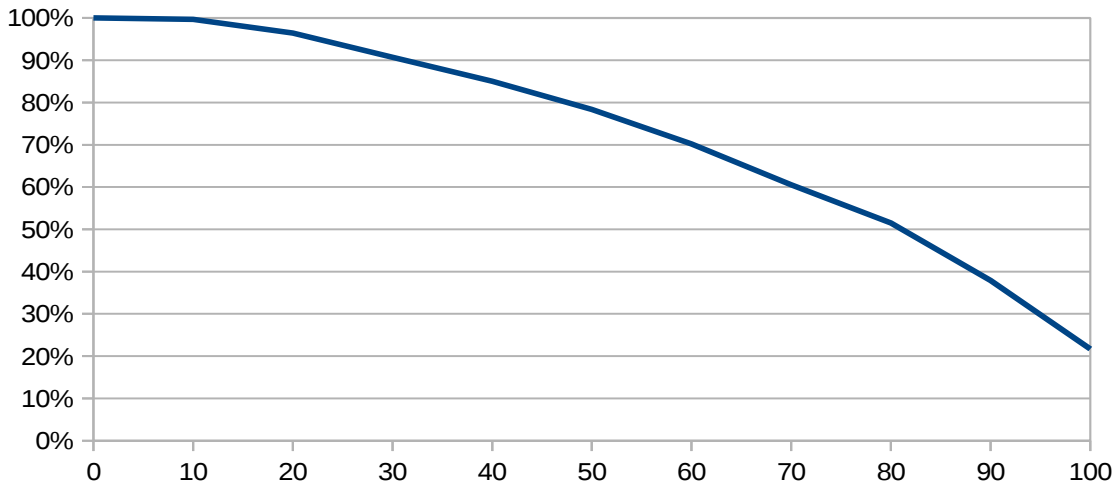


Figure 4.5.6: Influence of the percentage of parcels delivered at collection points (horizontal) on emissions (vertical).

Figure 4.5.7.

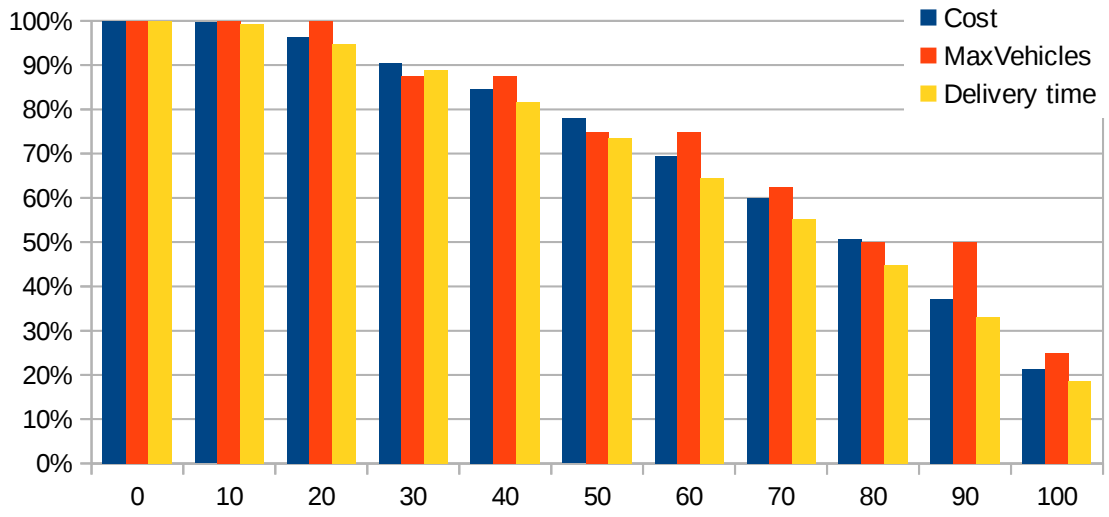


Figure 4.5.7: Influence of the percentage of parcels delivered at collection points (horizontal) on three KPIs (vertical).

It is noteworthy that the reimbursement of collection point personnel is not included in the cost computation and the actual cost decrease will, therefore, be somewhat lower.

4.5.2.3 Regional carrier monopolies: (sub)urban vs. rural

Two different regions were considered when simulating the effect of regional carrier monopolies. Region 1 is a medium-sized city and its surrounding suburban environment with 25,000 parcels requiring delivery over a period of one month, corresponding to a high parcel density. Parcel and depot data from two carriers was obtained, while only depot information was obtained from a third carrier for this region. The first two carriers have depots at the edge of the considered region,

whereas the third's is centrally located. In the reference scenario, Carriers 1 and 2 deliver parcels from their respective depots. Characteristics of the vehicles employed are those of a standard diesel van running at 30 km/h ((Sub)urban) and 50 km/h (Rural), as described in Table 4.5.3, and the algorithm applied is 1RPP + R&R heuristic.

Table 4.5.3: Vehicle properties for the default diesel van.

	City	Rural
Speed (diesel van)	30 km/h	50 km/h
Range[a]	300 km	500 km
Max route time	10 h	
CO_2 emission	3140 g/l	
Stop duration	4 minutes	
MultiStop duration	0.4 minutes per parcel	

[a]The vehicle's range is adjusted to reflect the speed limits and maximum route duration.

Figure 4.5.8 summarizes Region 1's results when all parcels are delivered by either Carrier 1, Carrier 2 or Carrier 3 in a monopoly scenario. Beware that all parcels' origins are assumed to be the monopoly holder's depot. Only one small advantage is to be made from assigning carrier monopolies in this type of dense region, with cost decreases of between 4% and 7%. All carriers' delivery routes appear to already be saturated and there is little advantage insofar as including additional parcels in their workload. Moving the depot from the region's edge to a central location (Carrier 3) results in greater KPI improvements, with a cost decrease of 14%, mainly due to the decrease in distance traveled per parcel. It may, however, still be questioned whether this decrease in delivery cost sufficiently compensates for the probable increase in costs and complexity in the long run since parcels must be redistributed outside current carrier logistic flows.

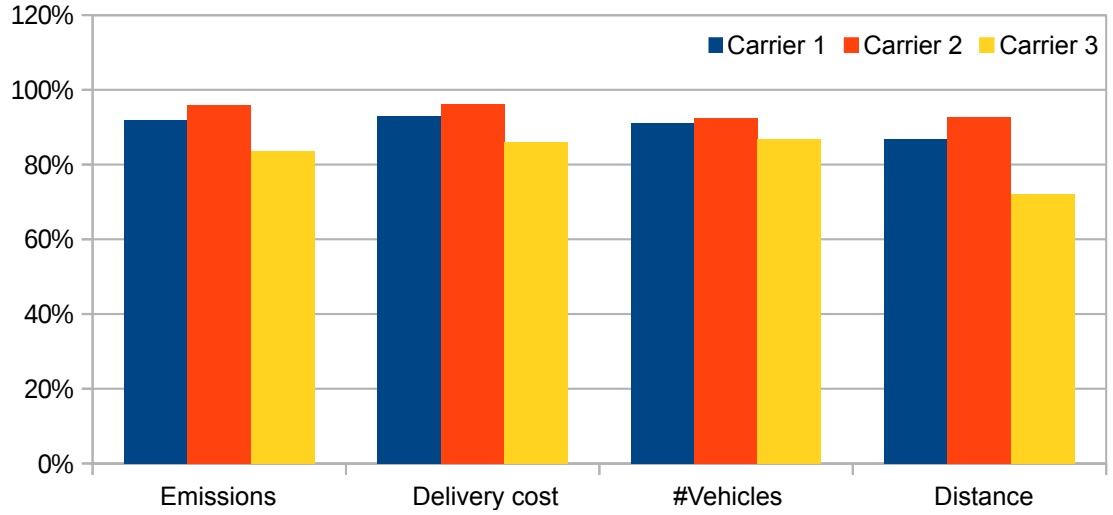


Figure 4.5.8: Relative delivery costs in case of an urban carrier monopoly (either Carrier 1, 2 or 3), compared to the situation where all three carriers have a share.

Region 2 is more rural with a total of 150 parcels requiring delivery over a period of five weeks, corresponding to a (very) low parcel density. Parcel and depot data was obtained from only

one single major carrier operating in this region. For investigating the effect of multiple carriers in the region, the delivery data have been divided into five sets. Subsequently, these sets have been assigned to separate fictitious carriers, thus mimicking the effect of five carriers operating simultaneously within the region. All parcels are assumed to be delivered within one week, albeit on their original delivery day of the week. In the reference scenario, parcels are delivered by five carriers independently. Four different scenarios simulate the impact of a rural carrier monopoly. The resulting costs and emissions are compared against the reference scenario.

Characteristics of the employed vehicles are those of a standard diesel van running at 50 km/h, as described in Table 4.5.3. Given how rural regions demand long travel distances per parcel, these are valid assumptions when compared against the 30 km/h for (sub)urban regions.

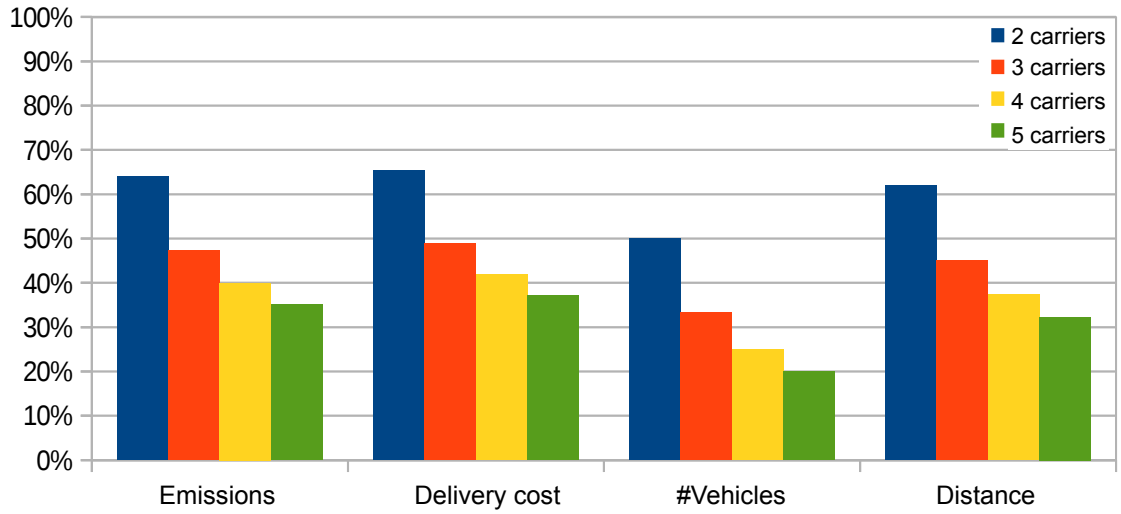


Figure 4.5.9: Relative KPIs in a rural carrier scenario where two, three, four or five carriers are merged into a monopoly, relfecting increasing delivery density. The reference scenario reflects all five carriers having their share.

Figure 4.5.9 summarizes the results of scenarios in which all activities of two, three, four or five carriers are merged into a monopoly home delivery. The results are compared against the reference scenario where all five carriers conduct their delivery share in the region.

By contrast to the results obtained in the (sub)urban region, the benefits of enforcing a regional carrier monopoly are considerable. Decreases in emissions, costs, required vehicles and distance per parcel of up to 80% were observed, while the average route duration increased from 4h40 to almost 8h, thereby highlighting the capability of operating far more efficient routes when under increasing density conditions.

Compared to urban deliveries, however, a monopolist's absolute rural delivery costs remain almost twice as high while the distance per parcel increases threefold, illustrating how low density regions prove commercially challenging even under monopoly conditions.

4.5.2.4 Alternative delivery policies

Impact of reduced service time and start-stop technology

The following experiments concern simulating the impact of reducing the time required to deliver a parcel to its destination (service time) and the application of start-stop technology during parcel delivery. 47,000 parcels are delivered by the default diesel van with a service time of 4 minutes for the reference scenario. The simulation scenarios, meanwhile, consider service times of 1 to 5 minutes and for each service time duration the influence of applying start-stop technology is investigated.

Several interesting conclusions may be extrapolated from the results presented in Figure 4.5.10. First and foremost, applying start-stop technology significantly reduces emissions during parcel delivery. Emissions are reduced by almost 40% for the base case (represented by the black line) where the service time is 4 minutes when switching off the engine during parcel delivery instead of letting the engine run idle (represented by the red line). Secondly, the graph clearly illustrates how service time must be halved to two minutes in order to obtain the same emission reductions as those obtained by applying start-stop technology when service time equals 4 minutes. Thus, from a purely ecological point of view, the application of start-stop technology represents a simple and immediate benefit since it is much easier to implement than delivery policy changes which halve service time.

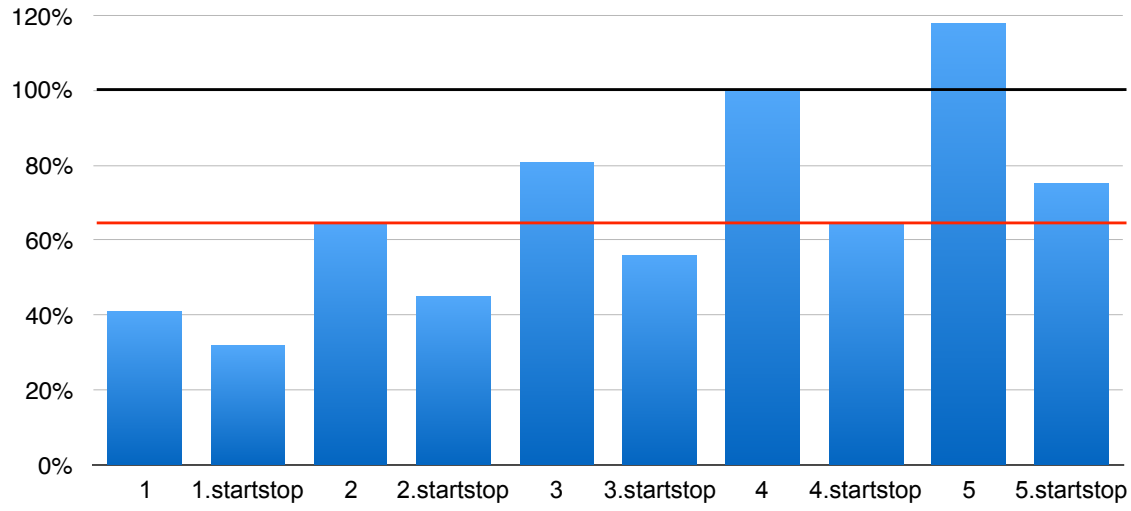


Figure 4.5.10: Influence of service times (horizontal (min)) and start-stop technology (startstop) on emissions (vertical). The reference scenario considers 4 minutes service time for standard vehicles without start-stop technology.

Figure 4.5.11, by contrast, visualizes the influence of start-stop technology and service time upon delivery costs. Given that the application of start-stop technology only affects total fuel cost (which is marginal compared to the driver's wage cost), it is unsurprising that the application of start-stop technology has only a negligible influence upon delivery costs. Therefore, from an economic perspective it may be a difficult decision concerning whether or not investing in additional start-stop technology for new vehicles is worthwhile. Much greater cost reductions

are obtained by reducing service time. Indeed, with high driver wage costs, any reduction in service time should incur significant delivery cost savings. These results demonstrate the need to further investigate the effects on service times by reducing parcel sizes and thereby increasing the percentage of parcels deliverable by mailbox.

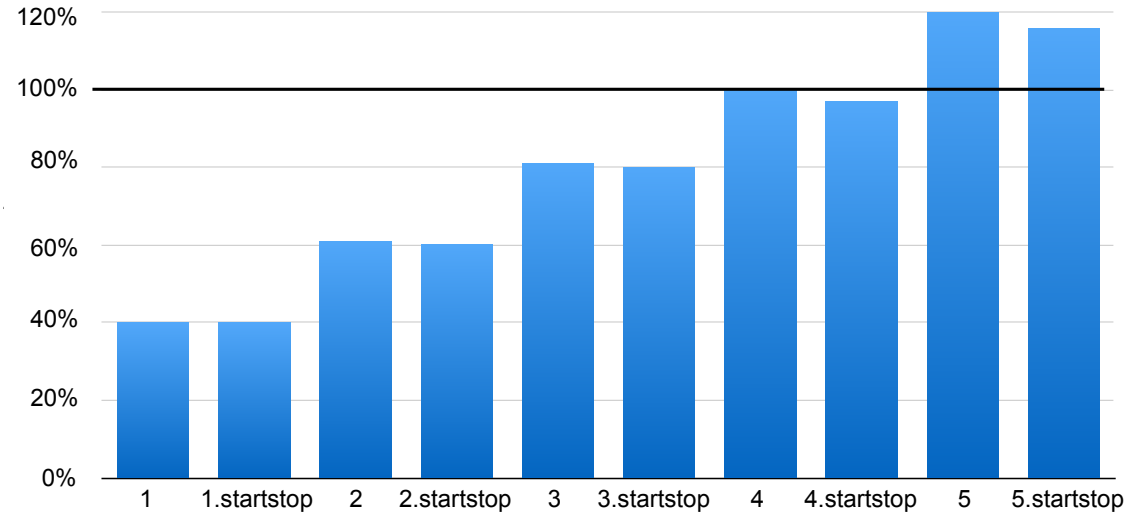


Figure 4.5.11: Influence of service times (horizontal (min)) and start-stop technology (startstop) on delivery cost (vertical). The reference scenario considers 4 minutes service time for standard vehicles without start-stop technology.

Impact of increased time windows

The following experiments simulate the impact of increasing time windows for parcel delivery in an urban environment. Default diesel vans with a 4 minute service time are employed to deliver over 47,000 parcels. The simulation investigates the impact of lengthening parcel delivery time windows from one to four days for 20%, 40% and 60% of the parcels.

Figure 4.5.12 presents the results, where the horizontal axis corresponds to the three scenario categories (20%, 40% and 60%) and the vertical axis details the level of emissions relative to the reference scenario (no extended time windows). The figure illustrates how increasing time windows does not yield the expected benefits. Indeed, increasing the time windows for 60% of the parcels by four days only results in emission decreases of 7%. Additionally, a time window increase for 20% of the parcels by two, three or four days result in similar emission decreases. A careful analysis of the results revealed that the real routes are densely loaded with deliveries. Regardless of any time window flexibilities, the most restrictive constraint appears to be the drivers' working time limits.

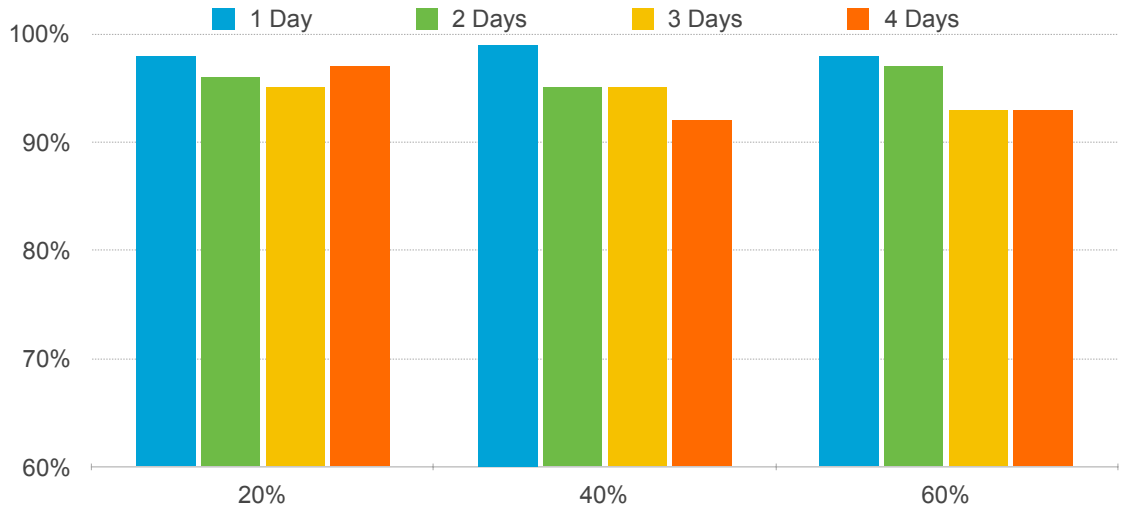


Figure 4.5.12: Impact of increased delivery time windows on emissions. The horizontal axis indicates the portion of deliveries with increased time windows and the vertical axis presents relative emissions. The reference scenario indicates emissions for the original time windows.

4.5.3 Discussion and insights

This paper's primary objective was to investigate the most profitable mode of e-commerce delivery operations, and it sought to reduce CO_2 emissions while satisfying customer service expectations. Computational experiments provided several insights which may be summarized as follows:

1. Contrary to intuition, regional carrier monopolies do not increase delivery efficiency, except in regions with very low parcel density.

2. Delivering parcels to a collection point instead of customers' homes significantly decreases both external (emissions) and internal costs (delivery cost, service time, fleet size). Moreover, customers collect their order at a delivery point whenever it proves convenient. This represents an indirect benefit for the transporter (no additional waiting times, no failed deliveries).

3. Significant benefits are obtained in all scenarios when the percentage of *bundled* deliveries is increased. Indeed, the number of orders requiring delivery to one point does not linearly increase service time at this destination.

4. Reducing delivery duration has the greatest impact on total cost, as expected, since it shortens the drivers' working time. For instance, if a customer is not at home, the driver must first wait for some time before writing a delivery notice.

5. Extending time windows does not yield the expected benefit, the primary reason being that most tours are saturated and cost reductions are marginal when compared against driver costs or those generated by to-the-door delivery.

4.6. Conclusion

The paper proposed a general optimization approach to the e-commerce delivery problem, enabling the assessment of various operational delivery policies on both costs and emissions. A mixed integer programming formulation, heuristic approaches and several delivery policies were presented. Computational results were obtained by applying the presented heuristics to a selection of anonymized real-world data gathered from e-commerce delivery transporters.

Interestingly, experimental results often proved counter-intuitive. For example, while one might assume that extending delivery time windows would reduce both costs and emissions, results indicate that such relaxations result in little or no profit. Given high personnel costs, one would have also assumed that reducing the driving time would be beneficial. Instead, it appears that, especially in urban areas, reducing delivery time potentially contributes more significantly towards cost and emission reductions than reducing the actual driving time. Results reveal how emission costs are decreased by up to 78% when utilizing a dense collection point network and by up to 80% when enforcing a regional carrier monopoly in rural areas. Another important observation is that switching off vehicle engines during individual parcel delivery reduces the environmental impact considerably.

Several interesting avenues exist for future research. Recently, many companies in the United States and India have begun implementing a *crowd-shipping* strategy (Archetti et al., 2016). By employing crowd-shipping, distributors ask customers who are collecting their orders to deliver those of other customers on their return journey in exchange for certain incentives. From a policy point of view, the environmental impact of integrating crowd-shipping into policies such as delivering to collection points or *carrier bundling* on home deliveries is worth exploring. This paper researches the impact of a few new policies, but many more remain to be investigated. Other research directions may focus on the realistic modeling of certain vehicle processes, such as energy consumption or CO_2 emissions. The vehicles' capacity constraints may represent yet another important issue that should be further investigated.

Appendix

4.A. Detailed computational results

Table 4.A.1: Comparison of CI, BRKGA and R&R over rural instances

Ins.	VT	#P	CI					BRKGA					CI + R&R					1RPP + R&R				
			Dis.	Em.	€	#R	time	Dis.	Em.	€	#R	time	Dis.	Em.	€	#R	time	Dis.	Em.	€	#R	time
R0D	D	17	13.50	4332.34	15.52	1.00	0.02	13.50	4332.34	15.46	1.00	41.57	13.50	4332.34	15.52	1.00	1.01	13.50	4332.34	15.52	1.00	0.80
R1D	D	76	5.22	1772.92	6.59	2.00	0.03	4.73	1622.12	6.02	2.00	516.46	4.68	1605.67	6.02	2.00	5.39	4.67	1605.46	6.01	2.00	3.99
R2D	D	5	25.23	7849.71	27.49	1.00	0.01	25.23	7849.71	27.46	1.00	3.05	25.23	7849.71	27.49	1.00	0.46	25.23	7849.71	27.49	1.00	0.30
R3D	D	25	5.85	1944.14	7.13	1.00	0.01	5.81	1932.90	7.05	1.00	43.68	5.85	1944.14	7.13	1.00	0.87	5.86	1947.95	7.15	1.00	0.90
R4D	D	10	13.02	4209.88	15.15	1.00	0.01	13.02	4209.88	15.09	1.00	16.38	13.02	4209.88	15.15	1.00	0.60	13.02	4209.88	15.15	1.00	0.59
R5D	D	6	23.69	7444.99	26.26	1.00	0.01	23.69	7444.99	26.21	1.00	5.33	23.69	7444.99	26.26	1.00	0.47	24.39	7659.40	27.01	1.00	0.37
R6D	D	5	31.57	9888.54	34.81	1.00	0.01	31.57	9888.54	34.74	1.00	5.69	31.57	9888.54	34.81	1.00	0.47	31.57	9888.54	34.81	1.00	0.36
R7D	D	9	12.50	3908.77	13.74	1.00	0.01	12.50	3908.77	13.72	1.00	2.80	12.50	3908.77	13.74	1.00	0.39	12.50	3908.77	13.74	1.00	0.27
R8D	D	7	28.60	8959.81	31.54	1.00	0.01	28.60	8959.81	31.48	1.00	7.53	28.60	8959.81	31.54	1.00	0.49	28.60	8959.81	31.54	1.00	0.42
R9D	D	3	61.19	19009.47	66.50	1.00	0.01	61.19	19009.47	66.42	1.00	4.58	61.19	19009.47	66.50	1.00	0.44	61.19	19009.47	66.50	1.00	0.32
R10D	D	13	11.61	3674.56	13.03	1.00	0.01	11.59	3668.66	12.98	1.00	8.56	11.59	3668.66	13.01	1.00	0.49	11.59	3668.66	13.01	1.00	0.42
R11D	D	18	8.93	2896.05	10.45	1.00	0.01	8.90	2887.03	10.37	1.00	27.64	8.93	2896.05	10.45	1.00	0.68	8.94	2898.95	10.46	1.00	0.71
R12D	D	20	7.79	2474.48	8.80	1.00	0.01	7.79	2474.48	8.77	1.00	7.96	7.79	2474.48	8.80	1.00	0.50	7.79	2474.48	8.80	1.00	0.41
R13D	D	41	4.59	1485.41	5.35	1.00	0.01	4.09	1332.25	4.80	1.00	25.49	4.13	1345.56	4.87	1.00	0.61	4.12	1342.48	4.86	1.00	0.63
R14D	D	9	13.51	4381.35	15.80	1.00	0.01	13.51	4381.21	15.73	1.00	15.63	13.51	4381.21	15.80	1.00	0.60	13.51	4381.21	15.80	1.00	0.59
R15D	D	14	13.89	4470.86	16.05	1.00	0.01	13.89	4470.86	15.98	1.00	27.62	13.89	4470.86	16.05	1.00	0.71	14.01	4508.38	16.18	1.00	0.74
R0C	C	17	13.50	4162.68	15.85	1.00	0.01	13.50	4162.68	15.79	1.00	42.84	13.50	4162.68	15.85	1.00	0.76	13.50	4162.68	15.85	1.00	0.80
R1C	C	76	5.22	1688.50	6.72	2.00	0.03	4.73	1542.48	6.14	2.00	411.63	4.68	1526.55	6.12	2.00	5.29	4.66	1522.07	6.11	2.00	4.10
R2C	C	5	25.23	7580.35	28.12	1.00	0.01	25.23	7580.35	28.09	1.00	2.80	25.23	7580.35	28.12	1.00	0.46	25.23	7580.35	28.12	1.00	0.27
R3C	C	25	5.85	1857.63	7.27	1.00	0.01	5.81	1846.74	7.19	1.00	43.68	5.85	1857.63	7.27	1.00	0.82	5.85	1859.13	7.28	1.00	0.89
R4C	C	1	13.02	4040.41	15.47	1.00	0.01	13.02	4040.41	15.41	1.00	16.14	13.02	4040.41	15.47	1.00	0.58	13.02	4040.41	15.47	1.00	0.59
R5C	C	6	23.69	7178.01	26.85	1.00	0.01	23.69	7178.01	26.80	1.00	5.34	23.69	7178.01	26.85	1.00	0.50	24.39	7385.61	27.61	1.00	0.37
R6C	C	5	31.57	9538.72	35.59	1.00	0.01	31.57	9538.72	35.53	1.00	5.74	31.57	9538.72	35.59	1.00	0.47	31.57	9538.72	35.59	1.00	0.39
R7C	C	9	12.50	3771.54	14.05	1.00	0.01	12.50	3771.54	14.03	1.00	3.78	12.50	3771.54	14.05	1.00	0.40	12.50	3771.54	14.05	1.00	0.29
R8C	C	7	28.60	8642.85	32.24	1.00	0.01	28.60	8642.85	32.19	1.00	7.88	28.60	8642.85	32.24	1.00	0.49	28.60	8642.85	32.24	1.00	0.42
R9C	C	3	61.19	18362.11	68.02	1.00	0.01	61.19	18362.11	67.95	1.00	4.52	61.19	18362.11	68.02	1.00	0.45	61.19	18362.11	68.02	1.00	0.31
R10C	C	13	11.61	3538.39	13.32	1.00	0.01	11.59	3532.69	13.27	1.00	8.57	11.59	3532.69	13.30	1.00	0.50	11.59	3532.69	13.30	1.00	0.41
R11C	C	18	8.93	2777.90	10.67	1.00	0.01	8.90	2769.17	10.59	1.00	27.64	8.93	2777.90	10.67	1.00	0.69	8.94	2781.52	10.68	1.00	0.69
R12C	C	20	7.79	2381.71	8.99	1.00	0.01	7.79	2381.71	8.96	1.00	8.39	7.79	2381.71	8.99	1.00	0.51	7.79	2381.71	8.99	1.00	0.43
R13C	C	41	4.59	1425.26	5.46	1.00	0.01	4.09	1276.96	4.90	1.00	27.32	4.13	1289.84	4.97	1.00	0.62	4.13	1288.26	4.96	1.00	0.64
R14C	C	9	13.51	4202.95	16.13	1.00	0.01	13.51	4202.82	16.07	1.00	14.32	13.51	4202.82	16.13	1.00	0.58	13.51	4202.82	16.13	1.00	0.58
R15C	C	14	13.89	4293.66	16.38	1.00	0.01	13.89	4293.66	16.33	1.00	27.50	13.89	4293.66	16.38	1.00	0.71	14.00	4325.80	16.50	1.00	0.75
R0M	M	17	13.50	4162.68	15.85	1.00	0.02	13.50	4162.68	15.79	1.00	25.02	13.50	4162.68	15.85	1.00	0.83	18.15	4054.08	20.77	1.90	0.87
R1M	M	76	7.56	1637.95	9.19	4.00	0.04	9.08	991.59	10.74	5.00	302.54	4.70	1068.98	6.11	2.00	2.06	5.77	1032.59	7.24	2.90	4.06
R2M	M	5	25.23	1096.96	27.68	1.00	0.01	25.23	1096.96	27.68	1.00	2.90	25.23	1096.96	27.68	1.00	0.72	25.23	1096.96	27.68	1.00	0.34
R3M	M	25	13.74	597.46	15.57	3.00	0.02	10.37	450.74	11.97	2.00	28.89	10.37	450.74	11.97	2.00	0.91	10.37	450.74	11.97	2.00	1.00
R4M	M	10	21.67	942.23	24.44	2.00	0.02	21.67	942.23	24.44	2.00	12.60	21.67	942.23	24.44	2.00	0.82	21.67	942.23	24.44	2.00	0.67
R5M	M	6	32.29	1404.25	35.61	2.00	0.01	32.29	1404.25	35.61	2.00	5.33	32.29	1404.25	35.61	2.00	0.80	32.29	1404.25	35.61	2.00	0.41
R6M	M	5	55.28	9878.54	60.85	2.00	0.01	31.57	9538.72	35.53	1.00	5.45	31.57	9538.72	35.59	1.00	0.76	31.57	9538.72	35.59	1.00	0.41
R7M	M	9	12.50	543.52	13.83	1.00	0.01	12.50	543.52	13.83	1.00	2.79	12.50	543.52	13.83	1.00	0.72	12.50	543.52	13.83	1.00	0.33
R8M	M	7	28.60	8642.85	32.24	1.00	0.01	28.60	8642.85	32.19	1.00	6.90	28.60	8642.85	32.24	1.00	0.78	28.60	8642.85	32.24	1.00	0.49
R9M	M	3	61.19	18362.11	68.02	1.00	0.01	61.19	18362.11	67.95	1.00	4.08	61.19	18362.11	68.02	1.00	0.74	61.19	18362.11	68.02	1.00	0.44
R10M	M	13	18.90	821.73	20.90	2.00	0.01	17.39	756.01	19.28	2.00	7.39	17.39	756.01	19.28	2.00	0.81	17.39	756.01	19.28	2.00	0.52
R11M	M	18	20.80	2798.91	23.30	3.00	0.01	19.98	2967.00	22.40	3.00	18.38	14.43	2611.89	16.50	2.00	0.84	13.42	2628.91	15.43	1.90	0.75
R12M	M	20	19.93	2769.90	21.94	3.00	0.01	7.79	2381.71	8.96	1.00	6.70	7.79	2381.71	8.99	1.00	0.80	7.79	2381.71	8.99	1.00	0.48
R13M	M	41	9.30	1315.60	10.47	3.00	0.01	4.09	1276.96	4.90	4.00	16.88	6.39	1264.30	7.36	2.00	0.84	6.38	1264.10	7.36	2.00	0.69
R14M	M	9	13.51	587.67	15.85	1.00	0.01	13.51	587.65	15.85	1.00	14.87	13.51	587.65	15.85	1.00	0.83	13.51	587.65	15.85	1.00	0.65
R15M	M	14	13.89	4293.66	16.38	1.00	0.02	13.89	4293.66	16.33	2.00	20.37	13.89	4293.66	16.38	1.00	0.84	20.47	4137.62	23.37	2.00	0.79

Table 4.A.2: Comparison of CI, BRKGA and R&R over urban instances

Ins.	VT	#P	CI					BRKGA					CI + R&R					1RPP + R&R				
			Dis.	Em.	€	#R	time	Dis.	Em.	€	#R	time	Dis.	Em.	€	#R	time	Dis.	Em.	€	#R	time
U0D	D	837	0.27	257.79	1.35	5.00	0.28	0.19	233.41	1.21	5.00	3478.22	0.25	251.38	1.33	5.00	4.62	0.18	229.05	1.25	5.10	4.11
U1D	D	681	0.28	250.67	1.30	4.00	0.17	0.20	227.44	1.17	4.00	2429.35	0.25	240.65	1.27	4.00	1.84	0.19	223.13	1.21	4.10	2.54
U2D	D	886	0.24	228.36	1.20	5.00	0.22	0.17	208.09	1.08	5.00	2857.47	0.22	221.72	1.18	4.90	3.91	0.16	203.28	1.11	4.50	3.51
U3D	D	612	0.00	0.00	0.00	0.00	0.00	0.22	249.49	1.29	4.00	2527.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
U4D	D	594	0.00	0.00	0.00	0.00	0.00	0.23	244.92	1.26	4.00	1682.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
U0C	C	837	0.27	221.42	1.35	5.00	0.26	0.20	199.33	1.22	5.00	3111.08	0.25	214.62	1.33	5.00	4.33	0.18	193.52	1.25	5.00	4.11
U1C	C	681	0.28	215.98	1.30	4.00	0.18	0.21	193.99	1.18	4.00	2131.97	0.24	204.94	1.26	4.00	1.68	0.19	189.44	1.21	4.00	2.49
U2C	C	886	0.24	196.06	1.20	5.00	0.23	0.18	178.62	1.09	5.00	2489.18	0.22	188.83	1.17	4.80	4.22	0.16	171.57	1.11	4.30	3.38
U3C	C	612	0.00	0.00	0.00	0.00	0.00	0.23	213.91	1.30	4.00	2142.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
U4C	C	594	0.00	0.00	0.00	0.00	0.00	0.23	209.13	1.26	4.00	1723.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
U0E	E	837	0.27	11.89	1.30	5.00	0.26	0.20	8.50	1.22	5.00	2744.44	0.25	11.05	1.28	5.00	4.61	0.18	7.75	1.20	5.00	4.10
U1E	E	681	0.28	12.15	1.25	4.00	0.17	0.20	8.88	1.17	4.00	1764.47	0.24	10.60	1.22	4.00	1.85	0.19	8.22	1.16	4.00	2.50
U2E	E	886	0.17	7.32	0.81	3.50	0.15	0.18	7.77	1.09	5.00	2132.38	0.15	6.53	0.79	3.30	2.76	0.11	4.84	0.75	3.10	2.44
U3E	E	612	0.00	0.00	0.00	0.00	0.00	0.22	9.73	1.29	4.00	1869.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
U4E	E	594	0.00	0.00	0.00	0.00	0.00	0.22	9.71	1.26	4.00	1471.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
U0M	M	837	0.27	11.89	1.30	5.00	0.26	0.19	8.32	1.21	5.00	2772.98	0.26	11.29	1.29	5.00	2.68	0.30	13.09	1.33	5.00	4.47
U1M	M	681	0.28	12.15	1.25	4.00	0.19	0.20	8.75	1.17	4.00	1765.54	0.25	10.78	1.22	4.00	1.28	0.24	10.86	1.22	4.10	2.73
U2M	M	886	0.00	0.00	0.00	0.00	0.00	0.17	7.45	1.08	5.00	2183.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
U3M	M	612	0.00	0.00	0.00	0.00	0.00	0.22	9.67	1.29	4.00	1838.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
U4M	M	594	0.00	0.00	0.00	0.00	0.00	0.23	9.79	1.26	4.00	1439.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 4.A.3: Comparison of CI, BRKGA and R&R over suburban instances

Ins.	VT	#P	CI					BRKGA					CI + R&R					1RPP + R&R				
			Dis.	Em.	€	#R	time	Dis.	Em.	€	#R	time	Dis.	Em.	€	#R	time	Dis.	Em.	€	#R	time
S0D	D	5194	1.66	709.47	2.99	55.00	7.08	1.65	706.39	2.92	53.40	27734.38	1.32	606.60	2.63	49.50	7476.69	0.94	488.61	2.22	42.90	500.09
S1D	D	5529	1.75	738.82	3.10	61.00	8.20	1.56	683.35	2.84	56.20	37197.88	1.38	626.13	2.71	54.00	9767.92	0.91	481.87	2.21	45.60	588.57
S2D	D	4466	1.76	747.23	3.14	50.00	5.70	1.73	737.07	3.04	52.80	19869.71	1.38	631.93	2.74	44.20	5284.41	0.97	504.93	2.30	38.10	350.95
S3D	D	4580	1.81	758.25	3.16	51.00	5.64	1.91	787.92	3.20	53.40	20802.09	1.39	628.63	2.71	45.00	5277.74	0.96	499.14	2.27	38.60	356.42
S4D	D	3743	2.04	841.02	3.48	46.00	4.37	1.76	755.36	3.12	55.00	13786.34	1.56	692.25	2.96	40.10	3478.97	1.05	538.26	2.43	33.90	249.27
S0C	C	5194	1.66	654.27	3.02	55.00	7.06	1.65	650.88	2.95	55.00	27615.20	1.30	547.16	2.63	49.10	7286.01	0.92	435.79	2.22	42.60	486.44
S1C	C	5529	1.75	682.27	3.13	61.00	8.21	1.53	619.52	2.85	55.80	37097.37	1.37	569.92	2.72	53.80	9624.12	0.90	432.80	2.22	45.40	598.19
S2C	C	4466	1.76	689.81	3.17	50.00	5.66	1.70	673.01	3.05	55.40	19866.75	1.39	579.91	2.77	44.20	5195.64	0.95	450.62	2.29	37.90	358.38
S3C	C	4580	1.81	701.12	3.20	51.00	5.76	1.91	731.66	3.26	53.20	20826.72	1.39	577.65	2.75	45.00	5482.66	0.96	448.30	2.27	38.50	356.38
S4C	C	3743	2.04	779.39	3.52	46.00	4.46	1.67	668.55	3.06	52.80	13565.85	1.51	623.17	2.95	39.50	3666.40	1.07	490.33	2.46	34.20	251.55
S0E	E	5194	1.35	58.56	2.61	56.00	7.00	1.57	68.28	2.85	56.40	27729.59	1.00	43.49	2.24	44.40	5735.09	0.91	39.49	2.14	43.00	509.56
S1E	E	5529	1.37	59.69	2.65	60.00	8.15	1.49	64.58	2.77	64.20	37192.35	1.01	43.95	2.26	47.60	8171.89	0.88	38.30	2.12	45.30	654.86
S2E	E	4466	1.55	67.22	2.86	54.00	5.79	1.57	68.49	2.89	55.40	19764.83	1.03	44.95	2.31	39.40	3945.61	0.94	40.68	2.21	38.00	392.21
S3E	E	4580	1.54	66.85	2.83	55.00	5.67	1.56	67.92	2.85	55.00	20578.25	1.02	44.27	2.27	39.70	4226.97	0.93	40.26	2.17	38.30	385.85
S4E	E	3743	1.83	79.39	3.20	53.00	4.33	1.76	76.47	3.13	56.20	13491.12	1.11	48.34	2.44	34.90	2616.27	1.02	44.45	2.34	33.90	306.83
S0M	M	5194	1.73	680.43	3.07	58.00	7.26	1.71	656.64	3.03	59.00	27314.66	1.30	535.74	2.61	49.30	8439.64	0.96	431.93	2.25	43.40	529.60
S1M	M	5529	1.71	675.85	3.06	61.00	8.29	1.48	596.70	2.82	57.80	35482.25	1.36	559.40	2.68	53.70	10676.80	0.94	432.76	2.24	46.10	626.92
S2M	M	4466	1.80	692.11	3.18	51.00	5.75	1.68	657.72	3.05	54.80	19850.22	1.37	552.34	2.72	44.10	5819.49	1.02	444.65	2.35	38.80	390.00
S3M	M	4580	1.78	686.82	3.14	52.00	5.71	1.81	690.41	3.16	55.20	20042.94	1.38	554.23	2.71	44.90	5994.38	0.99	434.21	2.29	39.00	402.42
S4M	M	3743	2.01	753.29	3.45	47.00	4.48	1.92	734.30	3.35	55.00	13667.59	1.52	591.20	2.92	39.60	3916.82	1.12	472.44	2.50	34.90	267.18

Chapter 5

Conclusions and future research

This dissertation investigated theory and practice of challenges within e-commerce logistics management from two perspective: efficiency and sustainability, thereby, making scientific, social and industrial contributions.

On the efficiency in e-commerce logistics, integration of decisions in warehouse scheduling was investigated. The majority of studies in warehouse scheduling focus on introducing new solution approaches for particular problem variations of either location assignment or solely crane scheduling. Therefore, important aspects of integrated optimization in warehouse scheduling are neglected, regarding both theoretical and practical aspects. Chapters 2 and 3 are dedicated to the integrated decision making in warehouse scheduling proposing new formulations and efficient solution approaches, each with considering realistic constraints. Specifically, Chapter 2 considers the Scheduling Crane Operations Problem (SCOP) in rail-rail transshipments terminals. There are two particular characteristics of the SCOP that had not been studied. *i*) The cranes operate side by side in their own designated working area and between two neighbouring cranes there is an overlap working area in which only one of the two cranes may be present, and *ii*) Unlike the container terminals there is no input or output points for containers. The containers are located on the trains and must be moved among them, which means moving a container from its position to its destination train opens room to move other containers. Chapter 3 introduced the integrated Crane-operated Warehouse Scheduling Problem (CWSP) to investigate the impact of integrating a location assignment problem (LAP) and a crane scheduling problem (CSP) in crane-operated warehouses. The CWSP, considers a different setting than in Chapter 2 in which the cranes move along the storage area to reach the respective input/output point during their operations. This means cranes can traverse the entire storage area provided safety distances between all cranes are respected. The new theoretical insights in this dissertation enable a deeper and more realistic understanding of scheduling problems in warehouses and container terminals, and have potential impact in the way such problems will be tackled in the future. Decision support systems, for example, could integrate location assignment and crane scheduling in a framework to efficiently find optimal solutions. The practical contributions directly impact the industry by facilitating practitioners towards applying academic results.

Furthermore, this dissertation studied the effects of alternative policies on carbon emissions in e-commerce deliveries. Despite valuable contributions in literature, there were unanswered questions regarding sustainable e-commerce deliveries. Chapter 4 explores sustainable decision making in e-commerce delivery and introduced an optimization approach to enable the assessment of various operational alternative delivery policies on both operations costs and ecological impact. The outcome of this research enables the assessment of various operational delivery policies on both costs and emissions.

It is worth mentioning that the formulations and solutions approaches developed in this thesis may be applied to other problems which share similar challenges as e-commerce logistics with a limited effort. Section 5.1 overviews the main conclusions and contributions of this dissertation, followed by ongoing and future research in Section 5.2.

5.1. Conclusions

This thesis started by focusing on warehouse scheduling in the RRTT. An in depth study of the existing challenges in RRTTs to classify the operational problems is conducted which resulted in a new classification of the problems in RRTTs which integrates different existing sub-problems into four integrated problems.

Then, a new mathematical formulation that employs continuous variables to represent the time was presented. This formulation enabled incorporating realistic constraints within the RRTT.

Finally, on the solution approach front for the SCOP new heuristics based on genetic algorithm were designed to minimize the total transshipment time and number of containers remaining on the yard. The proposed algorithms are able to solve the integrated SCOP in a yard with multiple cranes with overlapping working areas and buffer lines which employ ground transportation. Extensive experiments show how proposed algorithms improve solutions compare to typical dispatching rules by at least 43%. The insights also illustrate when trains' load factor is not high, the size of overlap area has no significant impact on total transshipment time.

More than proposing valid formulations and solution approaches to the warehouse scheduling in the RRTT, the thesis have enriched the line of research dedicated to include integrated decision making in warehouse scheduling. In addition, a set of instances designed and published to encourage future research in the field. The set of instances is inspired from a real-world RRTT which considers different characteristics of the problem.

The results of the aforementioned contributions are two research papers:

- Sam Heshmati, Zafeiris Kokkinogenis, Rosaldo J. F. Rossetti, Maria Antónia Carravilla, and José Fernando Oliveira. An agent-based approach to schedule crane operations in rail-rail transshipment terminals. In Raquel J. Fonseca, Gerhard-Wilhelm Weber, and João Telhada, editors, *Computational Management Science*, pages 91–97, Cham, 2016. Springer International Publishing. ISBN 978-3-319-20430-7

- Sam Heshmati, Maria Antónia Carravilla, and José Fernando Oliveira. Scheduling Crane Operations in Rail-Rail Transshipment Terminals to be submitted to Journal of Scheduling.

Inspired by the SCOP in RRTT, new setting of storage yard was considered. In crane-operated warehouses the cranes can move along the storage yard and they don't have dedicated working area, and there are input output point around the yard. New time-continuous mathematical formulation was presented which incorporates the cranes' movement considering the safety distance for both the integrated and sequential approaches. An efficient solution approach was developed based on the LAHC.

The results revealed a significant potential to minimize the storage cost and crane utilization, by simulating real-world cases and comparing the new approaches with typical dispatching rules. This work enriches the warehouse scheduling field by incorporating realistic constraints and introducing new mathematical formulations and solution approaches. Most or at least part of the challenges faced by the crane-operated warehouses are common to other storage areas which employ overhead cranes, thus the formulations and solutions approaches developed in these chapters may be applied to other problems with a limited effort. Hence, outcome of these chapters are applicable in smart warehouses, storage areas or container terminals where employ overhead cranes to move the products or containers with different crane settings. The aforementioned contributions are included in following research paper:

- Sam Heshmati, Túlio A.M. Toffolo, Wim Vancroonenburg, and Greet Vanden Berghe. Crane-operated warehouses: Integrating location assignment and crane scheduling. *Computers & Industrial Engineering*, 129:274–295, 2019. ISSN 0360-8352

Regarding sustainable decision making in e-commerce logistics, the EDP was presented. Extensive simulations employing proposed solution approaches and real-world data resulted in valuable insights. The insights of this work contributes directly to the e-commerce and parcel delivery industry. There was some counter intuitive findings such as extending delivery time windows result in little or no profit. Results reveal how emissions are reduced when utilizing a dense collection point network and also when enforcing a regional carrier monopoly in rural areas. Another important observation is that switching off vehicle engines during individual parcel delivery reduces the environmental impact considerably. To summarize, the contributions of this section include introducing the e-commerce delivery problem (EDP) and proposing efficient algorithms to solve different size instances from academic size to real-world size ones (up to 3 million parcels). As well as, presenting insights to find the most profitable mode of e-commerce delivery while reducing ecological impact and satisfying customer service expectations by providing a simulation framework to enable decision makers in industry evaluate alternative delivery policies. The results of the aforementioned contributions have been published in the following venue:

- Sam Heshmati, Jannes Verstichel, Eline Esprit, and Greet Vanden Berghe. Alternative e-commerce delivery policies. *EURO Journal on Transportation and Logistics*, 2018. ISSN 2192-4384

5.2. Future research

Considering the rapid development of information technology and advanced manufacturing technology, to enhance productivity and cope with the increasing demand, the logistics management is being transferred from digital to intelligent. This thesis proposed integrated optimization approach in crane-operated warehouses by considering realistic constraints in a deterministic and centralized framework. There are several interesting avenues of future research which may be considered and investigated. There are many scenarios that require to find a good schedule when there are uncertainties in requests' arrival times or any other sort of incomplete data in storage areas. Due to nature of operations in crane-operated warehouses an important research direction may include online scheduling or stochastic online scheduling (Vredeveld, 2012) to consider such uncertainties. Another approach to crane-operated warehouse scheduling problem may include distributed optimization approaches. With the advent of edge computing (Shi et al., 2016) that allows real-time distributed processing of large scale data, distributed computing and distributed decision making are becoming more realistic. The goal of distributed optimization is to solve an optimization problem where the objective function is formed by summation of local objective functions, and each processing node has access only to one local objective function. Distributed optimization approaches, with the ability to perform parallel computations, would have valuable contributions, both in terms of solution speed and the maximum problem size that can be addressed. By emerging smart manufacturing where real time decision making, uncertainty, and large-scale demand come hand in hand, approaches which employ integrated optimization, stochastic online scheduling and distributed optimization simultaneously may be beneficial. Furthermore, exact solution approaches would be a relevant contribution which may include proposing efficient lower bound methods.

With respect to alternative e-commerce delivery policies, future research may include both more realistic modeling of the problem and integrating new e-commerce technologies and innovations into environmental policies. To model the problem more realistically, considering certain vehicle processes, such as energy consumption or CO2 emissions would be a valuable contribution. The vehicles' capacity constraints may represent yet another important issue that should be further investigated. Integrating crowd-shipping as a newly emerged innovation within e-commerce industry into policies such as delivering to collection points or carrier bundling on home deliveries is worth exploring.

References

Knut Alicke. Modeling and optimization of the intermodal terminal mega hub. In Hans-Otto Günther and KapHwan Kim, editors, *Container Terminals and Automated Transport Systems*, pages 307–323. Springer Berlin Heidelberg, 2005.

Claudia Archetti, Martin Savelsbergh, and Maria Grazia Speranza. The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254(2):472–480, 2016. ISSN 0377-2217.

Tolga Bektaş and Gilbert Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011. ISSN 0191-2615.

Yvonne M. Bontekoning, Cathy Macharis, and Jan Jacob Trip. Is a new applied transportation research field emerging?—A review of intermodal rail–truck freight transport literature. *Transportation Research Part A: Policy and Practice*, 38(1):1–34, January 2004.

Nathalie Bostel and Pierre Dejax. Models and algorithms for container allocation problems on trains in a rapid transshipment shunting yard. *Transportation science*, 32(4):370–379, 1998.

Nils Boysen and Simon Emde. The parallel stack loading problem to minimize blockages. *European Journal of Operational Research*, 249(2):618–627, 2016. ISSN 0377-2217.

Nils Boysen and Malte Fliedner. Determining crane areas in intermodal transshipment yards: The yard partition problem. *European Journal of Operational Research*, 204(2):336–342, July 2010.

Nils Boysen and Konrad Stephan. A survey on single crane scheduling in automated storage/retrieval systems. *European Journal of Operational Research*, 254(3):691–704, 2016. ISSN 0377-2217.

Nils Boysen, Malte Fliedner, and Michael Kellner. Determining fixed crane areas in rail–rail transshipment yards. *Transportation Research Part E: Logistics and Transportation Review*, 46 (6):1005–1016, November 2010.

Nils Boysen, Florian Jaehn, and Erwin Pesch. Scheduling Freight Trains in Rail-Rail Transshipment Yards. *Transportation Science*, 45(2):199–211, May 2011.

Nils Boysen, Malte Fliedner, Florian Jaehn, and Erwin Pesch. Shunting yard operations: Theoretical aspects and applications. *European Journal of Operational Research*, 220(1):1–14, 2012a.

Nils Boysen, Florian Jaehn, and Erwin Pesch. New bounds and algorithms for the transshipment yard scheduling problem. *Journal of Scheduling*, 15(4):499–511, November 2012b.

Nils Boysen, Malte Fliedner, Florian Jaehn, and Erwin Pesch. A survey on container processing in railway yards. *Transportation Science*, 47(3):312–329, 2013.

Edmund K. Burke and Yuri Bykov. The late acceptance hill-climbing heuristic. *European Journal of Operational Research*, 258(1):70–78, 2017. ISSN 0377-2217.

Ann Melissa Campbell and Martin Savelsbergh. Incentive Schemes for Attended Home Delivery Services. *Transportation Science*, 40(3):327–341, 2006.

Lu Chen and Zhiqiang Lu. The storage location assignment problem for outbound containers in a maritime terminal. *International Journal of Production Economics*, 135(1):73–80, 2012.

Jan Christiaens and Greet Vanden Berghe. A fresh ruin & recreate implementation for the capacitated vehicle routing problem. *KU Leuven technical report*, 2016.

Paul Corry and Erhan Kozan. Optimised loading patterns for intermodal trains. *OR Spectrum*, 30 (4):721–750, November 2008.

G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6): 791–812, 1958.

Elias Dahlhaus, Peter Horak, Mirka Miller, and Joseph F. Ryan. The train marshalling problem. *Discrete Applied Mathematics*, 103(1–3):41–54, 2000.

Maryam Darvish and Leandro C. Coelho. Sequential versus integrated optimization: Production, location, inventory control, and distribution. *European Journal of Operational Research*, 268 (1):203–214, 2018. ISSN 0377-2217.

Rommert Dekker, Jacqueline Bloemhof, and Ioannis Mallidis. Operations research for green logistics - an overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, 219(3):671–679, 2012. ISSN 0377-2217. Feature Clusters.

Emrah Demir, Tolga Bektaş, and Gilbert Laporte. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346–359, 2012. ISSN 0377-2217.

Emrah Demir, Tolga Bektaş, and Gilbert Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014. ISSN 0377-2217.

Guy Desaulniers, Fausto Errico, Stefan Irnich, and Michael Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405, 2016.

Ulrich Dorndorf and Frank Schneider. Scheduling automated triple cross-over stacking cranes in a container yard. *OR Spectrum*, 32(3):617–632, 2010.

J.H.R. (Ron) Van Duin, Lorant Tavasszy, and Eiichi Taniguchi. Real time simulation of auctioning and re-scheduling processes in hybrid freight markets. *Transportation Research Part B: Methodological*, 41(9):1050–1066, 2007. ISSN 0191-2615.

Jan Fabian Ehmke, Ann Melissa Campbell, and Barrett W. Thomas. Vehicle routing to minimize time-dependent emissions in urban areas. *European Journal of Operational Research*, 251(2): 478–494, 2016. ISSN 0377-2217.

Nagham M. El-Berishy and Bernd Scholz-Reiter. Development and implementation of a green logistics-oriented framework for batch process industries: two case studies. *Logistics Research*, 9(1):1–10, 2016. ISSN 18650368.

Laureano Escudero. An inexact algorithm for the sequential ordering problem. *European Journal of Operational Research*, 37(2):236–249, 1988.

European Commission. CO_2 electricity per kWh. http://www.eea.europa.eu/data-and-maps/figures/co2-electricity-g-per-kwh/co2-per-electricity-kwh-fig-1_2010_qa.xls, 2011.

Stefan Fedtke and Nils Boysen. Gantry crane and shuttle car scheduling in modern rail–rail transshipment yards. *OR Spectrum*, 39(2):473–503, Mar 2017. ISSN 1436-6304.

Amir Hossein Gharehgozli, Yugang Yu, René de Koster, and Jan Tijmen Udding. A decision-tree stacking heuristic minimising the expected number of reshuffles at a container terminal. *International Journal of Production Research*, 52(9):2592–2611, 2014.

Amir Hossein Gharehgozli, Gilbert Laporte, Yugang Yu, and René de Koster. Scheduling Twin Yard Cranes in a Container Block. *Transportation Science*, 49(3):686–705, 2015.

Amir Hossein Gharehgozli, Floris Gerardus Vernooij, and Nima Zaerpour. A simulation study of the performance of twin automated stacking cranes at a seaport container terminal. *European Journal of Operational Research*, 261(1):108–128, 2017. ISSN 0377-2217.

Dominik Goeke and Michael Schneider. Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1):81–99, 2015. ISSN 03772217.

José Ángel Gonzalez, Eva Ponce, Carlos Matix, and Javier Carrasco. The Automatic Generation of Transhipment Plans for a Train–Train Terminal: Application to the Spanish–French Border. *Transportation Planning and Technology*, 31(5):37–41, 2008.

José Fernando Gonçalves and Mauricio G.C. Resende. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525, 2011.

Ronny S. Hansmann and Uwe T. Zimmermann. Optimal sorting of rolling stock at hump yards. In Hans-Joachim Krebs and Willi Jäger, editors, *Mathematics – Key Technology for the Future*, pages 189–203. Springer Berlin Heidelberg, 2008.

Shiwei He, Rui Song, and Sohail S. Chaudhry. An integrated dispatching model for rail yards operations. *Computers & Operations Research*, 30(7):939–966, 2003.

Sam Heshmati, Zafeiris Kokkinogenis, Rosaldo J. F. Rossetti, Maria Antónia Carravilla, and José Fernando Oliveira. An agent-based approach to schedule crane operations in rail-rail transshipment terminals. In Raquel J. Fonseca, Gerhard-Wilhelm Weber, and João Telhada, editors, *Computational Management Science*, pages 91–97, Cham, 2016. Springer International Publishing. ISBN 978-3-319-20430-7.

Sam Heshmati, Jannes Verstichel, Eline Esprit, and Greet Vanden Berghe. Alternative e-commerce delivery policies. *EURO Journal on Transportation and Logistics*, 2018. ISSN 2192-4384.

Sam Heshmati, Túlio A.M. Toffolo, Wim Vancroonenburg, and Greet Vanden Berghe. Crane-operated warehouses: Integrating location assignment and crane scheduling. *Computers & Industrial Engineering*, 129:274–295, 2019. ISSN 0360-8352.

Yixiao Huang, Lei Zhao, Tom Van Woensel, and Jean-Philippe Gross. Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95 (Supplement C):169–195, 2017. ISSN 0191-2615.

Florian Jaehn. Positioning of loading units in a transshipment yard storage area. *OR Spectrum*, 35 (2):399–416, 2012. ISSN 0171-6468.

Raka Jovanovic and Stefan Voß. A chain heuristic for the blocks relocation problem. *Computers & Industrial Engineering*, 75(Supplement C):79–86, 2014. ISSN 0360-8352.

Michael Kellner, Nils Boysen, and Malte Fliedner. How to park freight trains on rail–rail transshipment yards: the train location problem. *OR Spectrum*, 34(3):535–561, 2012.

Graham Kendall, Ruibin Bai, Jacek Błazewicz, Patrick De Causmaecker, Michel Gendreau, Robert John, Jiawei Li, Barry McCollum, Erwin Pesch, Rong Qu, Nasser Sabar, Greet Vanden Berghe, and Angelina Yee. Good laboratory practice for optimization research. *Journal of the Operational Research Society*, 67(4):676–689, Apr 2016. ISSN 1476-9360. doi: 10.1057/jors.2015.77. URL https://doi.org/10.1057/jors.2015.77.

Çari Koç and Ismail Karaoglan. The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing Journal*, 39:154–164, 2016. ISSN 15684946.

Çağrı Koç, Tolga Bektaş, Ola Jabali, and Gilbert Laporte. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. *Computers & Operations Research*, 64:11–27, 2015. ISSN 0305-0548.

Dusan Ku and Tiru S. Arthanari. On the abstraction method for the container relocation problem. *Computers & Operations Research*, 68(Supplement C):110–122, 2016. ISSN 0305-0548.

Valeria Leggieri and Mohamed Haouari. A practical solution approach for the green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 104:97–112, 2017. ISSN 1366-5545.

Jan K. Lenstra and Alexander H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981. ISSN 1097-0037.

Feiyue Li, Bruce Golden, and Edward Wasil. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & Operations Research*, 34 (10):2918–2930, 2007.

Wenkai Li, Yong Wu, M.E.H. Petering, Mark Goh, and Robert De Souza. Discrete time model and algorithms for container yard crane scheduling. *European Journal of Operational Research*, 198(1):165–172, oct 2009. ISSN 03772217.

Wenkai Li, Mark Goh, Yong Wu, M.E.H. Petering, R. de Souza, and Y.C. Wu. A continuous time model for multiple yard crane scheduling with last minute job arrivals. *International Journal of Production Economics*, 136(2):332–343, apr 2012. ISSN 09255273.

Xiyu Li, Alena Otto, and Erwin Pesch. Solving the single crane scheduling problem at rail transshipment yards. *Discrete Applied Mathematics*, 264:134 – 147, 2019. ISSN 0166-218X. Combinatorial Optimization: between Practice and Theory.

Canhong Lin, K.L. Choy, G.T.S. Ho, S.H. Chung, and H.Y. Lam. Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, 41(4, Part 1):1118–1138, 2014. ISSN 0957-4174.

Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016. ISSN 2214-7160.

Cathy Macharis and Yvonne M. Bontekoning. Opportunities for OR in intermodal freight transport research: A review. *European Journal of Operational Research*, 153(2):400–416, March 2004.

Fernando Marín Martínez, Isabel García Gutiérrez, Alberto Ortiz Oliveira, and Luis Miguel Arreche Bedia. Gantry crane operations to transfer containers between trains: A simulation study of a Spanish terminal. *Transportation Planning and Technology*, 27(4):261–284, 2004.

Roberto Montemanni, Derek H. Smith, Andrea-Emilio Rizzoli, and Luca Maria Gambardella. Sequential ordering problems for crane scheduling in port terminals. *International Journal of Simulation and Process Modelling*, 5(4):348–361, 2009. doi: 10.1504/IJSPM.2009.032597.

Andrés Muñoz-Villamizar, Jairo R. Montoya-Torres, and Javier Faulin. Impact of the use of electric vehicles in collaborative urban transport networks: A case study. *Transportation Research Part D: Transport and Environment*, 50:40–54, 2017.

Yuichi Nagata, Olli Bräysy, and Wout Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Comput. Oper. Res.*, 37(4):724–737, April 2010. ISSN 0305-0548.

Jenny Nossack, Dirk Briskorn, and Erwin Pesch. Container dispatching and conflict-free yard crane routing in an automated container terminal. *Transportation Science*, 52(5):1059–1076, 2018. doi: 10.1287/trsc.2017.0811.

Commision of the European Commiunities. Communication from the commission to the council and the European parliament; Towards a rail network giving priority to freight. Technical report, European Commision, Brussels, 2007.

Taejin Park, Ri Choe, Young Hun Kim, and Kwang Ryel Ryu. Dynamic adjustment of container stacking policy in an automated container terminal. *International Journal of Production Economics*, 133(1):385–392, 2011.

David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007. ISSN 0305-0548.

Purba Rao and Diane Holt. Do green supply chains lead to competitiveness and economic performance? *International Journal of Operations & Production Management*, 25(9):898–916, 2005.

Roberto Roberti and Min Wen. The electric traveling salesman problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 89:32–52, 2016. ISSN 1366-5545.

Harald Rotter. New operating concepts for intermodal transport: The mega hub in Hanover/Lehrte in Germany. *Transportation Planning and Technology*, 27(5):347–365, October 2004.

Michael Schneider, Andreas Stenger, and Dominik Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520, 2014.

Gerhard Schrimpf, Johannes Schneider, Hermann Stamm-Wilbrandt, and Gunter Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2):139–171, 2000.

Jing Shao, Hangjun Yang, Xiaoqiang Xing, and Liu Yang. E-commerce and traffic congestion: An economic and policy analysis. *Transportation Research Part B: Methodological*, 83:91–103, 2016. ISSN 0191-2615.

Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, Oct 2016. ISSN 2327-4662.

Wouter Souffriau, Pieter Vansteenwegen, GreetVanden Berghe, and DirkVan Oudheusden. Variable neighbourhood descent for planning crane operations in a train terminal. In Kenneth Sörensen, Marc Sevaux, Walter Habenicht, and Martin Josef Geiger, editors, *Metaheuristics in the Service Industry*, volume 624 of *Lecture Notes in Economics and Mathematical Systems*, pages 83–98. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-00938-9.

Robert Stahlbock and Stefan Voß. Operations research at container terminals: a literature update. *OR Spectrum*, 30(1):1–52, 2008.

Dirk Steenken, Stefan Voß, and Robert Stahlbock. Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26(1):3–49, 2004. ISSN 1436-6304.

Ali Serdar Tasan and Mitsuo Gen. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering*, 62(3):755–761, 2012. ISSN 03608352.

Terminet-Consortium. Final Report for Publication. Report for DG VII of the European Commission. Technical Report January 2000, European Commission, Brussels, 2001.

Rodrigo Franco Toso and Mauricio G.C. Resende. A C++ application programming interface for biased random-key genetic algorithms. *Optimization Methods and Software*, pages 1–13, 2014.

Thibaut Vidal, Teodor Gabriel Crainic, M Gendreau, N. Lahrichi, and W. Rei. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Operations Research*, 60(3): 611–624, 2012. ISSN 0030-364X.

Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1):475–489, 2013. ISSN 03050548.

Tjark Vredeveld. Stochastic online scheduling. *Computer Science - Research and Development*, 27(3):181–187, Aug 2012. ISSN 1865-2042.

Bart W. Wiegmans, Daniëlle T. Stekelenburg, Corné Versteegt, and Yvonne M. Bontekoning. Modeling rail-rail exchange operations: an analysis of conventional and new-generation terminals. *Transportation Journal*, 46(4):5–20, 2007.

Yong Wu, Wenkai Li, Matthew E. H. Petering, Mark Goh, and Robert de Souza. Scheduling multiple yard cranes with crane interference and safety distance requirement. *Transportation Science*, 49(4):990–1005, 2015.

Canrong Zhang, Tao Wu, Ming Zhong, Li Zheng, and Lixin Miao. Location assignment for outbound containers with adjusted weight proportion. *Computers & Operations Research*, 52: 84–93, 2014. ISSN 0305-0548.

Çağrı Koç, Tolga Bektaş, Ola Jabali, and Gilbert Laporte. The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, 70:239–254, 2014. ISSN 0191-2615.