ARPAN RIJAL

Managing External Temporal Constraints in Manual Warehouses

Managing External Temporal Constraints in Manual Warehouses

Managing External Temporal Constraints in Manual Warehouses

Het coördineren van extern opgelegde tijdsrestricties in manuele magazijnen

Thesis

to obtain the degree of Doctor from

Erasmus University Rotterdam

by the command of

rector magnificus

Prof.dr. R.C.M.E. Engels

and in accordance with the decision of the Doctoral Board.

The public defence shall be held on

Friday 4 September 2020 at 13:30 hours

by

Arpan Rijal

born in Jabdi, Nepal

Erasmus University Rotterdam

Doctoral Committee

Promoter:	Prof. dr. M.B.M. de Koster
Other members:	Prof. dr. A. Goel Prof. dr. A. Wagelmans Dr. M. Schmidt
Co-promoter:	Dr. M. Bijvank

Erasmus Research Institute of Management – ERIM
The joint research institute of the Rotterdam School of Management (RSM)
and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam
Internet: http://www.erim.eur.nl

ERIM Electronic Series Portal: http://repub.eur.nl/pub/

ERIM PhD Series in Research in Management, 506
ERIM reference number: EPS-2020-506-LIS
ISBN 978-90-5892-585-5

Design: PanArt en advies, www.panart.nl

This publication (cover and interior) is printed by Tuijtel on recycled paper, BalanceSilk®. The ink used is produced from renewable resources and alcohol free fountain solution. Certifications for the paper and the printing production process: Recycle, EU Ecolabel, FSC®C007225. More info: www.tuijtel.com

Acknowledgments

Looking back, there are many, many individuals and circumstances that have led to this point. First, my supervisors Dr. Marco Bijvank and Prof. René de Koster have my gratitude for the opportunity to come to RSM and perform research under their guidance. Marco, thank you for your patience and support during the entire course of the thesis trajectory. René, your infectious enthusiasm and guidance were constant. I am very fortunate to have had you as my promotor.

Prof. Asvin Goel, your guidance was instrumental in my decision to do a PhD. Your mentorship has grown into co-authorship, and I hope the working relationship grows for years to come. I would like to express by gratitude to Prof. Martin Savelsbergh for hosting me during my research visit at Georgia Tech. I am also thankful to Prof. Albert Wagelmans, Prof. Nils Boysen, Prof. Asvin Goel, Dr. Marie Schmidt, Dr. Remy Spliet and Dr. Ioannis Fragkos for agreeing to be part of the dissertation committee.

Dr. Paul Buijs was extremely helpful in guiding me through the bureaucratic requirements of SLIFF project. I would also like to remember Pieter Meints, Boyd Gerrits and Benno Jaspers from Albert Heijn for their enthusiastic support of my PhD projects. It was a great pleasure to have worked with you, and without your feedback and time, the relevance of the thesis would not be the same.

The five years I spent within the TOM group were undoubtedly some of the best years of my life. I am certain that I will forever compare any other workplace to the standards of TOM. Seminars and social events held within the department were special. Carmen, Ingrid, Cheryl and Lianne, thank you for your help. The engaging and exciting discussion and conversations with Tim, Jelmer, Xishu, Jun, Erik, Johan, Joshua, Alberto, Massoud, Alexandros and Anna made the five years of PhD feel a lot shorter. Frankie, Alp, Joydeep and Kaveh, thank you for the trips, dinners and the unending discussions. I will always cherish our friendship.

Christoph Meyer has my eternal gratitude for helping me study at Jacobs University, and the opportunities I had there. Bimersha, Samip, Shashank, Freddy, Vlad, Anuraag, Abhinandan and Monita thanks for the everlasting memories. Laxmi, Sagun, Chiran, Pramit and Suvod, I cannot thank you enough for helping me in the most trying of times. Andrea and Thijs, your friendship has brought a lot of joy in our life.

Dijus, Rashmi, Valiusha, Mihai, Paula, Stefan, Lidia, Victoras, Liuba, Victor, and Maria, I am lucky to be able call you family. Ama, Bua, and Ashish, thank you for always supporting me even when the barriers seemed impossible to overcome. Much of what I have done would remain distant dream without your sacrifice and support. For that, I am forever grateful. Jenny, you are a rock in my life. Your positivity, compassion and unwavering support keeps driving me. I wish a lifetime of happiness together. Eva, you are very special, and thank you for changing my life so wonderfully.

Arpan Rijal
Breda, 2020

Contents

Acknowledgments i

1 Introduction 1

1.1 Warehouse Processes . . . 2

1.2 Warehouse Resources . . . 3

1.3 Warehouses in a Network . . . 6

1.4 Contribution and Thesis Outline . . . 7

2 Integrated Scheduling and Assignment of Trucks at Unit-Load Cross-Dock with Mixed Service Mode Dock Doors 15

2.1 Introduction . . . 15

2.2 Literature Review . . . 19

2.2.1 Cross-dock operations with mixed-mode dock doors . . . 19

2.2.2 Cross-dock scheduling . . . 20

2.3 Problem Description, Notations and Assumptions . . . 25

2.4 Model Formulation . . . 28

2.5 Integrated Solution Approach . . . 33

2.5.1 Representation of solution . . . 34

2.5.2 Neighbourhood operators . . . 36

2.6 Results . . . 36

2.6.1 Generation of instances . . . 38

2.6.2 Intergated versus sequential solutions, small instances . . . 41

2.6.3 Performance of ALNS algorithm for integrated problem . . . 44

2.6.4 Proportion and position of mixed-mode dock doors . . . 54

2.6.5 Case study . . . 57

2.7 Conclusion . . . 59

2.8 Appendix . . . 62

2.8.1 Bodnar et al. (2017) - revisited . . . 62

2.8.2 Model formulation for sequential approach . . . 64

2.8.3 Integrated ALNS algorithm . . . 68
2.8.4 Additional numerical results . . . 78
2.8.5 Additional results case study . . . 79
2.8.6 Computational experiments for I-shaped cross-dock . . . 85

3 Workforce Scheduling with Order-Picking Assignments in Distribution Facilities 89

3.1 Introduction . . . 89
3.2 Literature Review . . . 92
3.3 Problem Description and Model Formulation . . . 97
3.4 Branch and Price Algorithm for OPSP . . . 104
3.4.1 Reduced master problem . . . 104
3.4.2 Pricing problem . . . 105
3.4.3 Branching . . . 112
3.5 Metaheuristic for OPSP . . . 112
3.5.1 Savings algorithm . . . 112
3.5.2 Large neighborhood search for improved solutions . . . 114
3.6 Results . . . 117
3.6.1 Instances in numerical test bed . . . 117
3.6.2 Algorithmic performance . . . 119
3.6.3 Flexible shift structures: A case study . . . 123
3.7 Conclusion . . . 128
3.8 Appendix . . . 131
3.8.1 Illustrative example . . . 131
3.8.2 Proof of Proposition 3.1 . . . 132
3.8.3 Additional constraints to OPSP formulation . . . 134
3.8.4 Network flow formulation . . . 138
3.8.5 Extensions to OPSP formulation . . . 141
3.8.6 Pseudocode of labeling algorithm . . . 146
3.8.7 Proof of Proposition 3.2 . . . 146
3.8.8 Optimal schedule for one order picker . . . 148
3.8.9 Detailed result of subsection 3.6.2 . . . 151
3.8.10 Including different break time structures in solution procedures 152
3.8.11 Detailed result of subsection 3.6.3 . . . 169
3.8.12 Preemptive breaks . . . 169

4 Impact of Warehouse Capacities on Vehicle Routing 177
4.1 Introduction . . . 177
4.2 Literature Review . . . 180
4.2.1 Routing problems with time windows . . . 181
4.2.2 Capacitated warehouse planning . . . 182
4.2.3 Integrated production and distribution . . . 183
4.3 Problem Description and Model Formulation . . . 184
4.3.1 Problem description . . . 184
4.3.2 Model formulation . . . 188
4.4 Solution Procedure . . . 194
4.4.1 Augmented graph . . . 196
4.4.2 Labels . . . 197
4.4.3 Resource extension functions . . . 198
4.4.4 Acceleration techniques . . . 203
4.5 Case study . . . 204
4.5.1 Generation of instances . . . 205
4.5.2 Performance of DP algorithm . . . 211
4.5.3 Impact of limited warehousing resources . . . 214
4.5.4 Managerial Insights . . . 215
4.6 Conclusion . . . 218
4.7 Appendix . . . 221
4.7.1 Pseudocode of dynamic programming algorithm . . . 221
4.7.2 Proof of Proposition 4.1 . . . 222
4.7.3 Determining the minimum number of order pickers in each shift 223
4.7.4 Detailed result for Figure 4.11 . . . 225

5 Conclusions and future outlook 227
5.1 Conclusions . . . 228
5.2 Future outlook . . . 231

Bibliography 233

About the author 251

Portfolio 253

Summary 255

Samenvatting (Summary in Dutch) **259**

ERIM Ph.D. Series Research in Management **263**

1 Introduction

Warehouses are intermediate facilities that store and process products between the location of their production and their consumption (Bartholdi & Hackman, 2019). In the broadest sense of the definition, the oldest form of warehouses known to us are granaries at D′hra modern day Jordan built in the pre-pottery Neolithic era approximately 11,000 years ago (Kuijt & Finlayson, 2009). These communal granaries were simple mud and stone constructions designed to store foraged food products. Since then, warehouses have evolved with the changing needs of humanity. In the second century B.C., large storage buildings for grains, olive oil, wine, paper and marble called *horrea* were built across Roman cities and ports to support city dwellers and militaries and to facilitate trade (Rickman, 2002). In the present day, we find examples of automated warehouses ready to dispatch orders at a click of a button to customers (Azadeh et al., 2019). These examples of warehouses from different periods in human history illustrate how warehouses have always evolved to fulfill the changing needs of humans.

Warehouses contribute to consumer welfare in several ways. Fundamentally, by serving as a place of storage, they provide a buffer between production and consumption of products and help absorb uncertainties in demand and supply (Lambert et al., 1998). In doing so, warehouses are able to bridge the time and space between the moment and location of production and consumption in a cost effective manner. In manufacturing, a large quantity of products can be manufactured in batches to realize production economies of scale. Similarly, in the agricultural industry, yield is excessive for immediate consumption and therefore has to be stored. Retailers procure products in large quantities to obtain volume discounts. In all of these settings, a warehouse provides a safe place for storage of excessive product quantities before demand is realized. Second, warehouses contribute to efficient transportation through better utilization of vehicles from suppliers and to customers in comparison to direct deliveries from suppliers to customers. At the same time, customers benefit by having to process fewer vehicles and deliveries compared to multiple direct deliveries

(Lambert et al., 1998). An extreme example of transportation economies of scale is achieved with cross-dock facility, a special warehouse where items are rapidly consolidated and shipped from the warehouse with minimal storage within the facility (Bartholdi III & Gue, 2000). Third, warehouses can also provide value added services such as postponed assembly, labeling and packaging (Heragu et al., 2005). Finally, warehouses can play an increasingly vital role in managing product returns especially with the rise of e-commerce activities (De Koster et al., 2007).

1.1 Warehouse Processes

Regardless of the form and functions of warehouses, warehouse processes can be divided into two broad categories of inbound and outbound processes (Bartholdi & Hackman, 2019). The storage function separates the linear chain of warehouse processes into inbound and outbound processes where every process before storage belongs to the inbound category, and the processes following storage are classified as outbound (see Figure 1.1).

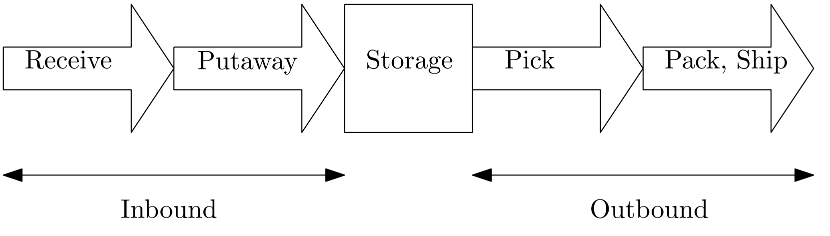


Figure 1.1: Warehouse processes adopted from Bartholdi & Hackman (2019)

The *inbound processes* start with receiving and processing shipments from suppliers or upstream facilities through warehouse dock doors. Shipments are unloaded and checked before they are put away in storage locations. Additionally, products returned from the customer can also be processed in the receiving area. Almost always, the form in which products are received and in which they leave the warehouse changes, except for cross-dock facilities (De Koster et al., 2007). For example, products may be received by the warehouse in pallets of identical products, but they may leave the warehouse as individual cases or pallets containing cases of multiple products. The problem of changing form is solved by the use of reserve and forward storage areas (Tompkins et al., 2010). *Forward storage area* is a part of the warehouse where products are stored in smaller forms such as cases or even individual units of products (Bartholdi & Hackman, 2008). This area is characterized by its convenient location for order picking, explained later. A *reserve storage area* is used to store products in relatively larger and more dense forms such as pallets and can be found in less accessible locations of the warehouse (Bartholdi & Hackman, 2008). When the forward storage area needs replenishment, they can be replenished internally from the reserve storage area or directly from the receiving area. The use of both reserve and forward storage helps to realize higher efficiency in the usage of storage spaces while maintaining easy accessibility of products in the forward storage locations.

The *outbound process* starts with order picking which is defined as the process of retrieving products from storage locations. After order picking, products are accumulated, sorted and packed before they are loaded onto vehicles and dispatched (Tompkins et al., 2010). When order sizes are small, for example in B2C e-commerce, an individual order picker can pick the entire customer order making accumulation and sorting of items redundant (Moons et al., 2017). However, when individual customer orders are larger and require picking by multiple order pickers, the picked items have to be accumulated and sorted first to ensure that the correct products are sent to the right customers (Gallien & Weber, 2010). The accumulation and sorting processes also provide a buffer time between the order picking and loading of vehicles, and they reduce the need to synchronize order picking activities with the loading of outbound vehicles. In the outbound process, value added services such as late customization, labeling and packaging of products may be included. Finally, vehicles are loaded with the products at the dock doors and dispatched from the warehouse.

1.2 Warehouse Resources

Management of warehouse processes in its essence involves decisions on two fundamental warehouse resources: space and time (labor hours) (Bartholdi & Hackman, 2019). Decisions on the usage of physical space in the form of shape, size, layout, and the type of storage systems used in the warehouses belong to the spatial decisions. On the other hand, temporal resources determine the usage of personnel and equipment to complete warehouse processes in a timely manner.

The usage of *spatial resources* is important for efficient warehouse management. Warehouses are usually built in industrial areas with close proximity to highways and infrastructure for other modes of transport (such as railways or a port). The land and building cost of a warehouse can be substantial (Gose, 2018). Once a warehouse is built, the shape, size and layout of the warehouse have direct a impact on temporal decisions and the time required to complete warehouse processes (Rouwenhorst et al., 2000). For example, the shape of cross-dock facilities (Bartholdi & Gue, 2004) and the layout in which dock doors are arranged (Gue, 1999) determine travel distances between dock doors and the labor cost to operate cross-dock facilities. The number of available dock doors and the modes in which they are operated (as inbound, outbound or mixed) have an impact on the timeliness of loading and unloading of trucks (Bodnar et al., 2017).

The utilization of *time resources* (such as labor hours) can be considered as a proxy for the cost of operating a warehouse because the amount of time used by the warehouse processes directly implies usage of valuable resources in the form of either people or equipment (Bartholdi & Hackman, 2019). In more automated settings, higher needs for time resources translate to larger needs of equipment and eventually a higher capital investment (Roodbergen & Vis, 2009). When employing humans, the utilization of time for any warehouse process is a direct cost driver (De Koster et al., 2007).

Minimizing the use of both the space and time resources contributes to efficient warehouse management. However, decisions on how space and time are utilized should not be made in isolation as they directly impact each other. When designing and controlling warehouses, this relationship has to be carefully evaluated.

Warehouse management decisions can also be classified on the scale of their time horizon and their demand on capital investments into strategic, tactical and operational decisions (Rouwenhorst et al., 2000). The *strategic decisions* are long-term decisions which have a substantial impact on capital investments. *Tactical decisions* are medium-term decisions which are made more frequently compared to strategic decisions and have comparatively a smaller impact on capital costs. *Operational decisions* are short-term decisions that are made frequently, some even multiple times in a day. Comparatively, operational decisions are less capital intensive but are crucial in maintaining high service levels.

Strategic decisions starts with the decision on usage of space such as location, shape and size of the warehouse. The number of products the warehouse needs to store with a given demand profile and the desired level of lead time and responsiveness determine the location (Baumol & Wolfe, 1958), size and shape of the warehouse (Gu et al., 2007). Another important strategic decision is related to the choice of storage and handling systems - the combination of storage and retrieval equipment and the methods of their usage (Dallari et al., 2009). Warehouses store a large variety of products with different storage requirements and serve multiple customer types with different demand requirements. To accommodate different storage needs of different products, warehouses use multiple warehousing systems simultaneously. The large variety of the available equipment and the methods to use them make it difficult for warehouse managers to choose appropriate warehousing systems for the strategic needs of any company.

In a next step, the strategic decisions extend to tactical decisions such as the allocation of space for different warehouse processes and associated systems (Heragu et al., 2005). Spatial decisions such as the amount of space allocated to forward versus reserve storage (Bartholdi & Hackman, 2008), usage of dedicated, random or cluster-based storage policy (De Koster et al., 2007) or the mode of dock door operation (Buijs et al., 2014) can be categorized as tactical decisions. Temporally, decisions on the type of shifts for order pickers or the number of order picking robots to use can also be considered as tactical decisions. Interestingly, given the warehousing contexts, several decisions on the tactical level can migrate to the strategic level, or even the operational level. For example, the number of robots in automated storage and retrieval systems (AS/RS) cannot be changed with ease once the warehouse has been built (Roodbergen & Vis, 2009). Therefore, the decisions on the number of AS/RS machines belongs to the strategic level. However, in parts-to-picker robotic systems such as Autostore (Azadeh et al., 2019), the number of robots used for order picking can be changed annually, monthly or even on an hourly basis.

Strategic and tactical decisions on the usage of space and time resources impose constraints on the operational decisions. Using the limited temporal and spatial resources determined on the strategic and tactical levels, warehouse managers have to ensure that the warehouse is able to satisfy customer requirements. Decisions such as batching of multiple orders (Gademann et al., 2001), scheduling of trucks at dock doors (Ladier & Alpan, 2016), routing of order pickers in the warehouse (Scholz et al., 2017) and assignment of orders to order pickers (Matusiak et al., 2017) belong to the operational level. Generally, these operational decisions are geared towards the objective of minimizing the usage of time resources (Rouwenhorst et al., 2000) while achieving high customer service requirements. By minimizing the time required to complete warehouse processes, the labor requirements of warehouses can be reduced. Additionally, minimizing the time needed to complete warehouse processes contributes even further to make the entire supply chain more efficient and responsive.

1.3 Warehouses in a Network

Warehouses are not isolated entities but part of a larger supply network with multiple stakeholders and varying demands and constraints on the warehouses (Rouwenhorst et al., 2000; Buijs et al., 2014). External to the warehouse other stakeholders (such as suppliers, other warehouses, transportation service providers, governmental organizations, insurance companies, fire policies and customers) impose restrictions that define the operating boundaries for a warehouse. These restrictions shape the manner in which warehouses should manage their usage of space and time resources.

Upstream from the warehouses, suppliers can impose constraints on the warehouses. By mandating the time window in which their vehicles have to be processed, suppliers dictate the number of dock doors (spatial) and personnel (temporal) required to ensure that vehicles are processed in time (Bodnar et al., 2017). The form in which shipments are received (in either pallets or boxes) determine the type of unloading equipment that can be used at the warehouse (Gu et al., 2007). If suppliers send shipments in mixed pallets with multiple products on one pallet, these products typically need to be sorted and restacked on single-SKU pallets before they are stored. The warehouse needs space for staging and sorting these products. Such restrictions from upstream facilities dictate the organization of inbound warehouse processes.

The stakeholders *downstream* of the warehouse (i.e., the customers) are retail stores, end-consumers or commercial facilities such as factories or cross-dock facilities. These entities have a diverse set of requirements that are imposed on the warehouse and determine how outbound warehouse processes are planned and configured. In B2C e-commerce, customers can more frequently indicate during what time window they want their online orders to be delivered to their homes (Agatz et al., 2011). Similarly, individual retail stores have their own time windows when warehouses can deliver and unload products to the stores. These time windows can be self imposed by retail store managers to ease the unloading of vehicles and shelving products in the stores (Spliet et al., 2018). Additionally, many Western European city governments impose time windows on when commercial vehicles can enter the city to reduce traffic congestion, improve safety and maximize welfare of city dwellers (Muñuzuri et al., 2005; Quak & de Koster, 2007). In all these contexts, the time windows and deadlines at downstream facilities temporally constrain the management of outbound processes at the warehouse. Additionally, the form in which products leave the warehouse dictates the type of loading equipment and personnel used at the warehouse. When the same warehouse processes different forms of outbound shipments such as pallets to downstream cross-dock facilities and cases to online shoppers, the warehouse has to employ more than one type of loading equipment. Furthermore, if the number of dock doors is limited (i.e., there are spatial constraints), and a large number of deliveries has to depart in quick succession, this requires buffer spaces (called staging areas) to hold products between the order picking and loading operations. Staging areas can be limited in space and themselves might need to be managed.

1.4 Contribution and Thesis Outline

In this dissertation, the focus is on *manual* warehouse operations from a network perspective where warehouse operations are constrained not only by the limited internal resources but also by external *temporal* constraints from upstream and downstream facilities. Specifically, mathematical models and algorithms are developed for the planning of warehouse processes that use manual labor, have temporal constraints and are constrained internally by limited personnel and space. The warehouse setting considered in this dissertation is illustrated in Figure 1.2.

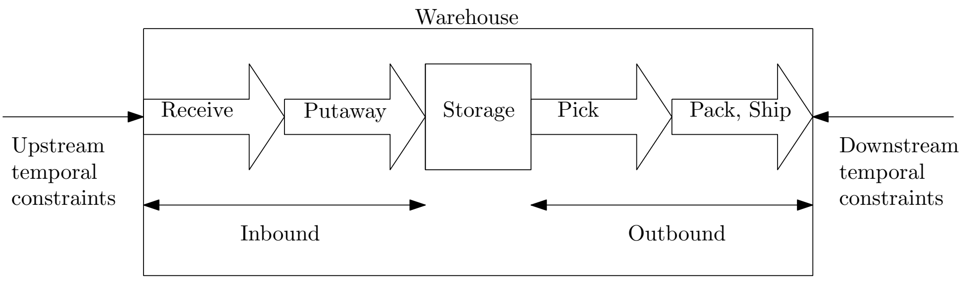


Figure 1.2: Warehouse setting and constraints considered in this thesis

The focus on manual warehouses is justified for several reasons. Even though automated warehouse solutions are being developed and adopted increasingly, most of the warehouse operations are still performed manually. De Koster et al. (2007) estimated that manual order picking systems, which make up the largest share of warehousing costs, made up 80 percent of the order picking systems in Western Europe. A more recent industry survey suggests that only 13 percent of warehouses have some level of automation (Michel, 2016). Usage of manual workers does not require large capital investments compared to automation, and manual operations can be scaled more flexibly compared to most automated warehousing solutions. However, the use of human workers in a warehouse has its own unique challenges. Warehouses with human workers have to abide by strict labor laws and union agreements which dictate the number of work shifts, lengths of shifts and work break periods (European Parliament, Council of the European Union, 2003). In automated warehouse settings, fewer workers are needed for critical operations, and planning of these personnel is less challenging. Despite the importance of humans in warehouses, the warehousing literature overlooks constraints that are specific to manual operations, namely, varying the number of personnel and capacity available at different times of the day (Henn et al., 2010; Henn, 2012; Scholz et al., 2017). This dissertation aims to fill this gap.

The network perspective on warehouse facilities also has its contribution. The warehouse management literature largely assumes that transportation planning is secondary and it focuses on the optimization of internal warehouse objectives such as minimizing the total order processing time (Gademann et al., 2001; Gademann & Velde, 2005; Matusiak et al., 2017) or minimizing late orders (Henn, 2015; Scholz et al., 2017). Alternatively, the transportation literature largely does not consider constraints at warehouses and optimizes purely transportation objectives such as minimizing travel distance (Solomon, 1987) or minimizing travel duration (Savelsbergh, 1992) of vehicles. As mentioned earlier, temporal constraints in warehouses originate from many sources and can have a significant impact on warehouse operations. The temporal constraints from upstream facilities can translate to processing time windows for inbound trucks at the warehouse. Departure deadlines or time windows for outbound orders may be necessary to ensure that orders are delivered to customers in time. If warehouses have an abundance of resources such as dock doors, staging space and sufficient order pickers, the temporal constraints can be ignored and warehouses can be planned as isolated entities. However, in practice, warehouses, especially in Western Europe, are severely constrained in the amount of space they can use for processing trucks or for staging orders. These factors necessitate efficient warehouse management to consider management of temporal and spatial resources when constrained by external temporal constraints.

The guiding research question for this thesis is: *How to efficiently manage resources in manual warehouses with external temporal constraints?* We conduct three studies to answer this research question. The first study investigates cross-dock operations where the inbound and outbound shipments have time windows and shipments are moved between the dock doors by workers driving fork lift trucks. In the second study, the temporal constraints at the warehouse appear in the form of *delivery due time windows* which dictate the duration within which an order can be picked and delivered to the staging area of a warehouse. The planning problem in this study has to determine the workforce schedule for order pickers to ensure all orders are picked and delivered within their time windows. In the final study of this thesis, the temporal constraints appear in the form of *time windows* at retail stores. This study goes beyond considering the temporal constraints as inputs for warehouse planning and optimizes the transportation routing and warehouse processes jointly when a warehouse has limited order picking, staging and loading capacity.

In the following, the individual studies that comprise this thesis are explained. A classification of the temporal constraints imposed on the warehouse and the internal resources considered in the studies is presented in Table 1.1.

Table 1.1: Classification of temporal constraints and warehouse resources studied in this thesis

	Warehouse resources		**External**
Study	**Spatial**	**Temporal**	**temporal constraints**
1	Dock doors, temporary storage area	Cross-dock workers	Hard inbound time windows, soft outbound time windows
2	Staging lanes	Flexible order pickers	Delivery due time windows
3	Staging area, dock doors	Order pickers, loaders	Customer time windows

Chapter 2: Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors

This chapter investigates the planning problem of assigning and scheduling trucks to dock doors in unit load cross-dock facilities with mixed-mode dock doors, i.e., dock doors which can process both inbound and outbound trucks. The cross-dock facility has limited spatial resources in the form of dock doors (Van Belle et al., 2012; Ladier & Alpan, 2016). The cross-dock operations are also constrained temporally by inbound and outbound time windows (Bodnar et al., 2017). Inbound shipments have to be processed within the predefined time window. Outbound trucks have soft time windows and can depart from the cross-dock facility with a delay but at penalty cost. The objective of the problem is to synchronize inbound and outbound vehicles in both space and time dimension as much as possible. When inbound trucks are stationed at the dock doors at the same time as when the corresponding outbound trucks are stationed at other dock doors, the shipments can be directly transferred from inbound trucks to outbound truck without being stored in a temporary storage area. Avoiding the usage of the temporary storage area minimizes extra travel and handling for workers (Bodnar et al., 2017). Additionally, when shipments are transferred directly from inbound to outbound vehicles, the aim is to process trucks at dock doors in close spatial proximity to minimize travel distances which we consider as a direct proxy for the operational costs of a cross-dock operations. Since both the dock-door assignment (Bartholdi III & Gue, 2000; Bozer & Carlo, 2008) and truck scheduling problems (Boysen et al., 2010; Bodnar et al., 2017) are NP-hard problems, the literature and practice take either a sequential solution approach to the individual problems (where trucks are scheduled first and assigned to dock doors second) or the problems are integrated for only the inbound or outbound operations. Our study shows that the sequential approach is not only suboptimal but can also lead to infeasible solutions that cannot be realized in practice. An integrated approach has the potential to remedy these shortcomings of the sequential approaches.

The research question addressed in this study is: *How to schedule and assign trucks to dock doors in cross-dock facilities with mixed-mode dock doors in an integrated manner?* To answer the question, we present a mathematical model for the problem and an adaptive large neighborhood search algorithm to solve the problem for real-sized instances. Extensive computational experiments suggest that scheduling and assigning trucks to dock doors in an integrated manner is superior compared to the sequential approach of scheduling first and assigning second by as much as 20-30%. Additionally, results suggest that both the proportion and position of mixed-mode dock doors have an impact on the temporal cost (i.e., delay of outbound trucks) and the spatial cost (i.e., travel distance of shipments between dock doors).

Chapter 3: Workforce scheduling with order-picking assignments in distribution facilities

The second study in this thesis investigates the scheduling of manual order pickers in warehouses where order picking operations are constrained temporally by predefined time windows for delivery of orders to the staging area of the warehouse. The staging area of the warehouse is organized in staging lanes, and it is limited in size compared to the volume of daily outbound orders. To ensure that orders arrive in time and the staging lanes are not overfilled with orders, each vehicle is allocated a due time window, and orders for a vehicle have to be picked and delivered at the right staging lane within its due time window. The arrival of orders before the beginning of the due time window is not possible because at that time the space is allocated for the orders for a different truck. The arrival of orders after the end of the due time window is forbidden as they will delay the delivery of orders at customer locations.

In this complex warehouse environment, the number of orders required to be picked at different times of the day can vary drastically. A large number of order pickers is required when many orders need to be picked in quick succession. Some of these order pickers may not be required soon after. Warehouse managers use *flexible order pickers* to solve the problem of a varying workforce requirement. These order pickers can be called on short notice and employed and paid for a variable duration of work. Planning this workforce requires warehouse managers to determine the number of order pickers to employ, the start and end times of their shifts, the length of shift duration and the break periods to minimize the duration during which flexible order pickers are employed. Furthermore, these decisions have to be made jointly with the assignment of orders to order pickers and their sequencing to ensure that the orders are picked and delivered within their due time windows.

Because of the interaction of workforce scheduling and order assignment and scheduling problems, the overall planning problem is a complex NP-hard problem. In practice, managers use their experience and intuition to determine the required workforce, and orders are scheduled and assigned to workers later on. The studies in the warehousing literature that investigate order picking with temporal constraints such as deadlines (Elsayed et al., 1993; Elsayed & Lee, 1996; Henn, 2015; Scholz et al., 2017) can only be applied to automated environments because they overlook shift related constraints which are relevant when humans are employed. Personnel scheduling literature is rich and several available studies provide models and solution approaches for planning of personnel with varying workloads (Goel & Irnich, 2017; Cordeau et al., 2010). However, the available studies cannot be used directly to optimize the use of *flexible workers* and assign orders. This study contributes to the warehousing literature by combining the richness of personnel scheduling problems when humans are employed with the temporal constraints of planning order pickers in warehouses.

The research question in this study is: *How can a flexible order picker workforce schedule be generated while assigning orders with due time windows to order pickers?*

The study presents two mathematical models for the problem and proposes two solution approaches - an exact solution approach based on a branch-and-price solution framework and a large neighborhood search algorithm. Furthermore, extensive experiments show that the solution approaches developed in this chapter can produce quality solutions for real-sized instances. The large neighborhood search algorithm is used in a case study to plan the order picker workforce of the warehouse operation of a Dutch retailer. The case study suggests that using shift structures with flexible breaks for order pickers, i.e., breaks that can start 15 minutes before or after the current break start time, can lead to savings of as much as 5% of the labor cost.

Chapter 4: Vehicle routing with warehouse considerations under time restrictions

This chapter investigates joint optimization of warehouse processes with limited resources and transportation planning in a holistic manner. In practice, transportation planning is generally done in advance, which forms the input for warehouse planning. However, it can be impossible or prohibitively expensive for a warehouse to ensure that vehicles can dispatch from the warehouse as needed by the transportation plans. There are several examples in industries where departure deadlines of vehicles imposed on the warehouse without consideration for the limited resources at the warehouse cannot be realized. A solution to this problem lies in holistically creating delivery routes while considering the limited resources at the warehouse. Dabia et al. (2019) investigate the impact of limited loading capacity on route planning and Moons et al. (2017, 2019) propose a problem of integrating order picking with routing. However, these available academic works consider only one warehouse process when generating routes which limits their applicability to warehouses where multiple constrained processes impact each other.

This chapter studies the interaction of three most important warehouse processes - order picking, staging and loading - and their impact on the routing of vehicles to customers with hard time windows. At different periods of the day, the warehouse has varying levels of limited order picking capacity because of the change in the shifts of order pickers and breaks. Additionally, the staging area and the number of dock doors are also constrained. Using the limited order picking, staging and loading capacity in the warehouse, orders have to be dispatched from the warehouse in time to ensure that they are delivered at the customer locations within the customer-specific time windows.

The research question addressed in this study is: *How to jointly plan transport and warehouse operations for warehouses with limited order picking, staging and loading capacity when deliveries have time windows?*

The study presents a mathematical model for the problem which minimizes the duration required to deliver orders to customers with time windows. The departure time of vehicles from the warehouse is constrained by a limited number of dock doors, staging space and order picking capacity. A solution approach for the problem is developed based on a dynamic programming approach. Experiments are conducted on instances generated from a data set of a Dutch retailer which suggest that constrained warehouse resources can have a significant impact on the route planning. When the size of the staging area and the availability of the order picking workforce is limited, the duration of routes can increase on average by 8.4% compared to the cases where routes are not constrained by warehouse resources. However, the best system wide performance is achieved when the fewest possible order pickers are used, because any savings that can be gained in routing costs is dominated by the cost of employing additional order pickers necessary to realize the departure times of optimal routes.

Research statement

This dissertation was written during my work at Erasmus University Rotterdam as a PhD candidate. The chapters in the thesis are self-contained papers written independently under the supervision of the doctoral advisors.

Chapter 2: This chapter is based on Rijal et al. (2019) published in *European Journal of Operational Research*. For this work, I collaborated with my copromotor Dr. Marco Bijvank and my promotor Prof. dr. René de Koster. Industry data used in this paper was provided by a large Dutch grocery retailer.

Chapter 3: I collaborated with my copromotor dr. Marco Bijvank, my promotor Prof. dr. René de Koster and Prof. dr. Asvin Goel for the study in this chapter. This chapter is based on a working paper which is being in second round of review in a journal at the time of writing. Data provided by a larger Dutch grocery retailer was used for the case study in this work.

Chapter 4: This chapter is based on a working paper being prepared for submission at the time of writing. For this work, I worked with my copromotor Dr. Marco Bijvank and my promotor Prof. dr. René de Koster. Data provided by a large Dutch grocery retailer was used to generate problem instances for the experiment.

2 Integrated Scheduling and Assignment of Trucks at Unit-Load Cross-Dock Terminals with Mixed Service Mode Dock Doors[1]

2.1 Introduction

Cross-docking is a material handling and distribution concept in which products received at a terminal are immediately unloaded from inbound trucks, sorted and consolidated based on their destinations, and loaded directly into outbound trucks for delivery to customers with little or no storage in between. Incoming shipments typically spend up to 24 hours inside a cross-dock terminal, while often not more than one hour (Bartholdi & Gue, 2004). Consequently, cross-docking has the potential to eliminate the storage and retrieval operation functions of a traditional warehouse. As it provides the best of the warehousing and direct delivery strategy, cross-docking has become a popular distribution strategy in practice. In a recent industry survey among 219 logistics, transportation, warehousing, and supply chain management practitioners, 68.5% of the survey respondents used cross-docking and 15.1% of the respondents are planning to use cross-dock terminals within two years (Saddle Creek Corporation, 2011).

A dispatcher of a cross-dock terminal faces two interrelated decisions: *where* and *when* the trucks should be processed at the dock doors of the terminal. In the literature, the location decision is called the *dock-door* (or *truck-to-door*) *assignment* problem and the timing decision is called the *truck scheduling* problem. Minimizing

[1]This chapter is based on Rijal, A., Bijvank, M., & de Koster, R. (2019). Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors. *European Journal of Operational Research*, 278(3), 752-771.

total travel distances within the terminal is the main objective of the dock-door assignment problem, whereas minimizing the total length of the planning horizon (i.e., the makespan) and any departure delays of outbound trucks (i.e., the tardiness) are the main objectives of the truck scheduling problem in the literature. When both decisions are considered simultaneously, this is called the *cross-dock scheduling* problem. As the truck scheduling problem and the dock-door assignment problem are both NP-hard, most literature focuses on the development of sequential approaches to solve these problems where trucks are usually scheduled first and then assigned to dock doors.

In most cross-dock terminals, an incoming truck is assigned to an *inbound door* (also called *receiving* or *strip door*) where the freight is unloaded and an outgoing truck is assigned to an *outbound door* (also called *shipping* or *stack door*) where the shipment is loaded to serve the customer destination. Boysen & Fliedner (2010) introduced the notion of a *mixed (or flexible) service mode* for dock doors, where both inbound and outbound trucks can be processed. More recently, Berghman et al. (2015) and Bodnar et al. (2017) propose solution techniques to include mixed-mode dock doors in truck scheduling problems at a cross-dock terminal. Both studies show numerically that including such flexible dock doors can reduce their objective function value (waiting time and operational costs, respectively) by as much as 18%. In an industry study by Ladier & Alpan (2016) there were five out of nine cross-dock terminals that indicated to use mixed-mode dock doors. This illustrates the relevance to study the use of mixed-mode dock doors.

The goal of our paper is to extend Bodnar et al. (2017) by including dock-door assignments to the truck scheduling problem when dock doors can operate in a mixed mode. The only two papers that study cross-dock scheduling with mixed-mode service doors are Shakeri et al. (2012) and Hermel et al. (2016). However, these authors restrict the service mode of each flexible dock door to be dedicated for either inbound or outbound operations at a particular time in the planning horizon (see Section 2.2 for more details). We use the mixed service mode in a broader sense, since both inbound and outbound trucks can be assigned to any flexible door that is available over the entire planning horizon.

The integration of the truck scheduling and dock-door assignment problem is non-trivial. In the literature, it is usually suggested to solve both problems sequentially (such as first-schedule-then-assign). The performance of such a sequential approach (compared to an integrated approach) depends on the size of the cross-dock terminal (measured as the number of dock doors) and the utilization of the dock doors (measured as the truck-to-dock-door ratio). The synchronization of the inbound and appropriate outbound trucks must be carefully scheduled over time when dock doors are heavily utilized. Furthermore, in cross-dock terminals where the distances between inbound and outbound trucks are substantial, the dispatcher needs to assign inbound trucks to receiving doors close to the designated outbound trucks. Reducing travel distance becomes then more important to improve the efficiency of the cross-dock operations. A sequential approach does not ensure that a dock door is available such that incoming shipments are assigned to a door that is close to the appropriate destination trucks to reduce travel distances without delaying the departure of outbound trucks.

Observations from practice reveal that cross-dock terminals can have extended time periods during which the dock-door utilization is 80% or higher (see Section 2.6.5 for details of our case study). As discussed above, this requires an integrated approach to solve the truck scheduling and dock-door assignment problem simultaneously. Furthermore, mixed-mode dock doors are more beneficial when dock doors are heavily utilized (Bodnar et al., 2017). However, with a higher utilization and mixed-mode dock doors, sequential approaches can lead to infeasible solutions (see Section 2.8.1). This strengthens our motivation to study cross-dock scheduling with mixed-mode dock doors. The research questions for this study are: (i) How can the integrated cross-dock scheduling problem be formulated and solved when some doors are exclusively dedicated to inbound or outbound operations and other doors are used in a mixed mode? (ii) To what extent is the integrated cross-dock scheduling approach better than the sequential approach? (iii) Does the ratio of the mixed-mode dock doors to dock doors with an exclusive service mode as well as the positioning of the mixed-mode dock doors have an impact on the operational performance of the cross-dock terminal? Ideally, all dock doors operate in a mixed service mode. However, in practice, it is preferred that not all dock doors have this flexibility since this service mode usually incurs additional operational overhead. Furthermore, mixed-mode dock doors can create confusion for employees working at the cross-dock facility since it might not always be clear whether a dock door is prepared for inbound or outbound operations. Such confusion does not exist for dock doors that operate in a dedicated service mode. Technological solutions that indicate, in real time, the mode of the dock door can minimize confusion among employees and are already in existence. We are aware of companies that started a pilot with such technologies to assist the material handling activities at dock doors. Operating a limited number of dock doors in the mixed service mode limits this operational overhead (Berghman et al., 2015). Furthermore, Bodnar et al. (2017) show that it is sufficient to convert only a fraction of the dock doors to a mixed service mode to gain the full benefits of mixed-mode dock doors (i.e., the operational benefit of additional dock doors operating in a mixed service mode is diminishing). However, the positioning of these mixed-mode dock doors can also have an impact on the cross-dock operations (besides their proportion). No previous literature has studied this aspect.

Our contributions to the existing literature are threefold. Firstly, we extend the formulation of the integrated scheduling and assignment problem to include mixed-mode dock doors. Even though the integration of the two problems has been recognized in the literature to generate operational efficiencies to cross-dock facilities, the use of dock doors with a mixed service mode has only been studied for either of the two problems in isolation and never in an integrated fashion. The integration of temporal and spatial decisions is a non-trivial extension and presents significant modeling challenges. In particular, we illustrate in Section 2.8.1 that the schedule resulting from the model proposed by Bodnar et al. (2017) can result in infeasible dock-door assignments. In addition to the integrated cross-dock scheduling model, we revise the formulation of the truck scheduling problem proposed by Bodnar et al. (2017) in Section 2.8.2 such that a feasible dock-door assignment can be found. This is used as one of the benchmark techniques in our numerical results. Secondly, we extend the adaptive large neighborhood search (ALNS) algorithm as proposed by Bodnar et al. (2017) to include dock-door assignment decisions, which requires a new solution representation and new neighbourhood operators. Thirdly, we perform rigorous numerical experiments and a case study which demonstrate that an integrated approach to the cross-dock scheduling problem with mixed-mode dock doors results in superior decision making compared to a sequential approach. In particular, we study different scenarios where we vary the dock-door utilization, the proportion of mixed-mode dock doors and the positioning of these dock doors.

The remainder of this paper is organized as follows. In the next section, we review related literature. In Section 2.3, we describe the integrated scheduling and assignment problem in more detail and introduce all notations. In Section 2.4, we develop a mathematical model for the cross-dock scheduling problem with mixed-mode dock doors. The outline of our ALNS algorithm for the integrated problem is presented in Section 2.5, whereas the details are provided in Section 2.8.3. We report our numerical results on computational experiments and a case study in Section 2.6, and conclude the work in Section 2.7 with our managerial insights.

2.2 Literature Review

Modeling cross-dock processes has been extensively studied in the last decade. Boysen & Fliedner (2010) classify the literature on truck scheduling based on the door environment (service mode and number of dock doors), operational characteristics (arrival and processing times, departure deadlines, availability of intermediate storage, etc.), and objective function. Van Belle et al. (2012) use a more general classification to categorize the literature based on the type of cross-docking problem: strategic decisions (location of cross-dock facilities and layout design), tactical decisions (flow of goods in a supply chain network with cross-dock terminals) and operational decisions (dock-door assignment, truck scheduling, vehicle routing, location of temporary storage). Buijs et al. (2014) identify interdependencies between these different levels of decision making, whereas Ladier & Alpan (2016) compare the literature on cross-docking decisions at the operational decision level to practices observed in industry. However, the literature on operational decision-making (such as truck scheduling and dock-door assignment) at cross-dock facilities where dock doors can have a mixed service mode is limited. Section 2.2.1 provides an overview on these papers. The second stream of literature related to our work is on cross-dock scheduling, which is addressed in Section 2.2.2.

2.2.1 Cross-dock operations with mixed-mode dock doors

There are only a few papers where dock doors can be used to process both inbound and outbound trucks. The early work by Brown (2003) and Bozer & Carlo (2008) studies a dock-door assignment problem in which origins and destinations (or trailers) are assigned to dock doors. Each door is exclusively used by the same trailer (i.e., by the same supplier for inbound trucks or by the same customer for outbound trucks) over the entire planning horizon, where the trailer can frequently visit the cross-dock facility. Consequently, these allocation decisions are tactical decisions rather than operational decisions. After being assigned, the door can operate only in an exclusive service mode for the scheduling decisions. In contrast, Miao et al. (2009) assign trucks to dock doors at an operational level, where a door can be used by different trucks. The schedule with the arrival and departure times of each truck is given as input parameter to the problem. Since no distinction is made between inbound and outbound operations, all dock doors can be seen as operating in a mixed service mode. The authors present two metaheuristics to solve the problem; a genetic algorithm

(GA) and a tabu search algorithm (TS). TS outperforms GA in their numerical experiments.

Only two papers study truck scheduling with mixed-mode dock doors. Berghman et al. (2015) present two model formulations for this problem where the objective is to minimize the weighted sojourn time of unit loads, i.e., the time between unloading and loading. A notable constraint in their model formulation is that an outbound truck cannot be docked unless all the inbound trucks that carry items for it are processed. The arrival and departure time windows are included as hard constraints. With numerical experiments on small-sized instances, they demonstrate that mixed-mode dock doors can generate substantial savings. In particular, only a limited number of mixed-mode dock doors provides the full benefit of flexible dock doors. Similar observations are obtained by Bodnar et al. (2017) who also study the truck scheduling problem with mixed-mode dock doors. In their objective function, they aim to minimize the usage of the temporary storage area and the delay of outbound trucks. Unlike the model by Berghman et al. (2015), outbound trucks can be processed before all inbound trucks that carry items for it are processed. This allows unit loads to be transferred directly to the outbound truck without incurring the additional cost of temporary storage. Commercial solvers are used to generate solutions within 1,800 seconds which are compared against the solutions from an ALNS algorithm designed for their problem. Extensive numerical experiments and a case study demonstrate the decreasing marginal returns when more flexible dock doors are included, and the additional benefit is negligible when more than 60% of the dock doors operate in a mixed service mode. In Section 2.8.1, we illustrate that the models formulated in both papers can result in infeasible solution when the scheduled trucks have to be assigned to dock doors in a sequential approach for the cross-dock scheduling problem.

2.2.2 Cross-dock scheduling

An overview of the limited literature to solve the cross-dock scheduling problem is provided in Table 2.1. The first six papers study cross-dock scheduling for only inbound operations. These studies assume that the outbound operations are already assigned and scheduled. Consequently, the dock doors under study are dedicated to unloading inbound trucks. This exclusive service mode for dock doors is assumed in most papers in the cross-dock literature. There are two exceptions as mentioned in the introduction. In Shakeri et al. (2012), each truck contains pallets that are unloaded, sorted, and then redistributed among the same trucks. Consequently, each truck and dock door is used first for inbound operations and then for outbound operations. Hermel et al. (2016) do not differentiate between service modes of doors and assign containers to dock doors such that each container, inbound or outbound, has a unique door. The same dock door is not used again by any other container during the planning horizon and retains this exclusive mode throughout the remainder of the planning period.

The third characteristic used to categorize the literature in Table 2.1 indicates whether each load from an inbound truck is already assigned to an outbound truck (i.e., pre-distribution cross-docking), or whether this assignment needs to be determined along with the door assignments and scheduling of trucks (i.e., post-distribution cross-docking). In the latter scenario, loads of a product from inbound trucks are interchangeable since they can be assigned to different outbound trucks. Acar et al. (2012) make no assumption about pre-distribution or post-distribution cross-docking.

Furthermore, all references assume that products are first unloaded and staged in front of the outbound door before they are loaded in the outbound truck, where there is an unlimited staging capacity. In reality, however, staging limitations around dock doors necessitate products to be moved to a temporary storage area when an inbound truck is unloaded before the corresponding outbound truck is docked at an outbound door (Yu & Egbelu, 2008; Bodnar et al., 2017). The extra operational efforts for this transferal of unit loads to and from the storage area is not included in the cross-dock scheduling literature. The only notable exception is Tootkaleh et al. (2016) who explicitly model the storage of products in a separate area when products of an inbound truck arrive after the outbound truck has already left the cross-dock terminal due to a departure deadline.

In the final two columns on the operational characteristics, we indicate whether there are specific arrival and departure time windows for either inbound and/or outbound trucks. If no arrival time is specified, it is assumed that all trucks are available at the yard at the beginning of the planning horizon and that they can be scheduled to enter the cross-dock terminal at any point in time. However, in practice, trucks arrive during the day and they can only enter the cross-dock terminal after they have arrived at the yard. Acar et al. (2012) and Konur & Golias (2013a,b) assume random arrival times of inbound trucks. Therefore, in their objective function, Acar et al. (2012) focus on evenly distributing the idle times between processing inbound trucks assigned to strip doors. In contrast, Konur & Golias (2013a,b) and Heidari et al. (2018) focus on minimizing waiting times at the yard (i.e., the time between arrival at the yard and entering the dock door). These three studies do not have restrictions on the departure times for outbound trucks. If there is a pre-determined time at which each outbound truck needs to leave the cross-dock terminal, this can be included to the problem either as a hard constraint or as a soft constraint. In the former case, all delayed loads are lost, kept in a temporary storage until the next outbound truck with the same destination departs or sent by an additional outbound truck (comparable to tardiness cost) (Boysen et al., 2013; Tootkaleh et al., 2016; Molavi et al., 2018). In the latter case, trucks can depart after their due time, but they are penalized in the objective function (Chmielewski et al., 2009; Liao et al., 2013; Van Belle et al., 2013; Assadi & Bagheri, 2016; Nassief et al., 2017).

The second and third last columns in Table 2.1 specify the size of the larger or real-world test instances that are reported: the number of inbound and outbound doors as well as the number of inbound and outbound trucks. The last column indicates whether the decision variables only specify the sequencing of the trucks at the dock doors (and then the actual enter and departure times are derived from this order and the given processing times), or whether the decision variables include an actual index for the time period at which the truck is docked at the terminal.

As mentioned in Section 2.1, the objective criterion for scheduling decisions include tardiness (i.e., amount of time trucks depart after their due time), earliness (i.e., when a truck is scheduled to depart before its due time), makespan (i.e., the time that the last outbound truck departs from the cross-dock terminal) and the waiting time for trucks to be processed. The dock-door assignment decisions are usually made based on travel distances between the dock doors of the inbound and corresponding outbound trucks. Table 2.1 provides an overview of the different single and multi-criterion objective functions used in the literature.

Most cross-dock scheduling problems are solved sequentially or with a metaheuristic. Acar et al. (2012) first assign doors based on a heuristic and then schedule the trucks. Boysen et al. (2013) use decomposition procedures (first-assign-then-schedule and first-schedule-then-assign), where heuristics are used for each stage in addition to a simulated annealing algorithm to achieve iterative improvements. Chmielewski et al. (2009) propose two solution approaches: decomposition-and-column generation and evolutionary algorithms. The first approach performs better but requires more computation time. Liao et al. (2013) use simulated annealing, tabu search, ant colony optimization, and (hybrid) decomposition evolution. Assadi & Bagheri (2016) propose a differential evolution algorithm and population-based metaheuristics with pairwise

Table 2.1: Summary of the related literature on cross-dock scheduling

Publications	Operational Characteristics						Objective	Size Numerical Instances		
	operational focus	service mode	distribution	temporary storage	arrival times	departure times		#doors IN, OUT	#trucks IN, OUT	#time periods
Acar et al. (2012)	IN	EXCL	-	no	yes	no	evenly spread idle times	10-50	unspecified	SEQ
Konur & Golias (2013a,b)	IN	EXCL	PRE	no	yes	no	waiting time, process cost	5-20	50-200	SEQ
Boysen et al. (2013)	IN	EXCL	PRE	no	no	yes	tardiness	10-30	48,15	SEQ
Nassief et al. (2017)	IN	EXCL	PRE	no	no	yes	tardiness, labor cost	20-40	60-100	unspecified
Liao et al. (2013)	IN	EXCL	POST	no	no	yes	tardiness	2-5	5-12	SEQ
Tootkaleh et al. (2016)	IN	EXCL	POST	no	no	yes	tardiness	1-7,2-8	10-40,2-8	2-8
Kuo (2013)	IN+OUT	EXCL	PRE	no	no	no	makespan	50-100,50-100	50-200,50-200	SEQ
Zhang et al. (2010)	IN+OUT	EXCL	PRE	no	yes	no	travel distance, waiting time, departure time	5,5	10,10	SEQ
Chmielewski et al. (2009)	IN+OUT	EXCL	PRE	no	yes	yes	travel distance, waiting time	20-40,20-60	30-80,20-40	4-8
Van Belle et al. (2013)	IN+OUT	EXCL	PRE	no	yes	yes	travel distance, tardiness	3,3	25-30,25-30	SEQ
Wisittipanich & Hengmeechai (2017)	IN+OUT	EXCL	POST	no	no	no	makespan	4-30,4-30	14-50,10-50	SEQ
Lee et al. (2012)	IN+OUT	EXCL	POST	no	no	no	throughput	2-10,2-10	10-55,10-55	SEQ
Assadi & Bagheri (2016)	IN+OUT	EXCL	POST	no	yes	yes	earliness, tardiness	8-20, 8-20	15-40,15-40	SEQ
Chen & Song (2009)	IN+OUT	EXCL	PRE	no	no	no	makespan	2-10,2-10	5-80,4-96	SEQ
Bellanger et al. (2013)	IN+OUT	EXCL	PRE	no	no	no	makespan	2-12,2-12	8-100,8-100	SEQ
Heidari et al. (2018)	IN+OUT	EXCL	PRE	no	yes	no	waiting time, process cost	2-20,2-20	5-200,5-200	SEQ
Molavi et al. (2018)	IN+OUT	EXCL	PRE	no	yes	yes	tardiness	2-12,2-12	2-30,2-30	SEQ
Fonseca et al. (2019)	IN+OUT	EXCL	PRE	no	no	no	makespan	2-10,2-10	20-80,16-96	SEQ
Hermel et al. (2016)	IN+OUT	MIXED	PRE	no	no	no	travel distance, makespan	40	100	120
Shakeri et al. (2012)	IN+OUT	MIXED	POST	no	no	no	makespan	100-250	200-2,000	SEQ
Our paper	IN+OUT	MIXED	PRE	yes	yes	yes	travel distance, tardiness, temporary storage	10-15,10-30	15-75,15-224	32

exchange and simulated annealing. Van Belle et al. (2013) solve the problem with tabu search where a swap operator and an insert operator are used. Kuo (2013) uses variable neighborhood search. Wisittipanich & Hengmeechai (2017) use particle swarm optimization with the local best solution and the near-neighbor best solution. Lee et al. (2012) use three different genetic algorithms. Konur & Golias (2013a,b) propose a genetic algorithm to find Pareto efficient schedules for the inbound trucks and recommend a hybrid approach which considers a combination of pessimistic and optimistic scenarios. Heidari et al. (2018) and Molavi et al. (2018) also use variations of a genetic algorithm. Tootkaleh et al. (2016) use simple heuristics based on greedy principles to solve the cross-docking problem. Nassief et al. (2017) and Zhang et al. (2010) solve their mathematical models to optimality. Chen & Song (2009), Bellanger et al. (2013) and Fonseca et al. (2019) use a different approach to tackle cross-dock scheduling problems. They model the problems as two-stage and three-stage hybrid flow shop scheduling problems with cross-docking constraints which prevents outbound truck from processing before all of its inbound shipments are processed and develop several heuristics, branch-and-bound algorithms and a hybrid Lagrangian metaheuristic to solve these problems.

The two papers that study mixed-mode dock doors use a sequential approach. Hermel et al. (2016) first assign inbound and outbound trucks to dock doors and then schedule the trucks according to resource constraints at the cross-dock terminal. Shakeri et al. (2012) use a two-phase decomposition heuristic, where trucks are sequenced first based on a ranking and then assigned to doors according to the sequence list.

It should be noted that cross-dock problems that are formulated as *flow shop scheduling problems* (Chen & Song, 2009; Bellanger et al., 2013; Fonseca et al., 2019) can be extended to include mixed-mode dock doors in a straightforward manner by designating some machines, equivalent to mixed-mode dock doors, as flexible machines that are capable of processing trucks in multiple stages of the flow shop. However, if the cost of the solution depends on the dock-door (or machine) combinations in which the truck is processed in the first (inbound) and second (outbound) stage, as is the case with our problem, the models and methods presented by Chen & Song (2009), Bellanger et al. (2013) and Fonseca et al. (2019) cannot be applied in this straightforward manner anymore. Furthermore, the usage of temporary storage, which is triggered when there is a mismatch between the time at which inbound trucks and the associated outbound trucks are processed, cannot be addressed by cross-dock models that are based on flow shop scheduling problems. This requires us to develop a new model and solution approaches for the problem.

Cross-docking can also be seen as part of a larger distribution network where vehicle routing decisions are included. A rich stream of literature aims at generating vehicle routes such that the routes for the trucks that need to visit a cross-dock facility are synchronized. Such problems are considered a subclass of vehicle routing problems with synchronization constraints (Drexl, 2012) or vehicle- and location-routing problems with intermediate stops (Schiffer et al., 2019). Since routing decisions are not included in the scope of our paper. We refer the interested reader to Agustina et al. (2014), Dondo & Cerdá (2014, 2015), Enderer et al. (2017) and Grangier et al. (2017).

2.3 Problem Description, Notations and Assumptions

Consider a cross-dock facility that needs to process a set of inbound and outbound trucks denoted by $\mathcal{I}$ and $\mathcal{O}$, respectively, over a given planning horizon $\mathcal{T}$. The planning horizon is discretized in $|\mathcal{T}|$ time periods of equal length. Time periods are indexed by t such that $t \in \mathcal{T}$. The dock doors at the cross-dock facility are divided into inbound, outbound and flexible (or mixed-mode) doors and represented by the sets $\mathcal{D}_I$, $\mathcal{D}_O$, and $\mathcal{D}_F$, respectively. If no truck is assigned to a flexible door $k \in \mathcal{D}_F$ at a certain time period, it can be assigned to process either inbound or outbound trucks. For simplicity, we denote all dock doors that can be assigned to unload inbound trucks as $\mathcal{D}_{IF} = \mathcal{D}_I \cup \mathcal{D}_F$ and all doors that can be used to load outbound trucks as $\mathcal{D}_{OF} = \mathcal{D}_O \cup \mathcal{D}_F$.

The time of arrival at the yard for truck $i \in \{\mathcal{I}, \mathcal{O}\}$ is denoted by $r_i \in \mathcal{T}$ (also called the ready time), whereas the departure time of truck i needs to be scheduled before or at due time $d_i \in \mathcal{T}$. Each inbound truck $i \in \mathcal{I}$ has to be processed within the time window $[r_i; d_i]$, whereas each outbound truck $j \in \mathcal{O}$ can leave the cross-dock terminal with a delay at a penalty cost γ for each time period that the truck is delayed. These costs for untimely deliveries include additional labor costs for the truck driver, longer usage of the truck, penalties for violating delivery time windows enforced by city authorities (Quak & de Koster, 2007), and additional costs by the customer to unload the truck upon arrival at the customer's location (Quak & de Koster, 2009). To account for any uncertainty in the arrival time, particularly for inbound trucks, the cross-dock operator can incorporate additional slack time to ensure that inbound trucks are available for processing at the beginning of the time window.

The cross-dock dispatcher has complete information about the contents of each inbound truck. The products are stacked on unit loads (e.g., pallets, roll cages), where all products in the same receiving unit load are destined to a specific outbound truck (i.e., pre-distribution cross-docking). Each inbound truck carries one or more unit loads for multiple outbound trucks. In particular, the number of unit loads arriving with inbound truck $i \in \mathcal{I}$ that needs to be unloaded, transferred and loaded to outbound truck $j \in \mathcal{O}$ is indicated as f_{ij}. It is assumed that sufficient personnel and equipment is available to perform these operations. Other internal operations (such as sorting and labeling) are not considered. The time required to unload all unit loads from an inbound truck is assumed to be fixed and exactly equal to the length of one time period. This is a common assumption when the transported freight is shipped in standardized cargo containers and the number of unit loads per inbound truck does not strongly differ (Boysen, 2010). Each inbound truck $i \in \mathcal{I}$ and outbound truck $j \in \mathcal{O}$ has to remain at the dock door for a minimum of g_i and h_j time periods, respectively. This is a generalization of the assumption made by Bodnar et al. (2017), where an inbound truck is at a dock door for exactly one time period. By allowing more flexibility, it gives the opportunity for inbound trucks to stay at a dock door longer to facilitate direct dock-to-dock transfers instead of using the temporary storage area. Furthermore, preemption of processing a truck is not allowed, i.e., a docked truck does not release its assigned door until it is completely unloaded or loaded.

Unloaded goods are transferred from the inbound truck either directly to the appropriate dock door where they are loaded onto an outbound truck or to the temporary storage area in the cross-dock terminal if the appropriate outbound truck is not yet docked at the terminal. Additionally, the unit loads can also be left at the inbound dock lane and moved to the outbound dock lanes at a later time period. When unit loads are placed at an inbound dock lane for more than one time period, the dock lane cannot be used by a different truck until the lane is cleared of the unit loads. This strategy enables a reduced number of unit loads that are moved to the temporary storage area, but it can lead to delays of outbound trucks. For retail cross-dock facilities, this is a reasonable representation as unit loads can stay at the inbound lanes for multiple time periods to facilitate direct dock-to-dock transfers.

We assume that the temporary storage area is sufficiently large and does not constrain cross-dock operations. This is also observed in the field investigation by Ladier & Alpan (2016). Furthermore, we have experienced that the cross-docking operations in many unit-load cross-dock facilities in Europe and Asia are performed in warehouse facilities that have significant space available for the temporary storage of items. Temporary storage is also no limitation in the four cross-dock facilities included in our case study (see Section 2.6.5). This form of storage is different from the floor-chain conveyor systems studied by Bartholdi III & Gue (2000) in less-than-truck load terminals where each dock door is reserved for one destination for the whole day. As a result, the space in front of a dock door can serve as a unique storage area for a dock alone and modeling temporary storage becomes redundant. If the amount of temporary storage would be constrained, we would have to consider the detailed use of temporary storage (Vis & Roodbergen, 2008), and track the use of each storage space individually. We do not include this level of granularity, as the amount of temporary storage space is usually sufficient. In addition, the number of unit loads in the problem would be too large to solve along with the scheduling and assignment decisions for trucks. In this paper, the storage space is only intended for items that cannot be directly transferred from an inbound truck to an outbound truck. This can be a necessity to create feasible solutions since we consider a high utilization of the dock doors. By penalizing the usage of the storage area in our objective function, our solution approach reduces the utilization of the storage area. Therefore, it is sufficient to model the intermediate storage in this basic manner.

The direct dock-to-dock transfer requires a travel distance m_{kp} between dock door $k \in \mathcal{D}_{IF}$ where the inbound truck is processed and dock door $p \in \mathcal{D}_{OF}$ where the outbound truck is processed. The cost associated with this direct dock-to-dock transfer is indicated by η per unit load per unit distance and consists of costs for personnel and equipment (such as forklifts). However, if inbound truck $i \in \mathcal{I}$ departs or the inbound dock lane is emptied before outbound truck $j \in \mathcal{O}$ is docked where $f_{ij} > 0$, the unit loads destined for outbound truck j have to be temporarily stored until the outbound truck is ready to be loaded. The cost of temporary storage for each unit load is indicated by β and consists of the additional transfer to and from the storage area as well as the costs incurred for double material handling. An overview of all notations is included in Table 2.2.

Most studies that integrate the truck scheduling and dock-door assignment problem have considered that the time needed to transfer goods from inbound to outbound trucks is directly proportional to the distance between the dock doors to which the trucks are assigned. Boysen et al. (2013) and Bodnar et al. (2017) both note that the time required to process an inbound trailer is approximately 30 to 45 minutes. Within such a time period, all transshipment operations can be performed in any cross-dock facility, even between the furthest dock doors in a large facility. Consequently, it is unnecessary to incorporate the actual transfer times in a cross-dock facility. Following this previous literature, we also assume that any direct dock-to-dock transfer or dock-to-storage transfer can be completed within one time period. Therefore, our

Table 2.2: Overview of the notation for the cross-dock scheduling problem

notation	description
$\mathcal{I}$, $\mathcal{O}$	set of inbound and outbound trucks
$\mathcal{D}_I$, $\mathcal{D}_O$, $\mathcal{D}_F$	set of inbound, outbound and mixed-mode doors
$\mathcal{T}$	set of time periods
r_i, d_i	arrival time and departure time of truck i
g_i	minimum processing time required by inbound truck i
h_j	minimum processing time required by outbound truck j
f_{ij}	number of unit loads to be transferred from inbound truck i to outbound truck j
m_{kp}	travel distance between dock doors k and p
η	transfer and material handling cost per unit load per unit distance associated with a direct dock-to-dock transfer
β	transfer and material handling cost per unit load associated with a transfer to and from the storage area
γ	penalty cost per unit load per unit time that a truck is delayed

objective is to minimize a weighted combination of three cost components: the transfer costs as function of internal travel distances between inbound and outbound doors, the usage of the temporary storage area and the total tardiness of outbound trucks with respect to their assigned due times. In the next section, we mathematically formulate the cross-dock scheduling problem that we study in more detail.

2.4 Model Formulation

The aim of this section is to formulate the multi-door cross-dock scheduling problem (that is, integrating the truck scheduling decisions and the dock-door assignment decisions) as a mixed-integer programming (MIP) model in which temporary storage and internal travel distances are minimized as well as the total tardiness with respect to the departure times of outbound trucks.

According to the 3-field notation system introduced by Boysen & Fliedner (2010) to classify operational cross-dock problems, our problem can be denoted as $[EM|r_i, d_i, p_i, t_p = 0|\sum T_i + \sum S_p + \sum DC]$. EM indicates that the cross-dock environment has both exclusive and mixed-mode dock doors. r_i, d_i, p_i indicates truck dependent arrival, departure, and processing times, respectively. $t_p = 0$ denotes that the transshipment (or transfer) time within the cross-dock facility is zero or, in other words, all transshipments are possible within the same time period. The final field indicates the objective as $\sum T_i + \sum S_p + \sum DC$, which is the minimization of the total tardiness, temporary storage, and total direct transfer distance costs. $\sum DC$ is a new notation we introduce specific to our problem. This was not considered in the nomenclature introduced by Boysen & Fliedner (2010). Note that p is used here in relation to the nomenclature of Boysen & Fliedner (2010). In the model formulation and the remainder of the paper, we use g and h to refer to the minimum processing time for inbound and outbound trucks, respectively (as described in Section 2.3).

The following parameters are used for simplifying purposes:

- $\mathcal{T}_i^E$ set of time periods before truck $i \in \mathcal{I} \cup \mathcal{O}$ arrives at the yard, $\mathcal{T}_i^E = \{t \mid t \in \mathcal{T}, t < r_i\}$
- $\mathcal{T}_i^L$ set of time periods after truck $i \in \mathcal{I} \cup \mathcal{O}$ is due to depart the terminal, $\mathcal{T}_i^L = \{t \mid t \in \mathcal{T}, t > d_i\}$
- $\mathcal{T}_i^N$ set of time periods within the processing time window of truck $i \in \mathcal{I} \cup \mathcal{O}$, $\mathcal{T}_i^N = \{t \mid t \in \mathcal{T}, r_i \leq t \leq d_i\}$
- δ_j^t delay for outbound truck $j \in \mathcal{O}$ if departed at time period $t \in \mathcal{T}$, $\delta_j^t = \max\{0, t - d_j\}$,
- λ_{ij} 1 if $f_{ij} > 0$, 0 otherwise

The following decision variables are used in our model formulation:

- x_{ik}^t 1 if inbound truck $i \in \mathcal{I}$ is assigned to door $k \in \mathcal{D}_{IF}$ at time period $t \in \mathcal{T}$, 0 otherwise
- $\hat{x}_{ik}^t$ 1 if inbound truck $i \in \mathcal{I}$ has finished unloading at door $k \in \mathcal{D}_{IF}$ at the end of time period $t \in \mathcal{T}$, 0 otherwise
- y_{jp}^t 1 if outbound truck $j \in \mathcal{O}$ is assigned to door $p \in \mathcal{D}_{OF}$ at time period $t \in \mathcal{T}$, 0 otherwise
- $\hat{y}_{jp}^t$ 1 if outbound truck $j \in \mathcal{O}$ has finished loading at door $p \in \mathcal{D}_{OF}$ at the end of time period $t \in \mathcal{T}$, 0 otherwise
- u_{ij}^t number of unit loads to unload from inbound truck $i \in \mathcal{I}$ destined for outbound truck $j \in \mathcal{O}$ after time period $t \in \mathcal{T}$
- l_{ij}^t number of unit loads loaded onto outbound truck $j \in \mathcal{O}$ originating from inbound truck $i \in \mathcal{I}$ after time period $t \in \mathcal{T}$
- s_{ij}^t 1 if unit loads destined for outbound truck $j \in \mathcal{O}$ are transferred from inbound truck $i \in \mathcal{I}$ to the temporary storage area at time period $t \in \mathcal{T}$, 0 otherwise
- $\hat{s}_{ij}^t$ number of unit loads at the temporary storage location at the end of time period $t \in \mathcal{T}$ that have been received from inbound truck $i \in \mathcal{I}$ and are destined for outbound truck $j \in \mathcal{O}$

$\hat{z}_{ijkp}$ 1 if inbound truck $i \in \mathcal{I}$ is completely unloaded at door $k \in \mathcal{D}_{IF}$ and outbound truck $j \in \mathcal{O}$ is completely processed at dock door $p \in \mathcal{D}_{OF}$ for dock-to-dock transfer of shipments between i and j, 0 otherwise

z_{ij}^t 1 if inbound truck $i \in \mathcal{I}$ and outbound truck $j \in \mathcal{O}$ are both docked at time period $t \in \mathcal{T}$ and j receives load from i, 0 otherwise

A_{ij}^t number of unit loads that have been received from inbound truck $i \in \mathcal{I}$ and are transferred from the temporary storage location to outbound truck $j \in \mathcal{O}$ at time period $t \in \mathcal{T}$

The integrated cross-dock scheduling problem can then be formulated as a MIP model as follows:

$$\min \quad \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{O}}\sum_{k\in\mathcal{D}_{IF}}\sum_{p\in\mathcal{D}_{OF}} \eta\hat{z}_{ijkp}m_{kp}f_{ij}+ \tag{2.1a}$$

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{O}}\sum_{t\in\mathcal{T}} \beta s_{ij}^t f_{ij}+ \tag{2.1b}$$

$$\sum_{j\in\mathcal{O}}\sum_{p\in\mathcal{D}_{OF}}\sum_{t\in\mathcal{T}_j^L} \gamma\delta_j^t\hat{y}_{jp}^t \tag{2.1c}$$

subject to

$$\sum_{k\in\mathcal{D}_{IF}} x_{ik}^t \leq 1 \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{2.2}$$

$$\sum_{p\in\mathcal{D}_{OF}} y_{jp}^t \leq 1 \qquad \forall j \in \mathcal{O}, t \in \mathcal{T} \tag{2.3}$$

$$\sum_{i\in\mathcal{I}} x_{ik}^t \leq 1 \qquad \forall k \in \mathcal{D}_I, t \in \mathcal{T} \tag{2.4}$$

$$\sum_{j\in\mathcal{O}} y_{jp}^t \leq 1 \qquad \forall p \in \mathcal{D}_O, t \in \mathcal{T} \tag{2.5}$$

$$\sum_{i\in\mathcal{I}} x_{ik}^t + \sum_{j\in\mathcal{O}} y_{jk}^t \leq 1 \qquad \forall k \in \mathcal{D}_F, t \in \mathcal{T} \tag{2.6}$$

$$\sum_{t\in\mathcal{T}_i^N}\sum_{k\in\mathcal{D}_{IF}} \hat{x}_{ik}^t = 1 \qquad \forall i \in \mathcal{I} \tag{2.7}$$

$$\sum_{t\in\mathcal{T}_j^N\cup\mathcal{T}_j^L}\sum_{p\in\mathcal{D}_{OF}} \hat{y}_{jp}^t = 1 \qquad \forall j \in \mathcal{O} \tag{2.8}$$

$$u_{ij}^0 = f_{ij} \quad \forall i \in \mathcal{I}, j \in \mathcal{O} \tag{2.9}$$

$$l_{ij}^0 = 0 \quad \forall i \in \mathcal{I}, j \in \mathcal{O} \tag{2.10}$$

$$\hat{s}_{ij}^0 = 0 \quad \forall i \in \mathcal{I}, j \in \mathcal{O} \tag{2.11}$$

$$u_{ij}^t \geq u_{ij}^{t-1} - (s_{ij}^t f_{ij}) - z_{ij}^t f_{ij} \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.12}$$

$$l_{ij}^t \leq l_{ij}^{t-1} + A_{ij}^t + z_{ij}^t f_{ij} \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.13}$$

$$s_{ij}^t \leq \sum_{k \in \mathcal{D}_{IF}} x_{ik}^t \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.14}$$

$$\hat{s}_{ij}^t = \hat{s}_{ij}^{t-1} + (s_{ij}^t f_{ij}) - A_{ij}^t \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.15}$$

$$x_{ik}^t \leq \hat{x}_{ik}^t + x_{ik}^{t+1} \quad \forall i \in \mathcal{I}, k \in \mathcal{D}_{IF}, t \in \mathcal{T} \tag{2.16}$$

$$y_{jp}^t \leq \hat{y}_{jp}^t + y_{jp}^{t+1} \quad \forall j \in \mathcal{O}, p \in \mathcal{D}_{OF}, t \in \mathcal{T} \tag{2.17}$$

$$\sum_{j \in \mathcal{O}} f_{ij} \Big(1 - \sum_{k \in \mathcal{D}_{IF}} \hat{x}_{ik}^t \Big) \geq \sum_{j \in \mathcal{O}} u_{ij}^t \quad \forall i \in \mathcal{I}, t \in \mathcal{T} \tag{2.18}$$

$$\sum_{i \in \mathcal{I}} \sum_{p \in \mathcal{D}_{OF}} f_{ij} \hat{y}_{jp}^t \leq \sum_{i \in \mathcal{I}} l_{ij}^t \quad \forall j \in \mathcal{O}, t \in \mathcal{T} \tag{2.19}$$

$$\sum_{k \in \mathcal{D}_{IF}} \sum_{t \in \mathcal{T}_i^E \cup \mathcal{T}_i^L} x_{ik}^t = 0 \quad \forall i \in \mathcal{I} \tag{2.20}$$

$$\sum_{p \in \mathcal{D}_{OF}} \sum_{t \in \mathcal{T}_j^E} y_{jp}^t = 0 \quad \forall j \in \mathcal{O} \tag{2.21}$$

$$g_i \sum_{k \in \mathcal{D}_{IF}} \hat{x}_{ik}^t \leq \sum_{k \in \mathcal{D}_{IF}} \sum_{t'=r_i}^{t} x_{ik}^{t'} \quad \forall i \in \mathcal{I}, t \in \mathcal{T}_i^N \tag{2.22}$$

$$h_j \sum_{p \in \mathcal{D}_{OF}} \hat{y}_{jp}^t \leq \sum_{p \in \mathcal{D}_{OF}} \sum_{t'=r_j}^{t} y_{jp}^{t'} \quad \forall j \in \mathcal{O}, t \in \{\mathcal{T}_j^N, \mathcal{T}_j^L\} \tag{2.23}$$

$$\lambda_{ij} \left(\sum_{t \in \mathcal{T}} \hat{x}_{ik}^t + \sum_{t \in \mathcal{T}} \hat{y}_{jp}^t - \sum_{t \in \mathcal{T}} s_{ij}^t \right) \leq 1 + \hat{z}_{ijkp} \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, k \in \mathcal{D}_{IF}, p \in \mathcal{D}_{OF} \tag{2.24}$$

$$\lambda_{ij} \left(\sum_{k \in \mathcal{D}_{IF}} x_{ik}^t + \sum_{p \in \mathcal{D}_{OF}} y_{jp}^t \right) \geq 2\, z_{ij}^t \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.25}$$

$$A_{ij}^t \leq \hat{s}_{ij}^{t-1} \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.26}$$

$$A_{ij}^t \leq f_{ij} \sum_{p \in \mathcal{D}_{OF}} y_{jp}^t \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.27}$$

$$\hat{z}_{ijkp} \in \{0,1\} \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, k \in \mathcal{D}_{IF}, p \in \mathcal{D}_{OF} \tag{2.28}$$

$$x_{ik}^t, \hat{x}_{ik}^t \in \{0,1\} \quad \forall i \in \mathcal{I}, k \in \mathcal{D}_{IF}, t \in \mathcal{T} \tag{2.29}$$

$$y_{jp}^t, \hat{y}_{jp}^t \in \{0,1\} \quad \forall j \in \mathcal{O}, p \in \mathcal{D}_{OF}, t \in \mathcal{T} \tag{2.30}$$

$$z_{ij}^t, s_{ij}^t \in \{0,1\} \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.31}$$

$$A_{ij}^t, u_{ij}^t, l_{ij}^t, \hat{s}_{ij}^t \geq 0 \quad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.32}$$

The first term (2.1a) in the objective function represents the transportation costs for shipments directly transferred from inbound to outbound trucks. The second term (2.1b) represents the costs of unit loads that are temporarily stored before being loaded onto the outbound truck. The third term (2.1c) represents the total tardiness costs of outbound trucks.

Constraints (2.2) and (2.3) ensure that each truck can only be docked at a dock door that is capable of processing the load at any time period. Constraints (2.4) to (2.6) ensure that the same dock door cannot be assigned to more than one truck at the same time period. Constraints (2.7) ensure that all inbound trucks are completely processed within their time window. All outbound trucks are completely processed between their arrival time period and the end of the planning horizon as per constraints (2.8). Constraints (2.9) specify the number of unit loads in each inbound truck at the beginning of the planning horizon. Each outbound truck and the temporary storage area is defined as empty at the beginning of the planning horizon by constraints (2.10) and (2.11), respectively.

Constraints (2.12) regulate the decrease in the number of unit loads that are left to be unloaded. Similarly, constraints (2.13) regulate the possible increment in the number of unit loads that are loaded onto each outbound truck at the end of each time period. Constraints (2.14) ensure that unit loads cannot be moved to the temporary storage area from an inbound truck unless the inbound truck is docked at a door. Constraints (2.15) are balancing equations that count the number of unit loads available in the temporary storage area.

Constraints (2.16) together with constraints (2.7) prevent preemption and they connect $\hat{x}_{ik}^t$ with x_{ik}^t. Similarly, each outbound truck cannot preempt according to constraints (2.17) in combination with constraints (2.8) and they also relate $\hat{y}_{jp}^t$ to y_{jp}^t. Constraints (2.18) and (2.19) force all inbound and outbound trucks to be completely unloaded and loaded, respectively, before their processing is completed in the previous constraints. Constraints (2.20) mandate that each inbound truck is not docked at the cross-dock terminal before its arrival or after its due time. Similarly, constraints (2.21) require that each outbound truck can only be docked after its arrival to the yard. Constraints (2.22) and (2.23) mandate that each inbound and outbound truck, respectively, has to remain at the dock door for at least a specified number of time periods once it is docked.

Constraints (2.24) regulate variable $\hat{z}_{ijkp}$, where it attains the value one when both inbound truck $i \in \mathcal{I}$ and outbound truck $j \in \mathcal{O}$ are docked at door $k \in \mathcal{D}_{IF}$ and $p \in \mathcal{D}_{OF}$, respectively, for a direct dock-to-dock transfer. For the variable z^t_{ij}, constraints (2.25) govern direct transfers of products between inbound and outbound trucks. Constraints (2.26) and (2.27) assign an appropriate value to A^t_{ij}. Specifically, constraints (2.27) do not allow unit loads to be moved from the temporary storage area unless the corresponding outbound truck is docked. Finally, constraints (2.28) to (2.32) specify the domain of each decision variable.

The following valid equalities are included into the model formulation to assist convergence to optimal solutions:

$$\sum_{k\in\mathcal{D}_{IF}} \sum_{p\in\mathcal{D}_{OF}} \hat{z}_{ijkp} + \sum_{t\in\mathcal{T}} s^t_{ij} = 1 \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, f_{ij} > 0 \tag{2.33}$$

Each transfer between inbound and outbound doors has to happen either through a direct dock-to-dock transfer, which is indicated by $\hat{z}_{ijkp}$, or by routing the transfer through the temporary storage area, which is represented by s^t_{ij}. Equation (2.33) exploits this information to define higher lower bounds on the objective value of fractional solutions and to help converge to the optimal solution more quickly.

Proposition 2.1. *The cross-dock scheduling problem* $[EM|r_i, d_i, p_i, t_p = 0|\sum T_i + \sum S_p + \sum DT]$ *is strongly NP-hard.*

Proof. $[EM|r_i, d_i, p_i, t_p = 0|\sum T_i + \sum S_p + \sum DT]$ is a generalization of $[EM|r_i, d_i, p_i, t_p = 0|\sum T_i + \sum S_p]$ which is NP hard in the strong sense (Bodnar et al., 2017).

2.5 Integrated Solution Approach

To solve the model formulated in Section 2.4, we propose to use adaptive large neighbourhood search (ALNS). In particular, we adapt the ALNS algorithm proposed by Bodnar et al. (2017) to include dock-door assignment decisions besides truck scheduling decisions. In this section, we only highlight the differences between our cross-dock scheduling ALNS algorithm and the truck scheduling ALNS algorithm of Bodnar et al. (2017), whereas all details of our integrated ALNS algorithm are described in Section 2.8.3.

The integrated cross-dock scheduling ALNS algorithm follows a decomposition approach where decisions pertaining to inbound trucks are changed by neighbourhood operators and the outbound decisions are made by solving a reduced problem to optimality (with the inbound decisions given as input parameters). Other decomposition approaches can be found in the literature, such as first-schedule-then-assign and first-assign-then-schedule (Boysen et al., 2013; Hermel et al., 2016). Furthermore, our decomposition approach is used in each iteration of our ALNS algorithm, rather than solving each decomposition step only once. The architecture of the cross-dock scheduling ALNS algorithm is presented in Figure 2.1. First, a set of initial feasible solutions is generated using constructive heuristics. The best solution is then used by the ALNS algorithm for modification and re-optimization. The scheduling and assignment decisions of inbound trucks are modified by neighbourhood operators, which are dynamically selected based on their individual performance to modify the scheduling and assignment decisions for the inbound trucks. Once the reduced problem finds the optimal scheduling and assignment decisions for the outbound trucks in the second step of the decomposition, the new solution is accepted using criteria from simulated annealing. This process is repeated until a stopping criterion is met, which terminates the algorithm and reports the best solution that is found.

The key difference between our cross-dock scheduling ALNS algorithm and the truck scheduling ALNS algorithm proposed by Bodnar et al. (2017) is that our ALNS algorithm simultaneously considers scheduling and assignment decisions rather than just scheduling decisions. As a result, our ALNS algorithm requires an extended solution representation and a new approach to generate initial solutions, as well as additional neighbourhood operators and an extended reduced problem formulation. In this section, we discuss the new solution representation and present an overview of the new neighbourhood operators. More details on each of the new aspects of the ALNS algorithm are presented in Section 2.8.3.

2.5.1 Representation of solution

As indicated above, the search algorithm will modify the decisions related to all inbound operations as part of the first step in the decomposition. Consequently, these decisions are represented as a matrix with the dimensions $|\mathcal{T}| \times |\mathcal{D}_{IF}|$. Each row of the matrix represents a dock door and each column represents a time period, where each individual element in the matrix indicates either the identification number of the inbound truck scheduled at that particular dock door at that time period or 0 if nothing is scheduled. An example of the solution representation is provided in Table 2.3. At dock door 1, inbound truck 1 is docked at the first two time periods. No

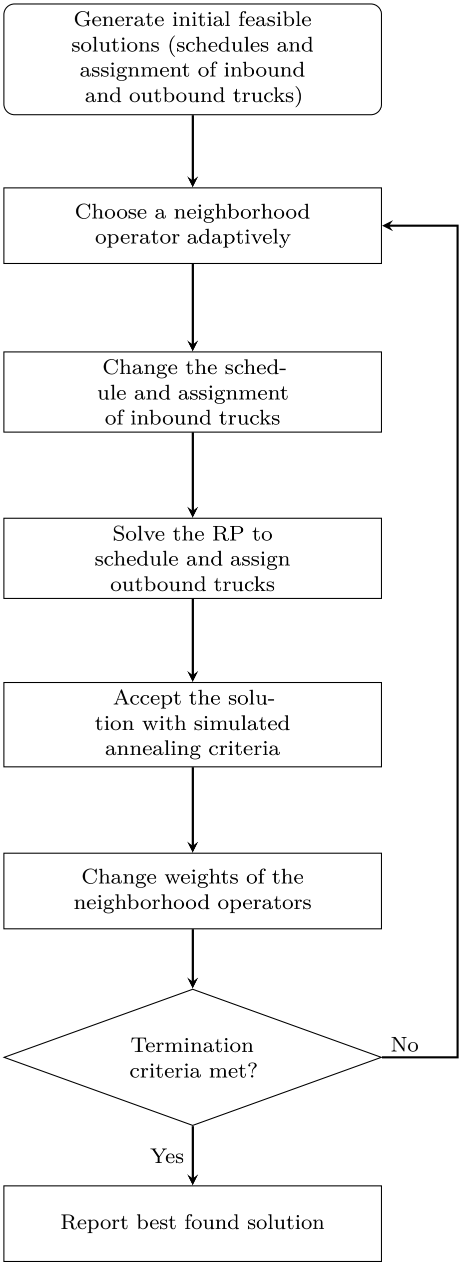


Figure 2.1: The architecture of the cross-dock scheduling ALNS algorithm

truck is assigned to door 1 in the third time period, and inbound truck 5 is assigned in the last three time periods.

Table 2.3: Example of the new solution representation

	1	2	3	$\cdots$	$\lvert\mathcal{T}\rvert - 2$	$\lvert\mathcal{T}\rvert - 1$	$\lvert\mathcal{T}\rvert$
1	1	1	**0**	$\cdots$	5	5	5
2	8	8	9	$\cdots$	11	**0**	**0**
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\lvert\mathcal{D}_{IF}\rvert - 1$	23	23	23	$\cdots$	2	2	2
$\lvert\mathcal{D}_{IF}\rvert$	55	55	56	$\cdots$	**0**	**0**	**0**

* **0** - no truck is assigned to the dock door

2.5.2 Neighbourhood operators

Removal and insertion neighbourhood operators are used, respectively, to destroy and to repair any scheduling and assignment decisions for the inbound operations in the solution representation introduced in the previous subsection. Table 2.4 gives an overview of all operators that are used by the cross-dock scheduling ALNS algorithm. It becomes clear that each operator is either a combination of a removal and insertion operator or the operator performs both tasks by itself. Furthermore, the last column in this table indicates whether the operator is designed to improve scheduling decisions (S), assignment decisions (A) or both types of decisions (S&A). The new operators are indicated in bold. Each of the neighbourhood operators is discussed in more detail in Section 2.8.3 and their performance is summarized in Section 2.8.4. From these results we conclude that the operators specifically designed to solve the integrated cross-dock scheduling problem improve the solution the most.

2.6 Results

In this section, the performance of the integrated cross-dock scheduling problem and the proposed solution approach are studied through numerical experiments, where the sequential approach (i.e., first-schedule-then-assign) is used as benchmark. In Section 2.8.2, we develop the MIP formulations for the truck scheduling and dock-door assignment problems in the sequential approach. We consider instances with 5, 10 or 20 dock doors. For most of the small instances with only five doors, we can find the optimal solution to both the cross-dock scheduling problem formulated in Section 2.4 as well as the truck scheduling and dock-door assignment problem of the sequential approach formulated in Section 2.8.2. For the instances with ten dock doors, we cannot find an optimal solution anymore for the integrated cross-dock scheduling problem

Table 2.4: The combinations of neighborhood operators used in the cross-dock scheduling ALNS algorithm. Operators indicated in bold are new operators. Each operator is designed to improve either scheduling decisions (S), assignment decisions (A) or both types of decisions (S&A).

Operator	Description	Focus
rRd & iBck	random removal and backward insertion	S
rRd & iFwd	random removal and forward insertion	S
rRd & **iSwap**	time and space positions of two trucks are randomly exchanged	S&A
rRd & **iUp**	random removal and insertion to adjacent dock door in one direction	A
rRd & **iDown**	random removal and insertion to adjacent dock door in different direction	A
rMx & iBck	random removal from the maxtime and backward insertion	S
rMx & iFwd	random removal from the maxtime and forward insertion	S
rCr & iBck	removal based on critical ratio and backward insertion	S
rCr & iFwd	removal based on critical ratio and forward insertion	S
rCr & **iSwap**	time and space positions of two trucks are exchanged based on the critical ratio	S&A
rSy & iBck	removal based on synchronization ratio and backward insertion	S
rSy & iFwd	removal based on synchronization ratio and forward insertion	S
rSy & **iSwap**	two trucks are exchanged based on the synchronization ratio	S&A
rI20	inbound truck processing is postponed to facilitate direct transfer	S
riextBck	inbound truck processing is extended backward to facilitate direct transfer	S
riextFwd	inbound truck processing is extended forward to facilitate direct transfer	S
rD2D	complete swap of the processing plans at two dock doors	A

and we resort to the ALNS algorithm to find good solutions. The sequential approach can still find optimal solutions for most instances when the utilization of the dock doors is rather low. However, when the utilization increases we only find solutions to both problems in the sequential approach with the use of metaheuristics proposed in the literature. The *truck scheduling problem* is then solved with the ALNS algorithm proposed by Bodnar et al. (2017), whereas the *dock-door assignment problem* is solved with the tabu search (TS) algorithm proposed by Miao et al. (2009). We use these two metaheuristics since they are the only available techniques in the literature for cross-dock terminals with similar characteristics as studied in our paper, which includes the use of mixed-mode dock doors. However, as mentioned before and illustrated in Section 2.8.1, the problem formulation by Bodnar et al. (2017) can result in schedules for which no feasible dock-door assignment exists without preemption. If this happens, we delay these shipments to the earliest timing such that a feasible solution can be created when assigning dock doors to the scheduled trucks. Furthermore, we note that the TS algorithm by Miao et al. (2009) is originally designed for a cross-dock terminal where all dock doors operate in a mixed service mode. Consequently, we have to make the following adjustments: inbound and outbound trucks are only assigned to the appropriate dock doors (i.e., to a door in $\mathcal{D}_{IF}$ and $\mathcal{D}_{OF}$, respectively), and the assignment of trucks to dock doors is only swapped between trucks of the same type (either inbound or outbound). The performance of these metaheuristics is compared to our ALNS algorithm in Section 2.6.3.

All experiments are conducted on a single thread of a i7-3520M, 2.9 GHz computer with 8 GB RAM. The ALNS algorithm is implemented in *Python* while the reduced problem is solved with *Gurobi 6.5.1* (Optimization, 2016). The termination criterion for the ALNS algorithm is set to 5,000 iterations (the same as Bodnar et al. (2017)). See Section 2.8.3 for details on the parameter settings for the ALNS algorithm.

This section is structured as follows. First, we describe the procedure to generate our instances. Second, we compare the optimal solutions of the integrated approach against those of the sequential approach for small instances. Third, we present the performance of the integrated ALNS algorithm against a sequential approach for medium and large-sized instances where we vary the utilization of the dock doors. Fourth, we analyze several strategies to position mixed-mode dock doors. The section concludes reporting the results of our case study where the integrated solutions are compared against the sequential solutions for each of the four cross-dock terminals of the retailer under study.

2.6.1 Generation of instances

The instances included in our testbed are similar to those studied by Bodnar et al. (2017). We consider cross-dock terminals with 5, 10 or 20 dock doors where 30, 60, 90, 120 or 150 trucks need to be scheduled and assigned. The average number of outbound trucks supplied by each inbound truck is either 4 or 6. This determines the utilization of the dock doors, measured by Bodnar et al. (2017) as

$$utilization(\%) = \frac{|\mathcal{I}| + |\mathcal{O}|}{|\mathcal{D}| \times |\mathcal{T}|} \times 100\%.$$

We use the same definition in our paper. Furthermore, the arrival and departure time windows are uniformly distributed over the planning horizon and have a length of 2 or 3 time periods. See Bodnar et al. (2017) for more details.

Since we extend the model formulation of Bodnar et al. (2017) with dock-door assignment decisions, we also have to specify travel distances between dock doors and indicate which of the doors operate in an exclusive or mixed service mode. For our comprehensive experiments we consider a U-shaped cross-dock terminal with dock doors on the outersides since this is the most common shape among unit-load cross-dock operations with temporary storage. Note that other shapes can be used as well. For instance, the layout of an I-shaped cross-dock facility is specified by two sequences of service modes (one for each side of the I shape). To illustrate that our integrated solution approach is also applicable to cross-dock terminals with another shape, we apply our methodology to an I-shaped cross-dock facility in Section 2.8.6. However, our main analysis in this section is restricted to U-shaped facilities only. Investigating all shapes in detail is merely a numerical exercise and beyond the scope of this paper.

The layout of a U-shaped cross-dock terminal is specified by a single sequence of service modes corresponding to the service modes for each of the dock doors. The proportion of mixed-mode dock doors ranges from 0% to 60% of the total number of dock doors (in multiples of 20%). Results by Berghman et al. (2015) and Bodnar et al. (2017) indicate that adding more flexible doors beyond the 60% has a negligible impact. We consider three configurations for the positioning of the inbound, outbound, and flexible dock doors. In each configuration the inbound dock doors are located in the centre with the outbound dock doors on either side (consistent with the observation of Bartholdi III & Gue (2000) and our case study in Section 2.6.5). The flexible dock doors are placed either in the center of the inbound doors, in the center of the outbound doors or at the outer limits of the cross-dock terminal. These strategies are referred to as the center, half-center and outer strategies, respectively. Each of the layout configurations is illustrated in Table 2.5.

Lastly, we need to specify the objective function coefficients. To express the travel distance in meters, we set the dock-to-dock transfer cost η equal to one and let the distance between two dock doors k, $p \in \mathcal{D}$ equal $m_{kp} = 3.6|k - p|$, since the travel distance between two adjacent dock doors in each layout configuration is 3.6 meters (which is equivalent to the length of 3 unit loads or pallets). The temporary storage cost β is set to twice the average dock-to-dock travel distance between all dock doors and approximates the travel distance of a unit load transferred between two randomly selected doors via the temporary storage area. In particular,

$$\beta = 2 \cdot 3.6 \cdot \sum_{i=1}^{|D|-1} i(i+1) / \left(|D| \cdot (|D| - 1)\right).$$

The delay cost γ is set to $2\,\beta$ or $10\,\beta$, which is the same as in Bodnar et al. (2017).

Table 2.5: The layout configurations of the inbound doors (i), outbound doors (o) and flexible doors (f) at a cross-dock terminal with either 5, 10 or 20 dock doors

layout	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{D}_I\rvert$	$\lvert\mathcal{D}_O\rvert$	$\lvert\mathcal{D}_F\rvert$	Mixed-mode Strategy	Layout Configuration
M1	5	2	3	0	None	$\Vert o\Vert o\Vert i\Vert i\Vert o\Vert$
M2	5	2	2	1	Center	$\Vert o\Vert i\Vert f\Vert i\Vert o\Vert$
M3	5	2	2	1	Half-center	$\Vert o\Vert f\Vert i\Vert i\Vert o\Vert$
M4	5	2	2	1	Outer	$\Vert f\Vert o\Vert i\Vert i\Vert o\Vert$
M5	5	1	2	2	Center	$\Vert o\Vert i\Vert f\Vert f\Vert o\Vert$
M6	5	1	2	2	Half-center	$\Vert o\Vert f\Vert i\Vert f\Vert o\Vert$
M7	5	1	2	2	Outer	$\Vert f\Vert o\Vert i\Vert o\Vert f\Vert$
M8	5	1	1	3	Center	$\Vert o\Vert f\Vert f\Vert f\Vert i\Vert$
M9	5	1	1	3	Half-center	$\Vert o\Vert f\Vert i\Vert f\Vert f\Vert$
M10	5	1	1	3	Outer	$\Vert f\Vert f\Vert o\Vert i\Vert f\Vert$
M11	10	5	5	0	None	$\Vert o\Vert o\Vert o\Vert i\Vert i\Vert i\Vert i\Vert i\Vert o\Vert o\Vert$
M12	10	4	4	2	Center	$\Vert o\Vert o\Vert i\Vert i\Vert f\Vert f\Vert i\Vert i\Vert o\Vert o\Vert$
M13	10	4	4	2	Half-center	$\Vert o\Vert f\Vert o\Vert i\Vert i\Vert i\Vert i\Vert o\Vert f\Vert o\Vert$
M14	10	4	4	2	Outer	$\Vert f\Vert o\Vert o\Vert i\Vert i\Vert i\Vert i\Vert o\Vert o\Vert f\Vert$
M15	10	2	2	4	Center	$\Vert o\Vert o\Vert i\Vert f\Vert f\Vert f\Vert f\Vert i\Vert i\Vert o\Vert$
M16	10	2	2	4	Half-center	$\Vert o\Vert f\Vert f\Vert o\Vert i\Vert i\Vert i\Vert f\Vert f\Vert o\Vert$
M17	10	2	2	4	Outer	$\Vert f\Vert f\Vert o\Vert o\Vert i\Vert i\Vert i\Vert o\Vert f\Vert f\Vert$
M18	10	2	2	6	Center	$\Vert o\Vert i\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert i\Vert o\Vert$
M19	10	2	2	6	Half-center	$\Vert o\Vert f\Vert f\Vert f\Vert i\Vert i\Vert f\Vert f\Vert f\Vert o\Vert$
M20	10	2	2	6	Outer	$\Vert f\Vert f\Vert f\Vert o\Vert i\Vert i\Vert o\Vert f\Vert f\Vert f\Vert$
M21	20	10	10	0	None	$\Vert o\Vert o\Vert o\Vert o\Vert o\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert o\Vert o\Vert o\Vert o\Vert o\Vert$
M22	20	8	8	4	Center	$\Vert o\Vert o\Vert o\Vert o\Vert i\Vert i\Vert i\Vert i\Vert f\Vert f\Vert f\Vert f\Vert i\Vert i\Vert i\Vert i\Vert o\Vert o\Vert o\Vert o\Vert$
M23	20	8	8	4	Half-center	$\Vert o\Vert o\Vert f\Vert f\Vert o\Vert o\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert o\Vert o\Vert f\Vert f\Vert o\Vert o\Vert$
M24	20	8	8	4	Outer	$\Vert f\Vert f\Vert o\Vert o\Vert o\Vert o\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert o\Vert o\Vert o\Vert o\Vert f\Vert f\Vert$
M25	20	6	6	8	Center	$\Vert o\Vert o\Vert o\Vert i\Vert i\Vert i\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert i\Vert i\Vert i\Vert o\Vert o\Vert o\Vert$
M26	20	6	6	8	Half-center	$\Vert o\Vert o\Vert f\Vert f\Vert f\Vert f\Vert o\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert o\Vert f\Vert f\Vert f\Vert f\Vert o\Vert o\Vert$
M27	20	6	6	8	Outer	$\Vert f\Vert f\Vert f\Vert f\Vert o\Vert o\Vert o\Vert i\Vert i\Vert i\Vert i\Vert i\Vert i\Vert o\Vert o\Vert o\Vert f\Vert f\Vert f\Vert f\Vert$
M28	20	4	4	12	Center	$\Vert o\Vert o\Vert i\Vert i\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert i\Vert i\Vert o\Vert o\Vert$
M29	20	4	4	12	Half-center	$\Vert o\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert o\Vert i\Vert i\Vert i\Vert i\Vert o\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert o\Vert$
M30	20	4	4	12	Outer	$\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert o\Vert o\Vert i\Vert i\Vert i\Vert i\Vert o\Vert o\Vert f\Vert f\Vert f\Vert f\Vert f\Vert f\Vert$

2.6.2 Intergated versus sequential solutions, small instances

For each instance, we report the relative performance of the integrated solution approach compared to a benchmark as well as the value of the objective function for the benchmark solution. The relative performance is calculated as $\Delta(\%) = ((z_A(x) - z_B(x))/z_B(x)) \cdot 100\%$, where $z_A(x)$ and $z_B(x)$ denote the objective function values corresponding to the solutions generated by solution technique A and B, respectively, where B is our benchmark. Note that a positive (negative) value for $\Delta(\%)$ indicates that technique A results in an inferior (superior) solution compared to the benchmark solution (which is given by technique B).

For the smallest instances, where $|D| = 5$, the benchmark solution is the best solution found by solving the MIP formulations of the scheduling and assignment problem in the sequential approach as defined in Section 2.8.2. We terminate the solution procedure for each problem if no optimal solution is found within 1,800 seconds, which is similar to Bodnar et al. (2017) (i.e., the solution procedure takes at most 3,600 seconds). The corresponding objective function value is reported as *Seq. obj.* We compare the performance of the best solution found by solving the MIP formulation of the cross-dock scheduling problem as defined in Section 2.4 and the performance of the solution found by our ALNS algorithm against the best solution found with the sequential approach. This is indicated by *Int.* $\Delta(\%)$ and *ALNS* $\Delta(\%)$, respectively. If the optimal solution for the integrated cross-dock scheduling problem is not found within 3,600 seconds, we terminate the solver and report the best found solution. The results are presented in Table 2.6. When *Gurobi* is able to find the optimal solution to the MIP formulations, the performance is reported in bold. The results show that the optimal solution for the sequential approach is found in 56 out of 80 instances. In contrast, the integrated approach finds the optimal solution in only 24 instances. However, the best solution to the integrated MIP formulation is for most instances superior to the best solution in the sequential approach, but there are also a number of instances (when there are more mixed-mode dock doors) where the integrated approach cannot find a good solution within 3,600 seconds. The results show that our ALNS algorithm is a good alternative solution approach to solve the integrated cross-dock scheduling problem with an average improvement of 2.12% compared to the best solution in the sequential approach.

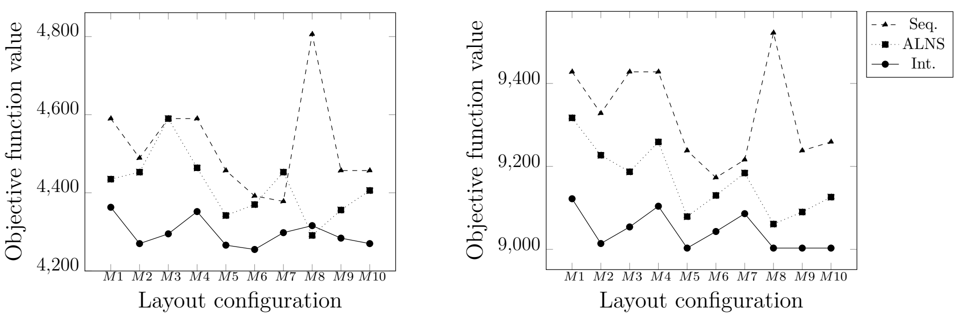
The interesting result from Table 2.6 is that the optimal solutions for the sequential approach cannot guarantee that the operational costs of a cross-dock terminal decrease when (more) mixed-mode dock doors are included. This contradicts the results from

Table 2.6: Performance of the integrated solution approach compared to sequential solution approach for small-sized instances with 5 dock doors

	$\eta = 1, \beta = 14.4, \gamma = 28.8$						$\eta = 1, \beta = 14.4, \gamma = 144$					
	$d_i - r_i = 2$			$d_i - r_i = 3$			$d_i - r_i = 2$			$d_i - r_i = 3$		
layout	Seq. obj.	Int. Δ(%)	ALNS Δ(%)	Seq. obj.	Int. Δ(%)	ALNS Δ(%)	Seq. obj.	Int. Δ(%)	ALNS Δ(%)	Seq. obj.	Int. Δ(%)	ALNS Δ(%)
(a) nr. of trucks = 30, average number of destinations per inbound truck = 4, utilization = 37.5%												
M1	**4590**	**-5%**	-3%	**4558**	-9%	-9%	**9428**	**-3%**	-1%	**11138**	-5%	-4%
M2	**4489**	**-5%**	-1%	4363	-5%	-5%	**9328**	**-3%**	-1%	**11009**	-5%	-3%
M3	**4590**	**-6%**	0%	4482	-6%	-3%	**9428**	**-4%**	-3%	**11077**	-5%	-3%
M4	**4590**	**-5%**	-3%	4558	-5%	-7%	**9428**	**-3%**	-2%	**11138**	-5%	-3%
M5	**4457**	-4%	-3%	4270	-6%	-6%	**9238**	**-3%**	-2%	**10951**	-5%	-5%
M6	**4392**	-3%	0%	4183	6%	1%	**9173**	**-1%**	0%	**10865**	-4%	-2%
M7	**4378**	-2%	2%	4262	6%	0%	**9216**	**-1%**	0%	**10944**	-3%	-2%
M8	**4806**	-10%	-11%	4583	1%	-8%	**9522**	**-5%**	-5%	**11344**	-6%	-6%
M9	**4457**	-4%	-2%	4222	7%	0%	**9238**	**-3%**	-2%	**10951**	-4%	-1%
M10	**4457**	-4%	-1%	4291	37%	-5%	**9259**	**-3%**	-1%	**10951**	-5%	-2%
(b) nr. of trucks = 30, average number of destinations per inbound truck = 6, utilization = 37.5%												
M1	**5472**	**-6%**	-4%	**6206**	2%	-4%	**11110**	**-2%**	-2%	**16564**	-1%	-1%
M2	**5353**	-6%	-4%	6026	9%	-1%	**11059**	**-3%**	-2%	**16448**	-1%	-1%
M3	**5429**	-6%	-3%	6206	0%	-4%	**11074**	**-2%**	-2%	**16560**	2%	-1%
M4	**5486**	-6%	-4%	6206	6%	-3%	**11124**	**-2%**	-2%	**16560**	0%	0%
M5	**5314**	-4%	-2%	5904	8%	-1%	**10919**	**-1%**	-1%	15915	3%	0%
M6	**5231**	-1%	0%	5904	4%	1%	**10894**	**-1%**	0%	15854	5%	1%
M7	**5238**	-1%	-1%	6044	7%	-1%	**11009**	**-1%**	-1%	16001	4%	1%
M8	**5569**	-7%	-6%	6070	12%	-3%	**11027**	1%	-2%	16092	2%	0%
M9	**5310**	-4%	-4%	5868	9%	0%	**10901**	**-1%**	0%	15926	27%	1%
M10	**5310**	-4%	-3%	5918	21%	0%	**10930**	**-2%**	-1%	15944	3%	1%

Seq. obj. is the objective function value of the best solution for the MIP formulations in the sequential approach, which serves as benchmark; *Int.* Δ(%) is the relative performance of the best solution for the MIP formulation in the integrated approach; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm. Any performance in bold indicates that the optimal solution is found for the MIP formulation.

Berghman et al. (2015) and Bodnar et al. (2017) when only scheduling decisions are considered. Figure 2.2 illustrates the objective function values for the instances with on average 4 outbound trucks per inbound truck and the length of the arrival and departure time window equals 2 periods. In both Figure 2.2a (low delay cost) and Figure 2.2b (high delay cost), the optimal solutions in the sequential approach for the instances with 60% mixed-mode dock doors (i.e., layout *M8*, *M9* and *M10*) are worse than the optimal solutions in the sequential approach for the instances with 40% mixed-mode dock doors (i.e., layout *M5*, *M6* and *M7*). In particular, the optimal solutions found with the sequential approach for layout *M8* are worse than the solutions for any other layout configuration (even compared to layout *M1* with no mixed-mode dock doors). The reason is that the sequential approach has no mechanism to consider the potential impact of a schedule on the assignment decisions in the second stage. As a result, the myopic scheduling decisions in the first stage substantially increase the total travel distance when the trucks are assigned to dock doors. In contrast, when the integrated approach is used to solve the cross-dock scheduling problem, we can claim that it is always beneficial to use more mixed-mode dock doors (consider Figure 2.2b where all solutions in the integrated approach are optimal). These results illustrate the value of our integrated approach over any sequential method.



(a) $\gamma = 2\beta$ **(b)** $\gamma = 10\beta$

Figure 2.2: The objective function values of the best solutions for the MIP formulations in the sequential and integrated approach as well as the solution with our ALNS algorithm for instances with 5 dock doors, 30 trucks, $d_i - r_i = 2$ and on average 4 outbound trucks per inbound truck

2.6.3 Performance of ALNS algorithm for integrated problem

For instances with more than five dock doors, we cannot solve the MIP formulation of the integrated cross-dock scheduling problem within 3,600 seconds. Therefore, we compare the performance of solutions found with our integrated ALNS algorithm against solutions of the sequential approach. We first present the results for instances where the dock-door utilization is low followed by instances where the dock-door utilization is high.

Low dock-door utilization
For the instances where $|\mathcal{D}| = 10$, the benchmark solution is the best solution found for the MIP formulations in the sequential approach (similar to Section 2.6.2). Besides the relative performance of the solution found with our ALNS algorithm (denoted by *ALNS* $\Delta(\%)$), we also report the relative performance of the solutions when existing metaheuristics from the literature are used to solve the problems in the sequential approach (denoted by *Seq MH* $\Delta(\%)$). For the instances where $|\mathcal{D}| = 20$, we cannot find good solutions for the MIP formulations in the sequential approach within a reasonable time. Consequently, the solution resulting from the existing metaheuristics becomes the benchmark solution, and the relative performance of our integrated ALNS algorithm against this benchmark solution is still denoted as *ALNS* $\Delta(\%)$. The results are presented in Table 2.7 and Table 2.8 for instances with 10 dock doors and 30 and 60 trucks, respectively. Table 2.9 and Table 2.10 presents results for the instances 20 dock doors and 60 and 120 trucks, respectively.

From the results in Table 2.7 and Table 2.8, it becomes clear that our ALNS algorithm finds solutions that outperform even the optimal solution to the MIP formulations in the sequential approach for most instances, with an average improvement of 4%. The integrated ALNS algorithm performs even better when the cost of delays is low ($\gamma = 2\beta$) rather than high ($\gamma = 10\beta$), with average improvements of 6.5% and 1.5%, respectively (ANOVA $p < 0.001$). This is to be expected as it is more cost effective to avoid delays when γ is higher. The first stage of the sequential approach (i.e., solving the scheduling problem) will exactly do this. The integrated approach will do something similar and also ends up favoring scheduling decisions, such that the benefits of simultaneously considering assignment decisions can be less substantial. Furthermore, as the utilization of the dock doors is rather low in these instances (18.75% and 37.5%), there is flexibility to find good dock-door assignment decisions in the sequential approach. Nevertheless, the solutions of our integrated ALNS algorithm are considerably better than the best solutions in the sequential approach.

The results in both tables indicate that solutions found with the existing metaheuristics in the sequential approach perform notably inferior. In Table 2.7 and Table 2.8, these solutions are on average 24.1% worse than the best solutions for the MIP formulations in the sequential approach, with a maximum optimality gap of 78.2%. Especially when there are mixed-mode dock doors, the metaheuristics in the sequential approach perform poorly. Furthermore, the average cost increase changes from 18% to 30.3% when the dock-door utilization increases from 18.75% to 37.5% (the difference is statistically significant, ANOVA $p < 0.001$). The metaheuristics in the sequential approach are also not performing well for the instances in Table 2.9 and Table 2.10, where the average improvement by using our ALNS algorithm to solve the cross-dock scheduling problem compared to the metaheuristics in the sequential approach results in a reduction of 24% for the objective function value. Similar to Table 2.7 and Table 2.8, the relative performance is even better when there are mixed-mode dock doors (the average improvement is then 26.4%)[2].

As discussed in Section 2.8.1, the problem formulation by Bodnar et al. (2017) only includes the number of trucks scheduled at dock doors in a certain service mode. This can result in a truck that needs to be assigned to a flexible door in one time period and to a dedicated door in another time period in order to find a feasible dock-door assignment. Since this type of preemption is not allowed in our problem formulation, we revised the problem formulation in Section 2.8.2 such that the truck scheduling problem generates schedules for which a feasible dock-door assignment can always be found with the sequential approach. Interestingly, our integrated ALNS algorithm finds solutions that are consistently superior compared to the solutions from the optimal MIP solutions in the sequential approach (as indicated in Table 2.7 and Table 2.8). For the instances in Table 2.9 and Table 2.10, the sequential approach can only use the metaheuristics to find a solution within a reasonable computational time. However, the schedule resulting from the ALNS algorithm of Bodnar et al. (2017) has to be adjusted (i.e., we delay activities) if no feasible assignment is possible. This is the main reason why the metaheuristics from the literature do not perform well in the sequential approach (in both tables). But even without this issue, our integrated ALNS algorithm finds significantly superior solutions compared to the solutions with

[2] To calculate the performance of our ALNS algorithm relative to the metaheuristics in the sequential approach in Table 2.7 and Table 2.8, we consider (*ALNS* Δ(%) - *Seq. MH* Δ(%)) $\cdot$ 100 / (100 + *Seq. MH* Δ(%)).

Table 2.7: Performance of our integrated ALNS algorithm and the existing metaheuristics from the literature for the sequential approach compared to the best solutions to the MIP formulations in the sequential approach for instances with 10 dock doors, 30 trucks and low dock-door utilization rates

	$\eta = 1, \beta = 26.4, \gamma = 52.8$						$\eta = 1, \beta = 26.4, \gamma = 264$					
	$d_i - r_i = 2$			$d_i - r_i = 3$			$d_i - r_i = 2$			$d_i - r_i = 3$		
layout	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)
(a) nr. of trucks = 30, average number of destinations per inbound truck = 4, utilization = 18.75%												
M11	**7528**	2%	-7%	**6542**	10%	-8%	**15493**	1%	-3%	**17533**	2%	-3%
M12	**6994**	10%	-4%	5717	64%	-7%	**14837**	9%	-3%	**16718**	18%	-3%
M13	**7224**	23%	-7%	6034	57%	-7%	**15154**	7%	-4%	**17063**	19%	-4%
M14	**7256**	10%	-6%	6185	58%	-7%	**15182**	7%	-3%	**17068**	35%	-4%
M15	**6895**	17%	-6%	**5587**	78%	-10%	**14428**	10%	-1%	**16274**	21%	-3%
M16	**7176**	16%	-9%	**6178**	50%	-13%	**14719**	4%	-2%	**16609**	16%	-3%
M17	**7180**	17%	-9%	**6178**	60%	-11%	**14737**	5%	-3%	**16609**	29%	-2%
M18	**6872**	7%	-7%	**5154**	36%	-1%	**14334**	2%	-2%	**16116**	18%	-1%
M19	**7002**	10%	-7%	**5438**	68%	-9%	**14456**	2%	-3%	**16404**	12%	-2%
M20	**7056**	7%	-8%	**5428**	62%	-6%	**14460**	6%	-2%	**16447**	23%	-1%
(b) nr. of trucks = 30, average number of destinations per inbound truck = 6, utilization = 18.75%												
M11	**8707**	3%	-9%	**9307**	1%	-5%	**18203**	0%	-3%	**26044**	1%	-3%
M12	**7892**	16%	-6%	8420	12%	-6%	**17497**	20%	-4%	**24367**	14%	-1%
M13	**8213**	10%	-8%	8816	14%	-8%	**17767**	12%	-4%	**24900**	4%	-2%
M14	**8278**	9%	-6%	8910	13%	-7%	**17821**	14%	-4%	**24900**	4%	-1%
M15	**7422**	15%	-2%	7921	18%	0%	**16764**	11%	-1%	**23617**	15%	1%
M16	**7685**	16%	-5%	8314	20%	-4%	**17077**	10%	-2%	**24128**	11%	-1%
M17	**7685**	28%	-6%	8314	24%	-5%	**17077**	10%	-1%	**24128**	9%	0%
M18	**7223**	18%	-1%	**7550**	32%	0%	**16712**	10%	-2%	**23718**	10%	0%
M19	**7597**	8%	-6%	**7925**	25%	-4%	**17018**	17%	-3%	**23905**	12%	0%
M20	**7597**	9%	-5%	**7925**	29%	-2%	**17018**	8%	-2%	**23905**	13%	-1%

Seq. obj. is the objective function value of the best solution for the MIP formulations in the sequential approach, which serves as benchmark; *Seq MH* Δ(%) is the relative performance of the solution found with metaheuristics in the sequential approach; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm. Any objective value in bold indicates that the optimal solution is found in the sequential approach.

Table 2.8: Performance of our integrated ALNS algorithm and the existing metaheuristics from the literature for the sequential approach compared to the best solutions to the MIP formulations in the sequential approach for instances with 10 dock doors, 60 trucks and low dock-door utilization rates

	$\eta = 1, \beta = 26.4, \gamma = 52.8$						$\eta = 1, \beta = 26.4, \gamma = 264$					
	$d_i - r_i = 2$			$d_i - r_i = 3$			$d_i - r_i = 2$			$d_i - r_i = 3$		
layout	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)
(a) nr. of trucks = 60, average number of destinations per inbound truck = 4, utilization = 37.5%												
M11	**14845**	4%	-3%	**12614**	7%	2%	**28478**	4%	-4%	**23478**	10%	0%
M12	13716	43%	1%	13218	13%	-12%	**26983**	56%	-1%	22042	34%	-1%
M13	14411	55%	-8%	14093	21%	-16%	**27530**	46%	-4%	22949	39%	-3%
M14	14497	42%	-6%	14042	8%	-15%	**27606**	25%	-3%	23064	39%	-2%
M15	13193	61%	-1%	12420	37%	-8%	**26134**	71%	3%	21620	53%	-1%
M16	13880	36%	-3%	12823	41%	-12%	**26699**	51%	1%	22380	55%	-3%
M17	13877	50%	-3%	12913	27%	-9%	**26699**	59%	0%	22380	56%	-3%
M18	13004	42%	0%	13115	17%	-19%	25880	49%	3%	21461	45%	4%
M19	12439	51%	6%	13374	9%	-14%	**26154**	42%	0%	21814	40%	-2%
M20	13451	39%	-1%	13414	23%	-13%	**26154**	58%	1%	21814	36%	1%
(b) nr. of trucks = 60, average number of destinations per inbound truck = 6, utilization = 37.5%												
M11	17345	4%	-1%	**16414**	3%	3%	**36408**	3%	0%	**31828**	2%	0%
M12	15924	31%	0%	18253	7%	-18%	**34926**	34%	-2%	28976	50%	-1%
M13	16622	19%	-1%	18606	16%	-14%	**35473**	31%	-2%	29376	31%	-1%
M14	16669	29%	-1%	18775	1%	-14%	**35668**	36%	0%	29502	40%	-2%
M15	16934	21%	-6%	18031	18%	-16%	**34208**	36%	1%	28020	33%	-1%
M16	17189	29%	-4%	18272	8%	-17%	**34637**	34%	-1%	28373	37%	0%
M17	17262	24%	-4%	18326	14%	-18%	**34651**	44%	-1%	28387	50%	-3%
M18	15605	35%	3%	16805	7%	-13%	34014	26%	2%	28490	27%	-1%
M19	15803	26%	0%	16848	17%	-12%	34230	26%	-1%	28782	13%	-3%
M20	15832	39%	-4%	16848	10%	-10%	34230	43%	-1%	28771	27%	-2%

Seq. obj. is the objective function value of the best solution for the MIP formulations in the sequential approach, which serves as benchmark; *Seq MH* Δ(%) is the relative performance of the solution found with metaheuristics in the sequential approach; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm. Any objective value in bold indicates that the optimal solution is found in the sequential approach.

the metaheuristics in the integrated approach. Just consider the results for layout $M21$, since these instances do not have any mixed-mode dock doors (and therefore do not suffer from the preemption issue). The results show that our integrated ALNS approach finds solutions that are on average 6.5% better than the metaheuristics solutions in the sequential approach.

High dock-door utilization

With 10 dock doors, when we increase the number of trucks to be processed at the cross-dock terminal within the same planning horizon to 90, 120 and 150 trucks, the dock-door utilization rate increases to 56.25%, 75% and 93.75%, respectively. The results for these instances are presented in Table 2.11, Table 2.12 and Table 2.13 . Similar to Table 2.9 and Table 2.10, the MIP formulations of the problem cannot be solved in either the sequential or the integrated approach. Therefore, the performance of the solutions from the integrated ALNS algorithm are compared to the solutions from the metaheuristics in the sequential approach. The results of *ALNS* $\Delta(\%)$ in Table 2.11, Table 2.12 and Table 2.13 are similar to the relative performance of the ALNS algorithm in Table 2.7, Table 2.8, Table 2.9 and Table 2.10. This illustrates the consistent superior performance of our integrated ALNS algorithm compared to the metaheuristics in the sequential approach. The only difference is when all dock doors operate in the exclusive service mode: the cost improvements are smaller in the integrated approach. Without mixed-mode dock doors, the integrated solutions are superior by 1.9% compared to 25% with mixed-mode dock doors.

Computational Effort

In Table 2.9, Table 2.10, Table 2.11, Table 2.12 and Table 2.13, we also report the computational effort to find a solution for the cross-dock scheduling problem with the metaheuristics in the sequential approach and the ALNS algorithm in the integrated approach. It becomes apparent that the ALNS algorithm requires significantly more CPU time. Although the computation times of the metaheuristics in the sequential approach are more favourable, the solution quality is significantly inferior. An alternative to improve the solution quality in the sequential approach (without adopting the integrated approach) is to generate multiple sequential solutions and choose the best one. However, such a strategy would make the computation times go up and it cannot guarantee an improved solution quality. Therefore, we propose our ALNS algorithm to solve the integrated cross-dock scheduling problem. One can reduce the number of iterations of the ALNS algorithm to find a solution faster. In Section 2.8.5 we explore the performance of the solution found with the ALNS

Table 2.9: Performance of our integrated ALNS algorithm compared to the existing metaheuristics from the literature for the sequential approach for instances with 20 dock doors, 60 trucks and low dock-door utilization rates

	$\eta = 1, \beta = 50.4, \gamma = 100.8$								$\eta = 1, \beta = 50.4, \gamma = 504$							
	$d_i - r_i = 2$				$d_i - r_i = 3$				$d_i - r_i = 2$				$d_i - r_i = 3$			
	Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS	
layout	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time
(a) nr. of trucks = 60, average number of destinations per inbound truck = 4, utilization = 18.75%																
M21	24826	299	-12%	2254	21046	340	-18%	1721	49604	205	-8%	2117	40111	317	-12%	2217
M22	29110	296	-35%	2600	25052	308	-40%	2119	53510	236	-18%	2547	49698	306	-34%	2618
M23	28357	295	-32%	2747	24325	323	-37%	2225	59072	240	-26%	2716	45000	281	-25%	2713
M24	29671	259	-34%	2654	25027	374	-33%	1922	51008	269	-14%	2687	56196	273	-41%	2765
M25	28267	287	-32%	2979	27101	427	-46%	1758	48834	259	-14%	3053	41540	312	-23%	3196
M26	26978	306	-30%	3105	27187	371	-44%	1610	51570	240	-15%	2924	43340	253	-27%	3317
M27	28537	374	-34%	3035	25988	374	-43%	1644	52405	298	-18%	3098	42491	302	-21%	2366
M28	23584	361	-23%	3291	26662	359	-44%	1893	48809	336	-12%	3317	40450	303	-20%	2400
M29	23641	313	-22%	3383	22428	320	-37%	1873	50216	265	-14%	2997	42430	358	-25%	2163
M30	27356	377	-33%	3006	22050	383	-35%	2789	51815	279	-18%	3311	43657	361	-27%	2304
(b) nr. of trucks = 60, average number of destinations per inbound truck = 6, utilization = 18.75%																
M21	28836	398	-7%	2203	25592	348	-13%	2275	63450	293	-4%	2165	50288	351	-8%	1686
M22	31072	295	-24%	2803	28613	404	-30%	2815	72846	277	-22%	2872	53291	354	-19%	1952
M23	33383	320	-27%	2737	31424	357	-34%	2836	72439	275	-20%	2686	52582	387	-17%	1953
M24	35147	320	-30%	2821	32789	366	-34%	2844	74858	290	-23%	2715	56664	358	-22%	2002
M25	28940	294	-22%	2704	25902	310	-28%	3212	66618	289	-16%	3154	59162	330	-27%	2145
M26	30611	301	-23%	2626	26986	301	-26%	3380	66636	255	-15%	3046	55285	370	-22%	2220
M27	32472	317	-27%	3028	30074	248	-35%	3383	71633	315	-21%	3054	54929	322	-21%	2155
M28	28400	307	-21%	3474	27482	277	-29%	3442	62708	269	-10%	3259	52348	332	-20%	2370
M29	29254	326	-23%	2668	29776	291	-33%	3767	64876	284	-13%	3408	53863	307	-23%	2411
M30	29646	384	-24%	2681	31619	309	-39%	3957	66467	269	-16%	3285	49763	315	-13%	2500

Seq. MH is the objective function value of the solution found with the metaheuristics in the sequential approach, which serves as benchmark; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm; Computation times (in seconds) are represented by *time*.

Table 2.10: Performance of our integrated ALNS algorithm compared to the existing metaheuristics from the literature for the sequential approach for instances with 20 dock doors, 120 trucks and low dock-door utilization rates

	$\eta = 1, \beta = 50.4, \gamma = 100.8$								$\eta = 1, \beta = 50.4, \gamma = 504$							
	$d_i - r_i = 2$				$d_i - r_i = 3$				$d_i - r_i = 2$				$d_i - r_i = 3$			
	Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS	
layout	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time
(a) nr. of trucks = 120, average number of destinations per inbound truck = 4, utilization = 37.5%																
M21	63346	1071	-6%	2495	48679	551	0%	3021	123070	683	0%	3260	84827	633	-4%	3080
M22	72382	813	-23%	3571	63760	776	-31%	3999	149036	518	-20%	4132	118807	841	-35%	4429
M23	83286	831	-33%	3539	61362	607	-25%	4130	149836	552	-20%	4407	123944	779	-36%	4279
M24	79279	736	-29%	3589	58331	608	-23%	3452	149576	549	-21%	4696	129280	800	-38%	4184
M25	74225	660	-29%	4183	61866	659	-33%	3616	154300	602	-25%	5122	132469	703	-42%	5164
M26	76583	904	-29%	4271	61420	532	-28%	3761	161568	476	-27%	5015	120715	909	-36%	5555
M27	79240	823	-32%	4024	64382	644	-34%	3880	171212	507	-33%	5515	127026	739	-38%	5050
M28	69926	835	-25%	4291	57809	554	-31%	4446	144972	539	-21%	5740	115297	667	-33%	5907
M29	76972	893	-30%	4646	63871	596	-35%	4317	142175	646	-18%	5800	118616	710	-35%	5860
M30	72983	715	-28%	4874	63450	621	-34%	4268	147838	541	-25%	5932	118631	696	-36%	5807
(b) nr. of trucks = 120, average number of destinations per inbound truck = 6, utilization = 37.5%																
M21	67324	1067	-4%	2723	61175	610	-3%	2170	134021	669	-4%	3314	107053	737	-1%	3188
M22	74160	971	-20%	3481	73264	658	-26%	2358	151020	573	-18%	4531	136159	743	-24%	3906
M23	74884	955	-18%	3509	72608	721	-24%	2377	151823	601	-18%	4107	135144	675	-25%	4124
M24	79268	814	-22%	3597	76896	607	-28%	2702	157982	626	-22%	4475	136843	691	-26%	4070
M25	76842	958	-24%	4019	74164	533	-28%	3173	165146	668	-26%	5010	142970	787	-29%	4849
M26	76021	936	-22%	4533	72324	644	-24%	2864	164542	623	-26%	4922	144212	754	-31%	4781
M27	74794	865	-20%	4003	72670	612	-27%	3421	155293	605	-21%	5075	138679	765	-28%	4860
M28	72918	937	-19%	3507	70931	677	-26%	3667	146689	742	-18%	5580	141811	922	-31%	5755
M29	74358	873	-21%	4873	73030	697	-27%	3382	147766	824	-18%	6054	143978	712	-31%	6193
M30	75427	672	-22%	3906	73591	633	-30%	3509	160060	652	-24%	5965	117695	671	-17%	5608

Seq. MH is the objective function value of the solution found with the metaheuristics in the sequential approach, which serves as benchmark; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm; Computation times (in seconds) are represented by *time*.

Table 2.11: Performance of our integrated ALNS algorithm compared to the existing metaheuristics from the literature for the sequential approach for instances with 10 dock doors, 90 trucks and high dock-door utilization rates

	$\eta = 1, \beta = 26.4, \gamma = 52.8$								$\eta = 1, \beta = 26.4, \gamma = 264$							
	$d_i - r_i = 2$				$d_i - r_i = 3$				$d_i - r_i = 2$				$d_i - r_i = 3$			
	Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS	
layout	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time
(a) nr. of trucks = 90, average number of destinations per inbound truck = 4, utilization = 56.25%																
M11	26360	254	-4%	784	24643	327	1%	785	47936	265	-1%	725	46818	259	-2%	756
M12	33019	272	-26%	1135	32905	340	-29%	1188	61042	228	-24%	1199	66763	326	-34%	1174
M13	32924	318	-25%	1231	33689	335	-30%	1122	67946	300	-31%	1180	63178	306	-32%	1317
M14	33586	277	-25%	1265	29592	384	-20%	1234	61913	324	-25%	1151	70777	281	-38%	1252
M15	35201	321	-30%	1348	32626	341	-31%	1335	64943	323	-29%	1328	69308	307	-40%	1551
M16	34122	270	-30%	1540	33961	317	-32%	1385	72426	328	-34%	1405	65867	324	-36%	1558
M17	33926	283	-28%	1626	31674	361	-28%	1495	67134	327	-31%	1426	64158	331	-35%	1607
M18	29267	313	-17%	1587	32414	372	-33%	1760	61567	340	-27%	1478	59945	307	-32%	1934
M19	32202	309	-26%	1674	30683	367	-26%	1586	66433	296	-31%	1531	58638	330	-30%	1477
M20	31369	293	-23%	1794	30050	363	-25%	1759	59774	254	-21%	1626	56501	313	-26%	1758
(b) nr. of trucks = 90, average number of destinations per inbound truck = 6, utilization = 56.25%																
M11	29279	307	0%	798	27826	375	-4%	910	60433	330	-2%	711	51035	307	-2%	814
M12	35432	296	-22%	1182	33056	442	-22%	1302	75108	315	-27%	1233	65498	317	-27%	1157
M13	35551	328	-20%	1152	35917	407	-27%	1309	73949	349	-26%	1374	75408	360	-37%	1207
M14	38646	326	-26%	1120	34136	437	-23%	1177	64914	303	-15%	1218	73296	373	-36%	1355
M15	38562	303	-30%	1434	36114	453	-29%	1459	78283	284	-31%	1470	80442	304	-43%	1252
M16	34661	306	-20%	1648	36144	390	-30%	1560	78241	263	-31%	1458	69916	343	-34%	1590
M17	37530	359	-26%	1564	37595	385	-32%	1569	73571	251	-26%	1524	77215	324	-39%	1460
M18	34885	339	-23%	1676	31698	387	-20%	1546	68033	284	-21%	1438	64974	336	-30%	1744
M19	33022	309	-18%	1790	31938	400	-20%	1739	73673	295	-27%	1755	62120	365	-27%	1812
M20	32531	387	-14%	1709	31459	402	-21%	1754	73646	331	-25%	1531	58673	438	-21%	1771

Seq. MH is the objective function value of the solution found with the metaheuristics in the sequential approach, which serves as the benchmark; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm; Computation times (in seconds) are represented by *time*.

Table 2.12: Performance of our integrated ALNS algorithm compared to the existing metaheuristics from the literature for the sequential approach for instances with 10 dock doors, 120 trucks and high dock-door utilization rates

	$\eta = 1, \beta = 26.4, \gamma = 52.8$								$\eta = 1, \beta = 26.4, \gamma = 264$							
	$d_i - r_i = 2$				$d_i - r_i = 3$				$d_i - r_i = 2$				$d_i - r_i = 3$			
	Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS	
layout	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time
(a) nr. of trucks = 120, average number of destinations per inbound truck = 4, utilization = 75%																
M11	41237	378	-2%	976	32184	393	-2%	1121	89737	481	-1%	1133	58907	407	-14%	1257
M12	47945	359	-16%	2204	42412	467	-27%	1867	106374	461	-19%	2142	82508	514	-40%	1626
M13	45100	429	-12%	1900	41688	405	-25%	1717	106481	399	-20%	2037	61343	378	-17%	1773
M14	47633	509	-15%	2207	41425	451	-25%	1722	104694	571	-15%	2098	73037	456	-30%	1980
M15	56663	477	-33%	2558	44578	567	-34%	2492	119461	481	-33%	2678	83621	610	-43%	2222
M16	52780	532	-28%	2463	47998	500	-36%	2095	115758	689	-32%	2726	92503	448	-46%	2666
M17	50558	383	-24%	2642	46483	447	-33%	2345	112938	546	-29%	2840	96434	594	-48%	2339
M18	47254	510	-23%	2436	39683	406	-27%	2643	101576	610	-24%	2991	72474	573	-35%	2710
M19	48726	464	-23%	2859	41495	508	-27%	3086	101728	561	-23%	2469	82963	555	-42%	2752
M20	48560	409	-23%	2839	42697	476	-31%	2608	104641	601	-27%	2306	85322	605	-43%	2648
(b) nr. of trucks = 120, average number of destinations per inbound truck = 6, utilization = 75%																
M11	44659	444	-3%	1389	39955	333	-6%	1825	100728	505	-3%	1052	79032	459	-5%	1182
M12	50776	486	-16%	1969	45400	399	-19%	1780	110530	629	-17%	1608	82223	466	-23%	1660
M13	54024	510	-22%	2373	44376	466	-20%	1820	114704	628	-22%	1782	75288	456	-16%	2102
M14	49777	498	-14%	2108	46548	516	-23%	1904	128279	629	-29%	1882	74988	372	-15%	2086
M15	52402	535	-22%	2772	49480	549	-30%	2896	117648	526	-28%	2132	89266	375	-31%	2315
M16	54953	575	-25%	2770	50346	511	-30%	2424	128179	618	-34%	2279	94090	417	-32%	2560
M17	55128	494	-25%	2058	46288	448	-23%	2631	131335	550	-34%	2360	88370	369	-30%	2157
M18	50058	517	-21%	2688	42383	624	-17%	3025	103033	665	-19%	2166	89356	448	-31%	3757
M19	50977	545	-21%	3028	43855	559	-21%	3185	120643	669	-32%	2502	80753	593	-26%	2861
M20	48577	581	-14%	2577	43560	490	-21%	2718	106051	614	-20%	3319	88697	445	-28%	2342

Seq. MH is the objective function value of the solution found with the metaheuristics in the sequential approach, which serves as the benchmark; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm; Computation times (in seconds) are represented by *time*.

Table 2.13: Performance of our integrated ALNS algorithm compared to the existing metaheuristics from the literature for the sequential approach for instances with 10 dock doors, 150 trucks and high dock-door utilization rates

	$\eta = 1, \beta = 26.4, \gamma = 52.8$								$\eta = 1, \beta = 26.4, \gamma = 264$							
	$d_i - r_i = 2$				$d_i - r_i = 3$				$d_i - r_i = 2$				$d_i - r_i = 3$			
	Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS		Seq. MH		ALNS	
layout	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time
(a) nr. of trucks = 150, average number of destinations per inbound truck = 4, utilization = 93.75%																
M11	52933	410	3%	1867	50834	426	1%	2030	127774	549	-4%	2155	114839	575	-1%	1845
M12	58507	571	-12%	2369	59464	499	-16%	2759	143333	615	-18%	2154	148992	769	-28%	2792
M13	58393	586	-13%	2147	60218	460	-18%	2104	132049	727	-14%	1841	139327	649	-24%	2738
M14	59324	498	-16%	2335	57634	415	-13%	2291	145838	674	-21%	1713	145698	704	-25%	2779
M15	66907	674	-27%	3181	64698	576	-25%	2892	161375	791	-34%	2134	150300	735	-32%	3012
M16	63884	624	-21%	2662	69227	612	-30%	2688	168138	849	-35%	2452	160240	755	-38%	2436
M17	64220	558	-25%	3068	68562	595	-29%	3591	166025	761	-34%	2133	140878	681	-26%	2755
M18	64787	664	-27%	4058	60164	691	-24%	3389	158116	808	-32%	3425	148692	751	-33%	3563
M19	62561	652	-23%	3362	60690	572	-22%	3998	148730	899	-28%	2223	153914	872	-36%	3866
M20	65729	658	-27%	3070	58106	655	-19%	3583	159475	846	-34%	2545	135200	935	-26%	3317
(b) nr. of trucks = 150, average number of destinations per inbound truck = 6, utilization = 93.75%																
M11	68056	441	0%	1487	56712	478	2%	1620	175016	533	3%	1311	135173	617	0%	1173
M12	71629	582	-12%	2469	63221	640	-13%	2470	186644	675	-17%	2460	150706	730	-14%	2095
M13	71075	477	-9%	2320	62941	735	-10%	2205	183739	707	-13%	2314	171755	664	-26%	2450
M14	69449	543	-8%	2154	63074	657	-11%	2580	185394	639	-15%	2364	158651	658	-19%	2487
M15	70754	744	-12%	2773	69137	884	-22%	2864	192102	918	-21%	2332	170486	746	-30%	2745
M16	77399	657	-21%	2902	70058	788	-23%	3124	197687	892	-24%	2768	167368	824	-29%	2289
M17	72880	625	-16%	2696	73157	754	-26%	3183	191472	825	-21%	2830	174204	770	-30%	2444
M18	68581	676	-14%	3029	58750	904	-11%	3816	186145	945	-22%	3481	142200	1023	-18%	2781
M19	69721	759	-13%	3423	64440	770	-18%	3654	178421	945	-20%	3394	151986	904	-22%	2799
M20	71956	660	-17%	3328	65246	816	-18%	3733	184830	923	-21%	3330	154349	846	-24%	2817

Seq. MH is the objective function value of the solution found with the metaheuristics in the sequential approach, which serves as the benchmark; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm; Computation times (in seconds) are represented by *time*.

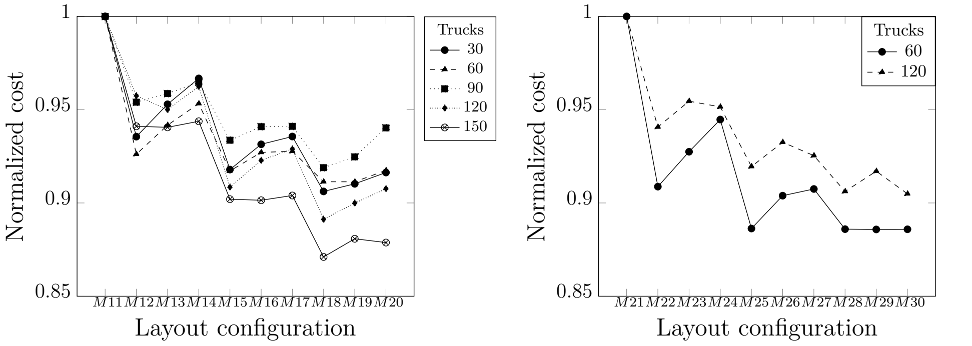
algorithm over the number of iterations. For consistency, all numerical results in this section were generated with 5,000 iterations.

Lastly, we also performed numerical experiments for instances with non-unit processing times (i.e., $h_j = 4$ for each outbound truck j, similar to Bodnar et al. (2017)). The results are presented in Section 2.8.4. Similar observations can be found as reported in this section: solutions from the integrated approach are more superior when the delay cost γ is low and when the utilization rate is high.

2.6.4 Proportion and position of mixed-mode dock doors

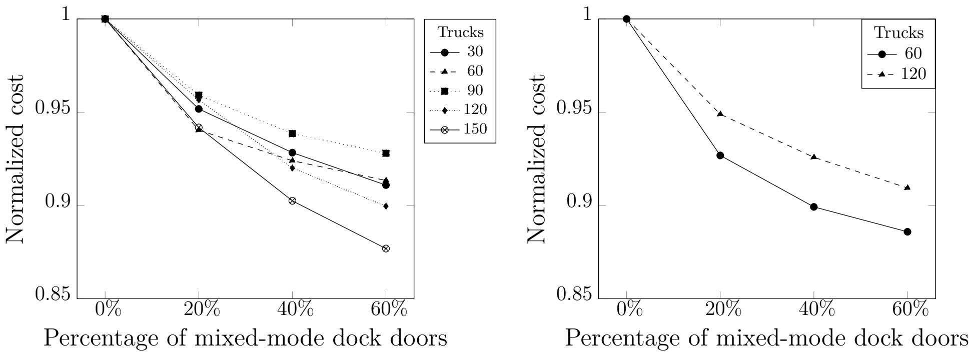
All numerical results in Table 2.6 until Table 2.13 report the performance when we vary the proportion of mixed-mode dock doors (i.e., 0% until 60% in steps of 20%) as well as the position of the mixed-mode dock doors (i.e., center, half-center and outer configuration strategies) indicated by the configuration layout. The results show that the number of mixed-mode dock doors and their position have a significant impact on the operational performance. Figure 2.3a and Figure 2.3b summarize the results for the integrated ALNS algorithm over the instances with 10 dock doors (i.e., Table 2.7 and Table 2.8, Table 2.11, Table 2.12 and Table 2.13) and 20 dock doors (i.e., Table 2.9 and Table 2.10), respectively. The performance of the solution for the configuration without any mixed-mode dock doors (M11 and M21, respectively) is used as benchmark, and the average relative performance for each configuration and dock-door utilization is reported. A value close to one indicates that the operational costs are similar to the layout without any mixed-mode dock doors. It becomes clear that the total operational costs decrease when more mixed-mode dock doors are included in the configuration. Figure 2.4 illustrates this by taking an average over the instances with 20%, 40% and 60% mixed-mode dock doors (i.e., average over M12-M14, M15-M17, M18-M20 in Figure 2.4a). This corresponds to an average cost reduction of 5.35%, 8.03% and 9.66%, respectively, compared to the total operational costs when no mixed-mode dock doors are used. The diminishing marginal returns in the proportion of mixed-mode dock doors is consistent with previous observations in the literature.

From our numerical results in the tables and Figure 2.3 we observe that not only the proportion of the mixed-mode dock doors has a significant impact on the operational performance of a cross-dock terminal, but also the position of these doors. To illustrates this observation, we take an average of the relative performance over the



(a) 10 dock doors **(b)** 20 dock doors

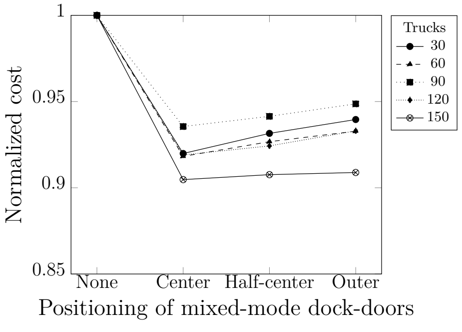
Figure 2.3: The average total operational costs for each configuration layout (i.e., proportion and position of mixed-mode dock doors) relative to the configuration without any mixed-mode dock doors



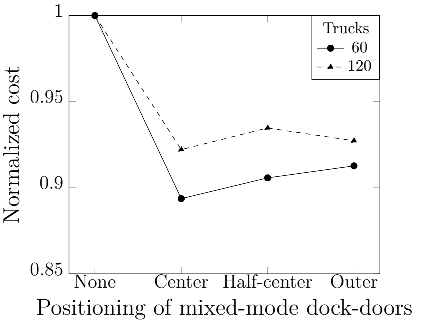
(a) 10 dock doors **(b)** 20 dock doors

Figure 2.4: The average total operational costs for each proportion of mixed-mode dock doors relative to the configuration without any mixed-mode dock doors

instances with the same configuration strategy regarding the position where mixed-mode dock doors are located, i.e., center, half-center and outer strategy (corresponding to the average over M12-M15-M18, M13-M16-M19, M14-M17-M20, respectively). The results are presented in Figure 2.5a and Figure 2.5b for the instances with 10 and 20 dock doors, respectively. The average cost reduction for each strategy is 8.4%, 7.6% and 7.1%, respectively. The pairwise t-tests with Bonferroni corrections (Benjamini & Yekutieli, 2001) and the Games-Howell post-hoc test (Field, 2013) both indicate that differences in some strategies are statistically significant[3]. The strategy without any mixed-mode dock doors performs the worst (p <0.001) for all comparisons. The statistical significance among the three strategies for the position of the mixed-mode dock doors depends on the dock-door utilization rate. First, we discuss the results for the instances with a low dock-door utilization (less than 50%): the center strategy outperforms the outer strategy (p <0.02), whereas the differences between the half-center and outer strategy, and the half-center and center strategy are not statically significant at a 0.05 significance level. However, for the instances with a high dock-door utilization, the differences in strategies are not statistically significant. For the instances with a low dock-door utilization, the total costs are dominated by assignment decisions, which is largely affected by the position of the mixed-mode dock doors. However, for the instances with a high dock-door utilization, the costs associated with the scheduling decisions (i.e., the temporary storage and delay costs) are substantially larger. Therefore, the overall objective function is less affected by the position of the mixed-mode dock doors compared to the proportion of these dock doors. Overall, the results indicate that the center strategy is a good layout design strategy to position the mixed-mode dock doors for unit-load cross-dock terminals where the dock-door utilization is either low or high.



(a) 10 dock doors



(b) 20 dock doors

Figure 2.5: The average total operational costs for each position of the mixed-mode dock doors relative to the configuration without any mixed-mode dock doors

[3]The Games-Howell test is used as the post-hoc test because Levene's test (Levene et al., 1960) for homogeneity of variance is statistically significant.

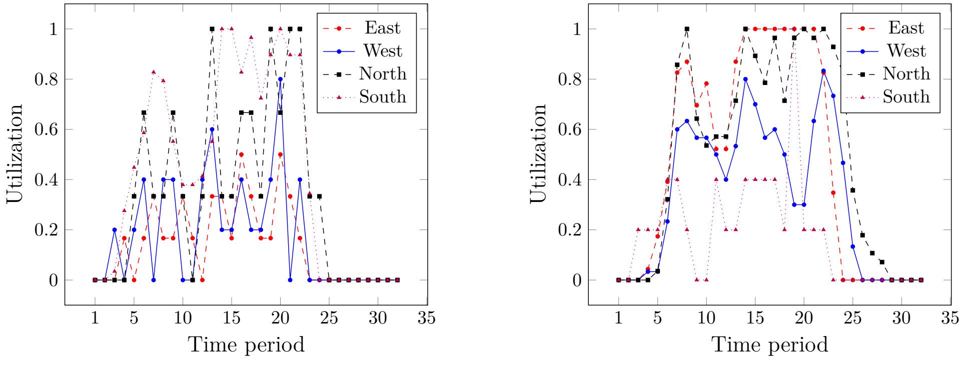
2.6.5 Case study

In this subsection, we apply the solution approaches to the cross-dock scheduling problem in a case study at the largest supermarket chain in the Netherlands. The retailer operates four cross-dock facilities where unit loads from suppliers and national distribution centers are consolidated before they are delivered to retail stores. Each cross-dock terminal has a U-shape and contains 29 to 35 dock doors. The inbound doors are positioned at the center of the terminal and the outbound doors are on either side of the inbound doors (as described in Section 2.6.1). At the beginning of this study, there were no mixed-mode dock doors. Based on the results in Section 2.6.4, we study four layout configurations at each cross-dock terminal: 0%, 20%, 40% and 60% mixed-mode dock doors, where the center strategy is used to position these doors.

Each cross-dock facility processes 187 to 256 trucks on a daily basis. The composition of each inbound and outbound truck is determined in advance (i.e., pre-distribution cross-docking). Each inbound truck carries unit loads for five to seven outbound trucks. All inbound trucks carry an equal number of unit loads and require similar processing times of approximately 30 minutes to unload and transfer their goods. Therefore, time periods have a length of 30 minutes. Unit loads are either directly transferred from the inbound staging lane onto the appropriate assigned outbound dock door (or even directly in the waiting truck), or stored temporarily until the outbound truck is docked at the terminal. The delivery time windows at the retail stores govern the due times of the outbound trucks at the cross-dock terminal, and they also pre-determine the route of each outbound truck.

To compare the performance of solutions generated by our integrated ALNS algorithm to the solutions of the metaheuristics in the sequential approach, we use the exact same data set as used by Bodnar et al. (2017) in their case study. The data set represents the usual dock-door utilization patterns as observed throughout the day. Figure 2.6a and Figure 2.6b show the actual utilization rate of the inbound and outbound dock doors, respectively, for each 30 minute time period. Please see Section 2.8.5 for a description of the current method used by the retailer to make truck scheduling and dock-door assignment decisions.

To determine the objective function coefficients, we use the following information to calculate the labor costs associated with the cross-dock operations. Each cross-dock terminal operates six days a week throughout the year. Each employee (driving either a pallet truck or an inbound/outbound truck) costs €35,000 per year, and (s)he works at 80% productivity for eight hours a day, five days a week and 48 weeks a



(a) Inbound dock-doors **(b)** Outbound dock-doors

Figure 2.6: Utilization rate of inbound and outbound dock-doors over a typical peak day of retailer's cross-dock

year. We assume that the internal transportation within the cross-dock terminal has a speed of two meters per second. Double handling of unit loads in batches of five requires two minutes for pick-up and drop-off operations. The travel distance between a dock door and the temporary storage area is approximated to be twice the average dock-to-dock travel distance, since the temporary storage area is at a substantial distance from the dock-door area in the cross-dock terminal. The cost of a direct dock-to-dock transfer (i.e., η) equals 0.003 Euro per meter[4]. The cost for each unit sent to the temporary storage area depends on the number of dock doors, and is therefore different among the cross-dock facilities: $\beta_{east} = 0.380$, $\beta_{west} = 0.425$, $\beta_{north} = 0.395$, $\beta_{south} = 0.418$[5]. The cost per time period that an outbound truck is delayed is the same for all cross-dock facilities and equals $\gamma = 11.393$[6].

To generate a solution with our integrated ALNS algorithm in a timely manner, we use either 5,000 iterations or 4 hours of computation time (whichever comes first). In Section 2.8.5, we show that more than 99% of the solution improvement compared to the initial solution occurs within the first 1,500 iterations, which is within the 4 hour runtime limit. The results of the case study are presented in Table 2.14, where

[4]Each employee costs €35,000/$(8 \times 5 \times 48 \times 3,600 \times 0.8) = 0.00633$ Euro/sec. and they travel two meters per second.

[5]For example, consider the East facility. It has 29 dock doors, which corresponds to an average travel distance of 10 doors. Since dock doors are 3.6 meters apart and the travel speed is 2 meters per second, it takes 18 seconds to transfer the products either to or from the storage area. Furthermore, the double handling takes an extra 24 seconds per unit. In total, it takes 60 seconds to move one unit from the inbound door to the outbound door through the storage area. This corresponds to a cost of €0.38 for this facility.

[6]The length of a period is 30 minutes and the labor cost for it is 0.00633 Euro/sec.

we report the values corresponding to each of the three components of the objective function for the solutions found with the metaheuristics in the sequential approach and our ALNS algorithm in the integrated approach. A comparison of the total operational costs is provided in Figure 2.7, where the solution found by the metaheuristics in the sequential approach for the instance without any mixed-mode dock doors is used as benchmark solution.

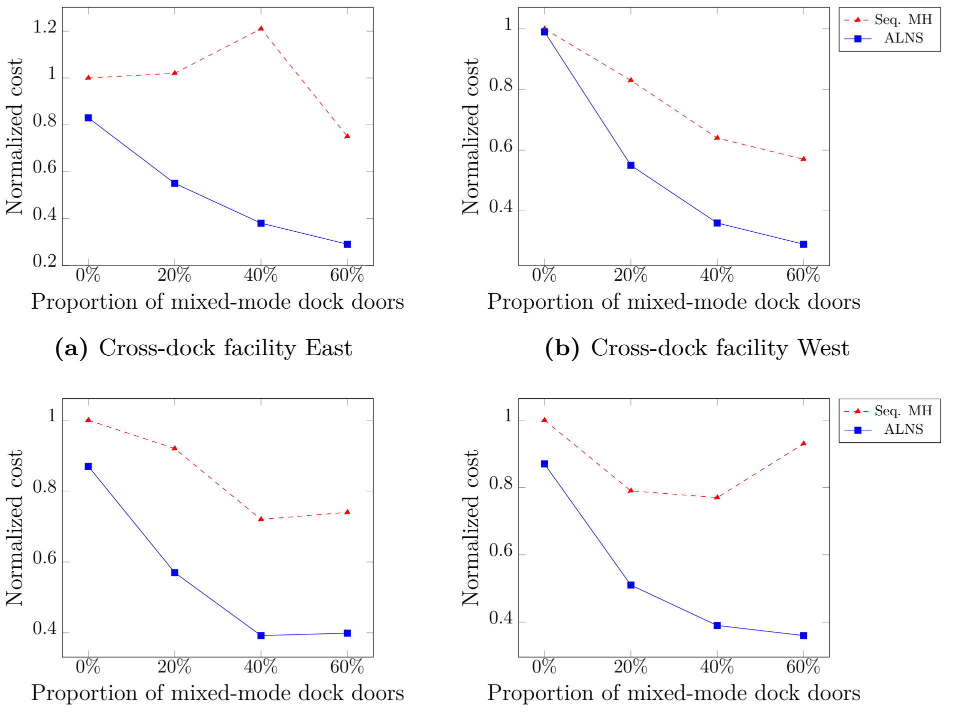
The solutions generated by our integrated ALNS algorithm consistently outperform those of the metaheuristics in the sequential approach. Furthermore, we notice that the cross-dock facilities can improve their operations when the retailer allows for more dock doors to operate in mixed mode, but only when the cross-dock scheduling problem is solved with our ALNS algorithm. Similar to our findings in Section 2.6.3, the metaheuristics in the sequential approach can find solutions that are worse when the proportion of mixed-mode dock doors increases. In particular, the solutions for the East and South cross-dock facility with, respectively, 40% and 60% mixed-mode dock doors, result in higher total operational costs than when less mixed-mode dock doors are included in the layout configuration. As we can see in Table 2.14, the total travel distance for dock-to-dock transfers or the number of unit loads that are temporarily stored increases for these instances. In contrast, the integrated ALNS algorithm finds solutions that consistently improve with the inclusion of more mixed-mode dock doors. However, the marginal benefit decreases, where the average cost savings for including 20%, 40% and 60% mixed-mode dock doors equals 46%, 61% and 66%, respectively. Therefore, we recommend the retailer to adopt at least 40% mixed-mode dock doors, and to locate them according to the center strategy where the integrated ALNS algorithm is used to make the truck scheduling and dock-door assignment decisions. This work has been further validated by a retailer who currently uses a decision support tool based on the methodology developed in this paper. The decision support tool is used to construct schedules for trucks and make tactical decisions on the layout of the cross-dock facility. The solutions have been implemented on multiple cross-docks locations operated by the retailer.

2.7 Conclusion

This paper studies scheduling and dock-door assignment decisions for inbound and outbound trucks at unit-load cross-dock terminals where dock doors can operate in a mixed service mode. In the literature, both problems are usually studied independently,

Table 2.14: The performance of the metaheuristics (MH) in the sequential approach and our ALNS algorithm in the integrated approach (measured as delay time for outbound trucks, usage of temporary storage and direct dock-to-dock travel distance) for each of the cross-dock facilities in our case study

Cross-dock Facility Location	Size of Instance					Performance MH in Sequential Approach			Performance ALNS in Integrated Approach		
	#trucks		#doors			Delayed outbound (min)	Nr. units in temporary storage	Internal travel (km)	Delayed outbound (min)	Nr. units in temporary storage	Internal travel (km)
	IN	OUT	IN	OUT	MIXED						
East	27	171	6	23	0	0	0	15.28	0	0	12.07
East	27	171	6	17	6	0	17	13.48	0	5	7.87
East	27	171	6	11	12	0	0	18.68	0	1	6.00
East	27	171	6	5	18	0	1	11.40	0	0	4.53
West	27	160	5	30	0	0	0	14.29	0	28	10.25
West	27	160	5	23	7	0	0	11.79	0	4	7.30
West	27	160	5	16	14	0	0	9.20	0	0	5.17
West	27	160	5	9	21	0	0	8.25	0	0	4.15
North	32	224	3	28	0	0	0	18.96	0	16	14.35
North	32	224	3	22	6	0	30	13.56	0	0	10.81
North	32	224	3	16	12	0	3	13.31	0	0	7.41
North	32	224	3	9	19	0	26	10.57	0	8	6.51
South	29	200	5	29	0	0	0	20.35	0	9	16.39
South	29	200	5	22	7	0	6	15.32	0	0	10.31
South	29	200	5	15	14	0	14	13.73	0	3	7.51
South	29	200	5	9	21	0	29	14.88	0	5	6.71



(a) Cross-dock facility East
(b) Cross-dock facility West
(c) Cross-dock facility North
(d) Cross-dock facility South

Figure 2.7: The relative performance of the metaheuristics in the sequential approach and our ALNS algorithm in the integrated approach for each of the cross-dock facilities in our case study

and it is proposed to combine them in a sequential manner such as a first-schedule-then-assign decomposition approach. We illustrate that an integrated approach where both types of decisions are considered simultaneously can result in substantially more efficient cross-dock operations. In particular, we show that the heuristic procedure to find a solution to the scheduling problem proposed by Bodnar et al. (2017) can result in infeasible solutions, which results in poor assignment decisions. There is no trivial solution to correct this. Therefore, we propose a new model formulation and heuristic procedure that integrates both decisions. The numerical results show that cost reductions of 20-30% can be obtained when our integrated approach is used compared to the best heuristic methods for the sequential approach.

This increased operational performance depends on the utilization rate of the dock doors as well as the proportion and position of the mixed-mode dock doors. We have investigated this impact for U-shaped cross-dock terminals in more detail. First, the marginal benefit of increasing the number of mixed-mode dock doors diminishes. We recommend managers to use at least 40% of the dock doors in a mixed service mode. Second, positioning mixed-mode dock doors at the center of the cross dock seems to outperform other layout configurations. Especially for cross-dock terminals where the dock-door utilization rates are relatively low, the positioning in the center has the most significant beneficial impact on the operational performance of the cross-dock terminal. A preliminary investigation of I-shaped cross-dock terminals shows that this conclusion does not seem to hold for such facilities.

In our analysis, we assume deterministic arrival and processing times of the trucks at the cross-dock facility. In future studies, the random nature of these aspects in the problem can be incorporated. We reduce the significance of this issue by considering slack time into the arrival time windows and using time periods of 30 to 45 minutes. More efficient time windows could be set when the uncertainty in the arrival times for the inbound trucks is studied. Furthermore, we mainly consider a U-shaped cross-dock terminal in our numerical results. The results in Section 2.8.6 demonstrate that our ALNS algorithm can be applied to I-shaped cross-dock facilities as well, but it is beyond the scope of this paper to investigate layout designs for various shapes of cross docks. Future work can investigate the best shape of cross-dock terminals when there are mixed-mode dock doors. Our study can serve the optimization component in a simulation-optimization approach for such a study.

2.8 Appendix

2.8.1 Bodnar et al. (2017) - revisited

The truck schedule resulting from the model proposed by Bodnar et al. (2017) can result in infeasible solutions for the dock-door assignment. Especially when all flexible doors are scheduled to be in use. This is because their model formulation does not include the service mode of the dock door at which a truck is processed (either inbound, outbound or flexible door). To illustrate this issue, consider the following small example. There are five inbound trucks denoted as $\mathcal{I} = \{i_1, i_2, i_3, i_4, i_5\}$, two outbound trucks $\mathcal{O} = \{o_1, o_2\}$, where $f_{i_1,o_1} = f_{i_2,o_1} = f_{i_3,o_1} = 1$ and $f_{i_3,o_2} = f_{i_4,o_2} = f_{i_5,o_2} = 1$, and value zero otherwise. The cross-dock terminal has three dock doors indicated by $\mathcal{D}_I = \{d_1\}$, $\mathcal{D}_O = \{d_2\}$ and $\mathcal{D}_F = \{d_3\}$, which are the inbound, outbound and flexible dock doors, respectively. The arrival time and due time of each trucks is included in Table 2.15.

Table 2.15: Arrival time r_i and due time d_i for each truck $i \in \mathcal{I} \cup \mathcal{O}$

	i_1	i_2	i_3	i_4	i_5	o_1	o_2
r_i	1	1	2	3	3	1	2
d_i	1	1	2	3	3	2	3

The optimal truck schedule according to the model formulated by Bodnar et al. (2017) is provided in Table 2.16. Since only the number of doors in use in a particular service mode is included in the model, a truck can be at a flexible door in one period and at an outbound door in the next period (since the flexible door might be needed to unload an inbound truck) or vice versa. The solution in Table 2.16 illustrates that outbound truck o_2 is docked at flexible door d_3 at time period $t = 2$ and at outbound door d_2 the next time period. This solution can only be feasible if trucks can preempt and switch dock doors while being processed. However, preemption is usually not allowed in the models from the literature (including ours). Since Berghman et al. (2015) also only consider the number of dock doors in use for each service mode, the same problem arises when a dock-door assignment has to be created from the schedule proposed by their model.

Table 2.16: The scheduling and assignment decisions for the inbound trucks $\mathcal{I}$ and outbound trucks $\mathcal{O}$ in the illustrative example

time	dock door		
period	d_1	d_2	d_3
1	i_1	o_1	i_2
2	i_3	o_1	o_2
3	i_4	o_2	i_5

In Section 2.8.2, we propose a new model formulation for the truck scheduling problem studied by Bodnar et al. (2017). This formulation does not allow preemption as discussed above. Furthermore, we extend their model formulation in the following sense: we allow inbound trucks to stay at the dock door for more than exactly one time period such that it can wait for outbound trucks to arrive while being docked. As a result, this new model formulation can accommodate more dock-to-dock transfers rather than using temporary storage in case the outbound truck is not yet docked at the terminal. In Section 2.4, we present the extension of the truck scheduling model proposed by Bodnar et al. (2017) into an integrated cross-dock scheduling model where the scheduling and assignment decisions are included simultaneously.

2.8.2 Model formulation for sequential approach

In the sequential approach, trucks are first scheduled and then assigned to dock doors as a decomposition mechanism. We formulate both problems when there are mixed-mode dock doors (or flexible dock doors) and a temporary storage area.

Scheduling problem

The following decision variables are used in the model formulation for the scheduling problem in the first stage of the decomposition mechanism:

x^t_{iIn} 1 if inbound truck $i \in \mathcal{I}$ is scheduled for processing at any inbound door at time period $t \in \mathcal{T}$, 0 otherwise

$\hat{x}^t_{iIn}$ 1 if inbound truck $i \in \mathcal{I}$ has finished unloading at any inbound door at the end of time period $t \in \mathcal{T}$, 0 otherwise

x^t_{iF} 1 if inbound truck $i \in \mathcal{I}$ is scheduled for processing at any flexible door at time period $t \in \mathcal{T}$, 0 otherwise

$\hat{x}^t_{iF}$ 1 if inbound truck $i \in \mathcal{I}$ has finished unloading at any flexible door at the end of time period $t \in \mathcal{T}$, 0 otherwise

y^t_{jOut} 1 if outbound truck $j \in \mathcal{O}$ is scheduled for processing at any outbound door at time period $t \in \mathcal{T}$, 0 otherwise

$\hat{y}^t_{jOut}$ 1 if outbound truck $j \in \mathcal{O}$ has finished loading at any outbound door at the end of time period $t \in \mathcal{T}$, 0 otherwise

y^t_{jF} 1 if outbound truck $j \in \mathcal{O}$ is scheduled for processing at any flexible door at time period $t \in \mathcal{T}$, 0 otherwise

$\hat{y}^t_{jF}$ 1 if outbound truck $j \in \mathcal{O}$ has finished loading at any flexible door at the end of time period $t \in \mathcal{T}$, 0 otherwise

u^t_{ij} number of unit loads to unload from inbound truck $i \in \mathcal{I}$ destined for outbound truck $j \in \mathcal{O}$ after time period $t \in \mathcal{T}$

l^t_{ij} number of unit loads loaded in outbound truck $j \in \mathcal{O}$ originating from inbound truck $i \in \mathcal{I}$ after time period $t \in \mathcal{T}$

s_{ij}^t 1 if unit loads destined for outbound truck $j \in \mathcal{O}$ are transferred from inbound truck $i \in \mathcal{I}$ to the temporary storage area at time period $t \in \mathcal{T}$, 0 otherwise

$\hat{s}_{ij}^t$ number of unit loads at the temporary storage location at the end of time period $t \in \mathcal{T}$ that have been received from inbound truck $i \in \mathcal{I}$ and are destined for outbound truck $j \in \mathcal{O}$

z_{ij}^t 1 if inbound truck $i \in \mathcal{I}$ and outbound truck $j \in \mathcal{O}$ are both docked at time period $t \in \mathcal{T}$, 0 otherwise

A_{ij}^t number of unit loads that have been received from inbound truck $i \in \mathcal{I}$ and are transferred from the temporary storage location to outbound truck $j \in \mathcal{O}$ at time period $t \in \mathcal{T}$

The scheduling problem can then be formulated as a MIP model as follows:

$$\min \quad \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{O}}\sum_{t\in\mathcal{T}} \beta s_{ij}^t f_{ij} + \sum_{j\in\mathcal{O}}\sum_{t\in\mathcal{T}_j^L} \gamma\delta_j^t(\hat{y}_{jOut}^t + \hat{y}_{jF}^t) \tag{2.34}$$

subject to

$$\sum_{i\in\mathcal{I}} x_{iIn}^t \leq |\mathcal{D}_I| \qquad \forall t \in \mathcal{T} \tag{2.35}$$

$$\sum_{j\in\mathcal{O}} y_{jOut}^t \leq |\mathcal{D}_O| \qquad \forall t \in \mathcal{T} \tag{2.36}$$

$$\sum_{i\in\mathcal{I}} x_{iF}^t + \sum_{j\in\mathcal{O}} y_{jF}^t \leq |\mathcal{D}_F| \qquad \forall t \in \mathcal{T} \tag{2.37}$$

$$\sum_{t\in\mathcal{T}_i^N} (\hat{x}_{iIn}^t + \hat{x}_{iF}^t) = 1 \qquad \forall i \in \mathcal{I} \tag{2.38}$$

$$\sum_{t\in\{\mathcal{T}_j^N,\mathcal{T}_j^L\}} (\hat{y}_{jOut}^t + \hat{y}_{jF}^t) = 1 \qquad \forall j \in \mathcal{O} \tag{2.39}$$

$$u_{ij}^0 = f_{ij} \qquad \forall i \in \mathcal{I}, j \in \mathcal{O} \tag{2.40}$$

$$l_{ij}^0 = 0 \qquad \forall i \in \mathcal{I}, j \in \mathcal{O} \tag{2.41}$$

$$\hat{s}_{ij}^0 = 0 \qquad \forall i \in \mathcal{I}, j \in \mathcal{O} \tag{2.42}$$

$$u_{ij}^t \geq u_{ij}^{t-1} - (s_{ij}^t f_{ij}) - z_{ij}^t f_{ij} \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.43}$$

$$l_{ij}^t \leq l_{ij}^{t-1} + A_{ij}^t + z_{ij}^t f_{ij} \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.44}$$

$$s_{ij}^t \leq x_{iIn}^t + x_{iF}^t \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \tag{2.45}$$

$$\hat{s}_{ij}^{t} = \hat{s}_{ij}^{t-1} + (s_{ij}^{t} f_{ij}) - A_{ij}^{t} \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \quad (2.46)$$

$$x_{iIn}^{t} \leq \hat{x}_{iIn}^{t} + x_{iIn}^{t+1} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \quad (2.47)$$

$$x_{iF}^{t} \leq \hat{x}_{iF}^{t} + x_{iF}^{t+1} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \quad (2.48)$$

$$y_{jOut}^{t} \leq \hat{y}_{jOut}^{t} + y_{jOut}^{t+1} \qquad \forall j \in \mathcal{O}, t \in \mathcal{T} \quad (2.49)$$

$$y_{jF}^{t} \leq \hat{y}_{jF}^{t} + y_{jF}^{t+1} \qquad \forall j \in \mathcal{O}, t \in \mathcal{T} \quad (2.50)$$

$$\sum_{j \in \mathcal{O}} f_{ij} \Big(1 - (\hat{x}_{iIn}^{t} + \hat{x}_{iF}^{t})\Big) \geq \sum_{j \in \mathcal{O}} u_{ij}^{t} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \quad (2.51)$$

$$\sum_{i \in \mathcal{I}} f_{ij} (\hat{y}_{jOut}^{t} + \hat{y}_{jF}^{t}) \leq \sum_{i \in \mathcal{I}} l_{ij}^{t} \qquad \forall j \in \mathcal{O}, t \in \mathcal{T} \quad (2.52)$$

$$\sum_{t \in \{\mathcal{T}_i^E, \mathcal{T}_i^L\}} (x_{iIn}^{t} + x_{iF}^{t}) = 0 \qquad \forall i \in \mathcal{I} \quad (2.53)$$

$$\sum_{t \in \mathcal{T}_j^E} (y_{jOut}^{t} + y_{jF}^{t}) = 0 \qquad \forall j \in \mathcal{O} \quad (2.54)$$

$$g_i (\hat{x}_{iIn}^{t} + \hat{x}_{iF}^{t}) \leq \sum_{t'=1}^{t} (x_{iIn}^{t'} + x_{iF}^{t'}) \qquad \forall i \in \mathcal{I}, t \in \mathcal{T}_i^N \quad (2.55)$$

$$h_j (\hat{y}_{jOut}^{t} + \hat{y}_{jF}^{t}) \leq \sum_{t'=r_j}^{t} (y_{jOut}^{t'} + y_{jF}^{t'}) \qquad \forall j \in \mathcal{O}, t \in \{\mathcal{T}_j^N, \mathcal{T}_j^L\} \quad (2.56)$$

$$\lambda_{ij} (x_{iIn}^{t} + x_{iF}^{t} + y_{jOut}^{t} + y_{jF}^{t}) \geq 2\, z_{ij}^{t} \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \quad (2.57)$$

$$A_{ij}^{t} \leq \hat{s}_{ij}^{t-1} \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \quad (2.58)$$

$$A_{ij}^{t} \leq f_{ij} (y_{jOut}^{t} + y_{jF}^{t}) \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \quad (2.59)$$

$$x_{iIn}^{t}, x_{iF}^{t}, \hat{x}_{iIn}^{t}, \hat{x}_{iF}^{t} \in \{0, 1\} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T} \quad (2.60)$$

$$y_{jOut}^{t}, y_{jF}^{t}, \hat{y}_{jOut}^{t}, \hat{y}_{jF}^{t} \in \{0, 1\} \qquad \forall j \in \mathcal{O}, t \in \mathcal{T} \quad (2.61)$$

$$z_{ij}^{t}, s_{ij}^{t} \in \{0, 1\} \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \quad (2.62)$$

$$A_{ij}^{t}, u_{ij}^{t}, l_{ij}^{t}, \hat{s}_{ij}^{t} \geq 0 \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, t \in \mathcal{T} \quad (2.63)$$

The first term in the objective function (2.34) represents the costs of unit loads that are temporarily stored before being loaded onto their destined outbound truck, and the second term represents the total tardiness costs of outbound trucks.

Constraints (2.35) ensure that the number of inbound trucks assigned to inbound dock doors is less than or equal to the number of inbound dock doors at any time period. Similarly, constraints (2.36) ensure that the number of outbound trucks assigned to outbound dock doors is less than or equal to the number of outbound dock doors at any time period. Constraints (2.37) ensure that the sum of the number of inbound and outbound trucks assigned to flexible dock doors is less than or equal to the number of flexible dock doors at any time period. Constraints (2.38) ensure that all inbound trucks are completely processed within their arrival and due time. All outbound trucks are completely processed at outbound or flexible doors within their arrival time period and the end of the planning horizon as per constraints (2.39).

The constraints (2.40) to (2.63) can be translated one-to-one from the constraints (2.9) to (2.32) in Section 2.4, with the exception that the variables with an index for individual dock doors are replaced by variables that represent either all inbound doors (x^t_{iIn} and $\hat{x}^t_{iIn}$), all outbound doors (y^t_{jOut} and $\hat{y}^t_{jOut}$) or all flexible doors (x^t_{iF}, $\hat{x}^t_{iF}$, y^t_{jF} and $\hat{y}^t_{jF}$), and the constraints (2.24) and (2.28) are removed since they specify the value for variable $\hat{z}_{ijkp}$ which we don't have in this scheduling problem.

Assignment problem

In the second stage of the decomposition mechanism, the trucks need to be assigned to dock doors given the truck schedule that was generated in first stage. The following decision variables are used in the model formulation for the assignment problem:

x_{ik}	1 if inbound truck $i \in \mathcal{I}$ is assigned to door $k \in \mathcal{D}_{IF}$, 0 otherwise
y_{jp}	1 if outbound truck $j \in \mathcal{O}$ is assigned to door $p \in \mathcal{D}_{OF}$, 0 otherwise
$\hat{z}_{ijkp}$	1 if inbound truck $i \in \mathcal{I}$ is completely unloaded at door $k \in \mathcal{D}_{IF}$ and outbound truck $j \in \mathcal{O}$ is completely processed at dock door $p \in \mathcal{D}_{OF}$ for dock-to-dock transfer of shipments between i and j, 0 otherwise

Furthermore, we use the values of the decision variables from the first stage as parameters in this stage, where we define the new parameters

$$\begin{aligned} \vartheta^t_i &= x^t_{iIn} + x^t_{iF} \quad \text{for } i \in \mathcal{I}, t \in \mathcal{T} \\ \varphi^t_j &= y^t_{jOut} + y^t_{jF} \quad \text{for } j \in \mathcal{O}, t \in \mathcal{T} \end{aligned}$$

The assignment problem can then be formulated as a MIP model as follows:

$$\min \quad \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{O}} \sum_{k \in \mathcal{D}_{IF}} \sum_{p \in \mathcal{D}_{OF}} \eta \hat{z}_{ijkp} m_{kp} f_{ij} \tag{2.64}$$

subject to

$$\sum_{i \in \mathcal{I}} (x_{ik}\, \vartheta_i^t) \leq 1 \qquad \forall k \in \mathcal{D}_I, t \in \mathcal{T} \tag{2.65}$$

$$\sum_{j \in \mathcal{O}} (y_{jp}\, \varphi_j^t) \leq 1 \qquad \forall p \in \mathcal{D}_O, t \in \mathcal{T} \tag{2.66}$$

$$\sum_{i \in \mathcal{I}} (x_{ik}\, \vartheta_i^t) + \sum_{j \in \mathcal{O}} (y_{jk}\, \varphi_j^t) \leq 1 \qquad \forall k \in \mathcal{D}_F, t \in \mathcal{T} \tag{2.67}$$

$$\sum_{k \in \mathcal{D}_{IF}} x_{ik} = 1 \qquad \forall i \in \mathcal{I} \tag{2.68}$$

$$\sum_{p \in \mathcal{D}_{OF}} y_{jp} = 1 \qquad \forall j \in \mathcal{O} \tag{2.69}$$

$$\lambda_{ij} \left(x_{ik} + y_{jp} - \sum_{t \in \mathcal{T}} s_{ij}^t \right) \leq 1 + \hat{z}_{ijkp} \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, k \in \mathcal{D}_{IF}, p \in \mathcal{D}_{OF} \tag{2.70}$$

$$x_{ik} \in \{0, 1\} \qquad \forall i \in \mathcal{I}, k \in \mathcal{D}_{IF} \tag{2.71}$$

$$y_{jp} \in \{0, 1\} \qquad \forall j \in \mathcal{O}, p \in \mathcal{D}_{OF} \tag{2.72}$$

$$\hat{z}_{ijkp} \in \{0, 1\} \qquad \forall i \in \mathcal{I}, j \in \mathcal{O}, k \in \mathcal{D}_{IF}, p \in \mathcal{D}_{OF} \tag{2.73}$$

The objective function (2.64) represents the minimization of the costs associated with all direct dock-to-dock transfers. Constraints (2.65) ensure that at most one inbound truck can be assigned to an inbound door at any time period. Similarly, constraints (2.66) ensure this for outbound trucks assigned to outbound doors and constraints (2.67) for either inbound or outbound trucks assigned to flexible doors. Constraints (2.68) and (2.69) ensure that each inbound and outbound truck is assigned to an appropriate dock door, respectively. These five constraints can be directly translated from the constraints (2.4) to (2.8) and they also prevent preemption of trucks. Constraints (2.70) regulate the value for the decision variable $\hat{z}_{ijkp}$ similar to constraints (2.24). Constraints (2.71) to (2.73) define the domain of the decision variables.

2.8.3 Integrated ALNS algorithm

In this appendix, we develop a metaheuristic to find a near-optimal solution for the cross-dock scheduling problem formulated in Section 2.4. The architecture of the metaheuristic is designed according to the Adaptive Large Neighborhood Search

(ALNS) principle and has a resemblance to the metaheuristics proposed by Bodnar et al. (2017).

As introduced in Section 2.5, a collection of feasible solutions is generated and the best solution is selected to be altered by neighborhood operators. A feasible solution π has corresponding total costs $z(\pi)$. After solution π is altered by an operator, the new solution is denoted as π' and has corresponding total costs $z(\pi')$. Whether π' replaces π is based on a simulated annealing principle. If the costs $z(\pi')$ are lower than the total costs $z(\pi^*)$ of best known solution π^*, π' is accepted as the new current solution π. Otherwise, π' is accepted with probability $e^{-(z(\pi')-z(\pi^*))/T}$, where temperature T is set based on Ropke & Pisinger (2006b). Initially, $T := -w \cdot z(\pi^*)/\ln 0.5$ and it is adjusted using the cooling parameter r as $T := rT$ with $0 < r < 1$. w is a user provided parameter. If the solution is not improved for n_T iterations, T is reset to its initial value to encourage the move from a local optimum.

Let N be the set of neighborhood operators that can be selected to modify π to π' in each iteration. The operator is randomly selected in each iteration, where the probability for each operator to be selected is proportional to the weight μ_n of an operator $n \in N$ (i.e., the probability equals $\mu_n / \sum_{n' \in N} \mu_{n'}$).

Initially, $\mu_n = 1$ for $n \in N$ and these weights are updated after 100 iterations (Ropke & Pisinger, 2006a). We use the updating scheme $\mu_n := (1-\rho)\mu_n + \rho(\zeta_n/\theta_n)$, if $\theta_n \neq 0$, where $\rho \in (0, 1)$ is user specified, θ_n is the number of times that operator n was selected to modify the solution in the last 100 iterations and ζ_n is an indicator for the performance of operator n over the last 100 iterations. In particular, the new solution π' can improve the best know solution π^* or the current solution π such that ζ_n is updated as follows after each of the 100 iterations:

$$\zeta_n := \begin{cases} \zeta_n + \delta^1 & \text{if } z(\pi') < z(\pi^*), \\ \zeta_n + \delta^2 & \text{if } z(\pi^*) \leq z(\pi') < z(\pi), \\ \zeta_n + \delta^3 & \text{if } z(\pi) \leq z(\pi'), \end{cases}$$

where $\delta^1 > \delta^2 > \delta^3$. After 100 iterations, the weights μ_n are calculated and they remain fixed for the next 100 iterations. Note that ζ_n and θ_n are reset to 0 for $n \in N$ after each 100 iterations (Stenger et al., 2013).

The architecture of the ALNS algorithm follows a decomposition scheme where we only destroy and repair the scheduling and assignment decisions of inbound trucks and we determine the optimal scheduling and assignment decisions for outbound trucks, where the inbound truck decisions are given. The decomposition scheme is inspired by

Algorithm 1 Pseudo code of the integrated ALNS algorithm

Construct a set of feasible solutions X
Select $\pi^* \in X$ such that $z(\pi^*) \leq z(\pi_0) \,\forall\, \pi_0 \in X$
Set the current solution $\pi \leftarrow \pi^*$
Construct inbound schedule and assignment plan x from π
while stop criterion is not met **do**
    Select a neighbourhood operator $n \in N$ based on adaptive weights μ_n
    Select an inbound processing plan $x' \in n(x)$
    Solve the reduced problem for x' given, resulting in π'
    if $z(\pi') < z(\pi^*)$ **then**
        Set $\pi^* \leftarrow \pi'$
    if $z(\pi') \leq z(\pi)$ **or** Uniform $[0;1) \leq e^{-(z(\pi')-z(\pi^*))/T}$ **then**
        Set $\pi \leftarrow \pi'$, $x \leftarrow x'$
    Update T
    Update the score ζ_n, θ_n and possibly μ_n for operator n
return π^*

the ALNS algorithm proposed by Bodnar et al. (2017). However, a crucial difference is that both scheduling and assignment decisions are determined simultaneously by the integrated ALNS algorithm (rather than just the scheduling decisions). The pseudocode for our ALNS algorithm is presented in Algorithm 1.

Initial feasible solution

Initial feasible solutions are created in two stages using a constructive heuristic based on a first-schedule-then-assign decomposition strategy (Boysen et al., 2013; Hermel et al., 2016).

Stage 1: Truck scheduling

In the first stage, inbound and outbound trucks are scheduled according to a modified version of the constructive heuristic proposed by Bodnar et al. (2017).

At a time period $t \in \mathcal{T}$, a subset of inbound and outbound trucks are docked at the terminal, denoted by $\mathcal{I}^t \subseteq \mathcal{I}$ and $\mathcal{O}^t \subseteq \mathcal{O}$, respectively. We initialize $\mathcal{I}^0 := \emptyset$ and $\mathcal{O}^0 := \emptyset$ since no truck is scheduled at the beginning of the planning horizon. For each time period $t \in \mathcal{T}$, we initialize these subsets according to the following rules: If a truck is docked at time period $t-1$ and it is not completely (un)loaded yet, it has to remain at the dock door in period t. Additionally, a truck needs to remain at the dock door for its corresponding minimum amount of time periods required (that is, g_i time periods for inbound truck i and h_j time periods for outbound truck j). Consequently, we initially set

$$\mathcal{I}^t := \{i \,|\, i \in \mathcal{I}^{t-1}, (i \text{ is not completed}) \vee (i \notin \mathcal{I}^{t-g_i})\} \tag{2.74}$$

$$\mathcal{O}^t := \{j \,|\, j \in \mathcal{O}^{t-1}, (j \text{ is not completed}) \vee (j \notin \mathcal{O}^{t-h_j})\} \tag{2.75}$$

Next, we generate a subset of inbound and outbound trucks that are candidates to be processed at time period t (i.e., to be included to $\mathcal{I}^t$ and $\mathcal{O}^t$). First, we specify the subset of outbound trucks as

$$C(\mathcal{O}^t) := \{j \,|\, j \in \mathcal{O}, r_j \le t, j \notin \cup_{t'=0}^{t-1} \mathcal{O}^{t'}\}, \tag{2.76}$$

such that only outbound trucks that have arrived at the yard and have not been docked before time period t are considered. The inbound trucks are selected based on similar rules, plus two addtional rules: an inbound truck i can only be included if it contains goods for an outbound truck j that is currently docked or considered to be docked at time period t (i.e., $j \in C(\mathcal{O}^t)$), or it can be included if it needs to be processed at time period t to fulfill the minimum processing time requirement g_i before it is due at time d_i:

$$C(\mathcal{I}^t) := \left\{ i \,|\, i \in \mathcal{I}, r_i \le t \le d_i, i \notin \cup_{t'=0}^{t-1} \mathcal{I}^{t'}, \left(\sum_{j \in \{C(\mathcal{O}^t), \mathcal{O}^t\}} f_{ij} > 0 \right) \vee (d_i - g_i) = t \right\} \tag{2.77}$$

The inbound trucks from $C(\mathcal{I}^t)$ are scheduled to be docked at t (i.e., included to $\mathcal{I}^t$) based on one of four strategies: (i) select the earliest arriving truck (FCFS), (ii) select the earliest due truck (EDD), (iii) select the truck that enables the largest direct dock-to-dock transfer (i.e., based on $\sum_{j \in \mathcal{O}^t} f_{ij}$), and (iv) select the truck that results in the most number of outbound trucks that are completely loaded and depart from the cross-dock terminal. Similar strategies can be used to select outbound trucks $j \in C(\mathcal{O}^t)$ to be included to $\mathcal{O}^t$. We refer the reader to Bodnar et al. (2017) for more details on each of the four strategies.

This constructive heuristic prescribes four decision rules to schedule inbound trucks and four strategies to schedule outbound trucks. Combining each of these rules, 16 different strategies can be employed to construct a schedule.

Stage 2: Dock-door assignment of scheduled trucks

In the second stage of the decomposition strategy, the scheduled trucks are assigned to dock doors based on a heuristic proposed by Van Belle et al. (2013).

Let $\mathcal{D}^t_{IF}$ and $\mathcal{D}^t_{OF}$ represent the sets of unassigned dock doors that are available to process inbound and outbound trucks, respectively, at time period $t \in \mathcal{T}$. At the beginning of the planning horizon, $\mathcal{D}^0_{IF} = \{\nu \,|\, \nu \in \mathcal{D}_{IF}\}$ and $\mathcal{D}^0_{OF} = \{\nu \,|\, \nu \in \mathcal{D}_{OF}\}$. We order the set $\mathcal{D}^t_{IF}$ in ascending order of the average travel distance from a door in $\mathcal{D}^t_{IF}$ to all doors in $\mathcal{D}_{OF}$. Similarly, $\mathcal{D}^t_{OF}$ is ordered in the ascending order of the average travel distance from a door to all doors in $\mathcal{D}_{IF}$.

For each inbound truck $i \in \mathcal{I}^t$, we calculate the number of unit loads that can directly be sent from inbound truck i to outbound truck j with a direct dock-to-dock transfer as

$$\omega_i = \sum_{j \in \mathcal{O}^t} f_{ij} \quad \forall\, t, i \in \mathcal{I}^t \tag{2.78}$$

Next, until each truck $i \in \mathcal{I}^t$ is assigned to a door, we iteratively select unassigned truck i as $i = \arg\max_{i' \in \mathcal{I}^t}\{\omega_{i'} \,|\, i' \text{ is not assigned}\}$ and assign this truck to the first door in $\mathcal{D}^t_{IF}$, denoted by door ν_i. Consequently, $\mathcal{D}^t_{IF} := \mathcal{D}^t_{IF} \setminus \{\nu_i\}$ to indicate that the door is occupied, and truck i is marked as assigned. If $\nu_i \in \mathcal{D}^t_{OF}$, then $\mathcal{D}^t_{OF} := \mathcal{D}^t_{OF} \setminus \{\nu_i\}$.

Similar to the inbound trucks, each outbound truck $j \in \mathcal{O}^t$ receives a score ω_j that represents the number of unit loads that it can receive directly from an inbound truck $i \in \mathcal{I}^t$:

$$\omega_j = \sum_{i \in \mathcal{I}^t} f_{ij} \quad \forall\, t, j \in \mathcal{O}^t \tag{2.79}$$

Until each outbound truck $j \in \mathcal{O}^t$ is assigned a dock door, we select the best unassigned outbound trucks $j = \arg\max_{j' \in \mathcal{O}^t}\{\omega_{j'} \,|\, j' \text{ is not assigned}\}$, and assign it to the first door in $\mathcal{D}^t_{OF}$, and denote this by ν_j. Next, truck j is marked as assigned, and the door ν_j is removed from $\mathcal{D}^t_{OF}$ and from $\mathcal{D}^t_{IF}$ if $\nu_j \in \mathcal{D}^t_{IF}$.

After all inbound trucks $i \in \mathcal{I}^t$ and outbound trucks $j \in \mathcal{O}^t$ are assigned a dock door at time period t, the subsets $\mathcal{D}_{IF}^{t+1}$ and $\mathcal{D}_{OF}^{t+1}$ are created from $\mathcal{D}_{IF}^t$ and $\mathcal{D}_{OF}^t$, respectively, by adding doors that become available in time period $t+1$, since the truck assigned to that door in period t is not scheduled in period $t+1$. In particular, $\mathcal{D}_{IF}^{t+1}$ consists of all doors in $\mathcal{D}_{IF}^t$, doors in $\mathcal{D}_{IF}$ that will be vacated by inbound trucks at the end of period t, and doors in $\mathcal{D}_F$ vacated by outbound trucks at the end of period t. Similarly, $\mathcal{D}_{OF}^{t+1}$ consists of doors in $\mathcal{D}_{OF}^t$, doors in $\mathcal{D}_{OF}$ vacated at the end of period t by outbound trucks, and flexible doors vacated by inbound truck at the end of period t.

$$\mathcal{D}_{IF}^{t+1} := \mathcal{D}_{IF}^t \cup \{\nu_i \,|\, \nu_i \in \mathcal{D}_{IF}, i \in \mathcal{I}^t, i \notin \mathcal{I}^{t+1}\} \cup \{\nu_j \,|\, \nu_j \in \mathcal{D}_F, j \in \mathcal{O}^t, j \notin \mathcal{O}^{t+1}\} \tag{2.80}$$

$$\mathcal{D}_{OF}^{t+1} := \mathcal{D}_{OF}^t \cup \{\nu_j \,|\, \nu_j \in \mathcal{D}_{OF}, j \in \mathcal{O}^t, j \notin \mathcal{O}^{t+1}\} \cup \{\nu_i \,|\, \nu_i \in \mathcal{D}_F, i \in \mathcal{I}^t, i \notin \mathcal{I}^{t+1}\} \tag{2.81}$$

Neighborhood operators

In accordance with the ALNS algorithm, a solution is broken up and then repaired in each iteration. The destroy operators remove truck(s) from the solution representation, and the repair operators insert the same truck(s) at a different dock door and/or time. We use the following *destroy operators* in our metaheuristics:

• **rRd**: An inbound truck is randomly selected to be removed from the solution matrix.

• **rMx**: An inbound truck is randomly selected to be removed from the time period at which the largest number of inbound trucks are scheduled. This operator aims to alleviate the bottleneck at the cross-dock facility by having fewer inbound trucks docked at the same time period and by making more dock doors available to process outbound trucks. Bodnar et al. (2017) use a similar operator.

• **rCr**: An inbound truck is randomly selected to be removed where the corresponding probability to do so is based on a critical ratio that captures the lateness or earliness of the inbound truck in comparison to the due time of the outbound trucks that receive goods from the inbound truck. This operator is also used by Bodnar et al. (2017).

• **rSy**: This operator aims at synchronizing the unloading of inbound trucks to ensure direct dock-to-dock transfers. An inbound truck is randomly selected to be removed where the corresponding probability to do so is based on a synchronization ratio that captures the earliness or lateness of the inbound truck in comparison to other inbound trucks that carry unit loads for the same outbound truck.

First, for each inbound truck $i \in \mathcal{I}$, we define the set of outbound trucks for which it carries load as $S_i = \{j \mid f_{ij} > 0\} \subseteq \mathcal{O}$. Let Q_i indicate the set of inbound trucks that carry goods for the same outbound trucks in S_i. That is, $Q_i := \{i' | \, i' \in \mathcal{I}, \, f_{i'j} > 0, j \in S_i\}$. Furthermore, let ψ_i be the earliest time period at which inbound truck i is currently scheduled, and $\chi_{i'}$ is the latest time period at which inbound truck $i' \in Q_i$ is scheduled. Define the synchronization ratio to schedule truck i earlier as $sy_i^{\leftarrow} = \sum_{i' \in Q_i} \max\{0, \psi_i - \chi_{i'}\}$, and to schedule truck i later as $sy_i^{\rightarrow} = \sum_{i' \in Q_i} \max\{0, \psi_{i'} - \chi_i\}$. High values for $sy_i^{\leftarrow}$ indicate that the inbound truck is scheduled later than other inbound trucks that also carry items for the same outbound trucks. Therefore, fewer unit loads might be directly transferred to outbound trucks. Similarly, $sy_i^{\rightarrow}$ indicates that the inbound truck is scheduled earlier than other inbound trucks supplying items to the same outbound trucks. Therefore, it is a good candidate for rescheduling to a later time period.

The *repair (or insert) operators* that we use to create new solutions are as follows:

• **iBck**: The removed inbound truck is scheduled to an earlier time period at the same dock door to which it was originally assigned.

• **iFwd**: The removed inbound truck is scheduled at a later time period at the same dock door to which it was originally assigned.

• **iSwap**: The removed inbound truck is swapped with a randomly selected inbound truck (not necessarily at a different dock door, as in Boysen et al. (2013)).

• **iUp**: The removed inbound truck is assigned to the next available dock door $i \in \mathcal{D}_{IF}$ with a higher index.

• **iDown**: The removed inbound truck is assigned to the next available dock door in $\mathcal{D}_{IF}$ with a lower index.

Besides individual operators that either destroy or repair a solution, we also introduce four *operators that both destroy and repair* a solution:

• **riI2O**: An inbound truck is removed from the schedule at a time period when no corresponding outbound truck is scheduled and inserted back into the schedule at the same dock door but at a later time period. This operator aims to achieve more direct dock-to-dock transfers.

• **riextBck**: An inbound truck is randomly selected and its scheduled time at the dock door is extended to one time period earlier. This operator is designed to facilitate more dock-to-dock transfers.

• **riextFwd**: An inbound truck is randomly selected and its scheduled time at the dock door is extended to one time period later. This operator is also designed to facilitate more dock-to-dock transfers.

• **riD2D**: Two dock doors are randomly selected from $\mathcal{D}_{IF}$ and their assigned trucks are swapped while maintaining the scheduling decisions.

For all operators that repair a solution, if a newly selected dock door for a truck is already occupied by another truck in the current solution π, this assigned truck and all trucks assigned to the same dock door thereafter are pushed back for processing by at least the minimum processing time of the newly assigned truck. As a result, some trucks might be pushed out of the planning horizon, which leads to an infeasible solution. Therefore, we try to repair the new solution to become feasible as follows: For each inbound truck $i \in \mathcal{I}$ missing in the new solution π', denote t^i as the time period at which truck i is scheduled in the current solution π. Assign this truck to the first available dock door at time period t^i or later. If the new schedule for the inbound operations remains infeasible, the new solution π' is rejected as infeasible.

Reduced problem

At this stage of the heuristic, we have determined a feasible truck schedule and dock-door assignment for the inbound operations. In the reduced problem, we assign and schedule outbound trucks only. The structure of this reduced problem resembles the reduced problem studied by Bodnar et al. (2017). However, we extend it in two aspects: (i) we include the total travel distance for dock-to-dock transfers into the objective function and (ii) we simultaneously schedule and assign the outbound trucks to dock doors.

Since the schedule and assignment of the inbound trucks is given for the reduced problem, we let parameter x_{ik}^t represent whether inbound truck i is assigned to dock door k at time period t (the same interpretation as our decision variable in the MIP model of Section 2.4, but now we know its value). Furthermore, we define the set of dock doors occupied by inbound trucks at time period $t \in \mathcal{T}$ as $\mathcal{D}_{IF}^t := \{k \,|\, x_{ik}^t = 1, \forall i \in \mathcal{I}, k \in \mathcal{D}_{IF}\}$ and the set of doors available to process outbound trucks at time period $t \in \mathcal{T}$ as $\mathcal{D}^t_{OF} = \mathcal{D}_{OF} \setminus \mathcal{D}^t_{IF}$. Additionally, the parameter $\bar{x}^t_i$ indicates with value one whether inbound truck i is scheduled until time period t, and zero otherwise.

We define the set of time periods during which any inbound truck with goods for outbound truck $j \in \mathcal{O}$ is scheduled to be docked at the terminal as $\mathcal{T}_j = \{t | \exists i \in \mathcal{I}, k \in \mathcal{D}_{IF}, t \in \mathcal{T} : x^t_{ik} = 1, f_{ij} > 0\}$. The first time period at which goods destined for outbound truck j become available from an inbound truck is denoted as $\alpha_j = \min_{t \in \mathcal{T}_j}\{t\}$, and the last time period at which goods destined for outbound truck j can be loaded with a direct dock-to-dock transfer equals $\omega_j = \max_{t \in \mathcal{T}_j}\{t\}$.

Consequently, outbound truck j should not be scheduled before time period $\alpha_j - h_j$. If outbound truck j is scheduled to be docked for the first time at or before time period $\omega_j - h_j$, the outbound truck has to remain scheduled and assigned to the same dock door until time period ω_j to guarantee that the truck is completely loaded before departing from the cross-dock terminal. If outbound truck j is scheduled to be docked for the first time after $\omega_j - h_j$, the outbound truck has to remain scheduled and assigned to the same dock door for exactly h_j time periods to satisfy the minimum processing time constraint.

In the reduced problem, the decision variable $\tilde{y}^t_{jp}$ attains the value one if outbound truck $j \in \mathcal{O}$ is scheduled at dock door $p \in \mathcal{D}_{OF}$ for the first time at time period $t \in \mathcal{T}$, and zero otherwise. When outbound truck j is scheduled for the first time at time period t', it needs to stay docked until time period $\max\{\omega_j, t' + h_j - 1\}$. Consequently, we specify parameter $\Phi^{t',t}_j$ as

$$\Phi^{t',t}_j = \begin{cases} 1 & \text{if } t' \leq t \leq \max\{\omega_j, t' + h_j - 1\} \\ 0 & \text{otherwise} \end{cases} \tag{2.82}$$

which indicates whether outbound truck j is still docked at time period t when it is first scheduled at time period t'. Note that $\Phi^{t',t}_j$ can have the value one only if $\sum_p \tilde{y}^{t'}_{jp} = 1$.

Furthermore, since we know until which time period outbound truck j is docked when it is first scheduled at time period t, we can directly calculate the length of the delay beyond its due time d_j. To indicate this, we specify

$$\Delta^t_j = \max\{0, \max\{\omega_j, t' + h_j - 1\} - d_j\} \tag{2.83}$$

Consequently, the parameter c_j^t is introduced as the sum of the tardiness costs and the total storage costs if outbound truck j is first scheduled at time period t, where

$$c_j^t = \gamma \Delta_j^t + \beta \sum_{t'=1}^{t-1} \sum_{i \in \mathcal{I}} \bar{x}_i^t f_{ij} \qquad \forall j \in \mathcal{O}, t \in \mathcal{T} \tag{2.84}$$

In addition to these two costs, we also include the total costs of direct dock-to-dock transfers to the objective function. Therefore, we introduce parameter $\bar{\bar{x}}_{ik}^t$, which indicates whether inbound truck i is docked at door k at time period t or later:

$$\bar{\bar{x}}_{ik}^t = \begin{cases} 1 & \text{if } \sum_{t' \geq t} x_{ik}^{t'} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.85}$$

A direct transfer between inbound truck i at dock door k and outbound truck j at dock door p can take place if there is a time period at which both $\bar{\bar{x}}_{ik}^t = 1$ and $\tilde{y}_{jp}^t = 1$. This leads to the following MIP model for the reduced problem (RP).

RP:

$$\min \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{O}} \sum_{k \in \mathcal{D}_{IF}} \sum_{p \in \mathcal{D}_{OF}} \sum_{t \in \mathcal{T}} \beta \bar{\bar{x}}_{ik}^t \tilde{y}_{jp}^t m_{kp} f_{ij} + \sum_{j \in \mathcal{O}} \sum_{p \in \mathcal{D}_{OF}} \sum_{t \in \mathcal{T}} c_j^t \tilde{y}_{jp}^t \tag{2.86}$$

subject to

$$\sum_{p \in \mathcal{D}_{OF}} \sum_{t \in \mathcal{T}} \tilde{y}_{jp}^t = 1 \qquad \forall j \in \mathcal{O} \tag{2.87}$$

$$\sum_{t' \in \mathcal{T}} \sum_{j \in \mathcal{O}} \tilde{y}_{jp}^t \Phi_j^{t',t} \leq 1 \qquad \forall p \in \mathcal{D}_{OF}, t \in \mathcal{T} \tag{2.88}$$

$$\sum_{t' \in \mathcal{T}} \sum_{j \in \mathcal{O}} \tilde{y}_{jp}^t \Phi_j^{t',t} = 0 \qquad \forall p \notin \mathcal{D}_{OF}^t, t \in \mathcal{T} \tag{2.89}$$

$$\tilde{y}_{jp}^t = 0 \qquad \forall j \in \mathcal{O}, p \in \mathcal{D}_{OF}, t \in \mathcal{T}_j^E \tag{2.90}$$

$$\tilde{y}_{jp}^t \in \{0,\, 1\} \qquad \forall j \in \mathcal{O}, p \in \mathcal{D}_{OF}, t \in \mathcal{T} \tag{2.91}$$

Constraints (2.87) ensure that each outbound truck is assigned exactly once. Constraints (2.88) impose non-preemption restrictions and ensure that a dock door is only assigned to one truck at most for each time period. Similarly, constraints (2.89) restrict the processing of outbound trucks to dock doors in $\mathcal{D}_{OF}$ at times not assigned to any inbound truck. Constraints (2.90) prevent outbound trucks to be scheduled before their arrival time. Constraints (2.91) define the decision variables as binary.

If an outbound truck j is processed at time $t \leq w_j - h_j$, it can be postponed until time period $w_j - h_j$ and still fulfill the minimum completion time requirement by the period w_j. If $t \leq w_j - h_j$ and $t \in \mathcal{T}_j$, the outbound truck can receive direct dock-to-dock transfers which justifies the early processing. However, if $t \leq w_j - h_j$ and $t \notin \mathcal{T}_j$, the processing of outbound truck j can be postponed without additional costs in the objective function. Constraints (2.92) formalize this condition. The constraint is an extension of the valid inequalities by Bodnar et al. (2017) to consider the dock-door assignment aspect as well.

$$\tilde{y}_{jp}^{t} = 0 \qquad \forall j \in \mathcal{O}, p \in \mathcal{D}_{OF}, t \in \{t' | t' \in \mathcal{T}, t' \leq w_j - h_j \text{ and } t' \notin \mathcal{T}_j\} \qquad (2.92)$$

Parameter settings for the computational experiments

For the ALNS algorithm, we use the same parameter vector $(r, n_T, \delta^1, \delta^2, \delta^3, \rho)$ as Stenger et al. (2013) and Bodnar et al. (2017), namely $(0.95, 200, 9, 3, 1, 0.3)$. We find that the solution quality and the computation time of the ALNS algorithm is largely determined by the choice for parameter w, which determines the initial temperature of the simulated annealing acceptance criterion. We choose the value $w = 0.05$. However, we cannot ensure the optimality of this parameter choice.

2.8.4 Additional numerical results

Performance neighborhood operators

As discussed in Section 2.8.3, the neighborhood operators to modify a solution are selected proportional to their performance. Table 2.17 and Table 2.18 present for each operator the frequency that it is selected and the relative improvement when it is selected, respectively. The operators that focus on either truck scheduling or dock-door assignment decisions are used equally and diversely. However, the assignment operator $rD2D$ is used more frequently when the size of the cross-dock terminal is larger. Among the scheduling operators, $rRd\&iFwd$, $rCr\&iBck$, $rMx\&iFwd$, $rI2O$ and $richFwd$ are used most frequently. In regard to the improvement to the solution, $rRd\&iFwd$, $rMx\&Bck$, $rCr\&Fwd$ and $rD2D$ generate solutions that result in the reduction of operational costs by 15.87%, 11.35%, 20.03% and 8.70%, respectively. Note that sometimes a small improvement by a different operator can be crucial to prevent the algorithm to end up in a local minimum.

Non-unit processing time

Similar to Bodnar et al. (2017), we have also studied the performance of the ALNS algorithm when the processing time h_j of each truck j equals four time periods (rather than $h_j = 1$). Table 2.19 and Table 2.20 present the results of this experiment for 20 and 30 trucks instances, respectively. The sequential MIP models in Section 2.8.2 is solved for 3,600 seconds in total (each model in the sequential approach is given 1,800 seconds) as in previous sections to generate the benchmark solutions (*Seq. obj.*). The solutions indicated in bold are optimal sequential solutions. Of the 160 instances, the solver is able to generate 114 (71.25%) optimal solutions within the termination limit. For the smaller instance with 20 trucks, all solutions are optimal. However, for the larger instances, the solver can generate optimal solutions only for 34 out of 80 instances. The ALNS solutions are within 6% of the optimal sequential solutions. On average the improvement of ALNS solution is 3.6% over the optimal sequential solution. At best, the ALNS solution is 16% better than the optimal sequential solution.

2.8.5 Additional results case study

Current solution approach

The retailer in our case study uses a *constructive heuristic* to schedule and assign trucks. First, inbound trucks are scheduled and assigned according to a first-come-first-served principle. When multiple dock doors are available, the assignment is done manually based on intuition. Next, outbound trucks are scheduled according their earliest due time. To assign the first outbound truck, the outbound door furthest to one end of the cross-dock terminal is selected. The second outbound truck is assigned to the available outbound door adjacent to the previously selected door, etc. This means that they assign outbound trucks from one end of the terminal until they reach the other end, and then they start all over again. Figure 2.8 shows a Gantt chart (similar to the solution representation introduced in Section 2.5.1) for the cross-dock scheduling problem on a particular day at the North facility, which has 31 dock doors.

Table 2.17: Frequency (%) that a neighborhood operator is selected

layout	rRd & iBck	rRd & iFwd	rRd& iSwp	rRd & iUp	rRd & iDwn	rMx & iBck	rMx & iFwd	rCr & iBck	rCr & iFwd	rCr & iSwp	rSy & iBck	rSy & iFwd	rSn & iSwp	rI20	rich Bck	rich Fwd	rD2D
(a) number of dock doors = 5																	
M1	6.08	5.64	5.89	7.92	5.7	4.18	4.49	5.93	4.23	4.66	4.79	4.17	4.24	2.97	4.11	3.93	21.07
M2	4.87	5.3	3.13	17.84	5.66	3.77	4.34	5.68	4.45	3.11	4.26	4.95	3.79	8.7	3.61	4.84	11.69
M3	3.89	3.72	3.22	10.6	4.74	3.67	3.3	4.41	3.55	2.79	3.26	3.38	3.01	2.71	2.62	3.27	37.85
M4	4.2	4.14	3.3	12.94	5.6	3.87	3.57	5.35	3.71	3.05	3.75	3.46	3.41	1.96	2.66	4.24	30.78
M5	4.93	4.31	2.78	10.1	4.63	3.93	4.01	5.5	4.2	4.21	3.73	4.16	3.98	13.42	2.98	5.03	18.1
M6	5.78	5.84	3.82	13.03	6	5.12	5.06	9.08	5.44	3.84	4.8	4.99	3.61	4.18	3.06	7.08	9.26
M7	6.12	5.78	4.72	12.74	5.76	5.61	4.68	7.77	4.68	4.44	5.02	4.71	2.85	4.14	3.77	7.23	9.98
M8	4.9	5.54	3.73	11.47	4.83	3.81	3.97	6.86	4.01	2.97	3.53	4.53	3.01	6.8	2.95	5.23	21.86
M9	4.03	5	3.14	12.28	3.89	3.15	3.11	4.43	4.15	3.15	2.93	2.92	2.72	17.71	2.57	4.41	20.41
M10	3.98	4.8	2.81	15.26	3.97	3.43	3.48	4.48	4.47	3.52	3.25	4.04	3.33	7.41	2.73	4.56	24.48
(b) number of dock doors = 10																	
M11	5.65	5.41	4.85	8.62	7.21	4.54	4.75	6.13	4.01	4.39	4.57	3.96	4.13	5.17	3.15	4.48	18.99
M12	4.93	4.67	3.57	12.3	6.7	4.7	4.76	5.26	4.26	3.11	4	4.18	3.75	2.39	2.72	4.19	24.52
M13	5.35	4.97	3.68	11.1	7.25	4.49	5.01	5.89	4.07	3.56	4.01	3.95	3.15	3.05	3.18	4.64	22.64
M14	5.25	4.95	3.57	11.67	7.08	4.87	4.76	5.27	4.37	3.7	4.07	4.07	3.44	3.45	3.05	5.04	21.38
M15	4.95	5.1	2.97	11.42	5.65	4.71	4.96	5.93	4.64	3.65	4.06	4.75	3.36	4.11	3.06	5.19	21.47
M16	5.29	5.19	3.37	10.25	6.25	4.61	5.43	5.99	4.83	3.2	4.03	4.46	3.17	2.98	2.7	5.44	22.81
M17	5.29	5.17	3.44	10.48	6.19	4.87	5.44	5.62	4.38	3.35	4.07	4.46	3.49	3.99	3.07	6.1	20.58
M18	4.63	5.39	3.69	9.83	5.76	4.24	5.47	4.67	4.83	3.02	4.02	4.83	3.05	10.23	2.73	6.66	16.93
M19	4.87	5.35	3.27	9.12	6.15	4.47	5.13	5.31	4.62	3.14	4.01	4.79	3.27	5.98	3.06	6.88	20.58
M20	5.23	5.63	3.47	10	6.12	4.8	5.94	5.94	4.96	2.96	3.78	4.92	3.58	5.29	2.96	6.96	17.46
(c) number of dock doors = 20																	
M21	6.65	6.1	3.78	8.38	7.87	7.15	6.78	6.25	4.37	2.92	5.15	4.59	3.01	3.78	3.14	4.41	15.68
M22	5.47	5.09	3.59	11.81	8.05	6.15	5.61	5.16	4.23	2.71	4.41	4.01	2.98	4.96	2.75	4.8	18.21
M23	5.7	5.89	2.97	10.78	7.42	6.63	5.77	6.39	4.61	3.03	4.51	4.47	3.19	4.15	2.84	4.37	17.28
M24	5.68	5.51	3.16	11.81	7.87	6.76	6.52	5.64	4.24	3.22	4.73	4.07	2.73	4.2	2.85	4.31	16.69
M25	5.13	5.04	3.17	11.79	8.01	5.19	5.49	5.06	4.21	3.02	4.57	4.11	2.79	7.58	2.74	5.1	17
M26	5.72	5.65	2.92	10.68	7.33	5.72	5.75	5.46	4.67	3.02	4.74	4.24	2.62	6	3.07	4.81	17.6
M27	5.55	5.64	3.11	11.93	7.57	6.35	6.33	5.43	4.58	3.01	4.83	4.51	2.53	4.93	2.89	5.04	15.77
M28	4.97	5.87	3.26	10.16	8.57	5	6.51	5	4.4	2.92	4.73	4.59	2.73	7.32	3.07	5.54	15.38
M29	4.9	5.65	3.3	10.13	7.49	5.83	5.84	5.56	4.36	2.91	4.19	4.33	2.56	9.27	2.7	4.71	16.3
M30	4.73	5.7	3.2	10.67	8.02	5.49	6.45	5.28	4.42	2.65	4.61	4.43	2.77	7.76	3.08	5.11	15.63

Table 2.18: Relative improvement (%) of the best known solution by the neighborhood operators

layout	rRd & iBck	rRd & iFwd	rRd& iSwp	rRd & iUp	rRd & iDwn	rMx & iBck	rMx & iFwd	rCr & iBck	rCr & iFwd	rCr & iSwp	rSy & iBck	rSy & iFwd	rSn & iSwp	rI20	rich bck	rich fwd	rD2D
(a) number of dock doors = 5																	
M1	0.93	0.43	0.45	1.8	2.15	0	0.15	1.46	0.41	0.67	84.44	0.72	1.79	1.58	0	1	2.04
M2	0.52	0.31	1.34	1.95	1.29	0.37	84.4	2.56	0.58	1.43	0.3	0.81	0.9	0	0.03	2.23	0.97
M3	1.43	1.56	0.68	1.15	1.67	1.13	87.7	1.76	0.12	0.26	0.11	0.69	0.1	0	0.31	0.32	1.04
M4	1.37	1.51	1.12	0.42	7.69	0.26	81.07	2.6	0.81	0	0.11	0.84	0.5	0.06	0	0	1.66
M5	0.79	3.51	0.01	2.27	1.99	0.7	0	1.33	80.45	0.5	0.18	2.51	0.39	0.95	0.19	0.42	3.8
M6	1.22	2.53	0.46	1.03	1.73	0.52	0.4	1.22	86.71	1.13	0.42	0.63	0.2	0	0.34	1.47	0
M7	2.66	2.14	1.17	1.87	5.35	1.11	0.72	3.11	71.86	0.07	0.11	0.77	0.04	0.45	0.47	4.33	3.77
M8	0.58	5.23	1.01	71.08	4.95	0.93	0.75	1.91	1.46	0.77	1.09	1.6	0.58	0.15	0.27	1.79	5.84
M9	2.7	0.99	0.75	77.53	2.09	1.08	1.3	0.4	0.21	1.15	0.8	0.83	1.05	1.26	0.35	3.61	3.91
M10	0.42	2.7	1.39	71.85	7.64	0.42	2.28	2.17	0.36	0.56	0.08	1.69	0.88	2.95	0	1.47	3.15
(b) number of dock doors = 10																	
M11	1.21	0.58	1.64	0.85	20.79	17.5	0.04	29.37	0.38	20.23	3.36	0.19	0.64	0.13	0.05	0.43	2.62
M12	0.92	21.1	0.43	0.83	0.75	21.27	0.81	0.95	3.95	0.32	0.25	0.54	0.46	0.04	0.03	29.47	17.88
M13	0.68	20.44	0.59	1.47	1.01	21.49	0.44	1	4.24	0.7	0.33	0.47	0.41	0.13	0.11	27.28	19.21
M14	1.67	19.75	0.81	1.38	0.84	21.78	0.39	0.78	4.51	0.4	0.15	0.32	0.55	0.03	0.13	26.92	19.59
M15	0.57	23.43	25.81	1.42	1.03	0.36	1.12	1.21	34.71	0.58	0.31	0.69	5.18	0.67	0.19	0.88	1.83
M16	0.92	23.27	26.77	2.03	1.45	0.48	0.81	1.14	33.26	0.43	0.39	0.87	4.05	0.22	0.05	0.95	2.91
M17	0.64	22.53	26.7	2.05	1.3	0.58	0.89	1.2	32.57	0.44	0.32	1.33	3.97	0.25	0.14	1.85	3.22
M18	0.99	16.11	4.83	2.5	22.25	17.82	1.19	1.26	24.25	0.31	0.56	0.62	0.3	1.87	0.1	1.8	3.24
M19	0.99	17.99	3.77	2.78	19.35	15.15	0.9	1.5	28.81	0.47	0.28	1.08	0.42	0.46	0.18	2.05	3.8
M20	0.93	16.91	3.96	3.17	19.2	15.16	1.55	1.55	29.15	0.28	0.58	1.15	0.47	0.17	0.16	2.21	3.39
(c) number of dock doors = 20																	
M21	2.37	1	1.21	3.06	2.25	41.25	0.91	1.09	0.77	27.87	3.27	0.56	0.58	0.23	0.15	2.31	11.14
M22	1.2	1.35	1.15	4.53	2.96	0.92	0.87	1.35	67.62	0.57	0.29	2.15	3.11	1.09	0.15	2.22	8.47
M23	1.35	1.58	1.42	4.46	2.37	0.79	1.39	1.47	63.12	0.66	0.83	0.75	3.3	0.2	0.41	3.16	12.75
M24	0.67	1.55	0.7	5.08	2.56	1.51	2.14	1.93	63.65	1.07	0.45	1.12	2.87	0.2	0.52	2.69	11.29
M25	0.81	1.71	0.86	3.31	2.14	0.39	1.89	1.27	1.54	0.79	0.24	27.52	0.63	1.16	38.93	2.38	14.42
M26	0.94	2.78	1.38	4.23	3.71	0.95	2.14	0.92	1.89	0.78	1.14	23.89	0.47	0.12	37.49	2.08	15.09
M27	0.84	1.7	1.18	5.26	3.03	0.78	1.24	0.87	2.81	1.27	0.83	24.71	0.69	0.59	37.18	3.13	13.9
M28	1.26	38.06	0.89	3.58	3.12	0.88	1.49	1.64	1.75	0.83	1.13	26.2	5.94	1.54	0.19	1.54	9.95
M29	1.12	39.36	1.46	4.92	3.74	1.03	1.97	1.11	2.62	0.72	0.89	23.13	3.21	0.16	0.73	2.67	11.17
M30	1.7	39.15	1.01	5.81	2.9	0.44	2.14	1.44	2.49	1.27	1.04	23.35	2.55	0.31	0.78	2.92	10.71

Table 2.19: Sequential solution compared to integrated ALNS solutions for instances with 10 dock doors, 20 trucks and non-unit processing time $\boldsymbol{h_j = 4 \forall j \in \mathcal{O}}$

	$\eta = 1, \beta = 50.4, \gamma = 100.8$								$\eta = 1, \beta = 50.4, \gamma = 504$							
	$d_i - r_i = 2$				$d_i - r_i = 3$				$d_i - r_i = 2$				$d_i - r_i = 3$			
	Seq.		ALNS		Seq.		ALNS		Seq.		ALNS		Seq.		ALNS	
layout	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time
(a) nr. of trucks = 20, average number of destinations per inbound truck = 4																
M11	**2693**	3	-6 %	288	**3713**	3	6 %	274	**9799**	3	-6 %	312	**11042**	9	-8 %	251
M12	**2244**	5	-1 %	356	**3259**	69	-2 %	340	**8203**	3	0 %	434	**9700**	72	-3 %	315
M13	**2244**	4	-3 %	362	**3803**	78	-6 %	312	**8491**	5	-2 %	511	**10132**	74	-4 %	323
M14	**2244**	3	-5 %	354	**3803**	76	-4 %	322	**8617**	4	-2 %	474	**10132**	80	-2 %	319
M15	**2218**	12	0 %	387	**3316**	51	-7 %	343	**8047**	11	0 %	559	**9257**	44	1 %	358
M16	**2218**	30	1 %	389	**3719**	58	-13 %	364	**8220**	27	0 %	450	**9674**	53	-2 %	341
M17	**2218**	19	-3 %	408	**3719**	65	-7 %	396	**8256**	18	-1 %	423	**9710**	55	-2 %	355
M18	**2218**	286	1 %	446	**3139**	20	-4 %	442	**7928**	285	1 %	558	**9089**	69	2 %	405
M19	**2218**	166	1 %	402	**3506**	27	-6 %	417	**8036**	167	0 %	640	**9384**	86	-2 %	395
M20	**2218**	141	0 %	410	**3528**	23	-9 %	441	**8044**	141	0 %	706	**9384**	73	-1 %	408
(b) nr. of trucks = 20, average number of destinations per inbound truck = 6																
M11	**3881**	6	-7 %	276	**3863**	11	-6 %	278	**11299**	4	-2 %	501	**9852**	9	-9 %	266
M12	**3538**	22	-6 %	333	**3314**	82	-9 %	340	**10038**	16	-1 %	541	**8712**	28	-10 %	349
M13	**3538**	22	-6 %	344	**3707**	79	-12 %	351	**10229**	13	-3 %	584	**8978**	48	-9 %	332
M14	**3538**	22	-7 %	331	**3714**	79	-9 %	342	**10283**	13	-3 %	569	**9029**	31	-7 %	339
M15	**3274**	13	-7 %	377	**3221**	24	-6 %	382	**9382**	9	-1 %	569	**8501**	21	-7 %	353
M16	**3274**	21	-8 %	386	**3696**	42	-16 %	376	**9551**	21	-2 %	496	**8976**	39	-11 %	371
M17	**3274**	15	-9 %	386	**3696**	32	-16 %	397	**9551**	13	-2 %	493	**8976**	29	-12 %	395
M18	**3089**	3	-5 %	406	**2959**	21	-2 %	436	**8818**	3	-1 %	440	**8028**	32	-4 %	426
M19	**3089**	3	-8 %	376	**3283**	39	-8 %	419	**8936**	3	-1 %	455	**8320**	85	-6 %	434
M20	**3089**	3	-6 %	423	**3305**	60	-9 %	449	**8936**	3	-1 %	399	**8320**	50	-6 %	434

Seq. obj. is the objective function value of the best solution for the MIP formulations in the sequential approach, which serves as benchmark; *ALNS* Δ(%) is the relative performance of the solution found with our*ALNS* algorithm. Any *Seq. obj.* value in bold indicates that the optimal solution is found in the sequential MIP formulation; Computation time of the solution approaches is represented by *time*.

Table 2.20: Sequential solution compared to integrated ALNS solutions for instances with 10 dock doors, 30 trucks and non-unit processing time $\boldsymbol{h_j = 4\,\forall j \in \mathcal{O}}$

	$\eta = 1, \beta = 50.4, \gamma = 100.8$								$\eta = 1, \beta = 50.4, \gamma = 504$							
	$d_i - r_i = 2$				$d_i - r_i = 3$				$d_i - r_i = 2$				$d_i - r_i = 3$			
	Seq.		ALNS		Seq.		ALNS		Seq.		ALNS		Seq.		ALNS	
layout	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time	obj.	time	Δ(%)	time
(a) nr. of trucks =30, average number of destinations per inbound truck = 4																
M11	**7498**	27	-6 %	337	**7601**	156	-6 %	392	**21878**	35	-1 %	328	**22098**	133	-2 %	372
M12	**6706**	436	-6 %	437	6613	1814	-5 %	549	18836	1819	-2 %	417	19813	1809	-2 %	427
M13	**6706**	438	-4 %	430	7128	1811	-7 %	518	19128	1816	-3 %	426	20399	1824	-4 %	479
M14	**6706**	433	-4 %	426	7121	1818	-6 %	521	19168	1819	-2 %	410	20384	1803	-1 %	449
M15	**6283**	88	-5 %	485	5930	1811	-5 %	651	**16909**	62	-3 %	491	18610	1833	-1 %	506
M16	**6283**	90	-4 %	463	6517	1841	-5 %	601	**17179**	53	-2 %	475	19063	1833	-1 %	503
M17	**6283**	84	-6 %	451	6517	1847	-7 %	587	**17244**	52	-2 %	466	19063	1830	-3 %	517
M18	**6046**	35	-3 %	547	5725	1820	-2 %	664	**16192**	29	-1 %	537	17520	1820	2 %	547
M19	**6046**	39	-6 %	518	6082	1846	-8 %	713	**16404**	27	-2 %	508	17876	1865	0 %	555
M20	**6046**	63	-6 %	533	6082	1843	-9 %	762	**16404**	31	-2 %	522	17876	1806	0 %	546
(b) nr. of trucks = 30, average number of destinations per inbound truck = 6																
M11	**9108**	127	-8 %	393	**11262**	742	-2 %	520	**26282**	120	-2 %	343	36169	1801	-1 %	354
M12	7946	1808	-6 %	473	10111	1835	-6 %	756	22886	1821	-3 %	400	32576	1812	-4 %	455
M13	7946	1822	-5 %	485	10506	1831	-4 %	645	23146	1828	-3 %	441	32915	1806	-3 %	450
M14	7946	1818	-7 %	568	10528	1822	-4 %	593	23228	1809	-3 %	436	32994	1810	-2 %	454
M15	**6785**	646	1 %	574	8995	1867	-1 %	636	**19877**	1199	-2 %	508	28898	1817	0 %	488
M16	**6785**	619	-5 %	556	9618	1853	-5 %	629	**20240**	1211	-2 %	503	29262	1811	-1 %	530
M17	**6785**	636	-2 %	578	9618	1839	-5 %	713	**20240**	1198	-2 %	508	29262	1821	-1 %	511
M18	**6310**	297	0 %	631	8365	1818	2 %	735	**18824**	208	2 %	526	28064	1810	2 %	576
M19	**6310**	299	-1 %	617	8642	1809	0 %	756	**19130**	213	1 %	525	28270	1805	-1 %	534
M20	**6310**	301	-1 %	627	8642	1810	3 %	684	**19130**	218	2 %	540	28270	1811	0 %	584

Seq. obj. is the objective function value of the best solution for the MIP formulations in the sequential approach, which serves as benchmark; *ALNS* Δ(%) is the relative performance of the solution found with our*ALNS* algorithm. Any *Seq. obj.* value in bold indicates that the optimal solution is found in the sequential MIP formulation; Computation time of the solution approaches is represented by *time*.

This Gantt chart provides an intuitive understanding of the constructive heuristic. Outbound trucks are scheduled and assigned in a cascading manner from door 1 to door 31, where the outbound trucks are numbered according to their due times. Once the furthest outbound door has been assigned, the assignment process is repeated from door 1 again.

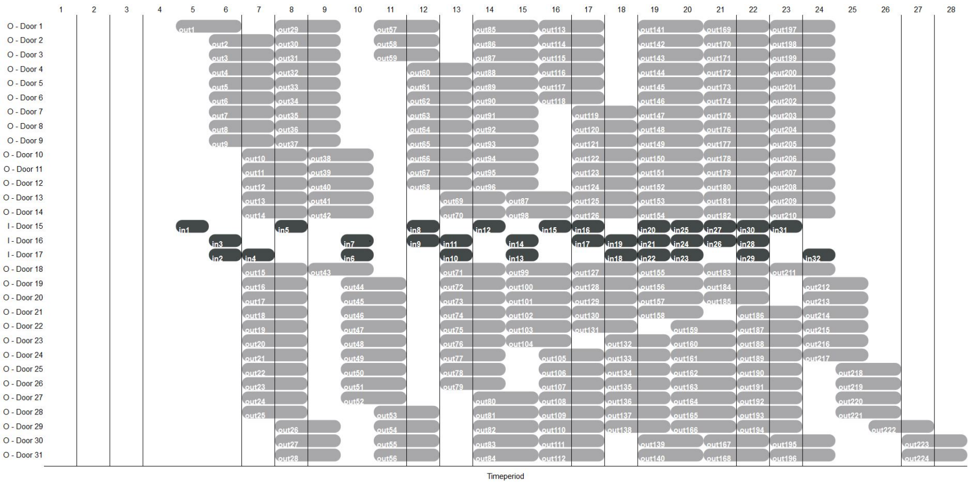


Figure 2.8: Gantt chart for the scheduling and assignment decisions of inbound (in) and outbound (out) trucks at the North cross-dock facility with 3 inbound doors (doors 15-17) and 28 outbound dock doors on a particular day

The performance of the constructive heuristic is presented in Table 2.21. It is clear that the retailer prioritizes the scheduling of outbound trucks based on their due times. In only one cross-dock facility, four outbound trucks (i.e., 0.53%) leave beyond their due time with a total delay of 480 minutes. Furthermore, the time windows to process the outbound trucks are quite large, which also prevents possible delays. However, the constructive heuristic ignores the inbound operations when the scheduling and assignment decisions are made for the outbound trucks. Consequently, the temporary storage area is highly utilized and the internal travel distance between dock doors is large. In total, 1,554 unit loads are stored temporarily per day. This corresponds to 51% of all truck loads being processed.

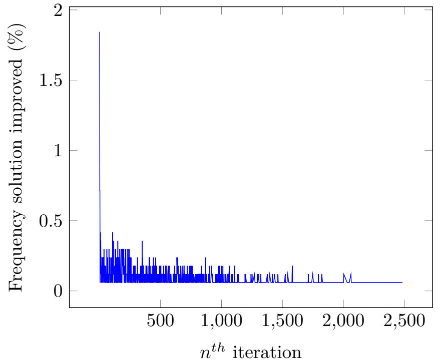
Performance ALNS algorithm

When the ALNS algorithm is used to solve the cross-dock scheduling problem at the retailer, Figure 2.9 illustrates that about 98% of the solution improvement is obtained

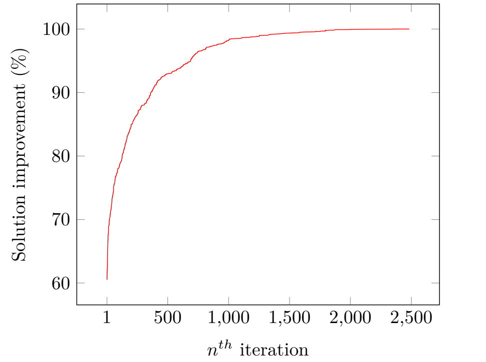
Table 2.21: Solution of the retailer's constructive heuristic

Cross-dock Facility Location	#trucks		#doors		Retailer's Constructive Heuristic		
	IN	OUT	IN	OUT	Delayed outbound (min)	#units in temporary storage	Internal travel (km)
East	27	171	6	23	480	222	30.16
West	27	160	5	30	0	387	44.05
North	32	224	3	28	0	502	46.17
South	29	200	5	29	0	443	49.72

in the first 1,000 iterations of the ALNS algorithm. Consequently, the number of iterations to find a solution to the problem can be reduced without harming the quality of the solution.



(a) Frequency of iterations with solution improvement



(b) Cumulative solution improvement by iteration

Figure 2.9: Frequency and proportion of the best solution improvement by iteration for the case study instances

2.8.6 Computational experiments for I-shaped cross-dock

In Section 2.6, all numerical results are based on layout configurations for a U-shaped cross-dock terminal. To illustrate that our integrated solution methodology of Section 2.5 can also be applied to cross docks with other shapes, we show the performance of our ALNS algorithm for different layouts of I-shaped facilities with 10 dock doors. The I-shaped terminal is represented by dock doors on two sides of the cross dock. The layout configurations are presented in Table 2.22. Unlike the U-shaped cross-dock terminal, the I shape requires two sides of the cross dock to be separated. In the representation of the layout, the notation **0** is used to differentiate between the two sides of the facility with dock doors. The distance between the two walls with dock doors is 50 meters. The adjacent dock doors on the same side are 3.6 meters apart from one another (the same as before). Consequently, the distance between two dock doors k and $p \in \mathcal{D}$ equals

$$m_{kp} = \begin{cases} 3.6\,|k-p| + 50 & \text{if } k,p \leq |\mathcal{D}|/2 \\ 3.6\,|k-p| + 50 & \text{if } k,p > |\mathcal{D}|/2 \\ 3.6\left(\left|\max\{k,p\} - |\mathcal{D}|/2 - \min\{k,p\}\right|\right) + 50 & \text{otherwise} \end{cases}$$

The cost for using the temporary storage and for tardiness of trucks is estimated in the same manner as for U-shaped terminals. In particular, the cost of temporary storage is twice the average dock-to-dock travel distance

$$\beta = 2 \cdot 3.6 \cdot \left(\frac{2 \cdot \sum_{i=1}^{|D|/2-1} i(i+1)}{|D| \cdot (|D|-1)} \right) + 50$$

and the tardiness cost is either twice or ten times the cost of temporary storage.

Table 2.22: The layout configurations of the inbound doors (i), outbound doors (o) and flexible doors (f) at an I-shaped cross-dock terminal with 10 dock doors

layout	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{D}_I\rvert$	$\lvert\mathcal{D}_O\rvert$	$\lvert\mathcal{D}_F\rvert$	Mixed-mode Strategy	Layout Configuration
I11	10	5	5	0	None	$\Vert i\Vert i\Vert i\Vert i\Vert i\Vert \mathbf{0}\Vert o\Vert o\Vert o\Vert o\Vert o\Vert$
I12	10	4	4	2	Center	$\Vert i\Vert i\Vert f\Vert i\Vert i\Vert \mathbf{0}\Vert o\Vert o\Vert f\Vert o\Vert o\Vert$
I13	10	4	4	2	Half-center	$\Vert i\Vert f\Vert i\Vert i\Vert i\Vert \mathbf{0}\Vert o\Vert f\Vert o\Vert o\Vert o\Vert$
I14	10	4	4	2	Outer	$\Vert f\Vert i\Vert i\Vert i\Vert i\Vert \mathbf{0}\Vert f\Vert o\Vert o\Vert o\Vert o\Vert$
I15	10	2	2	4	Center	$\Vert i\Vert i\Vert f\Vert f\Vert i\Vert \mathbf{0}\Vert o\Vert o\Vert f\Vert f\Vert o\Vert$
I16	10	2	2	4	Half-center	$\Vert i\Vert f\Vert i\Vert f\Vert i\Vert \mathbf{0}\Vert o\Vert f\Vert o\Vert f\Vert o\Vert$
I17	10	2	2	4	Outer	$\Vert f\Vert i\Vert i\Vert i\Vert f\Vert \mathbf{0}\Vert f\Vert o\Vert o\Vert o\Vert f\Vert$
I18	10	2	2	6	Center	$\Vert i\Vert f\Vert f\Vert f\Vert i\Vert \mathbf{0}\Vert o\Vert f\Vert f\Vert f\Vert o\Vert$
I19	10	2	2	6	Half-center	$\Vert f\Vert f\Vert i\Vert f\Vert i\Vert \mathbf{0}\Vert f\Vert f\Vert o\Vert f\Vert o\Vert$
I20	10	2	2	6	Outer	$\Vert f\Vert f\Vert i\Vert i\Vert f\Vert \mathbf{0}\Vert f\Vert f\Vert o\Vert o\Vert f\Vert$

The results are presented in Table 2.23 and Table 2.23 for instances with 30 and 60 trucks, respectively. We can compare these results to the results of the U-shaped terminal in Table 2.7 and Table 2.8, and we draw similar conclusions. The cost increase for using the metaheuristics in the sequential approach is lower for the I-shaped terminal, but our ALNS algorithm still outperforms these metaheuristics. More importantly, it still outperforms the optimal solution in the sequential approach for almost all instances. Furthermore, these initial results seem to suggest positioning the flexible doors according to the half-center configuration. More numerical results under different circumstances (similar to Section 2.6) are needed to confirm this finding for I-shaped terminals.

Table 2.23: Performance of our integrated ALNS algorithm and the existing metaheuristics from the literature for the sequential approach compared to the best solutions to the MIP formulations in the sequential approach for instances with 10 dock doors, 30 trucks and low dock-door utilization rates in an I-shaped cross-dock terminal

	$\eta = 1, \beta = 112.8, \gamma = 225.6$						$\eta = 1, \beta = 112.8, \gamma = 1128$					
	$d_i - r_i = 2$			$d_i - r_i = 3$			$d_i - r_i = 2$			$d_i - r_i = 3$		
layout	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)
(a) nr. of trucks = 30, average number of destinations per inbound truck = 4, utilization = 18.75%												
I11	**37020**	0%	-5%	**35340**	0%	-7%	**70631**	0%	-2%	**81242**	0%	-3%
I12	**36524**	14%	-5%	33856	16%	-5%	**69462**	8%	-2%	**80015**	10%	-3%
I13	**36531**	10%	-5%	33852	20%	-6%	**69462**	5%	-2%	**79964**	10%	-2%
I14	**36545**	1%	-5%	33863	5%	-7%	**69476**	8%	-2%	**80000**	6%	-3%
I15	**36398**	6%	-5%	**33210**	18%	-4%	**68126**	9%	-1%	**78799**	17%	-2%
I16	**36377**	6%	-5%	**33181**	30%	-4%	**68108**	4%	-2%	**78784**	23%	-2%
I17	**36438**	7%	-5%	**33242**	28%	-5%	**68166**	4%	-2%	**78802**	6%	-2%
I18	**36408**	2%	-5%	**32482**	33%	-2%	**68169**	0%	-2%	**78217**	15%	-1%
I19	**36364**	2%	-5%	**32503**	33%	-1%	**68122**	1%	-2%	**78210**	12%	-1%
I20	**36410**	2%	-5%	**32394**	34%	-2%	**68164**	0%	-2%	**78195**	11%	-2%
(b) nr. of trucks = 30, average number of destinations per inbound truck = 6, utilization = 18.75%												
I11	**40784**	0%	-8%	**44603**	1%	-6%	**81213**	0%	-3%	**115629**	0%	-2%
I12	**39272**	10%	-6%	43192	9%	-4%	**79842**	9%	-3%	**110774**	5%	-1%
I13	**39261**	7%	-5%	**43192**	9%	-5%	**79850**	4%	-3%	**110778**	5%	-1%
I14	**39265**	9%	-6%	**43206**	6%	-3%	**79878**	10%	-3%	**110796**	6%	-1%
I15	**37828**	16%	-1%	**42047**	12%	-3%	**76687**	8%	-1%	**108297**	10%	0%
I16	**37810**	13%	-1%	**42015**	12%	-3%	**76672**	9%	-1%	**108272**	10%	0%
I17	**37818**	13%	-3%	**42025**	12%	-1%	**76708**	9%	-1%	**108276**	10%	0%
I18	**37163**	10%	0%	**40960**	12%	0%	**77742**	11%	-2%	**108466**	11%	0%
I19	**37152**	11%	-1%	**40903**	12%	1%	**77710**	11%	-2%	**108416**	7%	0%
I20	**37137**	11%	0%	**40877**	12%	1%	**77698**	12%	-2%	**108422**	7%	0%

Seq. obj. is the objective function value of the best solution for the MIP formulations in the sequential approach, which serves as benchmark; *Seq MH* Δ(%) is the relative performance of the solution found with metaheuristics in the sequential approach; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm. Any objective value in bold indicates that the optimal solution is found in the sequential MIP formulation.

Table 2.24: Performance of our integrated ALNS algorithm and the existing metaheuristics from the literature for the sequential approach compared to the best solutions to the MIP formulations in the sequential approach for instances with 10 dock doors, 60 trucks and low dock-door utilization rates in an I-shaped cross-dock terminal

	$\eta = 1, \beta = 112.8, \gamma = 225.6$						$\eta = 1, \beta = 112.8, \gamma = 1128$					
	$d_i - r_i = 2$			$d_i - r_i = 3$			$d_i - r_i = 2$			$d_i - r_i = 3$		
layout	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)	Seq. obj.	Seq MH Δ(%)	ALNS Δ(%)
(a) nr. of trucks = 60, average number of destinations per inbound truck = 4, utilization = 37.5%												
I11	**72959**	1%	-4%	**65124**	3%	-1%	**130441**	2%	-4%	**109607**	3%	-2%
I12	71433	27%	-3%	73234	12%	-11%	127036	50%	-2%	107064	34%	1%
I13	71437	34%	-4%	64461	12%	-1%	127008	44%	-2%	107032	37%	-1%
I14	71469	33%	-2%	64472	17%	-2%	127047	15%	-2%	107050	28%	3%
I15	69825	35%	0%	66200	22%	-6%	124803	38%	2%	107749	37%	-1%
I16	69771	36%	1%	66204	23%	-5%	124767	20%	0%	108285	47%	-1%
I17	69557	30%	2%	66204	28%	-3%	124796	10%	1%	108263	22%	-1%
I18	70542	23%	-1%	64545	22%	-5%	124715	31%	2%	105685	32%	-1%
I19	70107	17%	-1%	64462	24%	-1%	124654	34%	-1%	105667	33%	0%
I20	70046	18%	0%	64469	12%	-3%	124632	37%	0%	108965	21%	-3%
(b) nr. of trucks = 60, average number of destinations per inbound truck = 6, utilization = 37.5%												
I11	**81793**	0%	-2%	**77446**	0%	0%	**162530**	1%	-1%	**143025**	1%	-2%
I12	78935	17%	1%	84135	15%	-10%	**160074**	39%	-1%	134394	21%	-1%
I13	78931	26%	-1%	84103	5%	-10%	**160088**	25%	-2%	134369	27%	-3%
I14	78985	18%	1%	84106	5%	-10%	**160113**	26%	-2%	134412	28%	-3%
I15	78788	24%	0%	91402	3%	-20%	**157354**	31%	0%	143826	20%	-9%
I16	78745	22%	-1%	91402	-1%	-19%	**157310**	34%	-1%	143776	18%	-12%
I17	78810	21%	-2%	91430	-1%	-19%	**157343**	31%	0%	143840	22%	-10%
I18	84186	5%	-4%	86900	3%	-14%	**157545**	22%	0%	129336	14%	0%
I19	84110	16%	-7%	86850	-1%	-15%	**157534**	19%	-1%	129324	20%	-1%
I20	84063	18%	-7%	86921	-4%	-17%	**157540**	29%	-2%	133213	23%	-4%

Seq. obj. is the objective function value of the best solution for the MIP formulations in the sequential approach, which serves as benchmark; *Seq MH* Δ(%) is the relative performance of the solution found with metaheuristics in the sequential approach; *ALNS* Δ(%) is the relative performance of the solution found with our ALNS algorithm. Any objective value in bold indicates that the optimal solution is found in the sequential MIP formulation.

3 Workforce Scheduling with Order-Picking Assignments in Distribution Facilities[1]

3.1 Introduction

Scheduling order pickers is one of the fundamental decision problems in manual picker-to-part warehouses, where order pickers walk (or drive) to the storage locations of items to retrieve all the items specified in a picking list. The order picking process is one of the most labour-, time- and capital-intensive activities in warehouses, responsible for more than 50% of the operating costs (Tompkins et al., 2010). Despite the rise of automated order picking, less than 3% of the warehouses are fully automated and less than 10% of the warehouses use automated parts-to-picker systems (Michel, 2016). Specifically, Azadeh et al. (2019) estimate that only 40 out of thousands of warehouses in Western Europe are fully automated. Consequently, manual order picking has been studied extensively in the literature and most research focuses on the development of travel time or distance models for various storage assignment, picking routing and order batching policies (Van Gils et al., 2018b). In contrast, the *order picker planning* problem which assigns and sequences orders to order pickers has hardly been studied (Van Gils et al., 2018b). This is an important problem for warehouses where orders have temporal restrictions such as deadlines. The assignment and the sequence of execution of orders have a direct impact on the tardiness of orders and on the costs associated with the order picking operations. Furthermore, as order pickers are humans, the order picker planning problem is further constrained by *shift scheduling decisions*, which include decisions regarding the start and end times of shifts and breaks, as well as the workforce level requirements for different shifts. The literature on order picker planning ignores these shift scheduling decisions and only

[1] This chapter is based on an article that is currently under review at a journal.

considers a single shift horizon (i.e., shifts with one start and end time for all order pickers) without the need for breaks (Matusiak et al., 2014; Henn, 2015; Scholz et al., 2017; Matusiak et al., 2017). Consequently, the available solution approaches in the literature can only be applied in a straightforward manner to manual order picker planning problems in warehouses where orders do not have temporal restrictions.

Many distribution centers in Western Europe face two main restrictions in the order picking operations: *due time windows* of orders and *flexible order pickers*. On-time retrieval of customer orders has become more important nowadays with companies offering deliveries to customers within a small time interval (e.g., one or two business days). To ensure that customer orders are delivered on time, trucks have departure deadlines from the warehouse. In retail logistics, these departure deadlines can also be imposed by strict city access time window regulations (Quak & de Koster, 2007) and contractual agreements with retail stores (Bodnar et al., 2017). Besides these temporal restrictions, there are also spatial restrictions due to limited capacity at the outbound staging areas of warehouses to consolidate orders that need to be delivered by the same truck. Consequently, every order has a *due time window* during which it needs to be picked and sent to the allocated staging lane. These due time windows present severe challenges to warehouse managers in maintaining the right order picking workforce at the appropriate times. To cover demand during peak periods, a large number of order pickers is required. These order pickers can become superfluous when the volume of order picking tasks decreases. To alleviate this problem, warehouses employ *flexible order pickers* who can be called upon to work on short notice. The shift start and end times vary for these employees, but they are guaranteed a minimum payment equal to the payment corresponding to the minimum compensated duration, which is defined as the duration of time an order picker is paid for even if the order pickers is asked to work in a shift with a shorter shift length. Labour laws in many countries specify a minimum compensation duration. For instance, employees in the United States, Canada and Australia must be paid for at least 3 hours each time they are required to report to work. Under these circumstances, the aim of the warehouse manager is to solve the order picker planning problem such that due time windows of orders are respected while minimizing the labor cost. This requires them to determine how many order pickers to schedule (including the start times, end times, and breaks for each order picker), assign the orders that need to be picked during each shift, as well as the sequence in which the orders are picked by the order pickers. We call this optimization problem the *order picker scheduling problem* (OPSP). Most warehouse managers rely on their experience and intuition to make these decisions. Even though our study is inspired by the largest grocery retail chain in The Netherlands, the use of flexible order pickers with minimum compensation and one or multiple break periods is common in many countries. Our definitions of flexible order pickers and break requirements are fully compliant with the current European Union (EU) Directive 91/533/EEC (European Parliament, Council of the European Union, 1991) as well as the new Directive (EU) 2019/1152 (European Parliament, Council of the European Union, 2019) that will replace the current Directive in 2022. An overview of other labor laws around the world are included in Section 3.8.5. Consequently, our study is generally applicable and relevant to many manual order picking warehouses where orders have tight due times and the resources to prepare orders (such as the number of staging lanes) and order pickers are limited.

In this paper, we combine the order picker planning problem with the shift scheduling problem to jointly determine the scheduling of start, end and break times for the shifts of flexible order pickers as well as the assignment and sequencing of orders with due time windows to these order pickers. The shift decisions have direct implications for the order picking process that should not be ignored. In Section 3.8.1, an illustrative example is given that highlights the importance of explicitly constructing shifts that take breaks into account when orders have due time windows and order pickers are flexible. The contributions of our work are four fold: (i) We introduce the OPSP to the order picking literature and formulate the OPSP as a mixed integer linear program (MILP); (ii) To solve the problem, we present an exact branch-and-price algorithm in combination with an efficient heuristic to generate tight upper bounds based on the savings algorithm; (iii) We propose a computationally efficient metaheuristic that is capable of producing near-optimal solutions for large instances; (iv) A case study is performed to investigate the practical impact of flexible shift structures and show the impact can be substantial.

The outline of this paper is as follows. Relevant literature is reviewed in Section 3.2. Section 3.3 presents the problem description and the model formulation of the problem. In Section 3.4, we present a branch-and-price algorithm to find optimal solutions for the problem. A metaheuristic to solve the problem is proposed in Section 3.5. Results from computational experiments and the case study follow in Section 3.6. Finally, Section 3.7 concludes the paper.

3.2 Literature Review

As identified in the previous section, the OPSP operates at the intersection of shift (or personnel) scheduling and order picker planning. More details on both research streams in the literature are provided in this section.

Order picker planning problem

When orders have temporal restrictions (such as due time windows) or when they result in penalties when completed early or late, the assignment of orders to order pickers and the sequencing to execute these orders have a direct impact on the feasibility of workforce schedules and the associated costs. Elsayed et al. (1993) and Elsayed & Lee (1996) are the first authors to study the *joint order batching and sequencing problem* (JOBSP) for a single automated storage and retrieval system (AS/RS) where the objective is to minimize the earliness and tardiness of orders. The authors suggest simple heuristic methods to generate solutions for the problem. Henn & Schmid (2013) and Henn (2015) extend this work to multiple order pickers, which is considered the *joint order batching, assignment and sequencing problem* (JOBASP). The authors suggest iterated local search and attribute-based hill climber, variable neighborhood search and variable neighborhood depth algorithms to solve this problem. Tsai et al. (2008) introduce a *joint order batching, assignment, sequencing, and routing problem* (JOBASRP), which is an extension of the JOBASP with routing decisions for the order pickers within the warehouse. Chen et al. (2015) and Scholz et al. (2017) propose heuristic solution approaches for this problem. Matusiak et al. (2014) investigate a variation of JOBASRP where the sequencing of batches is not relevant but the routing is part of the optimization problem which aims at minimizing the overall travel distance. In most applications, the storage racks are stationary, however, Boysen et al. (2017) consider an interesting variation with mobile rack warehouses, where an entire storage aisle may need to be moved to access items in it. Here, the objective is to sequence orders to minimize the number of aisle relocations.

In a recent review on order picking problems, Van Gils et al. (2018b) note that there is hardly any literature on the integration of the order assignment and sequencing decisions for order pickers (i.e., the order picker planning problem) while determining the order picking workforce (i.e., the shift scheduling problem). All work in the literature on scheduling manual order pickers assumes a single shift start and end time without the need for a break, which can be traced back to Elsayed et al. (1993)

and Elsayed & Lee (1996). This simplifying assumption is only valid for machine environments or for manual order picking environments where a fixed number of order pickers can start and end their shift at only one given time, no breaks are scheduled and orders do not have temporal restrictions (as discussed in Section 3.1). When order pickers have fixed employment contracts, the shift scheduling decisions are typically made at a tactical level or at least before any order assignment and sequencing decisions are made. However, when order pickers have flexible employment contracts, it is crucial to make shift scheduling decisions at the same time as the order assignment and sequencing decisions are made. Figure 3.1 illustrates the typical order in which decisions are made in the two types of employment contracts. These differences require us to review shift scheduling literature which is done in the following. Note that batching is decoupled in both of the contracts because integrating optimal order batching with other decisions is computationally prohibitive in realistic settings. Furthermore, an appropriate batching policy alone can explain much of the variance in travel times of order pickers compared to related decisions (storage, zoning, and routing) (Van Gils et al., 2018a).

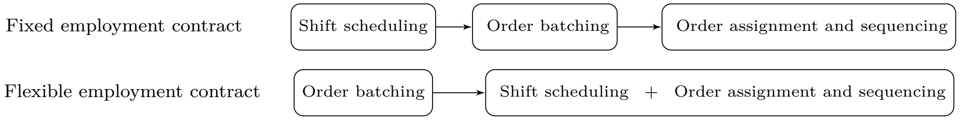


Figure 3.1: Sequence of decision problems with fixed and flexible employment contracts for order pickers

Shift (or personnel) scheduling problem

In contrast to the literature on order picking processes, the shift scheduling literature explicitly considers shift decisions as part of the planning problem. Shift scheduling is one of the oldest problems in the field of operations research. It dates back to Edie (1954) and Dantzig (1954) who scheduled toll booth operators and it has received a lot of attention in the literature since then (Ernst et al., 2004b,a; van den Bergh et al., 2013). Many of the mathematical formulations are based on a generalized set covering model where each possible shift (i.e., a combination of start time, end time, and break placement) is represented by a decision variable. The goal is to determine the optimal complement of shifts such that operational constraints are satisfied while optimizing some objective function. It has applications in many industries including airlines, public transportation, hospitality, military, health care and call centers.

Shift scheduling problems can be divided into two broad categories based on the type of workload they consider: *workload-coverage problems* or *task-coverage problems*. The main distinction between these two categories relies on what is known prior to performing the personnel planning. In *workload-coverage problems*, the actual tasks that need to be to executed during the planning horizon are not known by the time personnel is scheduled. Consequently, the *demand* for employees is forecasted based on expected workloads and workers are scheduled to cover these predicted personnel demands. Employees are usually scheduled to perform one type of task that can be preempted between employees working in different shifts (e.g., manning a cash register in a shop).

In contrast to workload-coverage problems, the actual *tasks* that need to be executed are known in *task-coverage problems*. Besides the creation of shifts, these problems also include assignment decisions of tasks to individual employees or shifts such that all, or as many as possible, tasks are completed. Consequently, task-coverage problems are generally more complicated to solve than workload-coverage problems. Task-coverage problems can be further divided into two subcategories: fixed task timing problems and flexible task timing problems. In *fixed task timing problems*, the timing when to execute each task is known *a priori* (therefore, sequencing decisions are not included). These problems aim to generate schedules that cover the fixed tasks with a minimum number of machines or shifts. Examples of these problems are fixed job scheduling problems (Fischetti et al., 1987, 1989), interval scheduling problems (Kroon et al., 1995; Kolen et al., 2007) and shift minimization personnel task scheduling problems (Krishnamoorthy et al., 2012). The navy personnel planning problem studied by Holder (2005) is a closely related problem. Another example of the fixed task timing problem is the *integrated task scheduling and personnel rostering problem*, which generates the roster of employees while explicitly considering the coverage of tasks (Smet et al., 2016). Beliën & Demeulemeester (2008) present a branch-and-price algorithm for the integrated rostering problem of nurses while incorporating the scheduling of tasks that arise from surgery schedules. When the planning horizon of the fixed task timing problem is divided in periods and the duration to execute each task is equivalent to the length of a period, this is known in the literature as the *multi-activity shift scheduling problem* (Côté et al., 2011; Elahipanah et al., 2013; Dahmen et al., 2018).

In *flexible task timing problems*, the timing when to execute tasks is a decision. Consequently, sequencing decisions have to be made besides the shift scheduling and task assignment decisions. For instance, home care workers are assigned to locations where tasks (such as cooking, cleaning and administering medicine) have to be performed within specific time windows (Rasmussen et al., 2012). Closely related problems include the *field workforce scheduling problem* where individual workers with appropriate skills are assigned to geographically distributed tasks (Alsheddy & Tsang, 2011) and the *technician task scheduling problem* where individuals with the correct skill mix are assigned to tasks of different priorities (Cordeau et al., 2010; Fırat & Hurkens, 2012).

Shift scheduling problem with breaks

The inclusion of breaks in the personnel scheduling literature is mainly limited to workload-coverage problems only. Thompson (1988) is one of the first authors to explicitly plan for breaks when shift schedules are generated. In its simplest form, the set covering formulation of Dantzig (1954) is extended with additional decision variables to represent breaks and reliefs. For problems involving a high degree of flexibility with respect to the timing of breaks, the number of enumerated shifts increases drastically and the resulting set covering problem can be very difficult to solve (if even possible). To overcome these challenges, Bechtold & Jacobs (1990) propose a compact formulation that implicitly considers breaks, but is tractable for realistic instances. This model is extended by Thompson (1995) to consider different types of breaks and even overtime. Aykin (1996) also presents a compact integer programming model that is capable of considering time windows for multiple breaks in one shift. Aykin (2000) shows that this model is computationally superior compared to the formulation in Bechtold & Jacobs (1990), who only consider one break in a shift. Sungur et al. (2017) present a goal programming approach for the same problem as studied by Aykin (1996).

The work of Bechtold & Jacobs (1990) is also extended by Brusco & Jacobs (2000) to introduce break and relief planning in *tour scheduling problems*. In this type of problem, the aim is to generate a schedule with multiple shifts for each employee as well as off days during which the employee is not working. Consequently, the planning horizon is longer compared to shift scheduling problems. Bard et al. (2007) also model a tour scheduling problem with break and labor rules but in a stochastic environment of a parcel sorting center. Gérard et al. (2016) present a heuristic that is based on column generation for a more extensive problem which simultaneously considers off days, shift scheduling, shift assignments and task assignments within shifts. A key difference of these problems compared to our OPSP is that the tasks have a fixed timing rather than a time window during which they need to be performed.

For flexible task timing problems, the scheduling of breaks is only included in the *truck driver scheduling problem*. In these problems, the sequence in which locations are visited by trucks has to be determined while satisfying appropriate time windows. The maximum amount of time a truck driver is allowed to be on the road is restricted such that breaks and rest periods have to be considered to satisfy the strict hours-of-service regulations (Goel, 2010; Goel & Kok, 2012). The truck driver scheduling problem is extended to vehicle routing decisions in Goel & Irnich (2017). In these studies, the objective of the problem is to minimize the travel distance. An alternative objective function for the problem is presented by Tilk & Goel (2020), where the problem aims to minimize the number of working days for a given route instead of the travel distance.

A comparison between OPSP and the available literature on shift scheduling problems can be found in Table 3.1. It becomes clear that the order picker planning problem does not consider shift scheduling decisions when orders have temporal restrictions. Furthermore, most of the flexible task timing problems in the shift scheduling literature do not consider the characteristics that are unique to warehouse environments (i.e., tasks with due time windows in combination with flexible workers who require breaks and a minimum payment).

In the shift scheduling literature with task assignments that have to be performed in a certain time window, only the truck driver scheduling problem considers breaks. It is therefore the closest related to our problem formulation. The break requirements for truck drivers considered in Goel & Irnich (2017) are similar to the break requirements for order pickers considered in the OPSP. However, a major difference is that Goel & Irnich (2017) focus on minimizing travel distances, whereas schedule durations do not play a role. The objective in Tilk & Goel (2020) is to minimize the sum of labor costs and distance-related costs whereas labor costs are related to the number of working days required to complete the route. The number of hours worked within a working day does not play a role and most schedules generated actually include long periods of waiting. The OPSP studied in our work combines elements of minimizing schedule duration with minimum compensated duration which make the OPSP structurally different from the aforementioned problems and it necessitates new solution approaches. Our work addresses this gap in the literature and combines the order picker planning literature and shift scheduling literature. In Section 3.4, we further explain the differences between our solution approach to solve the OPSP compared to other approaches in the literature.

3.3 Problem Description and Model Formulation

In this section, we explain the warehouse operations that define our order picker scheduling problem (OPSP) and we formulate the corresponding mixed integer linear program (MILP) model. Symmetry breaking constraints and additional constraints to tighten the model formulation are included in Section 3.8.3. An alternative formulation of the problem as a network flow problem is presented in Section 3.8.4. This formulation takes more computational effort to solve in our numerical experiments, it is therefore included as reference only. Extending the OPSP formulation with full-time order pickers and different types of break time constraints are discussed in Section 3.8.5.

Production facilities or retail stores place orders to receive items from a distribution warehouse based on their needs. An *order* is composed of multiple order lines, where each order line consists of a particular item and the corresponding requested quantity. The order lines that should be processed together create a *pick list*. The list contains all items that need to be picked and it guides the order picker through the warehouse. Items that are collected are put in roll cages such that products in the same roll cage are sent to a single customer. However, a customer's order can result in multiple roll cages picked by one or more order pickers. An order picker's tour finishes when all roll cages from the pick list are delivered to the corresponding staging lanes at the outbound docks. The total number of order lines and roll cages can exceed hundreds, which prohibits a joint optimization of the personnel scheduling and order batching problems within a reasonable computational effort. Consequently, we assume that order batching (i.e., the construction of pick lists) is done *a priori*. In the remainder of the paper, we use the term *batch* to refer to a pick list that is to be completed by a single order picker in a single pick tour.

Let I be the set of batches that are generated *a priori*. The time required to pick batch $i \in I$ is denoted by t_i. It includes the time for an order picker to travel between product locations of items in the batch, search for the items, place them in roll cages, and transport the filled roll cages to the staging lanes. We assume that t_i is independent of the order picker and its value is deterministic since the picking route is determined by the storage locations of the items in the batch and the routing strategy

Table 3.1: Comparison of OPSP to the shift scheduling and order picker planning literature

Type of problem (representative paper)	Coverage: Task	Coverage: Workload	Task timing: Fixed	Task timing: Flexible	Task timing: Window	Break timing	Minimum compensated duration
Parcel sorting center scheduling (Bard et al., 2007)		✓				✓	
Order assignment sequencing (Scholz et al., 2017)	✓			✓	✓		
Fixed job scheduling (Fischetti et al., 1989)	✓		✓				
Interval scheduling (Kroon et al., 1995)	✓		✓				
Shift minimization (Krishnamoorthy et al., 2012)	✓						
Nurse rostering and task scheduling (Beliën & Demeulemeester, 2008)	✓		✓				
Home care scheduling (Rasmussen et al., 2012)	✓			✓	✓		
Field workforce scheduling (Alsheddy & Tsang, 2011)	✓			✓	✓		
Technician task scheduling (Cordeau et al., 2010)	✓			✓			
Call center scheduling (Bhandari et al., 2008)		✓				✓	
Hotel staff scheduling (Thompson & Pullman, 2007)		✓				✓	
Navy personnel planning (Holder, 2005)	✓		✓			✓	
Tour scheduling (Brusco & Jacobs, 2000)	✓		✓			✓	
Multi-activity shift scheduling (Dahmen et al., 2018)	✓	✓				✓	
Truck driver scheduling (Goel & Irnich, 2017)	✓			✓	✓	✓	
Order picker scheduling problem	✓			✓	✓	✓	✓

of the warehouse. The company in our case study uses norm times that are set to pick a certain batch.

Each batch $i \in I$ has a corresponding delivery due time window $[r_i, d_i]$. All items in the batch have to be delivered to the designated staging area(s) within this time window. The values of r_i and d_i are determined based on the outbound truck departure schedule and the capacity of the staging lanes. The value of r_i usually corresponds to the departure time of the previous vehicle that departed from the same staging lane as where the vehicle for batch i is departing from, and d_i is the latest time batch i can be delivered at the staging lane for the vehicle to depart on time (i.e., without violating the delivery due time at the customer).

Let P represent the set of the flexible order pickers that can be employed by the warehouse, where $|P| = p_{max}$. Flexible workers are scheduled to work when needed, and as such, they are assigned one of a variety of possible shift lengths with different start times on any day. They are only compensated for the amount of time they spend at the warehouse. Although there is often no restriction on the minimum shift length for a worker, warehouses favor providing a minimum compensation if an employee is scheduled to work. This improves the working relation between the flexible order pickers and the warehouse to increase employee retention. The time corresponding to the minimum compensation duration is denoted by T_{min}. The maximum amount of time an employee can work per day is restricted by law and gives an upper limit on the shift length, which we denote by T_{max}.

There are also labor rules and union agreements on breaks for human order pickers. The amount of time an employee can work without a break is denoted by T_{break}. If an employee works for a duration that exceeds T_{break} time units she must be given an uninterrupted break of at least l_b time units. An employee can be entitled to more than one break in the same shift depending on the values of T_{break} and T_{max}. The length of the planning horizon is T_{day} time units. Formulations for alternative types of breaks are presented in Section 3.8.5. We assume that order picking is scheduled non-preemptively and breaks cannot interrupt this. Interrupting a pick tour and leaving picking equipment in the storage area creates congestion as well as safety and security hazards. Limited parking space for order picking equipment in the break areas and issues of theft or responsibility of already picked items may also prevent preemptive batch scheduling. In case breaks can preempt the order picking of a batch, we propose an updated solution framework and perform a numerical comparison in Section 3.8.12.

Even though flexible employees can potentially start and end their shifts at any time, many shift start and end times are an administrative and operational burden, and labor union agreements can prohibit this as well (Brusco & Jacobs, 1998). Furthermore, employees are paid in integral multiples of a certain duration (even if they completed the last task of their shift before the end of a certain time period). Therefore, we divide the planning horizon into W time periods of equal length, where each period consists of l time units. The set of admissible time periods to start or end a shift is denoted by S and E, respectively. Note that the discretization of the time horizon is only used for the start and end times of shifts. The actual tasks that need to be executed can still start and end at any point in time during the shift (i.e., they do not have to coincide with the time periods) and the same holds for breaks.

We make the assumption that all order picking operations associated with the batches in the planning horizon are performed within the same planning horizon, and we assume that all shifts of the order pickers start and end in the same planning horizon that they are scheduled for (i.e., there is no overlap between either order picking tasks of a batch or shifts of order pickers in different planning horizons).

Furthermore, we define a *task* as an activity that needs to be scheduled; either picking orders of a batch or taking a break. Arranging tasks in a sequence creates a shift, and each task in the sequence has a position (first, second and so on). This is illustrated with a Gantt chart in Figure 3.2. Employee 1 picks the items in batch 4 and 5 successively, then takes a break, and finally picks items in batch 10 before ending her shift. Note that the order picker completes four tasks but not necessarily consecutively (i.e., there can be an interruption or gap between two successive tasks), which is the case for Employee 2. Each order picker can perform at most $\bar{k}$ tasks in a shift. A summary of all parameters is provided in Table 3.2.

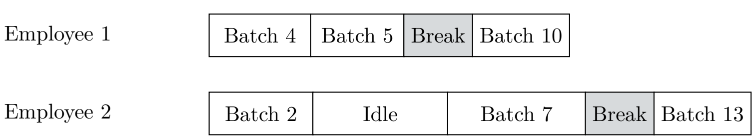


Figure 3.2: A Gantt chart to illustrate the concepts of tasks, shifts and task positions in a sequence

The following decision variables are used in our model formulation:

Table 3.2: Overview of the parameters for the order picker scheduling problem (OPSP)

notation	description
P	set of order pickers that can be scheduled, $\{1, \ldots, p_{max}\}$
I	set of batches that need to be picked
K	set of positions in which an order picker can perform a task, $\{1, \ldots, \bar{k}\}$
t_i	duration to pick and deliver the items of batch $i \in I$
r_i	earliest due time of batch $i \in I$
d_i	latest due time of batch $i \in I$, where $d_i \geq \max\{r_i, t_i\}$
T_{min}	minimum time an order picker needs to be compensated if scheduled
T_{max}	maximum shift length
T_{break}	maximum time duration an order picker can work consecutively without a break
T_{day}	length of the planning horizon
l	length of a time period
J	set of time periods, $\{1, \ldots, W\}$
S	set of time periods where a shift can start at the beginning of that period, $S \subseteq J$
E	set of time periods where a shift can end at the end of that period, $E \subseteq J$
l_b	duration of a break
M	a very large number

x_{ikp} is 1 if batch $i \in I$ is scheduled to be picked at the k^{th} position in the shift for order picker $p \in P$, where $k \in K$, else 0

y_{kp} is 1 if a break is scheduled at the k^{th} position in the shift for order picker $p \in P$, where $k \in K$, else 0

s_{jp} is 1 if order picker $p \in P$ starts the shift at the beginning of period $j \in S$, else 0

e_{jp} is 1 if order picker $p \in P$ ends the shift at the end of period $j \in E$, else 0

c_{kp} completion time of the task scheduled at the k^{th} position in the shift for order picker $p \in P$, where $k \in K$

m_p amount of time for which order picker $p \in P$ is compensated

The order picker scheduling problem (OPSP) is formulated as a MILP model as follows:

OPSP:

$$\min \sum_{p \in P} m_p \tag{3.1}$$

subject to

$$\sum_{i \in I} x_{ikp} + y_{kp} \leq 1 \qquad \forall k \in K, p \in P \tag{3.2}$$

$$\sum_{k \in K} \sum_{p \in P} x_{ikp} = 1 \qquad \forall i \in I \tag{3.3}$$

$$\sum_{i \in I} x_{i1p} + y_{1p} \leq \sum_{j \in S} s_{jp} \qquad \forall p \in P \tag{3.4}$$

$$\sum_{j \in S} s_{jp} = \sum_{j \in E} e_{jp} \qquad \forall p \in P \tag{3.5}$$

$$c_{1p} \geq \Big(\sum_{j \in S} (j-1) s_{jp} \Big) l + \sum_{i \in I} t_i x_{i1p} + l_b y_{1p} \qquad \forall p \in P \tag{3.6}$$

$$c_{kp} - c_{k-1,p} \geq \sum_{i \in I} t_i x_{ikp} + l_b y_{kp} \qquad \forall k \in K \setminus \{1\}, p \in P \tag{3.7}$$

$$\sum_{j \in E} (j e_{jp}) l \geq c_{\bar{k}p} \qquad \forall p \in P \tag{3.8}$$

$$c_{kp} + M(1 - x_{ikp}) \geq r_i \qquad \forall i \in I, k \in K, p \in P \tag{3.9}$$

$$c_{kp} - M(1 - x_{ikp}) \leq d_i \qquad \forall i \in I, k \in K, p \in P \tag{3.10}$$

$$c_{kp} - \Big(c_{hp} - \sum_{i \in I} t_i x_{ihp} \Big) - T_{break} \leq M \Big(\sum_{k'=h+1}^{k} y_{k'p} \Big) \qquad \forall h, k \in K, h < k, p \in P \tag{3.11}$$

$$\sum_{i \in I} x_{ik-1,p} + y_{k-1,p} \geq \sum_{i \in I} x_{ikp} + y_{kp} \qquad \forall k \in K \setminus \{1\}, p \in P \tag{3.12}$$

$$T_{min} \sum_{j \in S} s_{jp} \leq m_p \qquad \forall p \in P \tag{3.13}$$

$$\Big(\sum_{j \in E} j e_{jp} - \sum_{j \in S} (j-1) s_{jp} \Big) l \leq m_p \qquad \forall p \in P \tag{3.14}$$

$$c_{kp} \geq 0 \qquad \forall p \in P, k \in K \tag{3.15}$$

$$x_{ikp} \in \{0, 1\} \qquad \forall i \in I, k \in K, p \in P \tag{3.16}$$

$$y_{kp} \in \{0, 1\} \qquad \forall k \in K, p \in P \tag{3.17}$$

$$s_{jp} \in \{0, 1\} \qquad \forall j \in S, p \in P \tag{3.18}$$

$$e_{jp} \in \{0, 1\} \qquad \forall j \in E, p \in P \tag{3.19}$$

$$0 \leq m_p \leq T_{max} \qquad \forall p \in P \tag{3.20}$$

The objective function (3.1) expresses the minimization of the total labor cost over all order pickers who are scheduled to pick the items that need to be delivered during the planning horizon. Constraints (3.2) ensure that an order picker can perform at most one task in the k-th position of her shift. Constraints (3.3) ensure that each batch is picked exactly once. An order picker can only perform the first task in a shift if she is scheduled to start a shift according to constraints (3.4). Constraints (3.5) ensure that every order picker who starts a shift also has to end a shift (and vice versa).

Constraints (3.6) and (3.7) determine that the task in the k-th position of the order picker's shift can only be labeled as completed after it is executed. Constraints (3.8) ensure that the order picker can only finish her shift after completing the last assigned task. Constraints (3.9) and (3.10) require that batches are completed within their due time windows. Note that an order picker can have fewer than $\bar{k}$ tasks assigned to her shift. In that case, for all positions in a shift without an actual task assigned (i.e., for all k where $\sum_i x_{ikp} + y_{kp} = 0$), the completion times c_{kp} are set equal to the completion time of the last assigned task (i.e., $c_{kp} = c_{k-1,p}$).

Constraints (3.11) require that an order picker cannot work successively for a duration more than T_{break} time units without a break. The constraint specifies that the time between the start of the task at position h of the shift and the end of the task at position k, where $k > h$, has to be less than or equal to T_{break} in case no break is scheduled between these two tasks. Constraints (3.12) specify that a task can only be assigned to a position if there is also a task assigned to the previous position.

Constraints (3.13) ensure that an order picker is compensated for at least T_{min} time units if she is scheduled to work. Constraints (3.14) ensure that an order picker is compensated for at least the amount of time the order picker is scheduled to work (i.e., from the start time of the shift to the end time of the shift). Constraints (3.15) to (3.20) define the domain and range of the decision variables.

Proposition 3.1. *Generating a feasible solution for the OPSP is NP-hard in the strong sense.*

The proof for this proposition is presented in Section 3.8.2.

3.4 Branch and Price Algorithm for OPSP

This section outlines an exact procedure to solve the OPSP using a branch-and-price framework. In this solution approach, the linear relaxation in each node of a branch-and-bound tree is solved with column generation (Barnhart et al., 1998; Vanderbeck, 2000). A branch-and-price solution approach remains a successful and popular solution strategy for generating optimal solutions for problems in a variety of fields ranging from transport planning (Bertsimas et al., 2019), routing (Dellaert et al., 2018) to personnel scheduling (van den Bergh et al., 2013). We also develop a branch-and-price algorithm for the order picker scheduling problem. We first present the *reduced master problem* (RMP). The *pricing problem* to verify the optimality of an LP solution is presented in Section 3.4.2. The branching that occurs when the LP solution does not satisfy the integrality conditions is discussed in Section 3.4.3.

The proposed framework for the branch-and-price algorithm has similarities to the one used by Goel & Irnich (2017). However, because we use the schedule duration in the objective function (which includes employee waiting times between the performance of two tasks) and include the minimum compensated duration as constraints, the details of the building blocks for the branch-and-price algorithm are different from the algorithm in Goel & Irnich (2017). Specifically, the augmented graph for the pricing problem requires information on shift starting and ending times. The definitions of resources and resource extension functions that are used to solve the pricing problem also differ and are more comparable to those used for the minimum tour duration problem (MTDP) (Tilk & Irnich, 2017) rather than the truck driver scheduling problem. Furthermore, because of the constraints regarding the minimum compensated duration and flexible breaks, the problem suffers from significant issues of symmetry. Therefore, we develop a tailored acceleration strategy to address these issues (see the end of Section 3.4.2).

3.4.1 Reduced master problem

To present the reduced master problem for the OPSP in a column generation format, we first introduce the concept of a *column* as a *feasible shift schedule* that is specified by the start and end time as well as the assignment and sequence of tasks (both order picking and breaks) to be performed by a single order picker while respecting the due time windows of order picking tasks, maximum shift length T_{max} and maximum time between breaks T_{break}. Let Ω denote a set of all feasible schedules, where Ω' is a subset of Ω (i.e., $\Omega' \subseteq \Omega$). The cost for an individual schedule $q \in \Omega'$ is given by m_q. The parameter α_{iq} is set to 1 if batch i is processed (or picked) in schedule q, and zero otherwise. The decision variable θ_q represents the number of schedules of type q to be selected in the solution. The reduced master problem (RMP) can be formulated as a set covering problem:

RMP:

$$\min \sum_{q \in \Omega'} m_q \theta_q \tag{3.21}$$

subject to

$$\sum_{q \in \Omega'} \alpha_{iq} \theta_q \geq 1 \qquad \forall i \in I \tag{3.22}$$

$$\sum_{q \in \Omega'} \theta_q \leq p_{max} \tag{3.23}$$

$$\theta_q \geq 0 \qquad \forall q \in \Omega' \tag{3.24}$$

The objective in the RMP is the same as in the OPSP. Constraints (3.22) ensure that all batches are processed (or covered) with the selected schedules. Constraints (3.23) do not select more than p_{max} schedules to be performed by order pickers. The constraints (3.2) and (3.4) to (3.20) of the OPSP are included in the pricing problem where columns are generated that result in feasible schedules.

3.4.2 Pricing problem

The pricing problem for the OPSP can be formulated as an Elementary Shortest Path Problem with Resource Constraints (ESPPRC) (Feillet et al., 2004). This is a variation of the Shortest Path Problem with Resource Constraints (SPPRC) where cycles are not allowed, i.e., a node cannot be visited more than once. The SPPRC can be solved with pseudo-polynomial algorithms (Irnich & Desaulniers, 2005), whereas the ESPPRC is NP-hard in the strong sense (Dror, 1994). Nevertheless, ESPPRC is known to generate a superior lower bound compared to SPPRC when used as pricing problem (Contardo et al., 2015). A technique to solve the ESPPRC is a labeling algorithm based on dynamic programming (Feillet et al., 2004). This approach uses the concepts of resources in a graph and resource extension functions. A *resource* is an arbitrary one-dimensional piece of information that can be determined or measured at the vertices of a directed walk in a graph (e.g., cost, time, load). In this paper, time is the main resource. *Labels* are used to store the information on the resource values for partial paths. Labels reside at vertices and they are propagated via *resource extension functions* when they are extended along an arc. To keep the number of labels as small as possible, we define *dominance rules* to identify labels that need not be extended. We first introduce the graph structure, labels, resource extension functions and dominance rules.

Graph representation

Consider a subgraph $G = (V, A)$, where V is the set of vertices indicating the set of batches $i \in I$ that have to be picked and the arcs A indicate the subsequent sequence in which the batches are completed. The nodes in the sets S and E indicate the start and end times of a shift, respectively. Furthermore, dummy origin and destination nodes are indicated by $\texttt{o}$ and $\texttt{d}$, respectively. The complete set of all vertices is $V' := \{\texttt{o}\} \cup S \cup V \cup E \cup \{\texttt{d}\}$.

We introduce arcs between the dummy origin node $\texttt{o}$ and the shift start time nodes in S, between each vertex in S and V, between each vertex in V and E as well as between the shift end time nodes and the dummy destination node $\texttt{d}$. See Figure 3.3 for an example. The travel time for each arc is set to zero. The service time t_i at each node $i \in V$ equals the processing time of batch i, whereas the service time at the remaining vertices $V' \setminus V$ is zero.

The time windows for the origin and destination nodes are $[r_i\,;\,d_i] = [0\,;\,T_{day}]$ for $i \in \{\texttt{o}, \texttt{d}\}$ such that these nodes can be visited at any time during the time horizon. For the shift start time nodes $i \in S$, the value of $r_i = d_i$ equals the possible shift start times such that these nodes are visited at these specific times. Similarly, for the shift end time nodes $i \in E$, the value of $r_i = d_i$ equals the possible shift end times. A feasible schedule for an order picker comprises of a tour from node $\texttt{o}$ to node $\texttt{d}$ respecting the due time windows $[r_i\,;\,d_i]$ for $i \in V'$, maximum shift length T_{max} and the time until breaks T_{break}. As an illustrative example, Figure 3.3 represents a graph where there are three possible shift start and end times. The dashed arrow indicates a feasible schedule that starts at $s1$, then executes batch $i3$ and ends at $e1$.

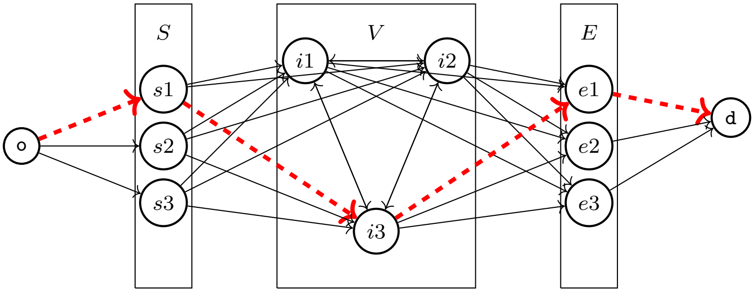


Figure 3.3: A representation of a graph structure for the pricing problem of the OPSP with 3 shift start and end times and 3 batches

Labels

A *partial schedule* corresponds to a partial path in the graph G. A partial schedule h where vertex i is visited as last node is defined by label $L_h^i = (i, c_h^i, T_i, (V_h^1, \ldots, V_h^{|V|}))$, where

- i is the last vertex that has been visited in the partial schedule
- c_h^i is the reduced cost of the partial schedule (i.e., the actual cost minus the dual values of the nodes visited, see below for more details)
- $T_i = (T_i^{time}, T_i^{dur}, T_i^{start}, T_i^{work}, T_i^{brk})$ indicates the resource vector, where the resource variables are
  - T_i^{time} is the time when the batch at node i is completed
  - T_i^{dur} is the minimum duration required to service all the nodes in the partial schedule including the waiting times if necessary to respect the due time windows
  - T_i^{start} is the latest possible start time of the shift to feasibly visit all of the vertices in the partial schedule while respecting the due time windows
  - T_i^{work} is the amount of time since the end of the last break
  - T_i^{brk} is the latest time to start picking the first batch after the previous break to ensure feasibility of the schedule
- V_h^v is 1 if node $v \in V$ is visited in the partial schedule or if it is infeasible to visit (due to the due time windows or maximum shift length), 0 otherwise

To guarantee elementarity of a (partial) path, it is sufficient to add the extra resources V_h^v for each node $v \in V$ indicating whether or not the node has been visited on the path. When this resource has the value one, it prohibits the path to re-enter previously visited nodes. Feillet et al. (2004) enhance this idea by observing that some nodes are not reachable due to the resource constraints, which they indicate by setting the resources V_h^v to one for these nodes without the path having to visit them. They use this to speed up the dominance check, which is explained later in this section.

The resource windows of resource vector T_i are given by $T_i^{time} \in [r_i\,;\,d_i]$, $T_i^{dur} \in [0\,;\,T_{max}]$, $T_i^{start} \in (-\infty\,;\,T_{day}]$, $T_i^{work} \in [0\,;\,T_{break}]$, and $T_i^{brk} \in (-\infty\,;\,\infty)$. A path is called resource-feasible if there exist resource vectors for each node in the path that satisfy their resource windows. Therefore, a feasible schedule is a resource-feasible path that starts in $\mathtt{o}$ and ends in $\mathtt{d}$. Furthermore, let $\mathscr{L}_i$ denote the set of all labels corresponding to partial schedules where node $i \in V'$ is the last visited node.

The initialization of a label is done at shift start nodes $i \in S$ as $L_h^i = (i, T_{min} - \psi, T_i, (V_h^1, \dots, V_h^{|V|}))$, where ψ is the dual variable associated with Constraint (3.23) of the RMP, $T_i^{time} = d_i$, $T_i^{dur} = 0$, $T_i^{start} = d_i$, $T_i^{work} = 0$ and $T_i^{brk} = \infty$, and $V_h^j = 0$ for all nodes $j \in V$.

Resource extension functions

A resource extension function (REF) is used to extend a label (or partial schedule) with an additional vertex such that all constraints related to the scheduling problem are still satisfied. There are two options to extend label L_h^i at vertex i to vertex j when $V_h^j = 0$. The first extension executes the batch in node j directly after finishing the batch in node i without a break. The second extension starts with a break before the execution of the batch in node j. Consequently, we consider the two resource extension functions $f(\cdot)$ and $g(\cdot)$, respectively.

The resource extension functions $f(T_i, j)$ for the extension of label L_h^i to node j without a break define the new resource variables of resource vector T_j as follows

$$T_j^{time} = f^{time}(T_i, j) := \max\{T_i^{time} + t_j, r_j\} \tag{3.25}$$

$$T_j^{dur} = f^{dur}(T_i, j) := \max\{T_i^{dur} + t_j, r_j - T_i^{start}\} \tag{3.26}$$

$$T_j^{work} = f^{work}(T_i, j) := \max\{T_i^{work} + t_j, r_j - T_i^{brk}\} \tag{3.27}$$

$$T_j^{brk} = f^{brk}(T_i, j) := \min\{d_j - (T_i^{work} + t_j), T_i^{brk}\} \tag{3.28}$$

Similarly, when label L_h^i is extended with a break before the order picking task is completed as indicated by node j, the resource extension functions $g(T_i, j)$ define the resource vector T_j as

$$T_j^{time} = \quad g^{time}(T_i, j) := \quad \max\{T_i^{time} + t_j + l_b, r_j\} \tag{3.29}$$

$$T_j^{dur} = \quad g^{dur}(T_i, j) := \quad \max\{T_i^{dur} + t_j + l_b, r_j - T_i^{start}\} \tag{3.30}$$

$$T_j^{work} = \quad g^{work}(T_i, j) := \quad t_j \tag{3.31}$$

$$T_j^{brk} = \quad g^{brk}(T_i, j) := \quad \min\{d_j - t_j, \infty\} \tag{3.32}$$

Note that the resource variable T_i^{start} is never updated after it is set in the shift start node. The REF for T_j^{time} is a classic REF from the routing literature (Irnich, 2008a). The REFs for T_j^{dur} and T_i^{start} bear resemblance to REFs from the MTDP (Tilk & Irnich, 2017). The REFs for T_j^{work} and T_j^{brk} are new and specifically designed to determine the amount of time elapsed since the last break. Desaulniers & Villeneuve (2000) use similar extension functions to estimate the cost of waiting at nodes for the shortest path problem with time windows and linear waiting costs.

The reduced cost for the partial schedule when label L_h^i is extended to node j, is given by $c_h^j := \max\{T_{min}, T_j^{dur}\} - \pi_j - \psi - \sum_{\hat{j} \in \hat{B}_h} \pi_{\hat{j}}$, where π_j is the dual value of constraints (3.22) for vertex j, ψ is the dual value associated with constraint (3.23), and $\sum_{\hat{j} \in \hat{B}_h} \pi_{\hat{j}}$ indicates the accumulated dual values associated with constraints (3.22) for the set of batches previously added to the partial schedule represented by the set $\hat{B}_h$. Note that the payment to pickers for the entire period (even if they work only for a fraction of the period) is accounted for by the use of shift end nodes which restrict the visit to the shift end nodes at the end of a period.

The resource V_h^j is set to one to prevent that vertex j is visited again. Furthermore, $V_h^{j'}$ is also set to one for any node $j' \in V'$ that cannot be visited anymore when node j is added to the partial path because of the resource constraints. The new label is then given by $L_h^j := (j, c_h^j, T_j, V_h)$, which is only feasible if the resource variables of the resource vector T_j fall within the associated resource windows.

Dominance

A dominance principle can be used to accelerate the solution technique by eliminating unnecessary labels. To define dominance in our pricing problem, we note that the REFs are either non-decreasing or non-increasing, such that an element-wise comparison can be made to determine dominance (Irnich & Desaulniers, 2005). A label L_h^i dominates a label $L_{h'}^i$ if both labels reside at the same vertex $i \in V'$ and if, for each feasible extension of $L_{h'}^i$ to $L_{h'}^j$, there exists a feasible extension of L_h^i to L_h^j where the value of each resource with a non-decreasing (or non-increasing) REF is less than (or larger than) or equal to the value of the resource in the extension of $L_{h'}^i$, i.e., $c_h^i \leq c_{h'}^i$; $T_{i,h}^{time} \leq T_{i,h'}^{time}$; $T_{i,h}^{dur} \leq T_{i,h'}^{dur}$; $T_{i,h}^{start} \geq T_{i,h'}^{start}$; $T_{i,h}^{work} \leq T_{i,h'}^{work}$; $T_{i,h}^{break} \geq T_{i,h'}^{break}$; $V_h^v \leq V_{h'}^v \, \forall \, v \in V$. Consequently, the partial schedule corresponding to label $L_{h'}^i$ cannot be part of the optimal solution. Note that the differentiation of time resources T_i for label h and h' is done for comparison required by dominance. For ease of notation, we do not use the differentiation of time resources for specific labels in the remainder of the paper.

Labeling algorithm

The pricing problem is solved by embedding the resource definitions, resource extension functions and dominance rules in the label correction algorithm by Feillet et al. (2004). The pseudocode for the labeling algorithm is presented in Section 3.8.6 .

Acceleration strategies

Acceleration strategies are commonly used to speed up branch-and-price algorithms and are key to successfully solving sizable problems (Kallehauge et al., 2005). We propose three acceleration strategies for the pricing problem.

Initial columns: It is known that column generation with good initial upper bounds accelerates the convergence of the linear relaxation at the root node (Desaulniers et al., 2002). Therefore, we first generate initial primal solutions with the savings algorithm outlined in Section 3.5.1. This algorithm aims to rapidly find a feasible solution. If the savings algorithm is not able to generate a feasible solution with at most p_{max} order pickers, the initial columns for column generation are initialized with an additional artificial column that covers all batches of the problem and has an arbitrarily high cost to ensure that this artificial column will not be part of the optimal solution.

Limited extension: To exploit the time windows and processing time information between order picking tasks to reduce the use of REFs, we first present the following proposition. The proof of this proposition is presented in Section 3.8.7.

Proposition 3.2. *If there is an optimal schedule with two batches i and j such that $r_i \geq r_j$, $d_i \geq d_j$, $t_i < t_j$ and i precedes j in the same order picker's schedule without a break in between the execution of the two batches, the execution order can be reversed with the same objective function value.*

For any partial schedule h ending with node i (i.e., presented by label L^i_h), if there is a node j for which $V^j_h = 0$ and the conditions in Proposition 3.2 satisfy, we only have to consider the extension with a break between the execution of the batches from node i and j. Consequently, we limit the extension of resources in the arc (i, j) with the resource extension function $g(\cdot)$ only. This particular strategy allows us to have fewer extensions and maintain a smaller set of labels while solving the pricing problem.

Limited discrepancy search: Desaulniers et al. (2008) and Spliet et al. (2018) show that the branch-and-price algorithm can be solved more efficiently when the pricing problem is solved with heuristics until no negative reduced costs are found (such that no new columns are added to the RMP). Along the same principle, as proposed by Feillet et al. (2007) and Goel & Irnich (2017), we use limited discrepancy search (LDS) to heuristically accelerate the generation of columns with a negative reduced cost.

LDS speeds up the pricing problem by maintaining a limited set of labels and heuristically removing so-called unpromising labels from the problem. In our pricing problem, labels with batches that require large waiting times and numerous breaks are considered unpromising labels. The waiting time between two nodes i and j is measured as the time window distance $TW_{distance}(i, j) := \max\{0, r_j - d_i\}$. The outgoing arcs from each node i are partitioned into two sets called *good arcs* and *bad arcs* based on $TW_{distance}$. An additional resource (denoted as l^{bad}) is included to the label that is increased by one if a label traverses through an arc from the set of bad arcs or if it is extended with a break. Only labels that have $l^{bad} \leq \Lambda$ are extended, where the threshold Λ is called the *discrepancy limit.* If the LDS is unable to find any columns with a negative reduced cost, the value of Λ increased by one and the LDS is repeated. When the discrepancy limit reaches an upper bound, the LDS terminates and the ESPPRC is solved with the labeling algorithm. Additionally, an iteration of LDS is terminated if 100 columns with negative reduced cost are generated. Note that the use of LDS does not impact the optimality of the branch-and-price technique since the last pricing problem at every node of the branch-and-bound tree is solved exactly with ESPPRC.

3.4.3 Branching

If the pricing problem cannot find columns with a negative reduced cost and the LP solution to the RMP is not integral, a node of the branch-and-bound tree is selected for branching. Branching is done on flow variables using the best-lower-bound-first strategy (Desaulniers, 2010).

A good upper-bound solution improves the efficiency of the branch-and-price technique by reducing the number of branch nodes in the search tree (Danna & Le Pape, 2005). In our solution procedure, before branching from the root node, we solve the MIP of the RMP where we only consider the columns that are generated at the root node. The solution to the MIP provides the upper bound before branching. If the root node is not solved within the time limit, the MIP of the RMP is solved with the available columns to derive the best known upper-bound solution for benchmarking purposes.

3.5 Metaheuristic for OPSP

Given the size of real-world instances of the OPSP and the computational complexity of the problem, even the branch-and-price technique developed in the previous section is not likely to be a viable solution approach in real-life applications. In this section, we present an efficient metaheuristic that adapts the classic savings principle by Clarke & Wright (1964) to generate an initial feasible solution and that solution is improved by a large neighborhood search algorithm (LNS) with simulated annealing (Pisinger & Ropke, 2010).

3.5.1 Savings algorithm

The savings algorithm iteratively combines two schedules into one schedule based on the savings principle (Clarke & Wright, 1964). The procedure begins by relaxing the maximum number of order pickers constraint and creating schedules that each consist of one batch to be picked. Then, it iteratively determines the saving in terms of the labor cost that is generated when two schedules are combined into one schedule (if possible). This saving is easy to calculate. Consider that schedule h' and h'' are combined in a feasible schedule h with the corresponding compensation $m_{h'}$, $m_{h''}$ and m_h, respectively, then the savings is $(m_{h'} + m_{h''}) - m_h$. Combining batches in two schedules into one schedule has the potential to overcome inefficiencies of individual schedules when these schedules have waiting times (or breaks) between tasks or the shift length is shorter than T_{min} time units.

To verify whether two (randomly) selected schedules can be combined into one schedule, we try to solve a simplified (or reduced) version of our OPSP, which is formulated as a MILP model in Section 3.8.8. Since the reduced problem finds the optimal schedule for only one order picker (or one shift) with a small number of order picking tasks, the MILP can be solved exactly in a reasonable amount of computation time. Even though the computation time of the MILP for the reduced problem is short, a set of infeasibility checks can be performed first as pre-processing step to easily verify whether the order picking tasks cannot be combined in a feasible schedule. See Section 3.8.8 for the infeasibility checks. If these checks do not rule out that a feasible schedule can be found, the reduced OPSP with one order picker is solved. If no feasible solution is found, it is concluded that the two schedules cannot be combined. Otherwise, the solution of the MILP model provides the combined schedule with the largest savings (i.e., it finds the optimal sequencing of the order picking batches).

In the classical savings algorithm by Clarke & Wright (1964), the savings of combining any given two schedules are calculated first before combining solutions in a given iteration of the algorithm. However, in this paper, if any two randomly selected schedules can be combined in a feasible schedule and result in a savings of at least T_{min} time units, the combined schedule is accepted immediately and the two individual schedules will not be considered for other savings in the same iteration of the savings algorithm. If no two schedules exist that can be combined in a feasible schedule that also results in sufficient savings of at least T_{min}, all possible combinations are first calculated and then the schedules are combined such that the maximum savings is achieved. The procedure continues until no savings can be realized while combining schedules. When no further savings can be realized and the number of schedules in the solution is less than p_{max}, a feasible solution is found that satisfies all constraints of the OPSP and the algorithm terminates. If no feasible solution is found, the savings algorithm enters the second phase, in which the batches of any pair of schedules are chosen to be combined in a new schedule that results in the largest savings (which can be the least negative savings or additional cost) until the number of schedules equals p_{max}.

3.5.2 Large neighborhood search for improved solutions

After a feasible solution for the OPSP is generated by the savings algorithm, this solution is improved with a large neighborhood search (LNS) procedure. Let us denote the feasible solution at the beginning of an iteration by π, where the corresponding cost (or objective function value) is $z(\pi) := \sum_{p \in P} m_p$. This solution is destroyed and then repaired in every iteration, which results in a new feasible solution π' with cost $z(\pi')$. Furthermore, let the best found solution so far be denoted by π^*. The decision whether π' becomes the starting solution in the next iteration is based on a simulated annealing principle: if $z(\pi') < z(\pi^*)$ then $\pi^* := \pi'$ and $\pi := \pi'$, otherwise π' is accepted as new solution π with probability $e^{-(z(\pi')-z(\pi^*))/\mathscr{T}}$, where $\mathscr{T}$ is the temperature that is initialized as $\mathscr{T} := -w \cdot z(\pi^*)/\ln(0.5)$ (Ropke & Pisinger, 2006a). The value is updated at the end of every iteration: $\mathscr{T} := \rho\mathscr{T}$, where $0 < \rho < 1$ is the cooling parameter. Consequently, it becomes less likely for worse solutions to be accepted as the starting solution in the next iteration when the number of iterations increases. If the best solution is not improved in n_T iterations, the temperature is reset to the inital value $(-w \cdot z(\pi^*)/ln(0.5))$, such that it is more likely to explore new areas in the feasible solution space.

Destruction and repair The LNS destroys and repairs the solution π in two stages. In the first stage, two order pickers are selected. The first order picker is selected probabilistically with a roulette wheel principle based on a *wastage ratio*. The wastage ratio of an order picker is the fraction of the amount of unproductive duration spent by the order picker compared to the total unproductive hours spent by all of the order pickers in the solution. The wastage ratio for order picker $p \in P$, who is assigned to complete the batches B_p with the cost m_p in solution π, is given by

$$w_p := \frac{m_p - \sum_{i \in B_p} t_i}{\sum_{p' \in P}(m_{p'} - \sum_{i \in B_{p'}} t_i)} \quad \forall p \in P. \tag{3.33}$$

If an order picker has a higher wastage ratio, she is likely to be chosen as the first picker. The second order picker is randomly selected among the remaining order pickers.

In the second stage, the batches previously assigned to the two selected order pickers are reassigned to generate a new (feasible) solution π'. For this purpose, we use one of two operators with equal probability. The *swap operator* exchanges a random subset of batches between the two order pickers. The *insert operator* randomly selects a subset of batches from the first order picker and assigns them to the second order picker. In the literature, swap and insert operators are typically designed to exchange or insert one job, task or trip at a time. The swap and insert operator in this work swaps and inserts multiple batches at a time. This allows us to generate new solutions that would otherwise require multiple operations with the traditional operators. The number of batches to swap or insert from each order picker is uniformly sampled between one and σ (which is a user-set parameter). If the best solution is not improved by n_σ iterations, the value of σ is reduced by 1.

After the batches are reassigned to the order pickers, the sequencing of the batches and scheduling of shifts for the order pickers is determined by solving the same MILP of the reduced problem as in the savings algorithm (see Section 3.8.8). Note that we also verify whether any of the infeasibility conditions is satisfied before solving the reduced problem. Rather than directly solving a MILP, other (more efficient) solution techniques can be proposed to find a solution for the reduced problem. For instance, in a dynamic programming approach a new graph can be created for each reduced problem, in which only the relevant batches assigned to a picker are included and a path starting at the dummy source, `o`, has to visit all batch nodes in the graph before returning to the dummy sink, `d`. At the dummy sink, the solution with the cheapest cost is selected and returned to the metaheuristic for evaluation.

Figure 3.4 illustrates how the two destroy operators work based on a simple example. The initial schedules of the two selected order pickers are represented by X^1 and X^2. With the swap operator, batch 2 and batch 6 are interchanged. The new batches assigned to the order pickers are indicated by $B^{1'}_{Swap}$ and $B^{2'}_{Swap}$, respectively. With the insert operator, batch 2 is unassigned from the first order picker and assigned to the second order picker. The new batches assigned to the order pickers are then indicated by $B^{1'}_{Insert}$ and $B^{2'}_{Insert}$, respectively. After solving the MILP as formulated in Section 3.8.8 for each of the two order pickers individually, we obtain the new schedules $X^{1'}$ and $X^{2'}$, respectively.

The LNS terminates if $z(\pi^*)$ does not exceed the lower bound formulated in Equation (3.43) (see Section 3.8.3), if the number of iterations exceeds a maximum threshold or if the run time exceeds a maximum threshold. Once the LNS terminates and time is available, we pass the LNS solution to the branch-and-price algorithm to improve the solution further by solving the pricing problem for one iteration without solving the ESPPRC exactly.

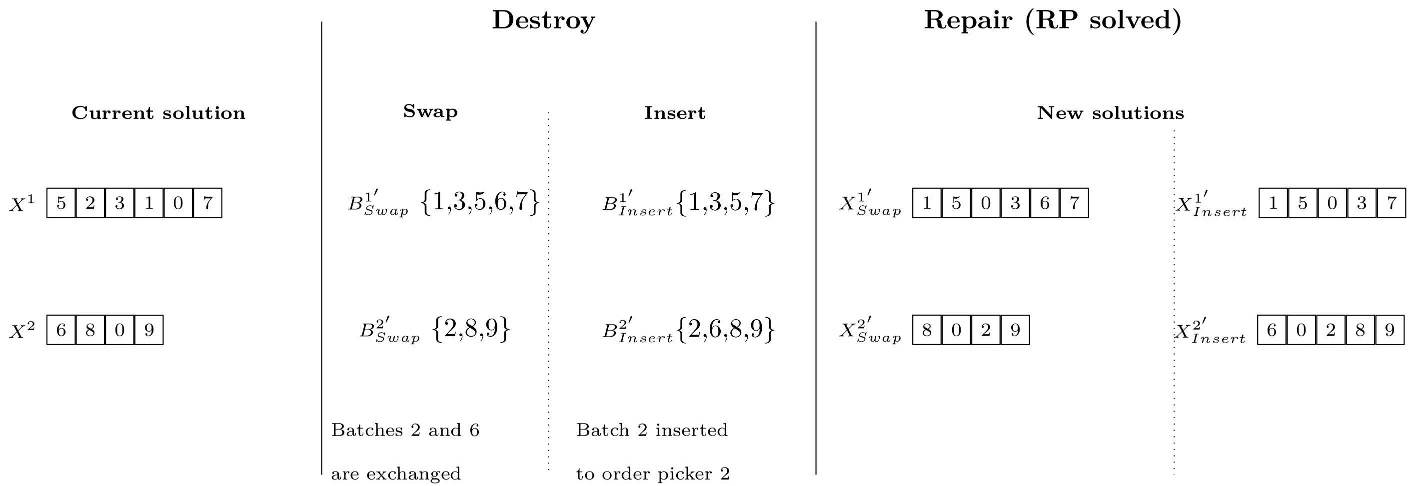


Figure 3.4: Illustration of the destroy operations in an iteration of the LNS algorithm, where 0 in a schedule represents a break

3.6 Results

This section presents a numerical comparison of the branch-and-price algorithm (Section 3.4), savings algorithm (Section 3.5.1) and metaheuristic (Section 3.5.2) to solve the OPSP. Since state-of-the-art commercial solvers such as *Gurobi 9.0.1* (Gurobi Optimization, 2020) are not able to generate an optimal solution for even the smallest instances and the branch-and-price algorithm outperforms *Gurobi* without exception, we do not report the performance of such commercial solvers here. See Section 3.8.9 for a comparison of the performance of commercial solver *Gurobi* and the branch-and-price algorithm. Furthermore, no existing solution procedures from the literature are included as benchmark since the authors are not aware of any other work that makes the same (or even similar) decisions and the objective function (see also Section 3.2).

All solution procedures are implemented in $C{+}{+}$ and run on an i7 3.60GHz machine with 16GB of RAM. For the parameters of the branch-and-price algorithm, the maximum number of *good* arcs from any node is set to 2 and the number of increments for the discrepancy limit in LDS (i.e., Λ) is set to 10. The parameters values for the metaheuristic are guided by the literature, where $\rho := 0.95$ and $n_T := 200$ (Bodnar et al., 2017; Stenger et al., 2013). The initial value of σ is set to 4 and n_σ to 1,250. Furthermore, $w := 0.1$ produced the best results in our numerical experiments, but we cannot guarantee optimality of this parameter value. The stopping criterion for the branch-and-price method is set to 1,800 seconds. For the metaheuristic, it is set to 360 seconds or 5,000 iterations (whichever comes first) to ensure that the method is suitable for practical applications.

3.6.1 Instances in numerical test bed

The instances are generated to mimic the operations of a retail grocery warehouse for which we had detailed data where the target departure times of the outbound trucks determine the staging lane operations as well as their earliest and latest due times. For all instances, we consider a 24-hour time horizon and we use minutes as our time unit. The warehouse operates 24 hours a day, and 7 days a week. However, all order picking tasks and shifts of order pickers are disjoint between different planning horizons as all shifts in a day start and end between 11pm and 11pm the next day.

Due time windows Two different patterns of due time windows are considered in our instances: *waved* and *waveless*. In waved instances, trucks arrive at the staging lanes at the same time and depart from the staging lanes at the same time (i.e., batches to be picked in the same wave have the same due time windows). Alternatively, in the waveless operations, the arrival and departure times of trucks at different staging lanes are not related. The deadline for each truck departure from a staging lane is taken from a uniform distribution in the range of [120,1425] minutes.

To make sure that there is sufficient time for the staging and loading operations of a truck, we push back deadlines (if needed) to guarantee at least 30 minutes between two consecutive departure due times of batches destined for the same dock door (or staging lane). The earliest due time of a batch is set to the latest due time of the previous batches at the same staging lane plus 15 minutes to ensure that loads of different trucks are not mixed up, and previous trucks have finished loading. The earliest due time of the first batch that is due at a staging lane is set to 0. The number of staging lanes in the instances varies from 1 to 8 (see below).

Processing time distributions The processing times of batches are taken either from one of the following uniform distributions: **U**[30,60], **U**[60,90] or **U**[90,120], or from an exponential distribution with the same corresponding average (i.e., 45, 75 or 105 minutes, respectively). The maximum processing time of a batch is restricted to 330 minutes to ensure that employees do not violate the break constraint ($T_{break} = 330$ minutes, see below).

Shift types In accordance with Dutch and European working hours laws, the maximum shift length to employ an order picker (i.e., T_{max}) is 540 minutes (or 9 hours), and the maximum time duration that an employee can work without a break (i.e., T_{break}) is 330 minutes (or 5.5 hours). The length of the break (i.e., l_b) has to be at least 45 consecutive minutes (European Parliament, Council of the European Union, 2003).

We consider six shift structures. In the shift structures SStr1, SStr2 and SStr3, shifts can start every 8 hours and T_{min} equals 8, 6 and 4 hours, respectively. In the shift structures SStr4, SStr5 and SStr6, shifts can start every 4 hours and T_{min} equals 8, 6 and 4 hours, respectively. In all shift structures, a shift can end at the end of any hour after T_{min}. Consequently, shift structure SStr1 is the most restrictive and SStr6 is the most flexible. Table 3.3 summarizes the six shift structures we consider.

Number of batches and staging lanes Each outbound truck requires exactly four order batches to be picked and the number of trucks departing the warehouse in the planning horizon equals either 10, 20 and 40 trucks. This results in instances with 40, 80 or 160 batches to be picked, respectively. The number of staging lanes in an

Table 3.3: Shift structures considered in our numerical experiments

Shift structure	Starting hours(S)	T_{min} (hours)
SStr 1	0, 8, 16	8
SStr 2	0, 8, 16	6
SStr 3	0, 8, 16	4
SStr 4	0, 4, 8, 12, 16, 20	8
SStr 5	0, 4, 8, 12, 16, 20	6
SStr 6	0, 4, 8, 12, 16, 20	4

instance is chosen such that the number of departures per staging lane is fixed at either 5, 10 or 20 trucks. As a result, the number of staging lanes ranges between 1 and 8 lanes. Note that instances with 10 trucks can only have 5 or 10 departures per staging lane. Furthermore, for instances were the number of trucks equals the number of truck departures in a staging lane, there is only one staging lane (grouped under *waved* in Table 3.4). We assume that sufficient order pickers are available to schedule with $p_{max} = 100$.

3.6.2 Algorithmic performance

Table 3.4 presents a summary of the results over all 504 instances in the test bed, whereas the results for the individual instances are presented in Section 3.8.9. For the branch-and-price algorithm, *Root solved* indicates the number of instances for which the column generation was able to solve the linear relaxation within the run time limit of 1,800 seconds. *Optimal solution* indicates the number of instances for which the optimal solution was found within this time limit. For those instances where the branch-and-price algorithm was not able to find the optimal solution, *Optimality gap %* presents the average relative percentage cost difference between the best lower bound found after branching and the best integer solution found after branching. The average time required to solve the root node and the overall branch-and-price algorithm is indicated by CPU^{LP} and CPU^{BP}, respectively. Note that CPU^{BP} also includes the time to generate an initial solution. The average relative performance gap between the solution generated by the savings algorithm and metaheuristic compared to the best branch-and-price solution is indicated by $\%\Delta^S$ and $\%\Delta^{MH}$, respectively[2], where a positive number indicates that the branch-and-price algorithm found a better

[2] $\%\Delta^S = (z(S) - z(BP))/z(BP) \times 100$ and $\%\Delta^{MH} = (z(MH) - z(BP))/z(BP) \times 100$, where $z(BP)$, $z(S)$ and $z(MH)$ denote the objective function value of the best integer solution found by the branch-and-price algorithm, savings algorithm and metaheuristic, respectively.

solution. The average computation time of the savings algorithm and metaheuristic is indicated by CPU^S and CPU^{MH}, respectively.

Table 3.4 shows that the branch-and-price algorithm is capable of solving reasonable size instances. However, the size of the instances adversely affects the performance of the exact approach. For the instances with 40, 80 and 160 batches, the root node can be solved in 100%, 58.9% and 26.3% of the instances, respectively, and the algorithm converges to an optimal solution within the run time for 60.2%, 36.1% and 19.4% of the instances, respectively. For the instances where the branch-and-price algorithm is not able to find an optimal solution, the average optimality gap is only 3.5%. Figure 3.5a shows the average optimality gap of the branch-and-price solutions for instances where we were able to solve the root node but the optimal solutions were not obtained.

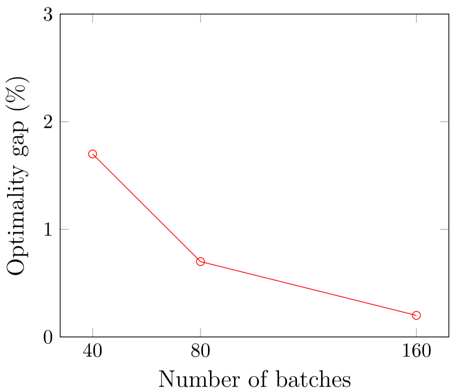
When we compare the number of instances for which the root node (i.e., the linear relaxation) is solved and the number of instances for which an optimal solution is found within the run time limit of 1,800 seconds, we make the following observations: First, waveless instances are more difficult to solve than waved instances. A reason why the branch-and-price algorithm can solve waved instances easier is because the *limited extension* property (see Proposition 3.2) exploits the fact that the batches have non-overlapping due time windows when solving the pricing problem. As a result, the labeling algorithm does not have to explore as many extensions between nodes, and it is capable of solving the pricing problem more efficiently for waved instances. Second, instances with exponentially distributed processing times are more difficult to solve than instances with uniformly distributed processing times. Instances with exponentially distributed processing times have many batches with short processing times. On average, the number of tasks that can be assigned to an order picker is higher with the exponentially distributed processing times. As a result, the labeling algorithm has to consider more potential solutions and labels when solving the pricing problem. Third, instances with more truck departures per staging lane are easier to solve than instances with fewer truck departures. When there are more trucks departing from the same staging lane, the average length of the due time windows is smaller (see Figure 3.11 in Section 3.8.9). As a result, the pricing problem needs to consider fewer extensions from any node as many potential solutions are not feasible. See Section 3.8.9 for a more detailed discussion on these observations.

The savings algorithm is able to quickly generate a feasible solution (on average within 4.3 seconds) for either the branch-and-price algorithm or the metaheuristic.

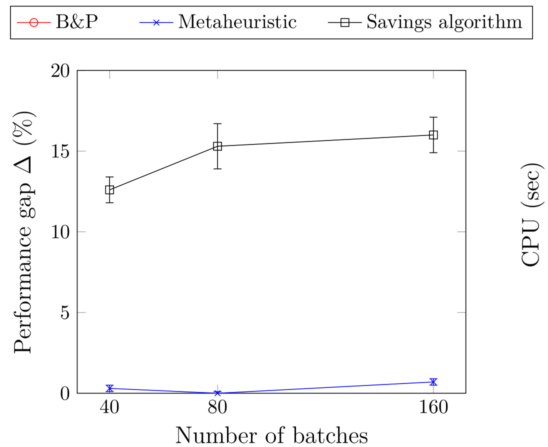
Table 3.4: Summary of results

			Branch-and-Price Algorithm					Savings Algorithm		Meta-heuristic	
Dep. per lane	Instance type	Batches	Number of instances			Average run time					
			Root solved	Optimal solution	Optimality gap %	CPU^{LP} (sec.)	CPU^{BP} (sec.)	$\%\Delta^S$	CPU^S (sec.)	$\%\Delta^{MH}$	CPU^{MH} (sec.)
5	Unif-Waved	40	18/18	11/18	4.00	10.5	809.7	11.6	2.5	0.2	48.2
		80	13/18	9/18	1.7	671.7	1,016.1	12.5	8.5	0.0	234.3
		160	3/18	1/18	0.6	1,601.5	1,758.4	12.1	33.8	0.7	361.4
	Unif-Waveless	40	18/18	10/18	5.0	54.9	839.2	17.9	2.3	0.5	110.3
		80	6/18	3/18	1.1	1,489.8	1,571.5	20.3	7.6	0.2	322.0
		160	1/18	0/18	0.8	1,956.9	1,800.0	17.4	26.2	1.9	362.2
	Exp-Waved	40	18/18	8/18	3.7	73.7	1,125.7	8.3	2.7	0.0	101.0
		80	4/18	2/18	1.5	1,496.3	1,603.9	10.5	9.0	0.0	302.8
		160	0/18	0/18	-	1,788.9	1,800.0	15.0	37.3	0.6	361.5
	Exp-Waveless	40	18/18	13/18	3.3	76.4	843.1	14.0	2.2	0.7	147.1
		80	4/18	0/18	1.4	1,637.2	1,800.0	14.7	7.9	-0.3	348.6
		160	0/18	0/18	-	1,889.1	1,800.0	12.9	28.3	1.1	363.3
10	Unif-Waved	40	18/18	16/18	2.1	1.7	204.5	12.2	1.8	0.6	30.7
		80	18/18	14/18	1.9	142.2	492.3	14.2	6.7	0.1	103.8
		160	14/18	11/18	0.4	507.2	785.1	11.3	26.1	0.2	210.0
	Unif-Waveless	80	12/18	6/18	2.5	759.8	1,278.9	18.8	6.2	-0.1	239.4
		160	6/18	5/18	0.3	1,402.5	1,447.3	17.8	21.9	0.3	336.3
	Exp-Waved	40	18/18	7/18	5.2	22.3	1,102.7	11.5	1.8	0.1	64.9
		80	12/18	8/18	3.2	758.2	1,030.2	15.0	6.5	0.4	205.0
		160	2/18	1/18	0.7	1,654.7	1,702.8	16.4	25.9	0.8	333.4
	Exp-Waveless	80	8/18	4/18	1.3	1,134.5	1,497.7	17.4	6.0	0.3	272.0
		160	1/18	0/18	3.6	1,727.7	1,800.0	20.0	21.3	1.2	364.2
20	Unif-Waved	80	17/18	14/18	1.2	108.1	410.0	13.0	4.9	0.0	66.4
		160	14/18	11/18	0.7	486.1	782.6	12.8	18.9	0.0	175.1
	Unif-Waveless	160	5/18	5/18	-	1,467.0	1,447.9	23.8	17.9	0.6	324.1
	Exp-Waved	80	12/18	5/18	1.7	671.1	1,316.7	16.5	5.0	0.0	215.0
		160	5/18	3/18	0.3	1,534.9	1,527.3	15.8	19.8	0.6	343.1
	Exp-Waveless	160	6/18	5/18	2.8	1,476.7	1,417.4	16.4	18.0	0.4	335.8

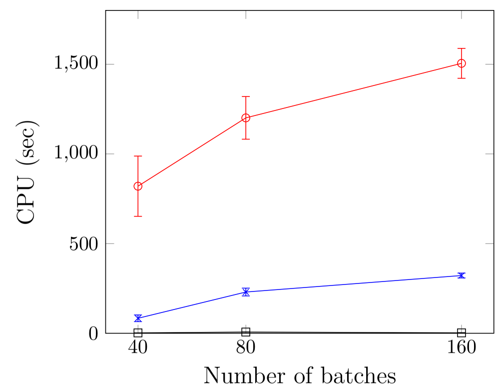
Note: Optimality gap % with "-" indicates that lower bound is not available for instances with non-optimal solutions.



(a) 95% CI of optimality gap



(b) 95% CI of performance deviation compared to B&P



(c) 95% CI of run times

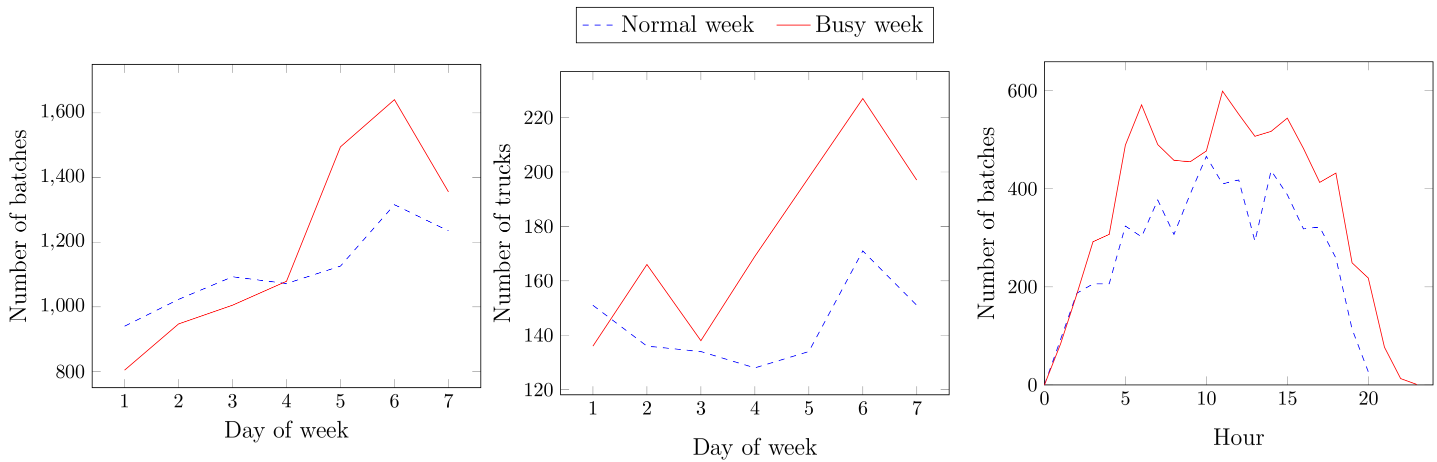
Figure 3.5: Performance comparision between solutions found with the branch-and-price algorithm (B&P), savings algorithm and metaheursitic

However, the quality of these solutions is poor, with an average cost deviation of 15.0% compared to the best solutions found with the branch-and-price algorithm. In contrast, the solutions with the metaheuristic have an average performance gap of less than 0.4% which is found within less than one-fifth of the computational time required for the branch-and-price algorithm. Figure 3.5b and Figure 3.5c present the performance gap of the heuristic procedures and the computational time for each of the three solution approaches, respectively.

3.6.3 Flexible shift structures: A case study

In this subsection, we apply the metaheuristic to the order picker scheduling problem at a warehouse with perishable products of a Dutch grocery retailer. The case study serves two purposes. First, it evaluates the usability of the metaheuristic that we propose to solve industrial instances. Second, the case illustrates some of the ways in which the methodology in this paper can be used to evaluate warehouse operating policies of interest to managers. In particular, we study the impact of the shift structures on the number of order pickers scheduled to perform the order-picking activities.

Description instances The retailer provided operational data regarding the processing times and due time windows of batches for two weeks of their operations. The first week represents a typical week in terms of the number of batches to be picked and shipped from the warehouse. The second week represents the busiest week of the year, which occurs during the Christmas season. There are 6.9% more batches to be picked in the busier week compared to the typical week (see Figure 3.6a, where day 1 is a Sunday). The warehouse has 53 staging lanes, and the number of trucks departing from the warehouse ranges between 128 and 227 trucks per day (see Figure 3.6b). When we consider the number of batches with a due deadline in a particular hour in Figure 3.6c, we identify two peak periods of operations: between hour 5 and hour 7, and between hour 10 and hour 12. In this figure, hour 0 corresponds to 11:00 pm since the warehouse starts its order-picking activities at that hour. The average processing time of a batch is around 41 minutes for both the busy and normal week, and the distribution of these processing times are similar in both weeks (see Figure 3.7a). The distribution of the duration of the due time windows is illustrated in Figure 3.7b. The larger due time windows in the right tail in this figure occur on days with fewer trucks departing from the warehouse (i.e., on day 1).

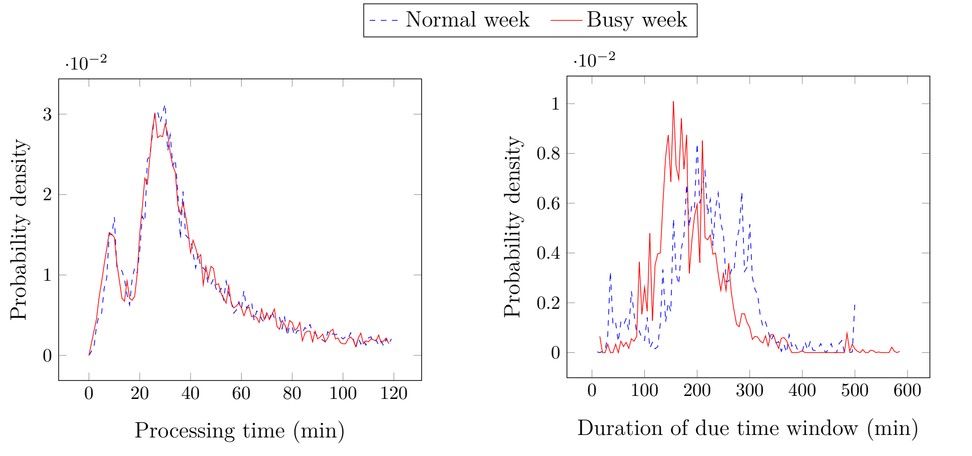


(a) Number of batches per day, where day 1 corresponds to Sunday

(b) Number of truck departures per day, where day 1 corresponds to Sunday

(c) Number of batches due per hour, where hour 0 corresponds to 11:00 pm

Figure 3.6: The workload in the case study



(a) Distribution of processing times for picking batches

(b) Distribution of the due time window durations to pick and deliver batches

Figure 3.7: Variability in the processing times and durations of due time windows for picking batches

Current shift structure The employees are hired to work at the warehouse through third party agencies. Their shifts can start at hour 0, 8 and 9 (i.e., at 11:00 pm, 7:00 am and 8:00 am). The flexible workers are allowed to work for at most 9 hours (i.e., $T_{max} = 9$ hours) and are compensated for at least 6 hours (i.e., $T_{min} = 6$ hours). In contrast to our MILP formulation in Section 3.3, the order pickers receive three breaks at fixed times after they start their shift: a 15-minute break after 2 hours, a 30-minute break after 3.5 hours and another 15-minute break after 6 hours. This shift structure is compliant with the EU and Dutch labor laws.

The warehouse manager has to determine the number of order pickers to schedule for each of the three shift start times, the shift duration of each order picker as well as the batches to be picked by each order picker. Currently, these decisions are made based on experience and intuition of warehouse managers. Due to data privacy concerns, the retailer was not willing to share the actual order picker schedules.

There are four interesting research questions in our case study with the retailer: (i) Can the metaheuristic that is developed in Section 3.5 be used in practice as a decision support tool? (ii) What is the value of flexible break times rather than fixed break times (that are currently used by the retailer)? (iii) What is the value of an additional shift start time? (iv) Can the retailer leverage flexible break times and an additional shift start time to offer a larger minimum compensation T_{min} without incurring higher labor costs? Especially the last question is of particular interest to the retailer since they believe that a larger minimum compensation helps to foster better working relationships with order pickers and to improve the retention rates of employees.

To answer these questions, we first conducted multiple rounds of consultation with the planners and managers to develop plausible and actionable scenarios. The scenarios can be distinguished along three dimensions. First is *flexible break times*. This means that the breaks for order pickers are scheduled at the current break start times ± 15 minutes. Second, an *additional shift start time* is introduced at hour 4 to account for the workload peak as illustrated in Figure 3.6c. We have also tried an additional start time at hour 3 and hour 5, but an additional shift start time at hour 4 resulted in the lowest objective function values. Third, the *minimum compensation time* can be increased to 7 hours or even 8 hours instead of 6 hours. Additionally, we introduce two shift structures that are in line with Section 3.3: a break of 20 minutes needs to be scheduled after at most 2 hours of work (i.e., $T_{break} = 2$ hours and $l_b = 20$ minutes). The minimum compensation time (i.e., T_{min}) is still 6 hours. This shift structure is comparable to the current shift structure in the sense that an employee is compensated for either two or three breaks in any shift, and the values of T_{min} and T_{max} are the same. An overview of the 12 different shift structure scenarios is provided in Table 3.5. Scenario 1 corresponds to the current shift structure, which serves as benchmark. Since the shift structure at the retailer is different than discussed in Section 3.3, we adapted the reduced problem of the metaheuristic to consider flexible break times (see Section 3.8.10 for details).

The overall cost savings as well as the impact on the average number of scheduled order pickers and on the average shift length are presented in Figure 3.8a, Figure 3.8b and Figure 3.8c, respectively, whereas the detailed results are presented in Section 3.8.11. By allowing 15 minutes of flexibility in the break times, the labor cost savings for the retailer are on average 8.8% (comparing scenario 2 to the base case of scenario 1). In particular, fewer employees have to be scheduled and the average shift length decreases as well. Example schedules under scenario 1, 2, 5 and 11 are presented in Section 3.8.11. When the minimum compensation time is increased from 6 hours to 7 or 8 hours (i.e., scenario 3 and 4), the retailer can still expect to have an average cost saving of 5.2% and 0.7%, respectively, by adopting flexible break times. The number of employees to schedule remains similar in the scenarios 2, 3 and 4, however, the average shift length increases. Interestingly, the average shift length in scenario 3 is comparable to scenario 1, i.e., the cost savings of 5.2% in scenario 3 are mainly due to the scheduling of fewer order pickers. Increasing the minimum compensation time to

Table 3.5: Shift structure scenarios to analyse

Shift Structure	Description	Flexible Break Times	Additional Shift Start	T_{min} (hours)
Scenario 1	Current scenario (base case)			6 hours
Scenario 2	Flexible breaks and $T_{min} = 6$ hours	✓		6
Scenario 3	Flexible breaks and $T_{min} = 7$ hours	✓		7
Scenario 4	Flexible breaks and $T_{min} = 8$ hours	✓		8
Scenario 5	Extra shift and $T_{min} = 6$ hours		✓	6
Scenario 6	Extra shift and $T_{min} = 7$ hours		✓	7
Scenario 7	Extra shift and $T_{min} = 8$ hours		✓	8
Scenario 8	Flexible breaks, extra shift and $T_{min} = 6$ hours	✓	✓	6
Scenario 9	Flexible breaks, extra shift and $T_{min} = 7$ hours	✓	✓	7
Scenario 10	Flexible breaks, extra shift and $T_{min} = 8$ hours	✓	✓	8
Scenario 11	Theoretical breaks and $T_{min} = 6$ hours	$T_{break} = 2$ hours, $l_b = 20$ minutes		6
Scenario 12	Theoretical breaks, extra shift and $T_{min} = 6$ hours	$T_{break} = 2$ hours, $l_b = 20$ minutes	✓	6

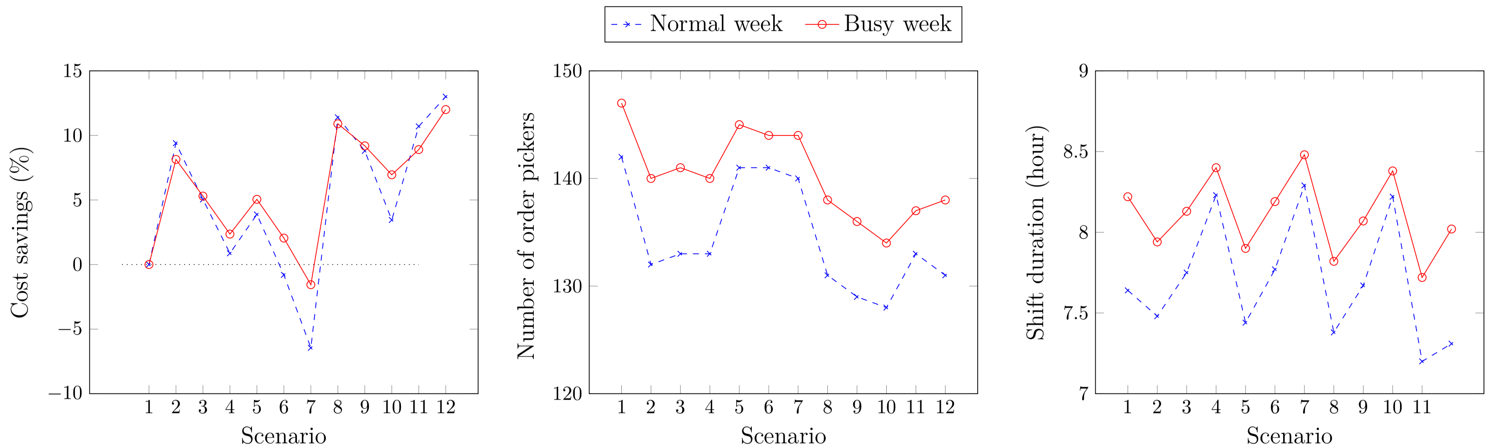
8 hours (in scenario 4) still results in cost savings. This is good news for the retailer, since the additional cost of an increased value of T_{min} is offset against 15 minutes of flexibility in the break start times. Allowing even more flexibility in scheduling breaks (in scenario 11), the average labor cost can decrease an additional 1% (the cost savings in scenario 11 is 9.8%, whereas this is 8.8% in scenario 2). However, this is considered not favorable by the retailer since there is less overlap between the breaks of employees in scenario 11 (see also Section 3.8.11), which is of social importance for the employees. Since the majority of the cost savings in scenario 11 are also captured by allowing 15 minutes of flexibility in the break times (as in scenario 2), these results provided sufficient motivation to initiate implementing this 15 minutes of flexibility.

The average cost savings of an additional shift start time at hour 4 is less substantial compared to flexible break times: 4.5% when T_{min} equals 6 hours, only 0.6% when T_{min} equals 7 hours and an average cost increase of 4% when T_{min} equals 8 hours (for scenario 5, 6 and 7, respectively). This is mainly because the number of order pickers that need to be scheduled decreases significantly less compared to flexible break times, whereas the average shift lengths are comparable.

When combining flexible break times and adding a shift start time at hour 4, the average labor cost can (obviously) decrease even further. What is interesting to observe is that the number of order pickers that is scheduled is actually decreasing as the minimum compensation time T_{min} increases from 6 to 7 hours and from 7 to 8 hours (comparing scenario 8, 9 and 10). Since the increase in average shift length is similar as before, the marginal decrease in average cost savings is less when T_{min} increases. The corresponding average cost savings in these scenarios are 11.1%, 9.0% and 5.2%, respectively. Finally, we observe that most of the cost savings of the flexible break times are captured by the 15-minute flexibility of the break times, since the cost savings in scenario 8 and 12 correspond to 11.1% and 12.5%, respectively (similar when comparing the cost savings between scenario 2 and 11). This reinforces our previous conclusion that it is sufficient to include only 15 minutes of flexibility when scheduling the break times.

3.7 Conclusion

In this paper, we study the order picker scheduling problem where order-picking tasks can be done flexibly but are constrained with due time windows. The problem intersects with the personnel scheduling literature. However, unlike the available



(a) Average cost savings relative to scenario 1

(b) Average number of order pickers scheduled

(c) Average shift length for an order picker

Figure 3.8: Performance of the different shift structure scenarios in the case study

literature, our problem minimizes the labor cost while considering the minimum promised pay to order pickers, the shift start and end times of employees as well as break times. Therefore, break times are explicitly included as scheduling variables as well as shift start and end times (with a minimum compensation time for each order picker). This is a common problem at warehouses with manual order pickers where batches of items need to be picked and delivered to outbound dock doors (or staging lanes) within the time windows that the trucks are scheduled to load the items. Since it combines the order picker planning problem and the shift scheduling problem, we call this the order picker scheduling problem. We present several formulations of the problem with a range of operational restrictions that are important to consider. Two methods are presented to solve the problem. First, an exact branch-and-price algorithm is developed. Since this algorithm can be prohibitive for practical applications, we also present an efficient metaheuristic that combines a savings algorithm and large neighborhood search. The results indicate that the heuristic has a stable performance and is capable of producing near-optimal solutions in a reasonable time for real-life instances.

In a case study, we show how the problem and solution approaches can be used to study different shift structures. In particular, the results show that the retailer can readily increase the minimum compensation duration for workers from 6 to 7 or 8 hours and still realize average labor cost savings of 5.2% or 0.7%, respectively, when a 15 minute flexibility in the scheduling of break times is implemented. By increasing the minimum compensation duration, order pickers might experience an improved job satisfaction to promote job retention. More cost savings of around 4-4.5% can be achieved when an additional shift start time is introduced. Inspired by the result, the retailer under study has decided to implement additional shifts and flexible breaks. Moreover, the findings are applicable beyond this grocery retailer as most retailers in Western Europe operate their warehouses constrained by staging time windows with flexible order pickers in a similar manner.

For the sake of brevity, we only consider identical order pickers in our study. Since the evaluation of a schedule in both our solution approaches to the problem is on the individual employee, order picker specific characteristics such as age and seniority-based breaks as well as restricted and preferred shift starting times can be added to the pricing problem for the branch-and-price algorithm and the reduced problem for the metaheuristic.

Furthermore, we assumed that shifts and order picking tasks are non-overlapping between different planning horizons (i.e., the shift start and end times of every shift are in the same planning horizon when batch orders can be picked). If this assumption were to be relaxed, we propose to use our solution methodology with a rolling horizon. In particular, we suggest to extend the planning horizon with an additional time period during which no new shifts are allowed to start but employees who started their shift in the original planning horizon can finish their shift in the extended time period and perform order picking tasks during that time period. Consequently, order pickers can be scheduled more efficiently at the end of the planning horizon and order picking tasks in the next planning horizon can be performed already. Such a rolling horizon approach results in feasible solutions that may not be optimal as there may not be sufficient order picking tasks available for the order pickers that start their shift in the next planning horizon. A shift scheduling problem that can dynamically include arrivals of new orders in an online environment could be of significant value for e-commerce companies.

Future research can take two additional trajectories within the offline retail environment. First, given the size of the instances in real-life business applications, order batching decisions are made *a priori* (similar to our approach). It can be worthwhile to jointly consider the order batching and shift scheduling problem. Second, we assume norm times to perform the order-picking activities in a deterministic manner. A compelling research direction would be to consider robust order picker scheduling problems with stochastic processing times of batches. These two research directions can be of significant value to both academia and practice.

3.8 Appendix

3.8.1 Illustrative example

Consider the following example that illustrates the importance of shift start, end and break timing decisions along with assignment and sequencing decisions when orders have due time window constraints.

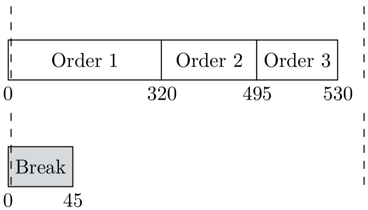
Example 3.1. Consider a warehouse that employs flexible order pickers who can work for at most 540 minutes and are guaranteed a minimum payment equal to 330 minutes of work (independent of the amount of work actually performed). There are 2 possible shift start times: time 0 and time 240 (in minutes). Labor laws require that employees are given a 45-minute uninterrupted break after 330 minutes of work. There are three batch of orders to be picked: the first batch has a processing time of 320 minutes and the due time window is [0,320] in minutes. The second batch has a processing time of 175 minutes and the due time window is [345,415] in minutes. The third batch has a processing time of 35 minutes and the due time window is [440,540] in minutes.

The available approach in the order picking literature suggests to implicitly include breaks as "work" that needs to be scheduled with fixed shift start and end times for all scheduled order pickers. This generic approach can result in a schedule with two order pickers as shown in Figure 3.9a. The first order picker picks all three batches consecutively and then the break is completed by a second order picker since the shift of the first order picker would violate the maximum shift length otherwise. The resulting shifts satisfy the shift duration limit as well as the due time windows to pick the batches. However, the first order picker has no scheduled break. Alternatively, when we assign breaks that comply to the labor laws but we only consider one shift start time, the order picking plan results in the schedule indicated in Figure 3.9b. In this schedule, the second order picker has to wait 170 minutes until she can start to pick items towards batch 2 because of the fixed shift start times. This schedule corresponds to 785 minutes of compensation for the scheduled order pickers. Finally, when we consider multiple shift start times, the shift of the second order picker can start at time 240, and she can immediately start to pick items for batch 2. See Figure 3.9c. This optimal schedule corresponds to 660 minutes of compensation for the scheduled order pickers. The distinction between these three examples clearly indicate the need to explicitly consider shift scheduling decisions in order picker planning problems.

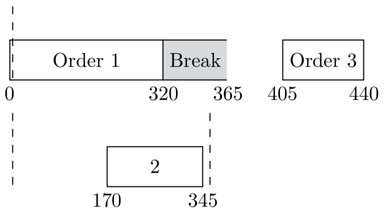
3.8.2 Proof of Proposition 3.1

In this proof, we show that the $P||C_{MAX}$ problem is a special case of the OPSP. Since the $P||C_{MAX}$ problem is known to be NP-hard in the strong sense (Garey & Johnson, 1979), this concludes that the OPSP is also NP-hard.

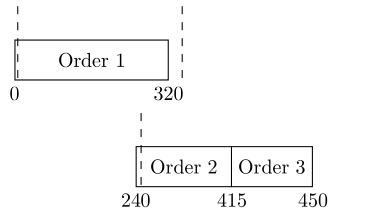
For a parallel machine scheduling problem, consider a set of jobs $N = \{1, 2, \ldots, n\}$ that need to be scheduled for processing on one of m identical parallel machines. The processing time of job i, $i = 1, \ldots, n$, is given by p_i and there are no deadlines or due time windows when each job has to be completed. The objective in the $P||C_{MAX}$



(a) Schedule with implicit breaks and fixed shift start and end times



(b) Schedule with explicit breaks and flexible shift end times



(c) Schedule with explicit breaks and flexible start and end times

Figure 3.9: Illustration of the importance of scheduling breaks and shifts in manual order picking operations with temporal constraints. The duration between the dotted lines represents the time interval for which the order picker is compensated.

problem is to assign all jobs to the machines such that the completion time of the last job among all machines (i.e., the makespan denoted as C_{MAX}) is minimized.

To relate this problem to the OPSP, let the machines be order pickers and the jobs are the batches to be picked. The due time window is given by $r_i = 0$ and $d_i = C_{MAX}$ for all batches $i \in I = \{1, 2, \ldots, n\}$. An order picker can process up to n batches. Shift start and end times for an order picker are not restricted and breaks play no role (i.e., $T_{break} = \infty$ or $l_b = 0$). Furthermore, the processing time of batch i equals $t_i = p_i$, $i = 1, 2, \ldots, n$, $T_{min} = 0$ and $T_{max} = \sum_{i \in N} p_i$. Then the two problems are the same.

3.8.3 Additional constraints to OPSP formulation

The formulation of our OPSP in Section 3.3 suffers from symmetry. This means that interchanging an entire schedule of batches and breaks between different order pickers leaves the objective function unchanged. This issue can be partially addressed with the use of lexicographic ordering constraints (Sherali & Smith, 2001; Jans, 2009). Constraints (3.34) impose a hierarchy between order pickers who are scheduled to work and those who are not. In particular, order pickers are scheduled sequentially based on the start time of their shift (i.e., the start time of order picker 1 cannot exceed the start time of order picker 2, etc.).

$$\sum_{j \in S} j s_{jp-1} \leq \sum_{j \in S} j s_{jp} \qquad \forall p \in P \setminus \{1\} \tag{3.34}$$

The second set of additional constraints provides a lower bound on the number of order pickers that should be scheduled to work between time periods j and j', where $j' > j$. Constraints (3.35) specify that the number of time periods during which an order picker needs to be scheduled to perform an order picking task between the two period j and j' needs to at least the minimum workload that needs to be executed during those time periods to ensure that all batches are picked within their due time windows.

$$\sum_{p\in P}\left[\sum_{\bar{j}=1}^{j'}\left(j'-\left(\max\left\{\bar{j},j\right\}-1\right)\right)s_{\bar{j}p}-\sum_{\bar{j}=1}^{j'}\left(j'-\max\left\{\bar{j},j-1\right\}\right)e_{\bar{j}p}\right]$$
$$\geq\sum_{i\in I}\frac{\left(t_i-\max\left\{\left(d_i-j'\cdot l\right)^{+}+\left((j-1)\cdot l-(d_i-t_i)\right)^{+},\mathbb{1}_{\{r_i\leq j'\cdot l\}}\cdot\left((j-1)\cdot l-(r_i-t_i)^{+}\right)^{+}\right\}\right)^{+}}{l}$$
$$\forall j<j',j\in J,j'\in J \tag{3.35}$$

where $(A)^{+}=\max\{A,0\}$ and $\mathbb{1}_{\{condition\}}$ is an indicator function which has the value 1 if the condition is true and 0 otherwise. The left-hand side of this constraint is the summation of all periods that order pickers are scheduled to work during the time interval from period j until period j'. The first summation equals all time periods from period $\max\{\bar{j},j\}$ until period j', where $\bar{j}\leq j'$ is the starting period of the shift for order picker p. However, the time periods after the shift ends until period j' need to be subtracted. Note that both summations are zero if the start time of a shift exceeds period j', both summations are equal if the start and end time of a shift preceeds period j, and the second summation is zero if the end time of a shift exceeds period j'. The term in between the brackets of the numerator of the right-hand side represents the minimum amount of execution time in the time interval from period j until period j' to pick the items of batch i. Note that period j starts at time $(j-1)\cdot l$ and period j' ends at time $j'\cdot l$. Consequently, the first term of the maximization is the maximum amount of time that batch i can be executed after period j' while still respecting the latest due time d_i and the second term is the maximum amount of time that batch i can be executed before period j while still respecting the earliest due time r_i. Subtracting this from the execution time t_i leaves the minimum amount of time units associated with the order picking task of batch i during the time periods j and j'. Consequently, the numerator in constraints (3.35) is equal to the minimum amount of time units that order pickers need to be scheduled during the time interval from period j until period j' to execute the batches $i\in I$. Since the denominator is the length of one time period, the ratio represents the minimum amount of time periods for which order pickers need to be scheduled to work between time period j and $j'>j$.

The final additional constraint is a lower bound on the objective function value in our OPSP formulation. This requires us to determine the minimum number of time units that order pickers need to be scheduled to perform $\sum_{i \in I} t_i$ time units of order picking tasks. For simplicity we assume that all order pickers have the same shift length, which is indicated by l_s time units. The scheduled time in each shift is assigned for order picking tasks and for breaks. To better understand the dynamics of order picking time and breaks during a shift, consider the following example: $l_s = 7.5$, $T_{break} = 3$ and $l_b = 1$. This means that the employee is scheduled for 6 time units to pick orders and 1.5 time units to take a break. However, when l_s changes to 4.5 time units, then the employee is scheduled for 3.5 time units to pick orders and 1 time unit to take a break. The following expression can be used to determine the number of time units that an employee is picking orders as a function of the shift length:

$$T_{work}(l_s) = \begin{cases} l_s - \left\lfloor \frac{l_s}{T_{break}+l_b} \right\rfloor \cdot l_b, & \text{if } l_s - \left\lfloor \frac{l_s}{T_{break}+l_b} \right\rfloor \cdot (T_{break} + l_b) \leq T_{break} \\ T_{break} \cdot \left(\left\lfloor \frac{l_s}{T_{break}+l_b} \right\rfloor + 1 \right), & \text{otherwise} \end{cases} \tag{3.36}$$

In the first condition, the shift stops while performing an order picking task and it stops while taking a break in the second condition. The minimum number of employees to cover $\sum_{i \in I} t_i$ time units of order picking tasks equals

$$N(l_s) = \left\lceil \frac{\sum_{i \in I} t_i}{T_{work}(l_s)} \right\rceil \tag{3.37}$$

The first $N(l_s) - 1$ order pickers are scheduled to perform order picking tasks for $T_{work}(l_s)$ time units and the additional worker only needs to be scheduled to cover the remaining work, which equals

$$\Delta(l_s) = \sum_{i \in I} t_i - (N(l_s) - 1) \cdot T_{work}(l_s). \tag{3.38}$$

If $\Delta(l_s) < T_{min}$, it can happen that it is not justifiable to assign this work to an N-th order picker since she has to be compensated for T_{min} time units. Instead it would be better to assign this work to the previous $N - 1$ order pickers by extending their shift length. The computations for this analysis depend on the actual shift length l_s, which we discuss first.

The question is what the best duration of the shift (i.e., l_s) needs to be such that the compensation for the scheduled order pickers is minimized. Note that it is most efficient to have a shift without a break. In that case, the productivity of the employee is 100% since a break is non-productive time that is scheduled. However, T_{min} can prevent a shift length of T_{break} time periods. Therefore, it is best to start with T_{break} time units of work and then add the minimum number of blocks of $l_b + T_{break}$ time units such that the total shift length equals or exceeds T_{min} time units for the first time. Consequently, the shift length cannot be extended without scheduling a break first. This corresponds to the following shift length:

$$l_s^{(1)} = \min\left\{T_{break} + \left\lceil \frac{(T_{min} - T_{break})^+}{l_b + T_{break}} \right\rceil (l_b + T_{break}), T_{max}\right\} \tag{3.39}$$

When $N(l_s^{(1)}) - 1$ order pickers are scheduled with a shift length of $l_s^{(1)}$ time units, we can analyze how to add the additional $\Delta(l_s^{(1)})$ time units to the schedule (as introduced above). If they are assigned to an N-th order picker, we need to make sure that this order picker is scheduled at least T_{min} time units. Alternatively, the additional $\Delta(l_s^{(1)})$ time units can be added to the shifts of the other $N-1$ order pickers. However, this would require assigning a break to these employees first. Therefore, the time units that need to be scheduled to cover the additional $\Delta(l_s^{(1)})$ time units of order picking tasks equals

$$l_{s'}^{(1)} = \min\left\{\max\left\{\Delta(l_s^{(1)}) + \left(\left\lceil \frac{\Delta(l_s^{(1)})}{T_{break}} \right\rceil - 1\right) \cdot l_b, T_{min}\right\}, \Delta(l_s^{(1)}) + \left\lceil \frac{\Delta(l_s^{(1)})}{T_{break}} \right\rceil \cdot l_b\right\} \tag{3.40}$$

Consequently, the minimum total number of time units that the order pickers need to be scheduled is given by $LB^{(1)} = (N(l_s^{(1)}) - 1) \cdot l_s^{(1)} + l_{s'}^{(1)}$.

There can be one exceptional scenario. If the shift with a duration of T_{min} time units ends while taking a break (i.e., the condition in Equation (3.36) is not satisfied when $l_s = T_{min}$), then we also need to analyze shift lengths of $l_s^{(2)} = T_{min}$ time units. This means that it can happen that it is better to compensate order pickers for a partial break at the end of their shift instead of adding an entire block of $l_b + T_{break}$ time units to the shift length. Assigning the additional $\Delta(l_s^{(2)})$ time units to the $N(l_s^{(2)}) - 1$ order pickers who are already scheduled is more complex than the previous scenario, since the order pickers with a shift length of T_{min} time units end their shift while taking a break. It requires $\lceil \Delta(l_s^{(2)})/T_{break} \rceil$ order pickers to extend their shift with the additional break time of l_b' time units, where

$$l_b' = l_b - \left(T_{min} - \left\lfloor \frac{T_{min}}{T_{break} + l_b} \right\rfloor \cdot (T_{break} + l_b) - T_{break}\right) \tag{3.41}$$

Therefore, the time units that need to be scheduled to cover the additional $\Delta(l_s^{(2)})$ time units of order picking tasks equals

$$l_{s'}^{(2)} = \min\left\{T_{min}, \Delta(l_s^{(2)}) + \left\lceil \frac{\Delta(l_s^{(2)})}{T_{break}} \right\rceil \cdot l_b' \right\} \tag{3.42}$$

The corresponding minimum total number of time units that the order pickers need to be scheduled in this second scenario is given by $LB^{(2)} = (N(l_s^{(2)}) - 1) \cdot l_s^{(2)} + l_{s'}^{(2)}$.

The smallest value of $LB^{(1)}$ and $LB^{(2)}$ should be added as lower bound on the objective function. However, since order pickers are paid (or compensated) in integral multiples of the period length l, we include the following lower bound, LB, to the original problem formulation

$$\sum_{p \in P} m_p \geq LB = \begin{cases} l \cdot \left\lceil LB^{(1)}/l \right\rceil, & \text{if } T_{min} - \left\lfloor \frac{T_{min}}{T_{break}+l_b} \right\rfloor \cdot (T_{break} + l_b) \leq T_{break} \\ l \cdot \left\lceil \min\{LB^{(1)}, LB^{(2)}\}/l \right\rceil, & \text{otherwise} \end{cases} \tag{3.43}$$

3.8.4 Network flow formulation

Consider a graph $G = (V, A)$, where V is the set of nodes and A is the set of arcs. Next, we introduce O and D as dummy source and sink nodes, respectively. The start and end times of shifts are also indicated by dummy nodes and denoted by S and E, respectively. Each batch in I is also a node. Consequently, $V := \{\mathtt{o}\} \cup \{\mathtt{d}\} \cup I \cup S \cup E$. Let $r_i := 0$ and $d_i := T_{day}$ for $i \in \{\mathtt{o}, \mathtt{d}\}$, whereas r_i andd_i present the start time of a shift for the nodes $i \in S$ and they present the end time of a shift for the nodes $i \in E$. The processing times t_i for the nodes $i \in V \setminus I$ are equal to zero. An example of the network graph is presented in Figure 3.10. In this network flow structure, traversing the graph is analogous to starting the shift at the time associated with node $i \in S$, executing the order picking tasks of the batches associated with nodes $i \in I$, and ending the shift at the time associated with node $i \in E$.

The same input parameters are used as summarized in Table 3.2, whereas the following decision variables are used in the network flow formulation:

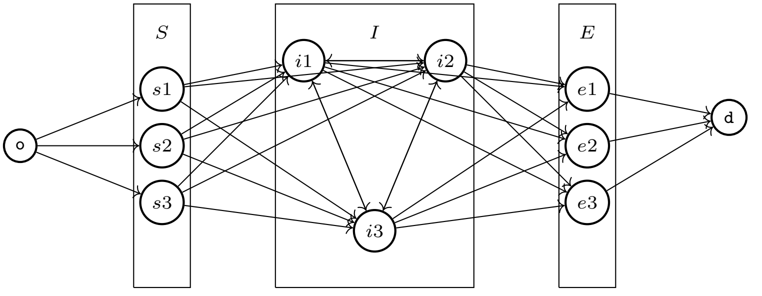


Figure 3.10: A representation of a network graph with three shift start and end times and three batches

x_{ij}^p	is 1 if node $i \in V$ is visited before node $j \in V$ by order picker $p \in P$, else 0
y_i^p	is 1 if order picker $p \in P$ takes a break before visiting node $i \in I$, else 0
c_i^p	completion time at node $i \in N$ by order picker $p \in P$
b_i^p	number of time units without a break before completing the processing time t_i at node $i \in V$ by order picker $p \in P$
m_p	amount of time for which order picker $p \in P$ is compensated

The order picker scheduling problem (OPSP) is formulated as a network flow problem as follows:

$$\min \sum_{p \in P} m_p \tag{3.44}$$

subject to

$$\sum_{j \in S} x_{\mathtt{o}j}^p \leq 1 \qquad \forall p \in P \tag{3.45}$$

$$\sum_{i \in E} x_{i\mathtt{d}}^p \leq 1 \qquad \forall p \in P \tag{3.46}$$

$$\sum_{p \in P} \sum_{i \in I \cup S} x_{ij}^p = 1 \qquad \forall j \in I \tag{3.47}$$

$$\sum_{p \in P} \sum_{j \in I \cup E} x_{ij}^p = 1 \qquad \forall i \in I \tag{3.48}$$

$$\sum_{j \in V} x_{ij}^p = \sum_{j \in V} x_{ji}^p \qquad \forall i \in V \setminus \{\mathtt{o}, \mathtt{d}\}, p \in P \tag{3.49}$$

$$c_j^p - c_i^p - t_j + (1 - x_{ij}^p)M \geq l_b y_j^p \qquad \forall i \in V, j \in V, p \in P \tag{3.50}$$

$$c_i^p + (1 - \sum_{j \in N} x_{ij}^p) M \geq r_i \qquad \forall i \in V, p \in P \tag{3.51}$$

$$c_i^p - (1 - \sum_{j \in N} x_{ij}^p) M \leq d_i \qquad \forall i \in V, p \in P \tag{3.52}$$

$$b_j^p + (1 - x_{ij}^p) M \geq t_j \qquad \forall i \in S, j \in I, p \in P \tag{3.53}$$

$$b_j^p - b_i^p + (1 - x_{ij}^p + y_j^p) M \geq (c_j^p - c_i^p) \qquad \forall i \in I, j \in I, p \in P \tag{3.54}$$

$$b_i^p + (1 - y_i^p) M \geq t_i \qquad \forall i \in I, p \in P \tag{3.55}$$

$$b_i^p \leq T_{break} \qquad \forall i \in I, p \in P \tag{3.56}$$

$$\sum_{i \in I \cup S} x_{ij}^p \geq y_j^p \qquad \forall j \in I, p \in P \tag{3.57}$$

$$T_{min} \sum_{j \in S} x_{\mathtt{o}j}^p \leq m_p \qquad \forall p \in P \tag{3.58}$$

$$c_j^p - c_i^p \leq m_p \qquad \forall i \in S, j \in E, p \in P \tag{3.59}$$

$$\sum_{p \in P} \sum_{j \in S} x_{\mathtt{o}j}^p \leq p_{max} \tag{3.60}$$

$$\sum_{p \in P} \Bigg(\sum_{j \in V \setminus S} x_{\mathtt{o}j}^p + \sum_{i \in S} \sum_{j \in V \setminus I} x_{ij}^p + \sum_{i \in I} \sum_{j \in V \setminus (I \cup E)} x_{ij}^p + \sum_{i \in E} \sum_{j \in V \setminus \{\mathtt{d}\}} x_{ij}^p + \sum_{j \in V} x_{\mathtt{d}j}^p \Bigg) = 0 \tag{3.61}$$

$$b_i^p, c_i^p \geq 0 \qquad \forall i \in V, p \in P \tag{3.62}$$

$$x_{ij}^p \in \{0, 1\} \qquad \forall i \in V, j \in V, p \in P \tag{3.63}$$

$$y_i^p \in \{0, 1\} \qquad \forall i \in I, p \in P \tag{3.64}$$

$$0 \leq m_p \leq T_{max} \qquad \forall p \in P \tag{3.65}$$

The objective function is the same as the model formulation described in Section 3.3. Constraints (3.45) and (3.46) allow for an order picker to start and end her shift at most once. Constraints (3.47) and (3.48) ensure that each batch node is visited (i.e., executed) exactly once. Constraints (3.49) are flow conservation constraints for all nodes in S, I and E.

Constraints (3.50) determine the completion times at all nodes. Constraints (3.51) and (3.52) prevent earliness and tardiness, respectively, at the corresponding nodes. Constraints (3.53) and (3.54) determine the duration at nodes since the last break (including the processing times at the nodes). Constraints (3.55) reset the duration since the last break if a break is scheduled before visiting node i. Constraints (3.56) ensure that an order picker cannot be scheduled to visit nodes consecutively for more than T_{break} time units without a break. Constraints (3.57) make sure that a break before visiting node j can only be assigned to the same order picker who actually visits that node.

Constraints (3.58) ensure that an order picker is compensated for at least T_{min} time units if she is scheduled to work. Constraints (3.59) track the number of time units that an order picker is scheduled to work from the start of the shift to the end of the shift. Constraints (3.60) restrict the maximum number of order pickers that can be scheduled (i.e., paths that can be visited in the graph).

Constraints (3.61) ensure that the paths in the network can only traverse through the nodes of the graph that are allowed by restricting the flow on the arcs that are not allowed to be zero. Constraints (3.62) to (3.65) define the domain and range of the decision variables.

Additional constraints

The symmetry breaking constraints and additional constraints to tighten the model formulation from Section 3.8.3 can easily be translated to the network flow formulation. For instance, the equivalent of constraints (3.34) is

$$\sum_{j \in S} j x_{\circ j p-1} \leq \sum_{j \in S} j x_{\circ j p} \qquad \forall p \in P \setminus \{1\} \tag{3.66}$$

Constraints (3.35) can be transformed similarly and constraints (3.43) can directly be included without any modifications.

3.8.5 Extensions to OPSP formulation

Our original OPSP formulation in Section 3.3 is generic, but can be extended with additional constraints to be applicable to planning requirements of individual warehouses. In this appendix, we discuss and model full time employees as well as common break and labor law requirements that can easily be included to the original problem formulation.

Warehouses can have numerous restrictions pertaining the use of human labor dependent on specific labor laws. The European Union stipulates that all employees working more than 6 hours in a shift must be given a break, but the length of the break is specified by individual countries (European Parliament, Council of the European Union, 2003). For example, the Dutch “Working Hours Act" mandates that each employee who works 5.5 hours must be given a break of 30 minutes, which could be reduced to 15 minutes conditional on agreements between the employer and labor unions[3] (Ministry of Social Affairs and Employment, 2010). There can also be differences between companies within the same country or region, based on labor union agreements and company cultures regarding working time, start times and break times.

In the United States, the Fair Labor Standards Act (FLSA) does not require to give employees a rest or meal break. However, individual states can carry break laws in their legislature. Common constructs include a 10-15 minute rest break for every 3.5 or 4 hours worked, and an uninterrupted meal break of 30 minutes for employees who work more than five hours in a day. Such meal breaks must usually be provided sometime after the first two hours of work and before the last two hours of work. In Japan, the labor laws dictate that employees are entitled to a break of at least 45 minutes when they work six to eight hours and to a break of at least one hour when the working hours exceed eight hours. According to the Employment Standards Act in Canada, employees are entitled to at least 30 consecutive minutes of break during every 5-hour work period. The Employment Relations Act in New Zealand prescribes a 30-minute meal break and multiple 10-minute rest breaks based on the number of hours worked. Even though the Fair Work Act in Australia does not provide a statutory entitlement to any work breaks, the Fair Work Ombudsman prescribes similar rules as New Zealand.

The Fair Labor Standards Act (FLSA) in the United States does not address flexible work schedules. Alternative work arrangements such as flexible work schedules are a matter of agreement between the employer and the employee. This seems to be the standard in many countries. However, there are labour laws that specify a minimum compensation duration. For instance, employees in the United States, Canada and Australia must be paid for at least 3 hours each time they are required to report to work.

[3] A working day cannot exceed 11 hours and if a shift is longer than 10 hours, the total break time has to be at least 45 minutes. The overall break time can be split in multiple breaks, but need to be at least 15 consecutive minutes.

3.8.5.1 Full time employees

In our original problem formulation, we assume that all employees are flexible and part-time workers. However, if there are full-time employees with a fixed contract, the problem formulation can easily be extended to accommodate these circumstances. Consider the set of employees P, where the first $\overline{p}$ employees are full-timers and the remaining $p_{max} - \overline{p}$ employees are part-timers. If the shift start and end times are included in an employee's contract, this means that s_{jp} and e_{jp} for $p = 1, \ldots, \overline{p}$ are predetermined values rather than decision variables. If only the shift length is included in the contract (let us denote this by l_s, which is not necessarily equal to T_{max}), then the duration between the start and end of the shift cannot exceed this:

$$\Big(\sum_{j \in E} j e_{jp} - \sum_{j \in S} (j-1) s_{jp}\Big) l \leq l_s, \qquad \forall p \in \{1, \ldots, \overline{p}\} \tag{3.67}$$

However, full-time workers have a fixed compensation specified in their contractual agreements. Therefore, the values of m_p for $p = 1, \ldots, \overline{p}$ are predetermined regardless of the tasks assigned to them.

Furthermore, there are often labor union agreements stipulating a maximum number of flexible employees that can be employed compared to the full-time employees (Bard et al., 2007). For instance, if there is a maximum ratio γ between the number of flexible order pickers divided by the number of full-time order pickers, then p_{max} is restricted:

$$\frac{p_{max} - \overline{p}}{\overline{p}} \leq \gamma \quad \iff \quad p_{max} \leq (\gamma + 1)\overline{p}. \tag{3.68}$$

3.8.5.2 Maximum number of order pickers per time period

The total number of order pickers that are scheduled to work at the same time period can be restricted, for instance if equipment has a limited availability (such as pick trucks in a warehouse). This can be included in the original problem formulation by adding the following constraints:

$$\sum_{j'=1}^{j} \sum_{p \in P} (s_{j'p} - e_{j'-1,p}) \leq p^{eqp} \qquad \forall j \in J \tag{3.69}$$

where $e_{0,p} = 0$.

3.8.5.3 Designated break times and break time windows

In the original problem formulation, it is assumed that each order picker can take a break at any time. However, break times can be restricted to certain hours in the day because of opening times of cafeterias or labor union agreements. Kniffin et al. (2015) observe that commensality (i.e., eating together) is positively correlated to the performance of employees, and it may offer the only valuable and rare occasion for employees to socialize. Therefore, consider a set of periods $J' \subseteq J$ at which employees are allowed to take a break. The decision variables y_{kp} also need to include the time period. Define y_{jkp} as a binary decision variable that has the value 1 if a break is scheduled at the start of period $j \in J'$ which is the k-th position in the shift for order picker $p \in P$ and 0 otherwise. The following constraints make sure that $c_{kp} = (j-1) \cdot l + l_b$ if $y_{jkp} = 1$ for $j \in J'$ and c_{kp} is unconstrained otherwise:

$$c_{kp} + M \cdot (1 - y_{jkp}) \geq (j-1) \cdot l + l_b \qquad \forall j \in J', k \in K, p \in P \tag{3.70}$$

$$c_{kp} - M \cdot (1 - y_{jkp}) \leq (j-1) \cdot l + l_b \qquad \forall j \in J', k \in K, p \in P \tag{3.71}$$

Note that y_{kp} needs to be replaced by $\sum_{j \in J'} y_{jkp}$ in the original problem formulation.

More general, a break can also be restricted to start within a time window rather than to the beginning of a time period. Consequently, define J' as the set of break time windows over the planning horizon and $[r_j^{break}, d_j^{break}]$ as the start and end time of break window $j \in J'$. The decision variables y_{jkp} have a similar interpretation as before, except that it refers to a break in the j-th break time window (not the start of a time period). The following constraints make sure that the break starts and finishes within the associated break time window if $y_{jkp} = 1$ for $j \in J'$ and c_{kp} is unconstrained otherwise:

$$c_{kp} + M(1 - y_{jkp}) \geq r_j^{break} + l_b \qquad \forall j \in J', k \in K, p \in P \tag{3.72}$$

$$c_{kp} - M(1 - y_{jkp}) \leq d_j^{break} \qquad \forall j \in J', k \in K, p \in P \tag{3.73}$$

3.8.5.4 Multiple types of breaks

Company culture and contractual agreements can require to give order pickers multiple types of breaks. For instance, several short breaks of up to 15 minutes are given during a shift in addition to a lunch break of 30 minutes. Let Φ indicate the set of various types of breaks. For each break type $\phi \in \Phi$, denote the maximum amount of time an order picker can work without a break of type ϕ by T^{ϕ}_{break}, and the length of a break of type ϕ by l^{ϕ}_{b}. We should note that breaks often have a hierarchy between them, i.e., a break can reset the working time accumulated for a different type of break. For example, employees are given a coffee break of 15 minutes after 3 hours of work and a lunch break of 30 minutes after 5 hours of work. The lunch break will reset the working time accumulated for the coffee break but not the other way around. Let Φ^{ϕ} denote the set of breaks that reset the accumulated working time for break type $\phi \in \Phi$. To continue the example, the coffee break is type 1 and the lunch break is type 2. Consequently, $\Phi = \{1, 2\}$, $T^{1}_{break} = \{3\}$, $T^{2}_{break} = \{5\}$, $\Phi^{1} = \{2\}$ and $\Phi^{2} = \varnothing$.

We introduce a set of binary decision variables y^{ϕ}_{kp} that indicate if the break of type ϕ is scheduled at the k-th position in the shift for order picker $p \in P$ and 0 otherwise. The following constraints include the hierarchy as discussed earlier:

$$c_{kp} - \left(c_{hp} - \sum_{i \in I} t_i x_{ihp} \right) \leq T^{\phi}_{break} + M \left(\sum_{\phi' \in \Phi^{\phi}} \sum_{k'=h+1}^{k} y^{\phi'}_{k'p} \right)$$

$$\forall k \in K, h < k, p \in P, \phi \in \Phi \tag{3.74}$$

Furthermore, y_{kp} needs to be replaced by $\sum_{\phi \in \Phi} y^{\phi}_{kp}$ and $l_b y_{kp}$ by $\sum_{\phi \in \Phi} y^{\phi}_{kp} l^{\phi}_{b}$ in the original problem formulation.

3.8.6 Pseudocode of labeling algorithm

Algorithm 2 Dynamic programming algorithm for the pricing problem

$\mathscr{L}_i := \{(i, T_{min} + \psi, T_i, (V_h^1 = 0, \ldots, V_h^{|V'|} = 0))\} \quad i \in S$ ▷ Initialization at shift start nodes
for $i \in V' \setminus S$ **do**
  $\mathscr{L}_i := \emptyset$ ▷ Initialization at remaining nodes
$\Delta := S$ ▷ Unvisited vertices
while $\Delta \neq \emptyset$ **do**
  for $i \in \Delta$ **do**
    for $j \in \delta(i)$ **do** ▷ Feasible nodes only
      for $L_h^i \in \mathscr{L}_i$ **do**
        if $V_{L_h^i}^j = 0$ **then** ▷ If j is not already visited
          $T_j := f(T_i, j)$ ▷ Extension without break
          if T_j is feasible **then**
            ADD $L_h^j = (j, c_h^j, T_j, V_h)$ to $\mathscr{L}_j$
          $T_j := g(T_i, j)$ ▷ Extension with break
          if T_j is feasible **then**
            ADD $L_h^j = (j, c_h^j, T_j, V_h)$ to $\mathscr{L}_j$
      if $\mathscr{L}_j$ changed **then**
        $\mathscr{L}_j \leftarrow DOMINANCE(\mathscr{L}_j)$ ▷ Eliminate dominated lables
        $\Delta \leftarrow \Delta \cup \{j\}$
    $\Delta \leftarrow \Delta \setminus \{i\}$
return $\arg\min_{L \in \mathscr{L}_D} \{c_L^D\}$

3.8.7 Proof of Proposition 3.2

Let x^* indicate a sequence of batches with cost $z(x^*)$ assigned to an order picker in the optimal solution where batch i precedes batch j. The completion times for batch i and batch j in x^* are presented by c_i^* and c_j^*, respectively. Furthermore, let γ_i^* be the earliest time the order picker can start picking items for batch i such that the sequence x^* remains optimal and q_j^* is the latest time the order picker can start to work on the batch succeeding batch j such that the sequence x^* remains optimal.

When the performance of the batches i and j is reversed, the new sequence is indicated by x' with cost $z(x')$ and the completion times of the batches i and j become c_i' and c_j', respectively. To proof Proposition 3.2, it is sufficient to show that the new solution is feasible and the cost $z(x')$ equals $z(x^*)$. There are four possible sequences to consider in regard to scheduling breaks before performing batch i and batch j.

Case I: batch i and j are performed without a break before i or between i and j (i.e., $i|j$)

Sub-case (i): $\gamma_i^* + t_i < r_i = c_i^*$ and $\gamma_i^* + t_i + t_j < c_j^* \leq q_j^*$ (i.e., there is a wait between γ_i^* and c_i^*)

In the new solution x', the order picker might need to wait before the completion of batch j. If the order picker needs to wait, $\gamma_i^* + t_j < r_j = c'_j$ and $\gamma_i^* + t_j + t_i < c'_i \leq q_j^*$. If the order picker does not need to wait, $\gamma_i^* + t_j \geq r_i = c'_j$ and $\gamma_i^* + t_j + t_i = c'_i \leq q_j^*$. In either case, q_j^* is not violated and the new sequence is feasible with the same cost.

Sub-case (ii): $\gamma_i^* + t_i = c_i^* \geq r_i$ and $\gamma_i^* + t_i + t_j = c_j^* \leq q_j^*$ (i.e., there is no wait between γ_i^* and c_i^*)

When the batches are reversed, there would still be no waiting time as $t_j > t_i$. Consequently, $\gamma_i^* + t_j = c'_j > r_i$ and $\gamma_i^* + t_j + t_i = c'_i \leq q_j^*$. Furthermore, q_j^* is respected with the reversal of the batches. Consequently, $z(x') = z(x^*)$.

Case II: batch i and j are performed with a break before i but not between i and j (i.e., $y|i|j$)

Sub-case (i): $\gamma_i^* + l_b + t_i < r_i = c_i^*$ and $\gamma_i^* + l_b + t_i + t_j < c_j^* \leq q_j^*$ (i.e., there is a wait between γ_i^* and c_i^*)

If there is a wait in the new solution x', then $\gamma_i^* + l_b + t_j < r_j = c'_j$ and $\gamma_i^* + l_b + t_j + t_i < c'_i \leq q_j^*$. If the order picker does not need to wait, then $\gamma_i^* + l_b + t_j \geq r_i = c'_j$ and $\gamma_i^* + l_b + t_j + t_i = c'_i \leq q_j^*$. In either case, q_j^* is not violated, and the new sequence is feasible with the same cost. Furthermore, the amount of work done between the end of the break and the completion of batch i does not increase.

Sub-case (ii): $\gamma_i^* + l_b + t_i \geq r_i = c_i^*$ and $\gamma_i^* + l_b + t_i + t_j = c_j^* \leq q_j^*$ (i.e., there is no wait between γ_i^* and c_i^*)

The reversal of the batches i and j does not introduce any waiting time for the order picker. Furthermore, the amount of time since the break remains the same.

Case III: batch i and j are performed without a break before i but between i and j (i.e., $i|y|j$)

Let the amount of work since the last break or since the start when the first batch is picked until the beginning when batch i is picked be denoted by b_r. If the sequence in which the batches i and j are performed is reversed, it is possible that $b_r + t_j > T_{break}$, which results in an infeasible solution. Therefore, batch j can not always precede batch i in this case.

Case IV: batch i and j are performed with a break before i and between i and j (i.e., $y|i|y|j$)

The same proof as Case I can be used, where the new value of t_i is increased by l_b and the new value of t_j by l_b.

3.8.8 Optimal schedule for one order picker

In the savings algorithm in Section 3.5.1 and the large neighborhood search in Section 3.5.2, the assignment of batches to an order picker is altered. In this appendix, we present an MILP formulation to optimally schedule the batches to an individual order picker such that the due time windows and break constraints are satisfied. Before we try to solve the reduced OPSP formulation with one order picker, we perform a set of infeasibility checks to verify whether a feasible solution can be found.

Infeasibility conditions

Even though the computation time of the MILP for the reduced problem is short, infeasibility checks can be performed first as pre-processing step to easily verify whether the order picking tasks cannot be combined in a feasible schedule. Let the set of batches to be included in the schedule be denoted by $B \subseteq I$. If any of the following infeasibility conditions is satisfied, the reduced problem is infeasible. The complexity of these checks are indicated within parentheses.

- $\sum\limits_{i \in B} t_i + \left(\left\lceil (\sum\limits_{i \in B} t_i)/T_{break} \right\rceil - 1\right) l_b > T_{max}$, (complexity: $\mathcal{O}(|B|)$), the total processing time of the batches and the necessary break times exceeds the maximum shift length.
- $\max_{i \in B}\{r_i\} - \min_{i \in B}\{d_i - t_i\} > T_{max}$, (complexity: $\mathcal{O}(|B|)$), the time difference between the latest earliest completion time and the earliest latest start time exceeds the maximum shift length (i.e., there are tasks that cannot be scheduled to be completed earlier or to start later such that the maximum shift length is not exceeded).

- $|B| + \left(\left\lceil (\sum_{i \in B} t_i)/T_{break} \right\rceil - 1\right) > \bar{k}$, (complexity: $\mathcal{O}(|B|)$), the minimum number of tasks (either order picking batches or breaks) to cover the workload in B exceeds the maximum number of tasks in a shift.

- $(\sum_{i \in B} t_i) > \max_{i \in B}\{d_i\} - (\min_{i \in B}\{r_i - t_i\})^+$, (complexity: $\mathcal{O}(|B|)$), the total processing time of all batches in B cannot be assigned to one order picker between the earliest start time and the latest completion time.

- $\exists i, j \in B : (r_i + t_j > d_j) \wedge (r_j + t_i > d_i)$, (complexity: $\mathcal{O}(|B|^2)$), the due time windows prohibit the order picker to perform batch j after i or to perform batch i after j (i.e., the due time windows are violated if both tasks need to be performed by the same order picker).

- Define the earliest start time of task $i \in B$ as $\gamma_i := r_i - t_i$. Let Γ_B and D_B denote the earliest start times and latest completion times of the batches in B, respectively. For any earliest start time $\gamma \in \Gamma_B$ and latest completion time $d \in D_B$ where $\gamma < d$, the set $I_{\gamma d}$ represents all batches in B that need to start at or after time γ and be completed at time d at the latest, i.e., $I_{\gamma d} := \{i \in B | (\gamma_i \geq \gamma) \wedge (d_i \leq d)\}$. The condition $\exists \gamma \in \Gamma_B, d \in D_B : \sum_{i \in I_{\gamma d}} t_i > (d - \gamma)$ indicates that the work load between γ and d cannot be completed by one order picker (complexity: $\mathcal{O}(|B|^3)$).

Reduced OPSP formulation with one order picker

When none of the infeasibility conditions are satisfied, we try to optimally sequence the tasks to the individual order picker while satisfying all constraints. This means that the original OPSP formulation of Section 3.3 is simplified by removing the multiple order pickers $p \in P$. Consequently, the set of batches that need to be scheduled for the single order picker is denoted by B and the decision variables in this reduced problem become the following:

x_{ik}	is 1 if batch $i \in B$ is scheduled to be picked at the k^{th} position in the shift for the order picker, where $k \in K$, else 0
y_k	is 1 if a break is scheduled at the k^{th} position in the shift for the order picker, where $k \in K$, else 0
s_j	is 1 if the order picker starts the shift at the beginning of period $j \in S$, else 0
e_j	is 1 if the order picker ends the shift at the end of period $j \in E$, else 0
c_k	completion time of the task scheduled at the k^{th} position in the shift of the order picker, where $k \in K$
m	amount of time for which the order picker is compensated

The reduced problem (RP) of the OPSP is formulated as a MILP model as follows:

RP:

$$\min m \tag{3.75}$$

subject to

$$\sum_{i \in B} x_{ik} + y_k \leq 1 \qquad \forall k \in K \tag{3.76}$$

$$\sum_{k \in K} x_{ik} = 1 \qquad \forall i \in B \tag{3.77}$$

$$\sum_{j \in J \setminus S} s_j + \sum_{j \in J \setminus E} e_j = 0 \tag{3.78}$$

$$\Big(\sum_{j \in S} (j-1) s_j\Big) l + \sum_{i \in B} t_i x_{i1} + l_b y_1 \leq c_1 \tag{3.79}$$

$$c_{k-1} + \sum_{i \in B} t_i x_{ik} + l_b y_k \leq c_k \qquad \forall k \in K \setminus \{1\} \tag{3.80}$$

$$\sum_{j \in E} (j e_j) l \geq c_{\bar{k}} \tag{3.81}$$

$$c_k + M(1 - x_{ik}) \geq r_i \qquad \forall i \in B, k \in K \tag{3.82}$$

$$c_k - M(1 - x_{ik}) \leq d_i \qquad \forall i \in B, k \in K \tag{3.83}$$

$$c_k - \Big(c_h - \sum_{i \in B} t_i x_{ih}\Big) - M\Big(\sum_{k'=h+1}^{k} y_{k'}\Big) \leq T_{break} \qquad \forall h, k \in K, h < k \tag{3.84}$$

$$\sum_{i \in B} x_{ik-1} + y_{jk-1} \geq \sum_{i \in B} x_{ik} + y_{jk} \qquad \forall k \in K \setminus \{1\} \tag{3.85}$$

$$(\sum_{j \in E} j e_j - \sum_{j \in S} (j-1) s_j) l \leq m \tag{3.86}$$

$$c_k \geq 0 \qquad \forall k \in K \qquad (3.87)$$

$$x_{ik} \in \{0,1\} \qquad \forall i \in B, k \in K \qquad (3.88)$$

$$y_k \in \{0,1\} \qquad \forall k \in K \qquad (3.89)$$

$$s_j, e_j \in \{0,1\} \qquad \forall j \in J \qquad (3.90)$$

$$\max\left\{T_{min}, \left\lceil \left(\sum_{i \in B} t_i + \left(\left\lceil (\sum_{i \in B} t_i)/T_{break} \right\rceil - 1 \right) l_b \right) / l \right\rceil l \right\} \leq m \leq T_{max} \qquad (3.91)$$

There is a one-to-one relation with the constraints in this RP formulation and the OPSP formulation in Section 3.3.

This reduced problem is strongly NP-hard as it is a generalization of the minimum tour duration problem (MTDP), which is known to be strongly NP-hard (Tilk & Irnich, 2017). However, we can use the overall architecture of the solution approaches for the MTDP to efficiently solve the MILP since the architecture allows us to incorporate our additional operational constraints. Furthermore, the scale of the reduced problem as a subroutine of the savings algorithm and the LNS is small enough to be practically solvable. The abovementioned infeasibility tests also limit the number of times that the reduced problem is solved unnecessarily.

3.8.9 Detailed result of Section 3.6.2

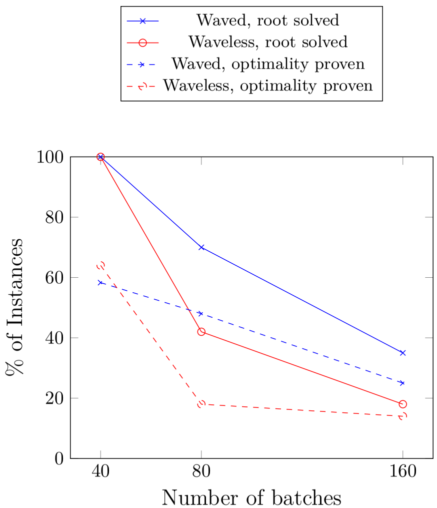
Table 3.6 until Table 3.19 provide the results of all individual numeral instances that are summarized in Table 3.4. For the branch-and-price algorithm, we report the optimal solution and best integer solution found by column generation at the root node (this corresponds to lower bound LB^{LP} and upper bound UB^{LP}, respectively) as well as the best lower bound from branching and the best integer solution after branching (denoted by LB^{IP} and UB^{IP}, respectively). The computational time to solve the root node, perform branching and perform the overall branch-and-price algorithm is indicated by CPU^{LP}, CPU^{IP} and CPU^{BP}, respectively (all in seconds). The relative cost increase of the solutions found by the savings algorithm, commercial solver Gurobi Optimizer and the metaheuristic compared to performance of the branch-and-price algorithm is indicated by $\%\Delta^{S}$, $\%\Delta^{GUR}$ and $\%\Delta^{MH}$, respectively, where $\%\Delta^{X} = (z(X) - z(BP))/z(BP) \times 100$ and $z(X)$ is the best integer solution found by solution procedure X. The computational time to perform solution procedure X is provided by CPU^{X} (in seconds) for solution procedure X. Note that we terminate the solution procedure when the computational time reaches 1,800 seconds.

As discussed in Section 3.6.2, the performance of the branch-and-price algorithm depends on the truck departure pattern (waved or waveless), processing time distribution (uniform or exponential), and number of truck departures per staging lane. Figure 3.11a shows that waved instances are more difficult to solve than waveless instances. Figure 3.11b shows that instances with exponentially distributed processing times are more difficult to solve than when the processing times have a uniform distribution. Finally, Figure 3.11c shows that instances with more truck departures per staging lane are easier to solve than instances with less truck departures per staging lane.

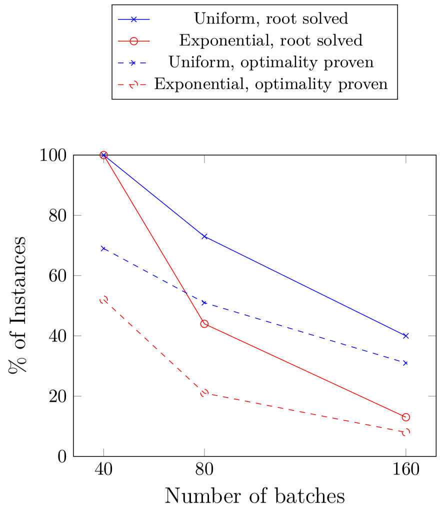
3.8.10 Including different break time structures in solution procedures

The shift structure to include breaks as discussed in Section 3.3 is very flexible, whereas there might be other break types as discussed in Section 3.8.5. This is actually the situation in our case study as discussed in Section 3.6.3, where there are three break types: two 15-minute breaks after 2 hours and 6 hours of starting the shift and one 30-minute break after 3.5 hours of starting the shift. To find a solution to the OPSP with the metaheuristic where these new break structures are included, we can directly include the constraints as formulated in Section 3.8.5 to the problem formulation. In particular, $\Phi = \{1, 2, 3\}$ where break type 1 and 2 indicate the two shorter breaks and break type 3 indicates the longer break. Consequently, the duration of the each break type is given by $l_b^1 = l_b^2 = 0.25$ and $l_b^3 = 0.5$, whereas the start time of each break type is given by $r^1 = 2$, $r^2 = 5$ and $r^3 = 3.5$. In the case where the start time of the breaks can have a 15-minute flexibility (i.e., scheduled at most 15 minutes earlier or later), we indicate the flexibility in the timing of the breaks by Ψ, which is set to 15 minutes (or $\Psi = 0.25$ hours). When there is no such flexibility (i.e., in the scenarios 1, 5, 6 and 7), the value of Ψ equals zero.

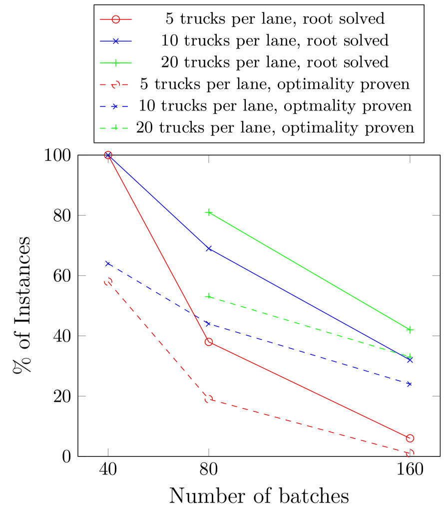
To use the metaheuristic, we have to reformulate the reduced problem to verify whether two schedules can be combined for one order picker (as formulated in Section 3.8.8). Let us first redefine some of the decision variables:



(a) Waved and waveless instances



(b) Uniform and exponential order processing times



(c) Number of truck departures per staging lane

Figure 3.11: Relative number of instances for which either the root node is solved or the optimal solution is found with the branch-and-price algorithm

Table 3.6: Results for Instances with 40 batches, 5 truck departures per staging lane, uniformly distributed processing time

	Branch & Price							S		GUR		MH	
	Linear Relaxation			Integer									
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{GUR}$	CPU^{GUR}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-40-45-SSTR1	4480.0	4800.0	1.5	4500.0	4800.0	1796.6	1800.0	0.0	1.9	0.0	1800.0	0.0	47.6
U-Waved-40-45-SSTR2	4200.0	4440.0	1.4	4260.0	4440.0	1795.1	1800.0	8.1	3.5	0.0	1800.0	-1.4	49.8
U-Waved-40-45-SSTR3	4152.0	4200.0	0.9	4200.0	4200.0	2.8	6.4	2.9	2.8	0.0	1800.0	0.0	44.8
U-Waved-40-45-SSTR4	4480.0	4800.0	5.2	4500.0	4800.0	1792.7	1800.0	0.0	2.1	0.0	1800.0	0.0	57.1
U-Waved-40-45-SSTR5	3480.0	3660.0	146.4	3480.0	3660.0	1650.4	1800.0	4.9	3.2	3.3	1800.0	0.0	195.2
U-Waved-40-45-SSTR6	2685.0	2760.0	4.7	2700.0	2700.0	100.3	107.8	4.4	2.8	4.4	1800.0	2.2	55.9
U-Waved-40-75-SSTR1	4980.0	4980.0	0.6	4980.0	4980.0	2.3	5.2	18.1	2.3	0.0	1800.0	0.0	31.2
U-Waved-40-75-SSTR2	4740.0	4740.0	0.8	4740.0	4740.0	2.5	5.9	10.1	2.5	3.8	1800.0	0.0	31.3
U-Waved-40-75-SSTR3	4660.0	4740.0	0.5	4680.0	4680.0	5.3	8.3	2.6	2.5	1.3	1800.0	1.3	32.2
U-Waved-40-75-SSTR4	4800.0	4800.0	0.8	4800.0	4800.0	2.5	5.7	21.3	2.5	0.0	1800.0	0.0	37.5
U-Waved-40-75-SSTR5	3840.0	3840.0	2.7	3840.0	3840.0	2.5	7.7	15.6	2.4	0.0	1800.0	0.0	40.3
U-Waved-40-75-SSTR6	3400.0	3480.0	1.1	3420.0	3420.0	5.4	9.0	12.3	2.5	5.3	1800.0	0.0	39.4
U-Waved-40-105-SSTR1	5600.0	5760.0	0.5	5640.0	5760.0	1797.5	1800.0	19.8	1.9	0.0	1800.0	0.0	25.4
U-Waved-40-105-SSTR2	5280.0	5400.0	3.4	5280.0	5400.0	1794.3	1800.0	22.2	2.4	5.6	1800.0	0.0	29.1
U-Waved-40-105-SSTR3	5160.0	5160.0	0.2	5160.0	5160.0	2.5	5.1	2.3	2.5	0.0	1800.0	0.0	31.2
U-Waved-40-105-SSTR4	5380.0	5520.0	5.9	5400.0	5520.0	1792.0	1800.0	31.5	2.0	5.4	1800.0	0.0	36.8
U-Waved-40-105-SSTR5	4936.0	5040.0	9.6	4980.0	4980.0	1792.1	1804.1	21.7	2.4	9.6	1800.0	1.2	45.2
U-Waved-40-105-SSTR6	4771.4	4800.0	3.4	4800.0	4800.0	2.8	8.9	11.3	2.7	8.8	1800.0	0.0	37.6
U-Waveless-40-45-SSTR1	3040.0	3360.0	1.5	3060.0	3360.0	1796.1	1800.0	1.8	2.4	0.0	1800.0	0.0	59.0
U-Waveless-40-45-SSTR2	2360.0	2520.0	2.0	2400.0	2520.0	1795.9	1800.0	9.5	2.1	0.0	1800.0	0.0	123.0
U-Waveless-40-45-SSTR3	2230.0	2340.0	12.3	2280.0	2280.0	99.4	114.8	10.5	3.0	7.9	1800.0	0.0	113.8
U-Waveless-40-45-SSTR4	2640.0	2940.0	117.2	2640.0	2940.0	1680.6	1800.0	16.3	2.3	0.0	1800.0	0.0	360.0
U-Waveless-40-45-SSTR5	2280.0	2340.0	477.3	2340.0	2340.0	2.9	483.1	25.6	2.9	7.7	1800.0	0.0	360.0
U-Waveless-40-45-SSTR6	2236.0	2280.0	10.3	2280.0	2280.0	2.5	15.3	13.2	2.5	7.9	1800.0	2.6	81.9
U-Waveless-40-75-SSTR1	4128.0	4320.0	7.4	4140.0	4320.0	1790.8	1800.0	22.2	1.8	0.0	1800.0	0.0	33.0
U-Waveless-40-75-SSTR2	3540.0	3600.0	3.8	3540.0	3540.0	17.9	23.6	39.0	2.0	6.8	1800.0	1.7	38.4
U-Waveless-40-75-SSTR3	3360.0	3420.0	4.4	3360.0	3360.0	37.2	43.5	12.5	1.9	8.9	1800.0	1.8	40.7
U-Waveless-40-75-SSTR4	3720.0	3900.0	125.3	3720.0	3900.0	1672.8	1800.0	24.6	2.0	10.8	1800.0	1.5	311.6
U-Waveless-40-75-SSTR5	3441.2	3540.0	207.6	3480.0	3540.0	1589.6	1800.0	10.2	2.8	6.8	1800.0	0.0	242.4
U-Waveless-40-75-SSTR6	3286.7	3360.0	15.1	3300.0	3360.0	1782.1	1800.0	12.5	2.8	7.1	1800.0	0.0	57.6
U-Waveless-40-105-SSTR1	6880.0	7200.0	0.4	6900.0	7200.0	1797.4	1800.0	13.3	2.3	0.0	1800.0	0.0	22.8
U-Waveless-40-105-SSTR2	5420.0	5460.0	0.4	5460.0	5460.0	2.3	5.0	20.9	2.3	1.1	1800.0	1.1	25.3
U-Waveless-40-105-SSTR3	4760.0	4800.0	0.5	4800.0	4800.0	1.8	4.1	5.0	1.8	0.0	1800.0	0.0	27.9
U-Waveless-40-105-SSTR4	5760.0	5760.0	1.6	5760.0	5760.0	2.0	5.5	41.7	1.9	0.0	1800.0	0.0	26.3
U-Waveless-40-105-SSTR5	5040.0	5040.0	0.9	5040.0	5040.0	2.1	5.0	28.6	2.1	16.7	1800.0	0.0	28.3
U-Waveless-40-105-SSTR6	4620.0	4620.0	0.8	4620.0	4620.0	2.3	5.3	15.6	2.2	10.4	1800.0	0.0	30.7

Table 3.7: Results for Instances with 40 batches, 5 truck departures per staging lane, exponentially distributed processing time

	Branch & Price							**S**		**GUR**		**MH**	
	Linear Relaxation			Integer									
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{GUR}$	CPU^{GUR}	$\%\Delta^{MH}$	CPU^{MH}
E-Waved-40-45-SSTR1	4140.0	4380.0	2.6	4140.0	4380.0	1795.1	1800.0	0.0	2.2	0.0	1800.0	0.0	65.0
E-Waved-40-45-SSTR2	3769.7	3960.0	1.4	3780.0	3960.0	1795.1	1800.0	4.5	3.6	4.5	1800.0	0.0	99.4
E-Waved-40-45-SSTR3	3718.3	3780.0	1.2	3720.0	3780.0	1795.6	1800.0	1.6	3.2	0.0	1800.0	0.0	69.1
E-Waved-40-45-SSTR4	4080.0	4320.0	2.2	4080.0	4320.0	1795.7	1800.0	11.1	2.2	0.0	1800.0	0.0	62.8
E-Waved-40-45-SSTR5	3156.6	3360.0	846.4	3180.0	3360.0	950.2	1800.0	8.9	3.3	7.1	1800.0	0.0	360.0
E-Waved-40-45-SSTR6	2464.5	2580.0	45.8	2520.0	2520.0	1757.1	1806.0	4.8	3.2	4.8	1800.0	0.0	173.9
E-Waved-40-75-SSTR1	5580.0	5880.0	0.7	5580.0	5880.0	1797.1	1800.0	0.0	2.2	0.0	1800.0	0.0	30.1
E-Waved-40-75-SSTR2	5046.0	5220.0	0.6	5100.0	5220.0	1796.7	1800.0	3.4	2.7	3.4	1800.0	0.0	31.5
E-Waved-40-75-SSTR3	5036.0	5100.0	4.5	5040.0	5100.0	1792.9	1800.0	2.4	2.7	2.4	1800.0	0.0	32.8
E-Waved-40-75-SSTR4	5460.0	5760.0	1.2	5460.0	5760.0	1796.9	1800.0	17.7	1.9	8.3	1800.0	0.0	29.9
E-Waved-40-75-SSTR5	4530.0	4620.0	1.3	4560.0	4620.0	1795.9	1800.0	9.1	2.9	9.1	1800.0	0.0	37.2
E-Waved-40-75-SSTR6	4260.0	4260.0	4.7	4260.0	4260.0	3.0	10.8	7.0	3.0	7.0	1800.0	0.0	34.2
E-Waved-40-105-SSTR1	5460.0	5460.0	0.1	5460.0	5460.0	2.2	4.5	7.7	2.2	7.7	1800.0	0.0	42.4
E-Waved-40-105-SSTR2	4980.0	4980.0	0.2	4980.0	4980.0	2.1	4.4	8.4	2.1	4.8	1800.0	0.0	37.7
E-Waved-40-105-SSTR3	4860.0	4860.0	0.2	4860.0	4860.0	2.8	5.8	3.7	2.8	3.7	1800.0	0.0	37.2
E-Waved-40-105-SSTR4	4800.0	4800.0	0.9	4800.0	4800.0	2.3	5.4	22.5	2.2	10.0	1800.0	0.0	40.1
E-Waved-40-105-SSTR5	3850.9	3900.0	349.6	3900.0	3900.0	2.6	354.7	21.5	2.5	7.7	1800.0	0.0	360.0
E-Waved-40-105-SSTR6	3448.2	3480.0	63.2	3480.0	3480.0	3.6	70.4	15.5	3.6	15.5	1800.0	0.0	273.4
E-Waveless-40-45-SSTR1	2920.0	2940.0	92.3	2940.0	2940.0	2.0	96.3	14.3	2.0	14.3	1800.0	0.0	235.0
E-Waveless-40-45-SSTR2	2352.3	2520.0	62.5	2400.0	2400.0	1743.2	1807.9	10.0	2.2	5.0	1800.0	5.0	140.0
E-Waveless-40-45-SSTR3	2101.8	2160.0	14.5	2160.0	2160.0	2.8	20.1	5.6	2.8	5.6	1800.0	0.0	113.9
E-Waveless-40-45-SSTR4	2720.0	2940.0	392.0	2760.0	2940.0	1406.0	1800.0	14.3	1.9	14.3	1800.0	0.0	360.0
E-Waveless-40-45-SSTR5	2349.9	2460.0	104.1	2400.0	2400.0	1700.6	1807.1	32.5	2.5	15.0	1800.0	5.0	357.6
E-Waveless-40-45-SSTR6	2111.8	2160.0	7.2	2160.0	2160.0	3.0	13.2	13.9	3.0	5.6	1800.0	0.0	123.6
E-Waveless-40-75-SSTR1	5070.0	5340.0	0.8	5100.0	5340.0	1797.5	1800.0	18.0	1.7	9.0	1800.0	0.0	28.8
E-Waveless-40-75-SSTR2	4121.5	4260.0	82.8	4140.0	4260.0	1714.9	1800.0	21.1	2.4	19.7	1800.0	0.0	194.9
E-Waveless-40-75-SSTR3	3983.4	4020.0	10.1	4020.0	4020.0	1.9	13.9	16.4	1.9	9.0	1800.0	0.0	53.1
E-Waveless-40-75-SSTR4	4294.5	4440.0	327.9	4320.0	4380.0	1470.3	1800.0	23.3	1.9	23.3	1800.0	1.4	360.0
E-Waveless-40-75-SSTR5	4056.1	4140.0	231.3	4080.0	4080.0	1446.5	1680.4	19.1	2.7	14.7	1800.0	1.5	351.1
E-Waveless-40-75-SSTR6	3936.5	4020.0	15.5	3960.0	3960.0	152.3	170.6	22.7	2.8	6.1	1800.0	0.0	51.3
E-Waveless-40-105-SSTR1	5280.0	5280.0	0.2	5280.0	5280.0	1.6	3.3	0.0	1.6	0.0	1800.0	0.0	34.3
E-Waveless-40-105-SSTR2	3960.0	3960.0	1.5	3960.0	3960.0	1.7	4.8	6.1	1.7	1.5	1800.0	0.0	34.8
E-Waveless-40-105-SSTR3	3402.0	3540.0	2.8	3480.0	3480.0	519.3	523.8	6.9	1.7	6.9	1800.0	0.0	35.1
E-Waveless-40-105-SSTR4	4800.0	4800.0	0.6	4800.0	4800.0	2.3	5.3	10.0	2.3	10.0	1800.0	0.0	40.7
E-Waveless-40-105-SSTR5	3900.0	3900.0	25.4	3900.0	3900.0	1.9	29.1	10.8	1.8	7.7	1800.0	0.0	82.4
E-Waveless-40-105-SSTR6	3400.9	3480.0	3.2	3420.0	3480.0	1794.8	1800.0	6.9	2.0	3.4	1800.0	0.0	48.3

Table 3.8: Results for Instances with 40 batches, 10 truck departures per staging lane

	Branch & Price							**S**		**GUR**		**MH**	
	Linear Relaxation			Integer									
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{GUR}$	CPU^{GUR}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-40-45-SSTR1	4800.0	4800.0	0.6	4800.0	4800.0	1.8	4.2	10.0	1.8	10.0	1800.0	0.0	30.5
U-Waved-40-45-SSTR2	4140.0	4140.0	0.3	4140.0	4140.0	2.1	4.5	8.7	2.1	1.4	1800.0	0.0	32.4
U-Waved-40-45-SSTR3	3660.0	3660.0	0.2	3660.0	3660.0	2.0	4.3	9.8	2.0	9.8	1800.0	0.0	33.1
U-Waved-40-45-SSTR4	2880.0	2880.0	16.4	2880.0	2880.0	2.0	20.3	50.0	2.0	33.3	1800.0	0.0	43.1
U-Waved-40-45-SSTR5	2700.0	3000.0	6.0	2760.0	2820.0	1791.8	1800.0	6.4	2.2	6.4	1800.0	6.4	61.7
U-Waved-40-45-SSTR6	2640.0	2820.0	1.3	2700.0	2760.0	1796.3	1800.0	2.2	2.4	2.2	1800.0	2.2	43.2
U-Waved-40-75-SSTR1	6240.0	6240.0	0.2	6240.0	6240.0	1.7	3.6	8.7	1.7	0.0	1800.0	0.0	22.9
U-Waved-40-75-SSTR2	5400.0	5400.0	0.2	5400.0	5400.0	1.7	3.5	6.7	1.7	1.1	1800.0	0.0	26.1
U-Waved-40-75-SSTR3	4920.0	4920.0	0.1	4920.0	4920.0	1.7	3.6	8.5	1.7	1.2	1800.0	0.0	23.8
U-Waved-40-75-SSTR4	4320.0	4320.0	0.4	4320.0	4320.0	1.8	3.9	11.1	1.7	0.0	1800.0	0.0	27.3
U-Waved-40-75-SSTR5	3780.0	3780.0	2.5	3780.0	3780.0	1.9	6.1	17.5	1.8	3.2	1800.0	0.0	32.2
U-Waved-40-75-SSTR6	3630.0	3660.0	1.0	3660.0	3660.0	2.0	4.9	11.5	2.0	6.6	1800.0	1.6	30.5
U-Waved-40-105-SSTR1	7680.0	7680.0	0.1	7680.0	7680.0	1.5	3.2	6.3	1.5	0.0	1800.0	0.0	21.3
U-Waved-40-105-SSTR2	6960.0	6960.0	0.1	6960.0	6960.0	1.5	3.2	8.6	1.5	4.3	1800.0	0.0	22.4
U-Waved-40-105-SSTR3	6480.0	6480.0	0.2	6480.0	6480.0	1.6	3.4	0.9	1.6	0.0	1800.0	0.0	23.1
U-Waved-40-105-SSTR4	5760.0	5760.0	0.5	5760.0	5760.0	1.5	3.6	26.0	1.5	0.0	1800.0	0.0	24.2
U-Waved-40-105-SSTR5	5220.0	5220.0	0.6	5220.0	5220.0	1.7	3.9	26.4	1.6	14.9	1800.0	0.0	27.2
U-Waved-40-105-SSTR6	5040.0	5040.0	0.3	5040.0	5040.0	1.9	4.0	0.0	1.9	0.0	1800.0	0.0	27.4
E-Waved-40-45-SSTR1	4000.0	4320.0	0.6	4020.0	4320.0	1797.3	1800.0	0.0	2.1	0.0	1800.0	0.0	43.1
E-Waved-40-45-SSTR2	3410.0	3600.0	0.6	3420.0	3600.0	1797.3	1800.0	0.0	2.1	0.0	1800.0	0.0	40.4
E-Waved-40-45-SSTR3	3110.0	3300.0	0.7	3120.0	3300.0	1797.0	1800.0	7.3	2.3	0.0	1800.0	0.0	40.9
E-Waved-40-45-SSTR4	2800.0	2940.0	3.3	2820.0	2940.0	1794.7	1800.0	16.3	2.1	0.0	1800.0	0.0	65.9
E-Waved-40-45-SSTR5	2600.0	2640.0	6.4	2640.0	2640.0	2.0	10.3	13.6	2.0	9.1	1800.0	0.0	66.0
E-Waved-40-45-SSTR6	2580.0	2640.0	4.7	2580.0	2640.0	1793.1	1800.0	6.8	2.3	4.5	1800.0	0.0	71.6
E-Waved-40-75-SSTR1	4680.0	5280.0	0.7	4800.0	5280.0	1797.6	1800.0	0.0	1.6	0.0	1800.0	0.0	33.7
E-Waved-40-75-SSTR2	3921.4	4260.0	0.9	3960.0	4260.0	1797.4	1800.0	18.3	1.7	9.9	1800.0	1.4	36.3
E-Waved-40-75-SSTR3	3441.4	3720.0	0.6	3480.0	3720.0	1797.5	1800.0	9.7	1.9	1.6	1800.0	0.0	40.7
E-Waved-40-75-SSTR4	3480.0	3840.0	5.7	3540.0	3840.0	1792.3	1800.0	12.5	2.0	0.0	1800.0	0.0	50.1
E-Waved-40-75-SSTR5	3180.0	3300.0	329.2	3240.0	3300.0	1468.8	1800.0	30.9	2.0	12.7	1800.0	0.0	360.0
E-Waved-40-75-SSTR6	2945.0	3060.0	26.2	3000.0	3060.0	1771.7	1800.0	13.7	2.1	11.8	1800.0	0.0	114.8
E-Waved-40-105-SSTR1	7260.0	7260.0	0.6	7260.0	7260.0	1.3	3.2	19.0	1.3	6.6	1800.0	0.0	22.6
E-Waved-40-105-SSTR2	6180.0	6180.0	11.8	6180.0	6180.0	1.3	14.4	16.5	1.3	6.8	1800.0	0.0	60.5
E-Waved-40-105-SSTR3	5760.0	5760.0	2.5	5760.0	5760.0	1.4	5.3	15.6	1.4	14.6	1800.0	0.0	34.3
E-Waved-40-105-SSTR4	6720.0	6720.0	0.1	6720.0	6720.0	1.3	2.8	7.1	1.3	0.0	1800.0	0.0	22.4
E-Waved-40-105-SSTR5	5760.0	5760.0	6.4	5760.0	5760.0	1.4	9.2	11.5	1.4	11.5	1800.0	0.0	36.1
E-Waved-40-105-SSTR6	5520.0	5520.0	0.5	5520.0	5520.0	1.6	3.7	7.6	1.6	7.6	1800.0	0.0	27.4

Table 3.9: Results for Instances with 80 batches, 5 truck departures per staging lane, uniformly distributed processing time

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-80-45-SSTR1		9,120.0	1,800.0				1,800.0	5.3	8.1	0.0	360.0
U-Waved-80-45-SSTR2		8,520.0	1,800.0				1,800.0	6.3	12.1	-0.7	360.0
U-Waved-80-45-SSTR3	8,143.0	8,220.0	994.3	8,160.0	8,220.0	795.8	1,800.0	2.9	9.9	0.0	360.0
U-Waved-80-45-SSTR4		9,120.0	1,800.0				1,800.0	10.5	7.7	0.0	360.0
U-Waved-80-45-SSTR5		7,020.0	1,800.0				1,800.0	12.0	10.1	0.0	360.0
U-Waved-80-45-SSTR6		5,280.0	1,800.0				1,800.0	9.1	10.0	0.0	360.0
U-Waved-80-75-SSTR1	10,020.0	10,020.0	26.1	10,020.0	10,020.0	6.8	39.7	17.4	6.8	0.0	95.0
U-Waved-80-75-SSTR2	9,540.0	9,540.0	25.2	9,540.0	9,540.0	8.9	42.8	17.6	8.8	0.0	148.7
U-Waved-80-75-SSTR3	9,400.0	9,420.0	14.3	9,420.0	9,420.0	8.4	31.0	3.8	8.4	0.0	71.5
U-Waved-80-75-SSTR4	9,600.0	9,600.0	21.8	9,600.0	9,600.0	7.3	36.3	15.0	7.3	0.0	102.9
U-Waved-80-75-SSTR5	7,650.0	7,680.0	488.1	7,680.0	7,680.0	8.0	504.1	14.1	7.9	0.0	360.0
U-Waved-80-75-SSTR6	6,759.0	6,780.0	321.3	6,780.0	6,780.0	8.1	337.4	14.2	8.0	0.0	265.3
U-Waved-80-105-SSTR1	11,200.0	11,520.0	11.3	11,220.0	11,520.0	1,782.2	1,800.0	13.0	6.5	0.0	43.6
U-Waved-80-105-SSTR2	10,560.0	10,800.0	55.4	10,560.0	10,800.0	1,736.8	1,800.0	15.6	7.8	0.0	72.5
U-Waved-80-105-SSTR3	10,280.0	10,320.0	4.8	10,320.0	10,320.0	8.1	21.0	2.3	8.1	0.0	48.2
U-Waved-80-105-SSTR4	10,880.0	11,040.0	119.1	10,920.0	11,040.0	1,673.5	1,800.0	35.3	7.3	0.0	358.3
U-Waved-80-105-SSTR5	9,963.5	10,020.0	982.5	10,020.0	10,020.0	8.8	1,000.0	21.0	8.7	0.0	360.0
U-Waved-80-105-SSTR6	9,628.0	9,660.0	58.7	9,660.0	9,660.0	9.6	78.0	9.9	9.6	0.0	118.8
U-Waveless-80-45-SSTR1	5,760.0	5,760.0	2.9	5,760.0	5,760.0	6.7	16.2	9.4	6.7	0.0	71.3
U-Waveless-80-45-SSTR2	4,620.0	4,680.0	78.9	4,620.0	4,680.0	1,713.7	1,800.0	15.4	7.3	-1.3	85.9
U-Waveless-80-45-SSTR3		4,320.0	1,800.0				1,800.0	6.9	7.7	1.4	360.0
U-Waveless-80-45-SSTR4		4,980.0	1,800.0				1,800.0	39.8	7.6	-1.2	360.0
U-Waveless-80-45-SSTR5		4,620.0	1,800.0				1,800.0	16.9	8.0	-1.3	360.0
U-Waveless-80-45-SSTR6	3,915.3	4,020.0	1,628.4	3,960.0	4,020.0	161.7	1,800.0	19.4	9.8	4.5	360.0
U-Waveless-80-75-SSTR1		7,440.0	1,800.0				1,800.0	36.3	6.5	-0.8	360.0
U-Waveless-80-75-SSTR2		6,900.0	1,800.0				1,800.0	19.1	7.0	0.0	360.0
U-Waveless-80-75-SSTR3		6,720.0	1,800.0				1,800.0	17.0	7.0	0.9	360.0
U-Waveless-80-75-SSTR4		7,140.0	1,800.0				1,800.0	37.0	8.7	0.8	360.0
U-Waveless-80-75-SSTR5		6,900.0	1,800.0				1,800.0	10.4	7.4	0.0	360.0
U-Waveless-80-75-SSTR6		6,480.0	1,800.0				1,800.0	12.0	8.7	0.0	360.0
U-Waveless-80-105-SSTR1		9,840.0	1,800.0				1,800.0	26.8	6.7	0.0	360.0
U-Waveless-80-105-SSTR2	9,202.7	9,240.0	1,177.1	9,240.0	9,240.0	6.4	1,189.8	27.9	6.4	0.0	360.0
U-Waveless-80-105-SSTR3	9,030.7	9,120.0	397.5	9,060.0	9,120.0	1,395.5	1,800.0	12.5	7.1	0.0	360.0
U-Waveless-80-105-SSTR4		9,540.0	1,800.0				1,800.0	28.3	7.0	0.0	360.0
U-Waveless-80-105-SSTR5		9,180.0	1,800.0				1,800.0	19.0	7.7	0.0	360.0
U-Waveless-80-105-SSTR6	8,684.4	8,700.0	62.3	8,700.0	8,700.0	9.2	80.6	11.0	9.1	0.0	191.4

Table 3.10: Results for Instances with 80 batches, 5 truck departures per staging lane, exponentially distributed processing time

	Branch & Price							**S**		**MH**	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
E-Waved-80-45-SSTR1		7,860.0	1,800.0				1,800.0	6.9	8.6	0.0	360.0
E-Waved-80-45-SSTR2		7,200.0	1,800.0				1,800.0	8.3	13.4	0.0	360.0
E-Waved-80-45-SSTR3	6,807.0	6,900.0	1,772.4	6,840.0	6,900.0	16.4	1,800.0	4.3	11.2	0.0	360.0
E-Waved-80-45-SSTR4		7,680.0	1,800.0				1,800.0	19.5	8.4	0.0	360.0
E-Waved-80-45-SSTR5		5,940.0	1,800.0				1,800.0	11.1	10.8	-1.0	360.0
E-Waved-80-45-SSTR6		4,620.0	1,800.0				1,800.0	13.0	13.0	0.0	360.0
E-Waved-80-75-SSTR1	12,131.3	12,180.0	31.6	12,180.0	12,180.0	6.5	44.7	3.9	6.5	0.0	88.5
E-Waved-80-75-SSTR2	10,487.9	10,740.0	20.7	10,500.0	10,740.0	1,771.8	1,800.0	3.9	7.5	0.0	78.9
E-Waved-80-75-SSTR3	10,030.0	10,080.0	8.6	10,080.0	10,080.0	8.7	26.0	2.4	8.7	0.0	53.4
E-Waved-80-75-SSTR4		9,660.0	1,800.0				1,800.0	26.1	7.1	0.0	166.0
E-Waved-80-75-SSTR5		8,520.0	1,800.0				1,800.0	19.0	7.9	0.0	360.0
E-Waved-80-75-SSTR6		8,160.0	1,800.0				1,800.0	5.9	8.5	0.7	360.0
E-Waved-80-105-SSTR1		12,600.0	1,800.0				1,800.0	2.9	7.5	0.0	360.0
E-Waved-80-105-SSTR2		10,740.0	1,800.0				1,800.0	1.7	8.1	0.6	360.0
E-Waved-80-105-SSTR3		10,020.0	1,800.0				1,800.0	0.0	9.0	0.0	360.0
E-Waved-80-105-SSTR4		9,600.0	1,800.0				1,800.0	30.6	7.2	0.0	360.0
E-Waved-80-105-SSTR5		8,040.0	1,800.0				1,800.0	17.2	9.0	-0.8	360.0
E-Waved-80-105-SSTR6		7,740.0	1,800.0				1,800.0	11.6	10.3	0.8	360.0
E-Waveless-80-45-SSTR1		4,800.0	1,800.0				1,800.0	30.0	6.9	0.0	360.0
E-Waveless-80-45-SSTR2		4,440.0	1,800.0				1,800.0	10.8	7.8	-4.2	360.0
E-Waveless-80-45-SSTR3		4,200.0	1,800.0				1,800.0	5.7	8.3	-1.4	360.0
E-Waveless-80-45-SSTR4		4,800.0	1,800.0				1,800.0	30.0	8.3	-5.3	360.0
E-Waveless-80-45-SSTR5		4,440.0	1,800.0				1,800.0	13.5	9.5	0.0	360.0
E-Waveless-80-45-SSTR6		4,020.0	1,800.0				1,800.0	10.4	12.4	0.0	360.0
E-Waveless-80-75-SSTR1	8,616.2	8,880.0	1,577.7	8,640.0	8,880.0	216.5	1,800.0	18.9	5.8	-0.7	360.0
E-Waveless-80-75-SSTR2		8,100.0	1,800.0				1,800.0	11.1	6.6	1.5	360.0
E-Waveless-80-75-SSTR3		7,800.0	1,800.0				1,800.0	6.2	7.1	0.0	360.0
E-Waveless-80-75-SSTR4		8,280.0	1,800.0				1,800.0	17.4	7.3	0.0	360.0
E-Waveless-80-75-SSTR5		8,100.0	1,800.0				1,800.0	11.1	7.2	0.7	360.0
E-Waveless-80-75-SSTR6		7,620.0	1,800.0				1,800.0	11.8	9.3	0.0	360.0
E-Waveless-80-105-SSTR1		8,460.0	1,800.0				1,800.0	14.2	6.0	0.0	360.0
E-Waveless-80-105-SSTR2	7,800.0	7,860.0	1,375.1	7,800.0	7,860.0	418.2	1,800.0	14.5	6.7	0.8	360.0
E-Waveless-80-105-SSTR3		7,560.0	1,800.0				1,800.0	12.7	8.8	1.6	360.0
E-Waveless-80-105-SSTR4	8,040.0	8,220.0	53.8	8,040.0	8,160.0	1,738.7	1,800.0	25.7	7.5	0.7	109.6
E-Waveless-80-105-SSTR5	7,800.0	7,860.0	1,273.0	7,800.0	7,860.0	519.0	1,800.0	9.9	8.1	1.5	360.0
E-Waveless-80-105-SSTR6		7,440.0	1,800.0				1,800.0	11.3	9.5	0.0	360.0

Table 3.11: Results for Instances with 80 batches, 10 truck departures per staging lane, uniformly distributed processing time

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-80-45-SSTR1	9,600.0	9,600.0	2.1	9,600.0	9,600.0	7.1	16.2	5.0	7.0	0.0	43.0
U-Waved-80-45-SSTR2	8,280.0	8,280.0	1.7	8,280.0	8,280.0	8.4	18.5	9.4	8.4	0.0	42.3
U-Waved-80-45-SSTR3	7,320.0	7,320.0	1.6	7,320.0	7,320.0	7.8	17.2	10.7	7.8	0.0	45.3
U-Waved-80-45-SSTR4	5,760.0	5,760.0	201.4	5,760.0	5,760.0	7.5	216.4	33.3	7.4	1.0	118.0
U-Waved-80-45-SSTR5	5,505.0	5,760.0	974.9	5,520.0	5,760.0	816.9	1,800.0	16.7	8.3	0.0	360.0
U-Waved-80-45-SSTR6	5,314.3	5,460.0	72.2	5,340.0	5,460.0	1,718.7	1,800.0	5.5	9.0	0.0	203.1
U-Waved-80-75-SSTR1	11,880.0	11,880.0	43.2	11,880.0	11,880.0	5.7	54.5	9.6	5.6	0.0	45.9
U-Waved-80-75-SSTR2	10,320.0	10,320.0	63.6	10,320.0	10,320.0	6.5	76.6	12.8	6.5	0.0	93.9
U-Waved-80-75-SSTR3	9,360.0	9,360.0	941.4	9,360.0	9,360.0	6.1	953.5	10.9	6.0	0.0	341.7
U-Waved-80-75-SSTR4	7,740.0	7,740.0	48.4	7,740.0	7,740.0	6.5	61.4	43.4	6.4	0.0	145.8
U-Waved-80-75-SSTR5	7,440.0	7,440.0	164.5	7,440.0	7,440.0	6.9	178.2	23.4	6.8	0.8	148.3
U-Waved-80-75-SSTR6	7,120.0	7,200.0	30.4	7,140.0	7,200.0	1,762.0	1,800.0	14.2	7.6	-0.8	76.4
U-Waved-80-105-SSTR1	15,360.0	15,360.0	0.9	15,360.0	15,360.0	5.0	10.8	10.2	5.0	0.0	28.8
U-Waved-80-105-SSTR2	13,860.0	13,860.0	0.9	13,860.0	13,860.0	5.6	12.1	5.2	5.6	0.0	30.1
U-Waved-80-105-SSTR3	12,900.0	12,900.0	0.7	12,900.0	12,900.0	5.7	12.1	0.9	5.7	0.0	30.0
U-Waved-80-105-SSTR4	11,520.0	11,580.0	2.5	11,520.0	11,580.0	1,791.9	1,800.0	25.9	5.5	-0.5	31.3
U-Waved-80-105-SSTR5	10,275.0	10,320.0	6.2	10,320.0	10,320.0	6.1	18.3	16.9	6.0	0.0	41.9
U-Waved-80-105-SSTR6	9,960.0	9,960.0	2.3	9,960.0	9,960.0	6.8	15.9	1.2	6.8	0.6	40.9
U-Waveless-80-45-SSTR1	6,240.0	6,240.0	3.5	6,240.0	6,240.0	6.1	15.6	7.7	6.1	0.0	54.4
U-Waveless-80-45-SSTR2	5,400.0	5,400.0	5.4	5,400.0	5,400.0	6.6	18.4	20.0	6.5	0.0	56.9
U-Waveless-80-45-SSTR3		5,280.0	1,800.0				1,800.0	4.5	6.8	0.0	360.0
U-Waveless-80-45-SSTR4	5,280.0	5,460.0	963.0	5,280.0	5,460.0	830.6	1,800.0	31.9	6.4	-2.2	179.9
U-Waveless-80-45-SSTR5		4,800.0	1,800.0				1,800.0	13.8	6.8	0.0	360.0
U-Waveless-80-45-SSTR6	4,513.3	4,800.0	160.8	4,560.0	4,800.0	1,630.9	1,800.0	7.5	8.3	-2.5	349.7
U-Waveless-80-75-SSTR1	8,800.0	9,120.0	274.7	8,820.0	9,120.0	1,520.1	1,800.0	15.8	5.2	0.0	360.0
U-Waveless-80-75-SSTR2		8,040.0	1,800.0				1,800.0	20.9	5.7	0.7	360.0
U-Waveless-80-75-SSTR3		7,980.0	1,800.0				1,800.0	11.3	6.4	0.0	360.0
U-Waveless-80-75-SSTR4		7,800.0	1,800.0				1,800.0	36.2	6.2	0.8	360.0
U-Waveless-80-75-SSTR5		7,020.0	1,800.0				1,800.0	19.7	6.0	0.8	360.0
U-Waveless-80-75-SSTR6	6,775.6	6,900.0	86.1	6,780.0	6,900.0	1,707.2	1,800.0	15.7	6.7	0.0	194.1
U-Waveless-80-105-SSTR1	11,026.7	11,160.0	3.1	11,040.0	11,160.0	1,791.9	1,800.0	30.1	5.0	0.0	34.1
U-Waveless-80-105-SSTR2	10,110.0	10,140.0	19.9	10,140.0	10,140.0	5.3	30.5	19.5	5.3	0.0	58.7
U-Waveless-80-105-SSTR3	10,035.0	10,080.0	17.1	10,080.0	10,080.0	5.4	27.8	11.9	5.4	0.0	66.7
U-Waveless-80-105-SSTR4	10,145.8	10,200.0	909.2	10,200.0	10,200.0	5.7	920.5	39.4	5.6	0.0	360.0
U-Waveless-80-105-SSTR5	9,564.0	9,600.0	395.9	9,600.0	9,600.0	6.0	407.8	21.9	6.0	0.0	360.0
U-Waveless-80-105-SSTR6	9,269.0	9,360.0	9.0	9,300.0	9,360.0	1,784.5	1,800.0	10.9	6.5	0.0	48.5

Table 3.12: Results for Instances with 80 batches, 10 truck departures per staging lane, exponentially distributed processing time

	Branch & Price							**S**		**MH**	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
E-Waved-80-45-SSTR1	7,356.0	7,680.0	321.7	7,380.0	7,680.0	1,471.7	1,800.0	7.0	6.6	0.0	85.0
E-Waved-80-45-SSTR2	6,354.0	6,660.0	678.2	6,360.0	6,660.0	1,114.2	1,800.0	12.6	7.6	0.9	360.0
E-Waved-80-45-SSTR3	5,814.0	6,060.0	1,091.9	5,820.0	6,060.0	700.7	1,800.0	13.9	7.4	1.0	199.8
E-Waved-80-45-SSTR4		5,580.0	1,800.0				1,800.0	32.3	7.0	4.1	360.0
E-Waved-80-45-SSTR5		5,460.0	1,800.0				1,800.0	15.4	8.1	0.0	360.0
E-Waved-80-45-SSTR6		5,160.0	1,800.0				1,800.0	18.6	8.8	1.1	360.0
E-Waved-80-75-SSTR1	10,140.0	10,140.0	3.1	10,140.0	10,140.0	6.2	15.5	13.6	6.2	0.0	47.3
E-Waved-80-75-SSTR2	8,610.0	8,640.0	10.0	8,640.0	8,640.0	6.7	23.2	17.4	6.6	0.0	360.0
E-Waved-80-75-SSTR3	7,830.0	7,860.0	2.2	7,860.0	7,860.0	6.5	15.2	5.3	6.5	0.0	52.6
E-Waved-80-75-SSTR4		6,960.0	1,800.0				1,800.0	37.9	6.5	0.0	84.3
E-Waved-80-75-SSTR5		6,600.0	1,800.0				1,800.0	29.1	6.7	1.8	360.0
E-Waved-80-75-SSTR6		6,480.0	1,800.0				1,800.0	14.8	7.6	-0.9	360.0
E-Waved-80-105-SSTR1	15,900.0	15,900.0	0.7	15,900.0	15,900.0	4.4	9.4	3.4	4.3	0.0	25.4
E-Waved-80-105-SSTR2	13,380.0	13,380.0	1.5	13,380.0	13,380.0	5.0	11.5	8.1	5.0	0.0	28.1
E-Waved-80-105-SSTR3	12,480.0	12,480.0	1.3	12,480.0	12,480.0	5.2	11.7	3.8	5.1	0.0	30.1
E-Waved-80-105-SSTR4	12,480.0	12,480.0	4.3	12,480.0	12,480.0	4.9	14.0	15.4	4.8	0.0	38.1
E-Waved-80-105-SSTR5	11,328.0	11,340.0	432.8	11,340.0	11,340.0	5.5	443.7	14.8	5.4	0.0	360.0
E-Waved-80-105-SSTR6	11,218.2	11,280.0	314.9	11,220.0	11,280.0	1,478.6	1,800.0	5.9	6.5	-0.5	206.6
E-Waveless-80-45-SSTR1	6,720.0	6,720.0	526.7	6,720.0	6,720.0	6.7	540.0	14.3	6.6	0.0	81.0
E-Waveless-80-45-SSTR2	5,660.0	5,700.0	953.3	5,700.0	5,700.0	6.2	965.6	11.6	6.1	0.0	118.7
E-Waveless-80-45-SSTR3		5,160.0	1,800.0				1,800.0	14.0	6.7	1.1	360.0
E-Waveless-80-45-SSTR4	6,080.0	6,240.0	15.9	6,120.0	6,240.0	1,777.7	1,800.0	8.7	6.4	0.0	151.3
E-Waveless-80-45-SSTR5		5,340.0	1,800.0				1,800.0	18.0	7.2	0.0	360.0
E-Waveless-80-45-SSTR6		4,980.0	1,800.0				1,800.0	18.1	7.7	0.0	360.0
E-Waveless-80-75-SSTR1		7,680.0	1,800.0				1,800.0	37.5	6.3	0.0	360.0
E-Waveless-80-75-SSTR2		6,660.0	1,800.0				1,800.0	34.2	6.2	1.8	360.0
E-Waveless-80-75-SSTR3		6,540.0	1,800.0				1,800.0	14.7	6.2	0.0	360.0
E-Waveless-80-75-SSTR4		7,200.0	1,800.0				1,800.0	40.0	6.3	0.8	360.0
E-Waveless-80-75-SSTR5		6,360.0	1,800.0				1,800.0	21.7	6.4	0.9	360.0
E-Waveless-80-75-SSTR6		6,180.0	1,800.0				1,800.0	20.4	7.2	0.0	360.0
E-Waveless-80-105-SSTR1	12,930.0	13,140.0	1.1	12,960.0	13,140.0	1,794.7	1,800.0	6.4	4.2	0.0	28.0
E-Waveless-80-105-SSTR2	11,030.0	11,160.0	3.6	11,040.0	11,160.0	1,791.7	1,800.0	8.1	4.7	0.0	36.8
E-Waveless-80-105-SSTR3	10,537.8	10,680.0	626.9	10,560.0	10,680.0	1,168.0	1,800.0	9.0	5.1	0.0	360.0
E-Waveless-80-105-SSTR4	12,060.0	12,060.0	5.2	12,060.0	12,060.0	4.9	14.9	20.4	4.8	0.0	39.6
E-Waveless-80-105-SSTR5		10,680.0	1,800.0				1,800.0	8.4	5.2	0.0	360.0
E-Waveless-80-105-SSTR6	10,342.4	10,380.0	226.7	10,380.0	10,380.0	5.8	238.3	7.5	5.8	0.0	360.0

Table 3.13: Results for Instances with 80 batches, 20 truck departures per staging lane

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-80-45-SSTR1	8,160.0	8,160.0	1.2	8,160.0	8,160.0	4.9	10.9	5.9	4.9	0.0	38.9
U-Waved-80-45-SSTR2	7,500.0	7,500.0	0.4	7,500.0	7,500.0	5.6	11.6	0.0	5.6	0.0	41.1
U-Waved-80-45-SSTR3	7,140.0	7,140.0	1.0	7,140.0	7,140.0	5.3	11.6	3.4	5.3	0.0	39.3
U-Waved-80-45-SSTR4	6,720.0	6,720.0	12.0	6,720.0	6,720.0	5.8	23.5	28.6	5.7	0.0	43.7
U-Waved-80-45-SSTR5	6,160.0	6,240.0	10.4	6,180.0	6,240.0	1,783.7	1,800.0	16.3	5.9	0.0	64.8
U-Waved-80-45-SSTR6	6,020.0	6,180.0	1.5	6,060.0	6,180.0	1,792.4	1,800.0	1.0	6.2	-1.0	41.4
U-Waved-80-75-SSTR1	10,320.0	10,320.0	1.6	10,320.0	10,320.0	4.3	10.1	12.2	4.2	0.0	26.0
U-Waved-80-75-SSTR2	9,600.0	9,600.0	1.0	9,600.0	9,600.0	4.6	10.2	10.6	4.6	0.0	30.3
U-Waved-80-75-SSTR3	9,120.0	9,120.0	1.4	9,120.0	9,120.0	4.5	10.4	7.2	4.5	0.0	28.8
U-Waved-80-75-SSTR4	8,160.0	8,220.0	87.6	8,160.0	8,220.0	1,707.8	1,800.0	35.8	4.6	0.0	204.4
U-Waved-80-75-SSTR5		7,740.0	1,800.0				1,800.0	21.7	5.0	1.5	360.0
U-Waved-80-75-SSTR6	7,625.0	7,680.0	20.0	7,680.0	7,680.0	5.5	30.9	16.4	5.4	0.0	92.1
U-Waved-80-105-SSTR1	13,440.0	13,440.0	0.4	13,440.0	13,440.0	3.8	7.9	8.0	3.8	0.0	25.8
U-Waved-80-105-SSTR2	12,240.0	12,240.0	0.9	12,240.0	12,240.0	4.3	9.5	13.7	4.3	0.0	27.8
U-Waved-80-105-SSTR3	12,240.0	12,240.0	0.3	12,240.0	12,240.0	4.3	8.9	4.4	4.3	0.0	26.1
U-Waved-80-105-SSTR4	11,760.0	11,760.0	1.3	11,760.0	11,760.0	4.2	9.6	28.6	4.1	0.0	28.3
U-Waved-80-105-SSTR5	10,725.0	10,740.0	3.8	10,740.0	10,740.0	4.5	12.8	12.3	4.5	0.0	39.1
U-Waved-80-105-SSTR6	10,260.0	10,260.0	2.3	10,260.0	10,260.0	5.1	12.5	7.0	5.1	0.0	35.0
E-Waved-80-45-SSTR1	6,240.0	6,240.0	4.9	6,240.0	6,240.0	5.1	15.0	23.1	5.0	0.0	49.9
E-Waved-80-45-SSTR2	5,700.0	5,700.0	3.7	5,700.0	5,700.0	5.6	14.9	6.3	5.6	0.0	58.2
E-Waved-80-45-SSTR3	5,400.0	5,400.0	2.8	5,400.0	5,400.0	5.4	13.5	18.9	5.3	0.0	59.3
E-Waved-80-45-SSTR4	5,208.0	5,340.0	47.4	5,220.0	5,340.0	1,746.9	1,800.0	34.8	5.7	0.0	168.8
E-Waved-80-45-SSTR5	5,025.0	5,160.0	239.8	5,040.0	5,160.0	1,554.2	1,800.0	14.0	6.1	0.0	360.0
E-Waved-80-45-SSTR6	4,980.0	4,980.0	223.7	4,980.0	4,980.0	6.7	237.1	15.7	6.7	0.0	360.0
E-Waved-80-75-SSTR1	9,840.0	10,080.0	4.0	9,840.0	10,080.0	1,791.9	1,800.0	14.3	4.1	0.0	32.1
E-Waved-80-75-SSTR2	8,890.0	9,120.0	36.9	8,940.0	9,120.0	1,758.8	1,800.0	4.6	4.3	0.0	55.6
E-Waved-80-75-SSTR3	8,320.0	8,520.0	40.6	8,340.0	8,520.0	1,755.1	1,800.0	7.0	4.2	0.0	88.5
E-Waved-80-75-SSTR4	9,120.0	9,180.0	5.3	9,120.0	9,180.0	1,790.3	1,800.0	20.9	4.5	-0.7	40.9
E-Waved-80-75-SSTR5		8,400.0	1,800.0				1,800.0	19.3	4.8	0.0	360.0
E-Waved-80-75-SSTR6	8,007.9	8,100.0	631.1	8,040.0	8,100.0	1,163.6	1,800.0	8.9	5.3	0.0	360.0
E-Waved-80-105-SSTR1		9,600.0	1,800.0				1,800.0	15.0	4.1	0.0	360.0
E-Waved-80-105-SSTR2		9,000.0	1,800.0				1,800.0	8.0	4.4	0.0	360.0
E-Waved-80-105-SSTR3		8,880.0	1,800.0				1,800.0	8.1	4.5	0.0	360.0
E-Waved-80-105-SSTR4		8,340.0	1,800.0				1,800.0	43.9	4.6	0.0	360.0
E-Waved-80-105-SSTR5		8,100.0	1,800.0				1,800.0	23.7	4.8	0.0	360.0
E-Waved-80-105-SSTR6	7,900.6	7,920.0	9.4	7,920.0	7,920.0	5.5	20.3	11.4	5.4	0.0	52.8

Table 3.14: Results for Instances with 160 batches, 5 truck departures per staging lane, uniformly distributed processing time

	Branch & Price							**S**		**MH**	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-160-45-SSTR1		17,760.0	1,800.0				1,800.0	0.3	29.3	0.0	360.0
U-Waved-160-45-SSTR2		16,620.0	1,800.0				1,800.0	7.9	48.9	0.0	360.0
U-Waved-160-45-SSTR3		16,380.0	1,800.0				1,800.0	2.9	41.8	-0.7	360.0
U-Waved-160-45-SSTR4		17,760.0	1,800.0				1,800.0	8.8	30.6	0.0	360.0
U-Waved-160-45-SSTR5		13,920.0	1,800.0				1,800.0	7.3	36.3	4.9	360.0
U-Waved-160-45-SSTR6		10,620.0	1,800.0				1,800.0	6.8	41.9	0.6	360.0
U-Waved-160-75-SSTR1		20,220.0	1,800.0				1,800.0	15.7	28.3	0.0	360.0
U-Waved-160-75-SSTR2		19,260.0	1,800.0				1,800.0	17.1	36.7	0.0	360.0
U-Waved-160-75-SSTR3		18,900.0	1,800.0				1,800.0	3.5	33.0	0.0	360.0
U-Waved-160-75-SSTR4		19,200.0	1,800.0				1,800.0	14.4	29.4	0.0	360.0
U-Waved-160-75-SSTR5		15,600.0	1,800.0				1,800.0	11.2	32.1	-1.2	360.0
U-Waved-160-75-SSTR6		13,740.0	1,800.0				1,800.0	7.9	32.7	0.4	360.0
U-Waved-160-105-SSTR1	22,400.0	22,560.0	406.3	22,440.0	22,560.0	1,367.8	1,800.0	23.1	25.9	0.0	360.0
U-Waved-160-105-SSTR2	21,120.0	21,240.0	700.8	21,120.0	21,240.0	1,068.6	1,800.0	22.6	30.6	2.8	360.0
U-Waved-160-105-SSTR3	20,520.0	20,520.0	988.9	20,520.0	20,520.0	31.1	1,051.0	6.4	31.0	0.6	360.0
U-Waved-160-105-SSTR4		21,840.0	1,800.0				1,800.0	30.2	29.2	2.2	360.0
U-Waved-160-105-SSTR5		20,220.0	1,800.0				1,800.0	18.4	32.6	1.2	360.0
U-Waved-160-105-SSTR6		19,320.0	1,800.0				1,800.0	12.4	37.3	1.2	360.0
U-Waveless-160-45-SSTR1		9,240.0	1,800.0				1,800.0	19.5	24.2	3.8	360.0
U-Waveless-160-45-SSTR2		8,400.0	1,800.0				1,800.0	12.9	26.3	4.1	360.0
U-Waveless-160-45-SSTR3		7,680.0	1,800.0				1,800.0	10.2	29.6	3.0	360.0
U-Waveless-160-45-SSTR4		9,240.0	1,800.0				1,800.0	25.3	26.6	1.3	360.0
U-Waveless-160-45-SSTR5		8,280.0	1,800.0				1,800.0	11.6	28.9	1.4	360.0
U-Waveless-160-45-SSTR6		7,620.0	1,800.0				1,800.0	7.9	34.4	3.1	360.0
U-Waveless-160-75-SSTR1		15,600.0	1,800.0				1,800.0	23.5	22.6	0.4	360.0
U-Waveless-160-75-SSTR2		13,560.0	1,800.0				1,800.0	19.0	23.5	3.4	360.0
U-Waveless-160-75-SSTR3		12,600.0	1,800.0				1,800.0	12.4	26.1	1.9	360.0
U-Waveless-160-75-SSTR4		15,540.0	1,800.0				1,800.0	27.0	24.9	0.4	360.0
U-Waveless-160-75-SSTR5		13,860.0	1,800.0				1,800.0	9.5	25.9	2.9	360.0
U-Waveless-160-75-SSTR6		12,480.0	1,800.0				1,800.0	14.9	29.4	1.9	360.0
U-Waveless-160-105-SSTR1	19,960.0	20,160.0	1,271.9	19,980.0	20,160.0	506.6	1,800.0	24.4	21.6	0.3	360.0
U-Waveless-160-105-SSTR2		18,480.0	1,800.0				1,800.0	27.3	23.6	1.0	360.0
U-Waveless-160-105-SSTR3		17,280.0	1,800.0				1,800.0	15.3	25.1	0.3	360.0
U-Waveless-160-105-SSTR4		19,740.0	1,800.0				1,800.0	25.5	23.8	3.5	360.0
U-Waveless-160-105-SSTR5		18,480.0	1,800.0				1,800.0	17.2	25.5	1.9	360.0
U-Waveless-160-105-SSTR6		17,160.0	1,800.0				1,800.0	10.5	30.2	0.3	360.0

Table 3.15: Results for Instances with 160 batches, 5 truck departures per staging lane, exponentially distributed processing time

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
E-Waved-160-45-SSTR1		13,260.0	1,800.0				1,800.0	13.1	32.6	-3.3	360.0
E-Waved-160-45-SSTR2		12,060.0	1,800.0				1,800.0	8.0	49.9	0.0	360.0
E-Waved-160-45-SSTR3		11,700.0	1,800.0				1,800.0	3.1	52.4	0.0	360.0
E-Waved-160-45-SSTR4		12,480.0	1,800.0				1,800.0	24.5	36.2	0.0	360.0
E-Waved-160-45-SSTR5		10,140.0	1,800.0				1,800.0	20.1	55.2	2.3	360.0
E-Waved-160-45-SSTR6		8,280.0	1,800.0				1,800.0	13.8	65.7	1.4	360.0
E-Waved-160-75-SSTR1		19,920.0	1,800.0				1,800.0	19.0	28.4	0.0	360.0
E-Waved-160-75-SSTR2		17,760.0	1,800.0				1,800.0	8.8	34.1	0.3	360.0
E-Waved-160-75-SSTR3		17,100.0	1,800.0				1,800.0	1.8	32.1	0.0	360.0
E-Waved-160-75-SSTR4		16,980.0	1,800.0				1,800.0	34.3	29.7	0.0	360.0
E-Waved-160-75-SSTR5		14,820.0	1,800.0				1,800.0	19.0	33.8	3.5	360.0
E-Waved-160-75-SSTR6		14,280.0	1,800.0				1,800.0	9.2	35.2	2.1	360.0
E-Waved-160-105-SSTR1		20,520.0	1,800.0				1,800.0	18.4	25.6	0.0	360.0
E-Waved-160-105-SSTR2		18,780.0	1,800.0				1,800.0	16.0	29.2	0.6	360.0
E-Waved-160-105-SSTR3		18,360.0	1,800.0				1,800.0	3.3	30.6	0.3	360.0
E-Waved-160-105-SSTR4		18,180.0	1,800.0				1,800.0	35.3	28.7	1.6	360.0
E-Waved-160-105-SSTR5		17,220.0	1,800.0				1,800.0	13.2	32.9	0.7	360.0
E-Waved-160-105-SSTR6		16,200.0	1,800.0				1,800.0	8.5	38.2	1.1	360.0
E-Waveless-160-45-SSTR1		10,140.0	1,800.0				1,800.0	18.3	26.8	0.0	360.0
E-Waveless-160-45-SSTR2		8,820.0	1,800.0				1,800.0	7.5	29.4	-0.7	360.0
E-Waveless-160-45-SSTR3		7,920.0	1,800.0				1,800.0	7.6	36.8	1.5	360.0
E-Waveless-160-45-SSTR4		9,840.0	1,800.0				1,800.0	17.7	31.3	1.2	360.0
E-Waveless-160-45-SSTR5		8,640.0	1,800.0				1,800.0	7.6	30.8	1.4	360.0
E-Waveless-160-45-SSTR6		7,740.0	1,800.0				1,800.0	8.5	41.7	2.3	360.0
E-Waveless-160-75-SSTR1		15,360.0	1,800.0				1,800.0	18.8	24.0	0.4	360.0
E-Waveless-160-75-SSTR2		14,220.0	1,800.0				1,800.0	11.4	24.6	1.3	360.0
E-Waveless-160-75-SSTR3		13,380.0	1,800.0				1,800.0	9.9	27.3	1.8	360.0
E-Waveless-160-75-SSTR4		14,760.0	1,800.0				1,800.0	21.1	25.8	3.5	360.0
E-Waveless-160-75-SSTR5		14,280.0	1,800.0				1,800.0	11.3	27.8	2.5	360.0
E-Waveless-160-75-SSTR6		13,320.0	1,800.0				1,800.0	7.7	31.4	1.8	360.0
E-Waveless-160-105-SSTR1		19,320.0	1,800.0				1,800.0	22.0	21.6	0.9	360.0
E-Waveless-160-105-SSTR2		16,980.0	1,800.0				1,800.0	15.9	22.2	1.0	360.0
E-Waveless-160-105-SSTR3		15,960.0	1,800.0				1,800.0	9.0	26.0	0.7	360.0
E-Waveless-160-105-SSTR4		18,660.0	1,800.0				1,800.0	19.3	23.7	-0.6	360.0
E-Waveless-160-105-SSTR5		16,860.0	1,800.0				1,800.0	12.1	26.5	1.1	360.0
E-Waveless-160-105-SSTR6		15,720.0	1,800.0				1,800.0	7.3	32.2	0.8	360.0

Table 3.16: Results for Instances with 160 batches, 10 truck departures per staging lane, uniformly distributed processing time

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-160-45-SSTR1	19,200.0	19,200.0	38.2	19,200.0	19,200.0	26.8	91.7	5.0	26.7	0.0	134.1
U-Waved-160-45-SSTR2	16,410.0	16,440.0	44.1	16,440.0	16,440.0	30.8	105.6	7.3	30.7	0.0	143.1
U-Waved-160-45-SSTR3	14,490.0	14,520.0	21.8	14,520.0	14,520.0	29.0	79.6	8.3	28.9	0.0	108.7
U-Waved-160-45-SSTR4		11,700.0	1,800.0				1,800.0	40.5	27.7	3.5	360.0
U-Waved-160-45-SSTR5		11,760.0	1,800.0				1,800.0	18.4	31.5	-0.5	360.0
U-Waved-160-45-SSTR6		10,800.0	1,800.0				1,800.0	5.0	36.7	0.6	360.0
U-Waved-160-75-SSTR1	24,480.0	24,480.0	7.7	24,480.0	24,480.0	22.3	52.2	4.2	22.2	0.0	78.1
U-Waved-160-75-SSTR2	21,510.0	21,540.0	39.1	21,540.0	21,540.0	23.2	85.3	6.7	23.1	0.0	97.4
U-Waved-160-75-SSTR3	19,590.0	19,620.0	11.7	19,620.0	19,620.0	25.8	63.3	5.2	25.8	0.0	102.6
U-Waved-160-75-SSTR4	16,800.0	16,800.0	333.4	16,800.0	16,800.0	26.1	385.3	20.4	25.8	0.0	263.1
U-Waved-160-75-SSTR5		15,120.0	1,800.0				1,800.0	21.0	26.7	0.0	360.0
U-Waved-160-75-SSTR6	14,080.0	14,220.0	218.0	14,100.0	14,220.0	1,552.0	1,800.0	18.1	30.0	-0.4	360.0
U-Waved-160-105-SSTR1	30,720.0	30,720.0	7.6	30,720.0	30,720.0	19.8	47.2	2.0	19.8	0.0	62.9
U-Waved-160-105-SSTR2	27,720.0	27,720.0	9.4	27,720.0	27,720.0	21.4	52.2	2.2	21.3	0.0	80.8
U-Waved-160-105-SSTR3	25,800.0	25,800.0	11.3	25,800.0	25,800.0	21.6	54.5	0.5	21.5	0.0	80.1
U-Waved-160-105-SSTR4	23,040.0	23,100.0	39.1	23,040.0	23,100.0	1,739.7	1,800.0	17.4	21.2	-0.3	97.0
U-Waved-160-105-SSTR5	20,340.0	20,340.0	469.0	20,340.0	20,340.0	23.4	515.5	20.6	23.2	0.0	360.0
U-Waved-160-105-SSTR6	19,890.0	19,980.0	706.2	19,920.0	19,980.0	1,067.8	1,800.0	0.9	26.1	0.0	360.0
U-Waveless-160-45-SSTR1		12,720.0	1,800.0				1,800.0	13.2	21.8	-0.5	360.0
U-Waveless-160-45-SSTR2		11,160.0	1,800.0				1,800.0	9.7	23.0	-3.3	360.0
U-Waveless-160-45-SSTR3		9,660.0	1,800.0				1,800.0	8.7	26.7	1.2	360.0
U-Waveless-160-45-SSTR4		11,820.0	1,800.0				1,800.0	21.8	24.1	3.0	360.0
U-Waveless-160-45-SSTR5		10,200.0	1,800.0				1,800.0	20.6	25.6	1.7	360.0
U-Waveless-160-45-SSTR6		9,180.0	1,800.0				1,800.0	12.4	31.6	1.3	360.0
U-Waveless-160-75-SSTR1		16,980.0	1,800.0				1,800.0	21.6	19.3	0.4	360.0
U-Waveless-160-75-SSTR2		14,280.0	1,800.0				1,800.0	14.3	19.7	0.4	360.0
U-Waveless-160-75-SSTR3		13,440.0	1,800.0				1,800.0	21.0	21.4	0.4	360.0
U-Waveless-160-75-SSTR4		17,040.0	1,800.0				1,800.0	24.3	20.2	0.0	360.0
U-Waveless-160-75-SSTR5		14,220.0	1,800.0				1,800.0	14.8	21.8	0.8	360.0
U-Waveless-160-75-SSTR6		13,200.0	1,800.0				1,800.0	18.6	24.5	0.5	360.0
U-Waveless-160-105-SSTR1	23,640.0	23,640.0	33.1	23,640.0	23,640.0	16.4	65.9	27.9	16.3	0.0	112.2
U-Waveless-160-105-SSTR2	19,440.0	19,440.0	202.5	19,440.0	19,440.0	16.8	236.0	25.3	16.7	0.0	315.1
U-Waveless-160-105-SSTR3	18,225.8	18,300.0	1,308.0	18,240.0	18,300.0	473.8	1,800.0	15.7	18.2	0.0	360.0
U-Waveless-160-105-SSTR4	23,640.0	23,640.0	82.5	23,640.0	23,640.0	19.3	120.9	13.2	19.2	0.0	186.9
U-Waveless-160-105-SSTR5	19,440.0	19,440.0	933.2	19,440.0	19,440.0	19.9	972.8	22.8	19.7	0.0	360.0
U-Waveless-160-105-SSTR6	18,204.3	18,240.0	1,208.9	18,240.0	18,240.0	23.8	1,256.5	14.5	23.7	0.0	360.0

Table 3.17: Results for Instances with 160 batches, 10 truck departures per staging lane, exponentially distributed processing time

	Branch & Price							S		MH	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
E-Waved-160-45-SSTR1	15,300.0	15,420.0	530.2	15,300.0	15,420.0	1,243.7	1,800.0	12.8	26.1	-0.4	360.0
E-Waved-160-45-SSTR2		13,320.0	1,800.0				1,800.0	18.0	31.0	0.0	360.0
E-Waved-160-45-SSTR3		12,120.0	1,800.0				1,800.0	19.3	30.4	-0.5	360.0
E-Waved-160-45-SSTR4		11,880.0	1,800.0				1,800.0	27.8	27.3	-0.5	360.0
E-Waved-160-45-SSTR5		11,460.0	1,800.0				1,800.0	13.1	30.8	0.5	360.0
E-Waved-160-45-SSTR6		10,260.0	1,800.0				1,800.0	9.9	36.9	1.2	360.0
E-Waved-160-75-SSTR1		16,560.0	1,800.0				1,800.0	19.2	23.2	1.1	360.0
E-Waved-160-75-SSTR2		14,820.0	1,800.0				1,800.0	18.2	25.1	1.2	360.0
E-Waved-160-75-SSTR3		13,800.0	1,800.0				1,800.0	19.6	25.3	0.4	360.0
E-Waved-160-75-SSTR4		13,260.0	1,800.0				1,800.0	31.7	26.6	5.6	360.0
E-Waved-160-75-SSTR5		13,020.0	1,800.0				1,800.0	18.4	27.5	2.7	360.0
E-Waved-160-75-SSTR6		12,300.0	1,800.0				1,800.0	13.2	32.8	1.4	360.0
E-Waved-160-105-SSTR1	26,520.0	26,520.0	14.1	26,520.0	26,520.0	18.3	50.6	7.2	18.2	0.0	101.0
E-Waved-160-105-SSTR2		22,620.0	1,800.0				1,800.0	13.3	19.6	-0.3	201.8
E-Waved-160-105-SSTR3		20,940.0	1,800.0				1,800.0	9.7	20.8	0.0	360.0
E-Waved-160-105-SSTR4		21,660.0	1,800.0				1,800.0	20.2	19.8	0.3	183.0
E-Waved-160-105-SSTR5		19,920.0	1,800.0				1,800.0	14.5	22.1	1.2	360.0
E-Waved-160-105-SSTR6		19,020.0	1,800.0				1,800.0	9.8	22.6	0.0	360.0
E-Waveless-160-45-SSTR1	11,180.0	11,640.0	294.6	11,220.0	11,640.0	1,485.3	1,800.0	19.6	20.1	0.0	360.0
E-Waveless-160-45-SSTR2		9,540.0	1,800.0				1,800.0	20.1	21.4	5.4	360.0
E-Waveless-160-45-SSTR3		9,060.0	1,800.0				1,800.0	14.6	24.8	1.9	360.0
E-Waveless-160-45-SSTR4		11,100.0	1,800.0				1,800.0	34.6	22.6	0.5	360.0
E-Waveless-160-45-SSTR5		9,240.0	1,800.0				1,800.0	29.2	26.6	4.9	360.0
E-Waveless-160-45-SSTR6		8,760.0	1,800.0				1,800.0	19.2	29.1	2.7	360.0
E-Waveless-160-75-SSTR1		15,000.0	1,800.0				1,800.0	25.2	19.7	0.0	360.0
E-Waveless-160-75-SSTR2		12,720.0	1,800.0				1,800.0	20.8	20.0	0.9	360.0
E-Waveless-160-75-SSTR3		12,180.0	1,800.0				1,800.0	18.7	23.3	0.5	360.0
E-Waveless-160-75-SSTR4		14,640.0	1,800.0				1,800.0	29.1	21.6	-0.4	360.0
E-Waveless-160-75-SSTR5		12,660.0	1,800.0				1,800.0	24.2	23.0	1.9	360.0
E-Waveless-160-75-SSTR6		11,760.0	1,800.0				1,800.0	15.3	26.5	0.5	360.0
E-Waveless-160-105-SSTR1		22,260.0	1,800.0				1,800.0	16.7	15.4	0.0	360.0
E-Waveless-160-105-SSTR2		18,900.0	1,800.0				1,800.0	15.9	16.4	0.6	360.0
E-Waveless-160-105-SSTR3		18,000.0	1,800.0				1,800.0	13.0	17.4	0.7	360.0
E-Waveless-160-105-SSTR4		21,360.0	1,800.0				1,800.0	19.7	17.2	0.3	360.0
E-Waveless-160-105-SSTR5		18,840.0	1,800.0				1,800.0	15.6	18.0	0.6	360.0
E-Waveless-160-105-SSTR6		17,820.0	1,800.0				1,800.0	8.4	19.8	0.7	360.0

Table 3.18: Results for Instances with 160 batches, 20 truck departures per staging lane, uniformly distributed processing time

	Branch & Price							**S**		**MH**	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
U-Waved-160-45-SSTR1	13,920.0	13,920.0	30.9	13,920.0	13,920.0	18.6	68.1	10.8	18.5	0.0	91.7
U-Waved-160-45-SSTR2	13,200.0	13,200.0	4.2	13,200.0	13,200.0	20.9	46.0	0.5	20.9	0.0	88.7
U-Waved-160-45-SSTR3	12,810.0	12,840.0	3.5	12,840.0	12,840.0	21.0	45.5	2.3	21.0	-0.5	73.2
U-Waved-160-45-SSTR4		11,340.0	1,800.0				1,800.0	40.7	21.3	-0.5	180.0
U-Waved-160-45-SSTR5		10,860.0	1,800.0				1,800.0	18.8	23.1	2.2	360.0
U-Waved-160-45-SSTR6	10,397.1	10,620.0	53.7	10,440.0	10,620.0	1,720.1	1,800.0	11.3	26.1	0.0	194.7
U-Waved-160-75-SSTR1	20,940.0	20,940.0	12.9	20,940.0	20,940.0	16.7	46.1	8.3	16.6	0.0	85.0
U-Waved-160-75-SSTR2	19,320.0	19,320.0	3.2	19,320.0	19,320.0	19.2	41.4	7.5	19.1	0.0	56.9
U-Waved-160-75-SSTR3	18,600.0	18,660.0	9.7	18,600.0	18,660.0	1,773.2	1,800.0	4.2	17.1	0.0	84.1
U-Waved-160-75-SSTR4	16,800.0	16,860.0	421.4	16,800.0	16,860.0	1,360.2	1,800.0	22.8	18.4	-0.4	360.0
U-Waved-160-75-SSTR5		15,840.0	1,800.0				1,800.0	14.8	19.5	-0.4	360.0
U-Waved-160-75-SSTR6		15,300.0	1,800.0				1,800.0	17.3	20.1	0.4	360.0
U-Waved-160-105-SSTR1	26,040.0	26,040.0	6.0	26,040.0	26,040.0	14.7	35.3	11.3	14.7	0.0	55.0
U-Waved-160-105-SSTR2	23,760.0	23,760.0	14.4	23,760.0	23,760.0	16.1	46.5	8.1	16.0	0.0	48.4
U-Waved-160-105-SSTR3	23,640.0	23,640.0	3.4	23,640.0	23,640.0	16.1	35.5	4.1	16.0	0.0	52.9
U-Waved-160-105-SSTR4	22,440.0	22,440.0	17.0	22,440.0	22,440.0	15.7	48.3	29.1	15.6	0.0	97.6
U-Waved-160-105-SSTR5	20,820.0	20,820.0	175.1	20,820.0	20,820.0	17.2	209.3	11.2	17.0	0.0	221.6
U-Waved-160-105-SSTR6	20,200.0	20,220.0	826.7	20,220.0	20,220.0	19.4	865.3	7.7	19.3	0.0	360.0
U-Waveless-160-45-SSTR1		10,860.0	1,800.0				1,800.0	24.3	18.3	0.0	360.0
U-Waveless-160-45-SSTR2		10,260.0	1,800.0				1,800.0	19.9	19.8	0.0	360.0
U-Waveless-160-45-SSTR3		9,660.0	1,800.0				1,800.0	18.6	20.4	0.0	360.0
U-Waveless-160-45-SSTR4		11,400.0	1,800.0				1,800.0	31.1	20.1	-3.3	360.0
U-Waveless-160-45-SSTR5		10,380.0	1,800.0				1,800.0	20.2	21.9	2.3	360.0
U-Waveless-160-45-SSTR6		8,940.0	1,800.0				1,800.0	16.1	23.3	2.6	360.0
U-Waveless-160-75-SSTR1	16,200.0	16,200.0	1,298.1	16,200.0	16,200.0	15.9	1,329.6	27.4	15.6	0.7	360.0
U-Waveless-160-75-SSTR2		14,940.0	1,800.0				1,800.0	25.3	17.4	1.2	360.0
U-Waveless-160-75-SSTR3		14,400.0	1,800.0				1,800.0	19.6	17.4	0.8	360.0
U-Waveless-160-75-SSTR4		15,240.0	1,800.0				1,800.0	41.7	17.5	1.9	360.0
U-Waveless-160-75-SSTR5		14,220.0	1,800.0				1,800.0	23.6	18.8	2.5	360.0
U-Waveless-160-75-SSTR6		13,380.0	1,800.0				1,800.0	19.3	20.7	0.4	360.0
U-Waveless-160-105-SSTR1	22,800.0	22,800.0	18.9	22,800.0	22,800.0	13.3	45.4	31.3	13.2	0.3	88.0
U-Waveless-160-105-SSTR2	20,220.0	20,220.0	24.4	20,220.0	20,220.0	14.4	53.1	26.7	14.3	0.0	73.8
U-Waveless-160-105-SSTR3	19,350.0	19,380.0	43.1	19,380.0	19,380.0	14.6	72.2	11.5	14.5	0.0	125.7
U-Waveless-160-105-SSTR4		20,340.0	1,800.0				1,800.0	37.2	14.8	0.9	360.0
U-Waveless-160-105-SSTR5		18,840.0	1,800.0				1,800.0	25.8	15.7	0.6	360.0
U-Waveless-160-105-SSTR6	17,892.0	17,940.0	1,125.4	17,940.0	17,940.0	17.9	1,161.0	9.4	17.8	0.0	360.0

Table 3.19: Results for Instances with 160 batches, 20 truck departures per staging lane, exponentially distributed processing time

	Branch & Price							**S**		**MH**	
	Linear Relaxation			Integer							
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{S}$	CPU^{S}	$\%\Delta^{MH}$	CPU^{MH}
E-Waved-160-45-SSTR1		13,020.0	1,800.0				1,800.0	11.1	20.1	-5.9	360.0
E-Waved-160-45-SSTR2		11,640.0	1,800.0				1,800.0	17.5	20.8	1.0	360.0
E-Waved-160-45-SSTR3		11,100.0	1,800.0				1,800.0	14.6	21.0	2.1	360.0
E-Waved-160-45-SSTR4		10,800.0	1,800.0				1,800.0	38.3	21.8	5.3	360.0
E-Waved-160-45-SSTR5		10,020.0	1,800.0				1,800.0	21.6	24.3	1.8	360.0
E-Waved-160-45-SSTR6		9,600.0	1,800.0				1,800.0	16.3	26.2	0.6	360.0
E-Waved-160-75-SSTR1	17,760.0	17,760.0	88.8	17,760.0	17,760.0	17.9	124.5	13.9	17.8	0.0	163.6
E-Waved-160-75-SSTR2	16,260.0	16,260.0	230.8	16,260.0	16,260.0	18.7	268.2	17.0	18.6	0.4	221.8
E-Waved-160-75-SSTR3	15,900.0	15,900.0	60.5	15,900.0	15,900.0	19.2	98.8	14.7	19.1	0.4	360.0
E-Waved-160-75-SSTR4		15,780.0	1,800.0				1,800.0	25.9	19.3	3.3	360.0
E-Waved-160-75-SSTR5		15,420.0	1,800.0				1,800.0	15.6	19.8	0.8	360.0
E-Waved-160-75-SSTR6		15,060.0	1,800.0				1,800.0	10.4	23.1	0.0	360.0
E-Waved-160-105-SSTR1	21,600.0	21,660.0	46.3	21,600.0	21,660.0	1,738.2	1,800.0	6.4	15.6	0.0	115.6
E-Waved-160-105-SSTR2	19,560.0	19,620.0	108.6	19,560.0	19,620.0	1,675.3	1,800.0	10.7	16.1	0.0	155.6
E-Waved-160-105-SSTR3		18,960.0	1,800.0				1,800.0	9.8	17.5	0.0	360.0
E-Waved-160-105-SSTR4		18,780.0	1,800.0				1,800.0	18.2	17.4	0.6	360.0
E-Waved-160-105-SSTR5		17,640.0	1,800.0				1,800.0	14.6	17.9	0.7	360.0
E-Waved-160-105-SSTR6		17,220.0	1,800.0				1,800.0	8.0	19.2	0.3	360.0
E-Waveless-160-45-SSTR1		11,820.0	1,800.0				1,800.0	21.8	18.4	-1.0	360.0
E-Waveless-160-45-SSTR2		10,560.0	1,800.0				1,800.0	13.1	19.0	0.6	360.0
E-Waveless-160-45-SSTR3		9,600.0	1,800.0				1,800.0	21.9	20.1	2.4	360.0
E-Waveless-160-45-SSTR4		10,560.0	1,800.0				1,800.0	40.9	21.4	0.0	360.0
E-Waveless-160-45-SSTR5		10,140.0	1,800.0				1,800.0	17.2	21.2	1.7	360.0
E-Waveless-160-45-SSTR6		9,120.0	1,800.0				1,800.0	21.1	23.4	2.6	360.0
E-Waveless-160-75-SSTR1	16,920.0	17,400.0	128.3	16,920.0	17,400.0	1,656.3	1,800.0	15.9	15.4	0.0	360.0
E-Waveless-160-75-SSTR2		15,060.0	1,800.0				1,800.0	13.1	16.2	0.0	360.0
E-Waveless-160-75-SSTR3		14,100.0	1,800.0				1,800.0	18.3	16.6	0.0	360.0
E-Waveless-160-75-SSTR4		16,980.0	1,800.0				1,800.0	22.3	17.5	0.0	360.0
E-Waveless-160-75-SSTR5		14,700.0	1,800.0				1,800.0	14.3	18.9	0.4	360.0
E-Waveless-160-75-SSTR6		13,740.0	1,800.0				1,800.0	11.8	20.2	0.4	360.0
E-Waveless-160-105-SSTR1	22,620.0	22,620.0	31.7	22,620.0	22,620.0	14.8	61.3	10.3	14.8	0.0	137.7
E-Waveless-160-105-SSTR2	19,950.0	19,980.0	154.5	19,980.0	19,980.0	14.9	184.3	9.0	14.9	0.0	360.0
E-Waveless-160-105-SSTR3	19,106.4	19,140.0	1,217.6	19,140.0	19,140.0	15.5	1,248.5	6.6	15.4	0.0	360.0
E-Waveless-160-105-SSTR4	20,700.0	20,700.0	17.6	20,700.0	20,700.0	16.1	49.7	15.9	16.0	0.0	65.4
E-Waveless-160-105-SSTR5		18,300.0	1,800.0				1,800.0	14.4	16.2	0.3	360.0
E-Waveless-160-105-SSTR6	17,993.2	18,000.0	532.2	18,000.0	18,000.0	19.0	570.1	6.7	18.9	0.0	360.0

x_{ik} is 1 if batch $i \in B$ is scheduled to be picked at the k^{th} position in the shift for the order picker, where $k \in K$, else 0

y_k^{ϕ} is 1 if a break $\phi \in \Phi$ is scheduled at the k^{th} position in the shift for the order picker, where $k \in K$, else 0

s_j is 1 if the order picker starts the shift at the beginning of period $j \in S$, else 0

e_j is 1 if the order picker ends the shift at the end of period $j \in E$, else 0

c_k completion time of the task scheduled at the k^{th} position in the shift of the order picker, where $k \in K$

m amount of time for which the order picker is compensated

The reduced problem (RP) of the OPSP for the case study is then reformulated as a MILP model as follows:

RPCase:

$$\min m \tag{3.92}$$

subject to

$$\sum_{i \in B} x_{ik} + \sum_{\phi \in \Phi} y_k^{\phi} \leq 1 \qquad \forall k \in K \tag{3.93}$$

$$\sum_{k \in K} x_{ik} = 1 \qquad \forall i \in B \tag{3.94}$$

$$\sum_{j \in J \setminus S} s_j + \sum_{j \in J \setminus E} e_j = 0 \tag{3.95}$$

$$c_1 - \Big(\sum_{j \in S} (j-1) s_j\Big) l \geq \sum_{i \in B} t_i x_{i1} + \sum_{\phi \in \Phi} l_b^{\phi} y_k^{\phi} \tag{3.96}$$

$$c_k - c_{k-1} \geq \sum_{i \in B} t_i x_{ik} + \sum_{\phi \in \Phi} l_b^{\phi} y_k^{\phi} \qquad \forall k \in K \setminus \{1\} \tag{3.97}$$

$$\sum_{j \in E} (j e_j) l \geq c_{\bar{k}} \tag{3.98}$$

$$c_k + M(1 - x_{ik}) \geq r_i \qquad \forall i \in B, k \in K \tag{3.99}$$

$$c_k - M(1 - x_{ik}) \leq d_i \qquad \forall i \in B, k \in K \tag{3.100}$$

$$\sum_{k \in K} y_k^1 = \sum_{k \in K} y_k^3 = 1 \tag{3.101}$$

$$M \sum_{k \in K} y_k^2 \geq m - 360 \tag{3.102}$$

$$c_k + M(1 - y_k^{\phi}) - \Big(\sum_{j \in S} (j-1) s_j\Big) l \geq r^{\phi} - \Psi \qquad \forall k \in K, \phi \in \Phi \tag{3.103}$$

$$c_k + M(1 - y_k^{\phi}) - \Big(\sum_{j \in S}(j-1)s_j\Big)l \leq r^{\phi} + \Psi \qquad \forall k \in K, \phi \in \Phi \qquad (3.104)$$

$$\sum_{i \in B} x_{ik-1} + \sum_{\phi \in \Phi} y_k^{\phi} \geq \sum_{i \in B} x_{ik} + \sum_{\phi \in R} y_k^{\phi} \qquad \forall k \in K \setminus \{1\} \qquad (3.105)$$

$$(\sum_{j \in E} je_j - \sum_{j \in S}(j-1)s_j)l \leq m \qquad (3.106)$$

$$c_k \geq 0 \qquad \forall k \in K \qquad (3.107)$$

$$x_{ik} \in \{0,1\} \qquad \forall i \in B, k \in K \qquad (3.108)$$

$$y_k^{\phi} \in \{0,1\} \qquad \forall k \in K, \phi \in \Phi \qquad (3.109)$$

$$s_j, e_j \in \{0,1\} \qquad \forall j \in J \qquad (3.110)$$

$$T_{min} \leq m \leq T_{max} \qquad (3.111)$$

Most of the constraints in this reformulation are the same as the original reduced problem formulation in Section 3.8.8. Here, we only explain the constraints that are different. Constraint (3.101) enforces that the first short break and the long break have to be scheduled, whereas constraint (3.102) indicates that the second short break only has to be scheduled if the order picker is employed for more than 6 hours. Constraints (3.103) and (3.104) regulate the start and completion times of individual breaks according to the flexibility Ψ for the break times.

3.8.11 Detailed result of Section 3.6.3

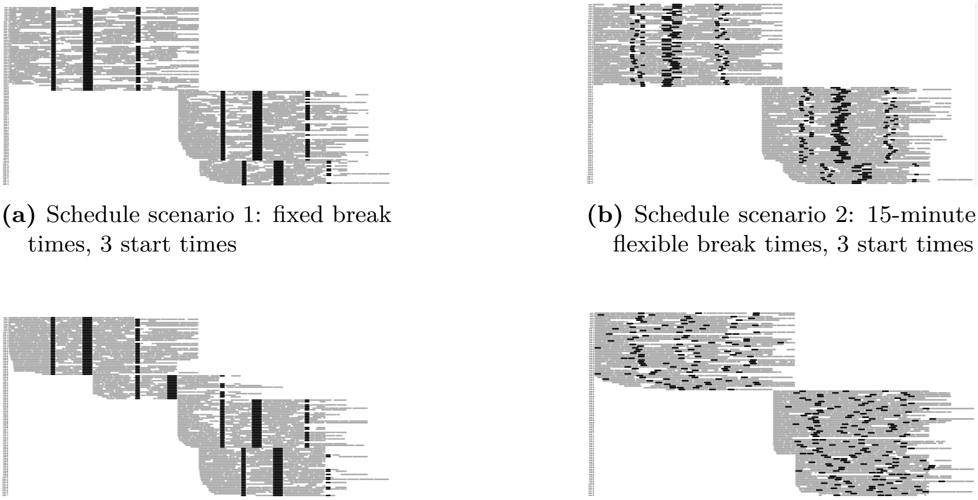
The schedules resulting from the OPSP on Day 1 of the normal week are illustrated in Figure 3.12 for the scenarios where T_{min} equals 6 hours. The total number of hours that the order pickers are compensated for is indicated in Table 3.20, whereas as the number of scheduled order pickers is presented in Table 3.21. When dividing these two numbers for each instance, we obtain the average shift length. The number of order pickers who are compensated for exactly T_{min} time unit is included in Table 3.22. A summary of these results is included in Figure 3.8.

3.8.12 Preemptive breaks

The problem formulation in Section 3.3 does not allow preemptive order picking, i.e., the order picking of a batch has to be completed before the order picker can take a break. In this section, we study the operational environment where the items of a

Table 3.20: Number of hours that scheduled order pickers need to be compensated for per scenario

		Scenario											
Week	Day	1	2	3	4	5	6	7	8	9	10	11	12
Normal	1	897	800	851	916	813	828	910	779	791	831	783	750
	2	933	839	899	984	928	996	1,056	838	885	912	861	844
	3	1,071	991	1,023	1,091	1,074	1,090	1,163	956	1,002	1,072	970	932
	4	1,004	915	966	1,010	958	1,023	1,064	892	917	1,010	899	870
	5	1,131	1,021	1,053	1,123	1,069	1,119	1,144	994	1,017	1,049	971	971
	6	1,297	1,177	1,221	1,264	1,221	1,293	1,315	1,144	1,166	1,203	1,149	1,114
	7	1,283	1,158	1,213	1,274	1,262	1,339	1,455	1,148	1,161	1,274	1,170	1,146
Busy	1	783	726	758	816	753	819	861	705	712	777	706	686
	2	929	835	835	879	854	854	906	811	830	830	823	779
	3	990	901	901	967	921	945	945	860	873	876	878	829
	4	1,063	971	1,013	1,014	1,006	1,054	1,090	955	977	975	972	945
	5	1,544	1,420	1,495	1,495	1,454	1,454	1,574	1,367	1,392	1,398	1,435	1,393
	6	1,713	1,625	1,639	1,668	1,669	1,744	1,730	1,584	1,619	1,651	1,620	1,612
	7	1,412	1,288	1,375	1,375	1,374	1,391	1,447	1,256	1,280	1,350	1,293	1,239



(a) Schedule scenario 1: fixed break times, 3 start times

(b) Schedule scenario 2: 15-minute flexible break times, 3 start times

(c) Schedule scenario 5: fixed break times, 4 start times

(d) Schedule scenario 11: completely flexible break times, 3 start times

Figure 3.12: Gantt charts illustrating the scheduling of order picking tasks (in grey) and breaks (in black) for the four scenarios where $T_{min} = 6$ hours

Table 3.21: Number of scheduled order pickers per scenario

		Scenario											
Week	Day	1	2	3	4	5	6	7	8	9	10	11	12
Normal	1	115	104	108	111	111	105	110	106	104	101	110	105
	2	125	117	118	120	128	132	129	115	116	112	117	124
	3	140	135	133	134	144	139	140	131	132	131	133	132
	4	133	123	127	123	132	133	130	123	122	123	126	123
	5	147	137	136	136	139	144	138	134	133	127	135	135
	6	167	153	155	153	161	166	157	151	151	146	154	150
	7	170	153	155	154	170	171	174	155	147	154	156	151
Busy	1	101	96	97	98	100	102	103	97	91	92	96	93
	2	111	110	103	105	109	105	108	106	102	100	105	104
	3	120	114	112	115	117	120	111	110	109	105	109	109
	4	131	124	126	121	128	130	130	122	121	117	126	124
	5	185	175	182	177	188	176	186	172	173	166	173	176
	6	206	199	199	197	204	206	201	196	197	195	195	201
	7	172	160	167	165	170	170	170	161	159	163	159	162

batch can be partially picked before the order picker takes a break. We can derive an exact algorithm for the preemptive scheduling case after making the necessary changes to the resource extension functions of the pricing problem for the non-preemptive problem presented in Section 3.3. All other components of the pricing problem(e.g., the resources, their definition, feasibility windows and dominance rules) remain the same as in the non-preemptive scheduling problem.

Table 3.22: Number of scheduled order pickers that get compensated for T_{min} time units per scenario

		Scenario											
Week	Day	1	2	3	4	5	6	7	8	9	10	11	12
Normal	1	24	22	48	83	37	46	80	18	42	64	51	41
	2	41	47	64	96	50	76	105	42	72	94	47	59
	3	30	43	66	115	52	61	97	48	65	98	45	48
	4	36	40	76	97	52	71	106	49	68	91	49	49
	5	39	45	67	101	35	71	98	53	72	90	52	52
	6	32	27	59	113	44	71	98	33	68	103	28	35
	7	56	44	70	112	62	81	111	40	45	83	52	35
Busy	1	28	28	46	66	36	35	66	14	31	54	41	33
	2	8	21	32	66	25	31	66	29	39	62	18	21
	3	13	20	37	68	24	36	54	27	39	65	21	27
	4	12	24	38	75	28	38	80	26	36	69	31	29
	5	13	25	44	98	37	39	100	21	40	88	25	28
	6	20	22	48	105	30	30	79	32	54	96	27	21
	7	19	22	32	110	27	39	83	21	37	102	30	32

The resource extension functions without breaks are given by the same functions $f(\cdot)$ as specified in Section 3.4.2 4.2. The only difference is in extension functions with breaks. A distinction has to be made between two conditions. First, if $T_i^{work} + t_j$ is greater than T_{break}, then batch j cannot be completed before the order picker takes a break. The portion of batch j that can be completed before the break is scheduled has a duration of $T_i^{work} + t_j - T_{break}$ time units and the remaining items of the batch can be picked after the break. The resource extension functions under this first condition (denoted as $g'(\cdot)$) are defined as:

$$T_j^{time} = g'^{time}(T_i, j) := \max\{T_i^{time} + t_j + l_b, r_j\} \tag{3.112}$$

$$T_j^{dur} = g'^{dur}(T_i, j) := \max\{T_i^{dur} + t_j + l_b, r_j - T_i^{start}\} \tag{3.113}$$

$$T_j^{work} = g'^{work}(T_i, j) := T_i^{work} + t_j - T_{break} \tag{3.114}$$

$$T_j^{brk} = g'^{brk}(T_i, j) := \min\{d_j - T_i^{work} - t_j + T_{break}, \infty\} \tag{3.115}$$

Under the second condition, if $T_i^{work} + t_j$ is less than or equal to T_{break}, then the complete batch can be picked before the order picker takes a break. The resource extension functions under this second condition are denoted by $g''(\cdot)$ and defined as follows:

$$T_j^{time} = g''^{time}(T_i, j) := \max\{T_i^{time} + t_j + l_b, r_j\} \quad (3.116)$$

$$T_j^{dur} = g''^{dur}(T_i, j) := \max\{T_i^{dur} + t_j + l_b, r_j - T_i^{start}\} \quad (3.117)$$

$$T_j^{work} = g''^{work}(T_i, j) := 0 \quad (3.118)$$

$$T_j^{brk} = g''^{brk}(T_i, j) := \min\{d_j, \infty\} \quad (3.119)$$

For a numerical test bed with instances of 40 batches and 5 or 10 truck departures per staging lane, we present the performance of the branch-and-price algorithm when preemptive order scheduling is allowed in Table 3.23 and Table 3.24, respectively. Note that the counterparts with non-preemptive order scheduling are presented in Table 3.6, Table 3.7 and Table 3.8, respectively. The last column in these two tables indicates the relative cost increase of using preemption against non-preemption, which is expressed as $\%\Delta^P = ((z^P - z^{NP})/z^{NP}) \times 100$, where z^P and z^{NP} are the best branch-and-price solutions for preemtive and non-preemptive order scheduling, respectively.

The results illustrate the adaptability of our solution framework. For three of the instances, the algorithm for preemptive order picking was able to generate only inferior solutions compared to the algorithm for non-preemptive order picking within the limited run time. Excluding these three instances, the cost savings of preemptive order scheduling are on average 0.4% compared to non-preemptive order scheduling, with the largest cost savings of 8.3% However, the problem formulation with preemptive order scheduling assumes that breaks occur instanteneously and they ignore the time it takes to travel between the location of the last picked item in the pick tour and the location where the order picker takes a break. A more accurate approach to include preemptive order picking would be to include this extra travel time. Consequently, preemptive order scheduling can be more expensive than non-preemptive order scheduling. Therefore, studying preemptive order scheduling can have a high practical and academic relevance but is beyond the scope of our work.

Table 3.23: Results for Instances with 40 batches, 5 truck departures per staging lane, preemptive order picking

	Branch & Price							
	Linear Relaxation			Integer				
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{P}$
U-Waved-40-45-SSTR1	4,480.0	4,800.0	4.7	4,500.0	4,800.0	1,793.3	1,800.0	0.0
U-Waved-40-45-SSTR2	4,200.0	4,440.0	3.4	4,200.0	4,440.0	1,793.0	1,800.0	0.0
U-Waved-40-45-SSTR3	4,152.0	4,200.0	2.1	4,200.0	4,200.0	2.8	7.7	0.0
U-Waved-40-45-SSTR4	4,480.0	4,800.0	4.4	4,500.0	4,800.0	1,793.4	1,800.0	0.0
U-Waved-40-45-SSTR5	3,480.0	3,660.0	148.4	3,480.0	3,660.0	1,649.0	1,800.0	0.0
U-Waved-40-45-SSTR6	2,685.0	2,760.0	17.2	2,700.0	2,700.0	1,636.2	1,656.2	0.0
U-Waved-40-75-SSTR1	4,980.0	4,980.0	1.8	4,980.0	4,980.0	2.1	6.0	0.0
U-Waved-40-75-SSTR2	4,740.0	4,740.0	1.4	4,740.0	4,740.0	2.4	6.2	0.0
U-Waved-40-75-SSTR3	4,660.0	4,740.0	1.2	4,680.0	4,680.0	8.0	11.5	0.0
U-Waved-40-75-SSTR4	4,800.0	4,800.0	2.0	4,800.0	4,800.0	2.3	6.6	0.0
U-Waved-40-75-SSTR5	3,840.0	3,840.0	40.0	3,840.0	3,840.0	2.4	44.7	0.0
U-Waved-40-75-SSTR6	3,400.0	3,420.0	6.1	3,420.0	3,420.0	2.4	10.8	0.0
U-Waved-40-105-SSTR1	5,600.0	5,760.0	1.1	5,640.0	5,760.0	1,796.9	1,800.0	0.0
U-Waved-40-105-SSTR2	5,280.0	5,400.0	0.8	5,280.0	5,400.0	1,797.0	1,800.0	0.0
U-Waved-40-105-SSTR3	5,160.0	5,160.0	0.4	5,160.0	5,160.0	2.5	5.5	0.0
U-Waved-40-105-SSTR4	5,320.0	5,460.0	6.9	5,340.0	5,460.0	1,790.7	1,800.0	-1.1
U-Waved-40-105-SSTR5	4,936.0	5,040.0	14.3	4,980.0	5,040.0	1,783.0	1,800.0	1.2
U-Waved-40-105-SSTR6	4,771.2	4,800.0	3.5	4,800.0	4,800.0	2.8	8.9	0.0
U-Waveless-40-45-SSTR1	3,040.0	3,360.0	7.5	3,060.0	3,360.0	1,790.4	1,800.0	0.0
U-Waveless-40-45-SSTR2	2,360.0	2,520.0	14.1	2,400.0	2,520.0	1,783.9	1,800.0	0.0
U-Waveless-40-45-SSTR3	2,181.2	2,280.0	27.0	2,220.0	2,280.0	1,770.7	1,800.0	0.0
U-Waveless-40-45-SSTR4	2,640.0	2,940.0	335.7	2,640.0	2,940.0	1,462.0	1,800.0	0.0
U-Waveless-40-45-SSTR5	2,280.0	2,340.0	480.6	2,280.0	2,340.0	1,317.1	1,800.0	0.0
U-Waveless-40-45-SSTR6	2,168.2	2,280.0	29.7	2,220.0	2,280.0	1,767.9	1,800.0	0.0
U-Waveless-40-75-SSTR1	3,960.0	3,960.0	16.1	3,960.0	3,960.0	1.7	19.5	-8.3
U-Waveless-40-75-SSTR2	3,505.3	3,600.0	59.8	3,540.0	3,540.0	383.4	445.0	0.0
U-Waveless-40-75-SSTR3	3,360.0	3,420.0	11.4	3,360.0	3,360.0	986.5	1,000.3	0.0
U-Waveless-40-75-SSTR4	3,720.0	3,900.0	129.7	3,720.0	3,900.0	1,668.3	1,800.0	0.0
U-Waveless-40-75-SSTR5	3,441.1	3,540.0	268.4	3,480.0	3,540.0	1,529.6	1,800.0	0.0
U-Waveless-40-75-SSTR6	3,281.1	3,360.0	18.0	3,300.0	3,360.0	1,779.9	1,800.0	0.0
U-Waveless-40-105-SSTR1	6,830.0	6,840.0	0.7	6,840.0	6,840.0	1.6	3.8	-5.0
U-Waveless-40-105-SSTR2	5,382.9	5,400.0	0.6	5,400.0	5,400.0	1.7	3.9	-1.1
U-Waveless-40-105-SSTR3	4,740.0	4,800.0	0.8	4,740.0	4,800.0	1,797.5	1,800.0	0.0
U-Waveless-40-105-SSTR4	5,760.0	5,760.0	3.5	5,760.0	5,760.0	1.9	7.2	0.0
U-Waveless-40-105-SSTR5	5,040.0	5,040.0	2.2	5,040.0	5,040.0	1.9	6.0	0.0
U-Waveless-40-105-SSTR6	4,620.0	4,620.0	1.5	4,620.0	4,620.0	2.1	5.7	0.0
E-Waved-40-45-SSTR1	4,140.0	4,380.0	6.5	4,140.0	4,380.0	1,791.5	1,800.0	0.0
E-Waved-40-45-SSTR2	3,769.7	3,960.0	3.5	3,780.0	3,960.0	1,793.0	1,800.0	0.0
E-Waved-40-45-SSTR3	3,718.3	3,780.0	2.3	3,720.0	3,780.0	1,794.5	1,800.0	0.0
E-Waved-40-45-SSTR4	4,080.0	4,320.0	5.6	4,080.0	4,320.0	1,792.2	1,800.0	0.0
E-Waved-40-45-SSTR5	3,156.6	3,360.0	1,069.5	3,180.0	3,360.0	727.7	1,800.0	0.0
E-Waved-40-45-SSTR6	2,464.5	2,580.0	55.8	2,520.0	2,580.0	1,741.1	1,800.0	2.4
E-Waved-40-75-SSTR1	5,580.0	5,880.0	1.5	5,580.0	5,880.0	1,796.3	1,800.0	0.0
E-Waved-40-75-SSTR2	5,010.0	5,160.0	1.2	5,040.0	5,160.0	1,796.5	1,800.0	-1.1
E-Waved-40-75-SSTR3	5,000.0	5,040.0	3.4	5,040.0	5,040.0	2.3	8.1	-1.2
E-Waved-40-75-SSTR4	5,460.0	5,760.0	4.7	5,460.0	5,760.0	1,793.2	1,800.0	0.0
E-Waved-40-75-SSTR5	4,530.0	4,620.0	5.1	4,560.0	4,620.0	1,792.6	1,800.0	0.0
E-Waved-40-75-SSTR6	4,260.0	4,260.0	2.1	4,260.0	4,260.0	2.3	6.6	0.0
E-Waved-40-105-SSTR1	5,460.0	5,460.0	0.4	5,460.0	5,460.0	2.2	4.8	0.0
E-Waved-40-105-SSTR2	4,980.0	4,980.0	0.6	4,980.0	4,980.0	2.5	5.5	0.0
E-Waved-40-105-SSTR3	4,860.0	4,860.0	0.5	4,860.0	4,860.0	2.7	5.8	0.0
E-Waved-40-105-SSTR4	4,800.0	4,800.0	3.2	4,800.0	4,800.0	2.4	7.9	0.0
E-Waved-40-105-SSTR5	3,846.3	3,900.0	727.3	3,900.0	3,900.0	2.9	732.9	0.0
E-Waved-40-105-SSTR6	3,442.2	3,480.0	162.1	3,480.0	3,480.0	3.1	168.2	0.0
E-Waveless-40-45-SSTR1	2,920.0	2,940.0	163.1	2,940.0	2,940.0	1.9	166.9	0.0
E-Waveless-40-45-SSTR2	2,352.3	2,520.0	165.9	2,400.0	2,400.0	1,060.3	1,228.2	0.0
E-Waveless-40-45-SSTR3	2,099.1	2,160.0	27.4	2,100.0	2,160.0	1,770.6	1,800.0	0.0
E-Waveless-40-45-SSTR4	2,720.0	2,940.0	691.1	2,760.0	2,940.0	1,106.9	1,800.0	0.0
E-Waveless-40-45-SSTR5	2,349,9	2,520.0	168.1	2,400.0	2,460.0	1,641.1	1,800.0	2.5
E-Waveless-40-45-SSTR6	2,111.8	2,160.0	23.8	2,160.0	2,160.0	2.6	28.8	0.0
E-Waveless-40-75-SSTR1	5,070.0	5,340.0	5.0	5,100.0	5,340.0	1,793.3	1,800.0	0.0
E-Waveless-40-75-SSTR2	4,120.0	4,260.0	128.9	4,140.0	4,260.0	1,669.4	1,800.0	0.0
E-Waveless-40-75-SSTR3	3,983.4	4,020.0	25.3	4,020.0	4,020.0	1.8	28.9	0.0
E-Waveless-40-75-SSTR4	4,279.1	4,380.0	535.8	4,320.0	4,380.0	1,262.5	1,800.0	0.0
E-Waveless-40-75-SSTR5	4,035.0	4,080.0	267.0	4,080.0	4,080.0	2.1	271.0	0.0
E-Waveless-40-75-SSTR6	3,925.7	3,960.0	19.3	3,960.0	3,960.0	2.2	23.7	0.0
E-Waveless-40-105-SSTR1	5,280.0	5,280.0	0.6	5,280.0	5,280.0	1.6	3.7	0.0
E-Waveless-40-105-SSTR2	3,960.0	3,960.0	4.5	3,960.0	3,960.0	1.6	7.7	0.0
E-Waveless-40-105-SSTR3	3,402.0	3,480.0	8.7	3,420.0	3,480.0	1,789.6	1,800.0	0.0
E-Waveless-40-105-SSTR4	4,800.0	4,800.0	2.8	4,800.0	4,800.0	1.9	6.5	0.0
E-Waveless-40-105-SSTR5	3,900.0	3,900.0	51.9	3,900.0	3,900.0	1.9	55.6	0.0
E-Waveless-40-105-SSTR6	3,399.4	3,480.0	6.5	3,420.0	3,480.0	1,791.6	1,800.0	0.0

Table 3.24: Results for Instances with 40 batches, 10 truck departures per staging lane, preemptive order picking

	Branch & Price for Preemptive Order Picking							
	Linear Relaxation			Integer				
Instance	LB^{LP}	UB^{LP}	CPU^{LP}	LB^{IP}	UB^{IP}	CPU^{IP}	CPU^{BP}	$\%\Delta^{P}$
U-Waved-40-45-SSTR1	4,800.00	4,800.00	4	4,800.00	4,800.00	1.8	7.6	0.0
U-Waved-40-45-SSTR2	4,140.00	4,140.00	0.4	4,140.00	4,140.00	2.1	4.6	0.0
U-Waved-40-45-SSTR3	3,660.00	3,660.00	0.6	3,660.00	3,660.00	2	4.6	0.0
U-Waved-40-45-SSTR4	2,880.00	3,360.00	21.8	2,880.00	2,880.00	192	216.1	0.0
U-Waved-40-45-SSTR5	2,700.00	3,000.00	70.6	2,700.00	2,700.00	1,115.10	1,185.70	-4.3
U-Waved-40-45-SSTR6	2,640.00	2,760.00	4.1	2,640.00	2,760.00	1,793.50	1,800.00	0.0
U-Waved-40-75-SSTR1	6,240.00	6,240.00	0.2	6,240.00	6,240.00	1.5	3.2	0.0
U-Waved-40-75-SSTR2	5,400.00	5,400.00	0.6	5,400.00	5,400.00	1.7	4	0.0
U-Waved-40-75-SSTR3	4,920.00	4,920.00	0.4	4,920.00	4,920.00	1.7	3.8	0.0
U-Waved-40-75-SSTR4	4,320.00	4,320.00	1.6	4,320.00	4,320.00	1.7	4.9	0.0
U-Waved-40-75-SSTR5	3,780.00	3,780.00	4.5	3,780.00	3,780.00	1.8	8	0.0
U-Waved-40-75-SSTR6	3,630.00	3,660.00	3.3	3,660.00	3,660.00	2.1	7.4	0.0
U-Waved-40-105-SSTR1	7,680.00	7,680.00	0.1	7,680.00	7,680.00	1.4	2.8	0.0
U-Waved-40-105-SSTR2	6,960.00	6,960.00	0.2	6,960.00	6,960.00	1.5	3.2	0.0
U-Waved-40-105-SSTR3	6,480.00	6,480.00	0.2	6,480.00	6,480.00	1.5	3.2	0.0
U-Waved-40-105-SSTR4	5,760.00	5,760.00	0.9	5,760.00	5,760.00	1.5	3.9	0.0
U-Waved-40-105-SSTR5	5,190.00	5,220.00	1.1	5,220.00	5,220.00	1.7	4.5	0.0
U-Waved-40-105-SSTR6	5,040.00	5,040.00	0.3	5,040.00	5,040.00	2	4.2	0.0
E-Waved-40-45-SSTR1	4,000.00	4,320.00	2.9	4,020.00	4,320.00	1,795.20	1,800.00	0.0
E-Waved-40-45-SSTR2	3,410.00	3,600.00	3.3	3,420.00	3,600.00	1,794.60	1,800.00	0.0
E-Waved-40-45-SSTR3	3,110.00	3,300.00	1.4	3,120.00	3,300.00	1,796.40	1,800.00	0.0
E-Waved-40-45-SSTR4	2,760.00	2,940.00	102.7	2,760.00	2,940.00	1,695.40	1,800.00	0.0
E-Waved-40-45-SSTR5	2,600.00	2,640.00	481.8	2,640.00	2,640.00	2.2	486	0.0
E-Waved-40-45-SSTR6	2,580.00	2,640.00	98.2	2,580.00	2,640.00	1,699.60	1,800.00	0.0
E-Waved-40-75-SSTR1	4,680.00	5,280.00	2.8	4,680.00	5,280.00	1,795.60	1,800.00	0.0
E-Waved-40-75-SSTR2	3,921.40	4,260.00	6.5	3,960.00	4,260.00	1,791.80	1,800.00	0.0
E-Waved-40-75-SSTR3	3,441.40	3,720.00	2.1	3,480.00	3,720.00	1,796.10	1,800.00	0.0
E-Waved-40-75-SSTR4	3,480.00	3,840.00	22.4	3,480.00	3,840.00	1,775.80	1,800.00	0.0
E-Waved-40-75-SSTR5	3,180.00	3,300.00	387.1	3,180.00	3,300.00	1,411.00	1,800.00	0.0
E-Waved-40-75-SSTR6	2,943.50	3,060.00	45.6	3,000.00	3,060.00	1,752.20	1,800.00	0.0
E-Waved-40-105-SSTR1	6,840.00	6,840.00	0.8	6,840.00	6,840.00	1.3	3.3	-5.8
E-Waved-40-105-SSTR2	5,991.10	6,000.00	73.1	6,000.00	6,000.00	1.4	75.9	-2.9
E-Waved-40-105-SSTR3	5,567.10	5,580.00	35.9	5,580.00	5,580.00	1.4	38.6	-3.1
E-Waved-40-105-SSTR4	6,300.00	6,300.00	0.5	6,300.00	6,300.00	1.4	3.2	-6.3
E-Waved-40-105-SSTR5	5,753.30	5,820.00	11.3	5,760.00	2,760.00	17.3	33.2	0.0
E-Waved-40-105-SSTR6	5,480.00	5,520.00	3	5,520.00	5,520.00	1.6	6.2	0.0

4 Impact of Warehouse Capacities on Vehicle Routing

4.1 Introduction

Deliveries of products to customers within predefined time windows are a common service requirement in many modern supply chains. For example, a timely delivery of shipments is important in B2C distribution channels, such as online groceries (Agatz et al., 2011), retail stores (Spliet & Gabor, 2014) and shipment consolidations in cross-dock facilities (Dellaert et al., 2018). Consequently, many companies impose time windows on the delivery of orders. The way in which these time windows are defined and distributed across customers has an impact on the cost and performance of delivery operations (Quak & de Koster, 2009; Akyol & De Koster, 2013, 2018). Delivery time windows also have significant implications on warehouse operations. Warehouse managers have to ensure that items are picked from warehouse locations and prepared for shipment in a timely manner so that vehicles can depart from the warehouse at an appropriate time. The planning of warehouse operations is particularly challenging because multiple constrained warehouse processes interact, and these processes have to be carefully planned to ensure a timely departure of orders from the warehouse and consequently a timely order delivery at customer locations. However, most literature on vehicle routing overlooks the internal warehouse constraints and only focuses on the distribution of items between the warehouse and delivery locations, either computationally (Laporte, 2009; Spliet & Gabor, 2014) or empirically (Quak & de Koster, 2009; Akyol & De Koster, 2013, 2018). The inherent underlying assumption in these works is that a secondary distribution is more expensive than other related processes in the supply chain, and warehouse managers have sufficient capacity at their disposal to easily absorb and accommodate any routing needs without any additional financial burden. In practice, however, vehicles may not be ready to be loaded when warehouse constraints are not taken into consideration (Rijal et al., 2020). As a result, we have observed cases in practice where the warehouse processes have become overwhelmed to execute the vehicle routes that have been generated without consideration of these warehouse operations. In some cases, the distribution plan was even rendered infeasible as it violated warehouse capacity constraints. These circumstances highlight the conflicting objectives of transportation managers and warehouse managers. Additionally, the costs incurred at warehouses make up a larger share of the overall supply chain costs than transportation (Alicke & Lösch, 2010). This is particularly the case in the retail industry, which is largely affected by delivery time windows (Akyol & De Koster, 2013). Therefore, there exists a need and opportunity for warehouse and transportation managers to jointly optimize warehouse and transportation activities such that more efficient system-wide operations are achieved. The goal of this paper is to determine under which operating conditions managers should optimize warehouse operations jointly or even before the transportation processes.

In the following, we describe the three most important warehouse processes that managers need to consider when constructing vehicle routes with time windows that dictate the execution of these warehouse processes:

i. **Order picking** is the process of retrieving items from storage locations in the warehouse to fulfill customer orders (De Koster et al., 2007). The order picking capacity is restricted by the number of available robots in automated warehouses (Azadeh et al., 2019), or by the number of scheduled order pickers at different times of the day in warehouses where order picking is performed manually (De Koster et al., 2007). Especially in the latter type of warehouses, this capacity fluctuates during the day due to different shifts of employees and break times.

ii. **Staging** is the activity of preparing the items that have been picked for the shipment. It is normally identified for the purpose of consolidating shipments to a single destination. Consequently, this process negates the need for perfect synchronization of order picking and loading of vehicles. The staging area is usually located directly behind the dock doors of a warehouse. In many Asian and Western European warehouses this space is severely restricted in comparison to the daily order picking output of the warehouse. Once a staging area is fully occupied, order picking operations have to be halted and vehicles have to be loaded and dispatched before additional items can be sent to the staging area.

iii. **Loading** is the process of physically moving items onto vehicles and is therefore the last process performed at the warehouse. It serves as the interface between warehouse operations and route planning of delivery vehicles. The loading capacity of vehicles is restricted by the number of available dock doors as well as the number of loading personnel and available equipment in the warehouse. Vehicles can only be dispatched from the warehouse once they are loaded with all items that have to be delivered on their route. Upon vehicle dispatch, the staging space is released for other items to be picked and staged.

The interactions between the warehouse processes and transportation planning is visualized in Figure 4.1. As the staging process is at the heart of the warehouse operations, the size of the staging area as well as the speed at which it can be filled (through order picking) and be vacated (through loading) determine whether these warehouse processes need to be considered when constructing vehicle routes. In particular, a small staging space requires the joint planning of order picking and loading. Furthermore, if vehicles are required to be loaded earlier in order to vacate staging space for later shipments, any additional costs associated with the early use of the vehicle and truck driver cannot be overlooked. Therefore, when warehouses have a limited order picking capacity, a small staging area, or a limited number of loading docks, it becomes necessary to use a more holistic approach to the planning of order deliveries and vehicle routing (Rijal et al., 2020).

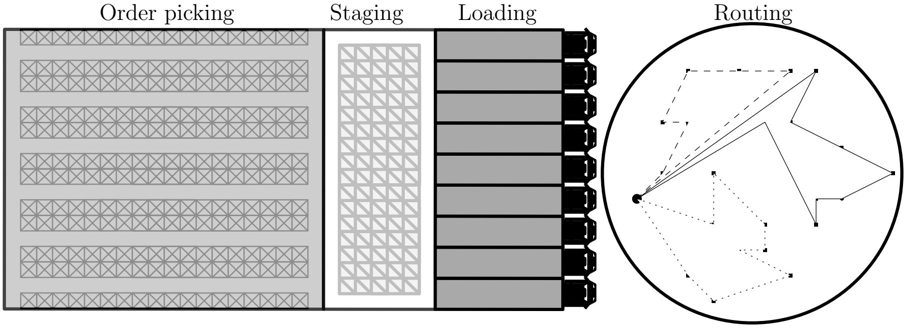


Figure 4.1: Warehousing and transportation processes considered in this paper

The guiding research question for this paper is: “Under which conditions should managers jointly plan warehouse operations and transportation for warehouses with limited order picking, staging and loading capacity such that orders are delivered in their respective time windows?" A handful of papers have partly considered warehouse operations when constructing vehicle routes. Dabia et al. (2019) integrate a limited loading capacity at warehouses in vehicle routing problems with time windows. In their problem formulation, the warehouse has shift- (or time-) dependent loading capacities that are taken into consideration when the vehicle departure from the warehouse is scheduled. They do not consider any additional warehousing processes such as staging or order picking limitations. From the warehousing perspective, Moons et al. (2017, 2019) propose a model that integrates order picking and vehicle routing with time windows. In their problem, customer orders become known to the warehouse dynamically, and the warehouse manager needs to determine a sufficient number of temporary employees to ensure that orders are picked and delivered within their respective time windows. The operational context of their work may appear closely related to ours. However, staging and loading are two important processes that are overlooked in their approach. In e-commerce, packages are small and large staging spaces or loading capacities may not be required. However, staging and loading constraints present significant challenges in many warehouses (and particularly in retail warehouses). Using the terminology from machine scheduling, the works of Dabia et al. (2019) and Moons et al. (2017, 2019) can be considered as single machine scheduling problems (either order picking or loading) together with vehicle routing. In contrast, our problem is equivalent to a two-stage machine scheduling problem (i.e., order picking and loading) with a finite buffer (i.e., staging). This requires different models and solution procedures. We contribute to the literature by developing a more general vehicle routing model that incorporates order picking, staging and loading processes. In addition, we provide a solution methodology that is based on dynamic programming, which can be adapted readily to many applications where warehouse operations are constrained and have an impact on transportation.

The outline of our paper is as follows. In Section 4.2, the relevant literature is discussed. The model formulation of the problem and the algorithm to solve the problem are presented in Section 4.3 and Section 4.4, respectively. Computational experiments and managerial insights generated from a case study are reported in Section 4.5. The paper concludes in Section 4.6 with our final remarks and a discussion how to extend our work.

4.2 Literature Review

In this section, we provide an overview of three streams in the literature that are relevant to our study: vehicle routing with time windows, capacitated warehouse planning, and integrated production and distribution.

4.2.1 Routing problems with time windows

The vehicle routing literature is vast and varied. In this section, we only discuss vehicle routing problems with hard time windows (VRPTW), which require that customers have to be visited within a specified time interval. In particular, we emphasize studies that combine warehouse processes with the routing of vehicles.

Several studies have empirically investigated the impact of hard time window constraints on secondary distribution costs. One of the main insights is that tighter and overlapping time windows across cities lead to a non-linear increase in transportation costs (Muñuzuri et al., 2005; Quak & de Koster, 2007). However, most studies focus on computationally solving VRPTW with the development of exact algorithms that find an optimal solution (Desrochers et al., 1992; Kohl et al., 1999; Kallehauge et al., 2005; Desaulniers et al., 2008; Baldacci et al., 2011) or heuristic procedures that find close-to-optimal solutions (Savelsbergh, 1985; Solomon, 1987; Cordeau et al., 2001; Ropke & Pisinger, 2006a; Vidal et al., 2012). In these studies, the aim is to minimize the total travel distance of vehicles. An exception includes Savelsbergh (1992), who presents an edge-exchange mechanism for the problem where the objective is to minimize the total travel duration since vehicles might have to wait at customer locations before they can unload the orders. Similarly, Tilk & Irnich (2017) propose a dynamic programming approach for the same problem but with only one vehicle. They refer to the problem as the minimum tour duration problem (MTDP).

Regardless of the objective function, whether it minimizes distance or duration, all of the aforementioned studies assume decoupled warehousing and distribution decisions. These problems assume that all units that have to be shipped are already picked and available for shipment at the beginning of the planning horizon (i.e., the *ready time* equals $t = 0$), and the authors ignore the interaction with warehousing processes. A separate class of vehicle routing problems exists where orders have heterogeneous ready times (or release times) (Cattaruzza et al., 2016). However, the release times in these problems are input parameters that are determined *a priori* and are not part of the decision-making process.

Hempsch & Irnich (2008) study routing problems where multiple vehicle routes compete for limited resources. For example, vehicles require loading capacity at the warehouse. As discussed in Section 4.1, this directly affects the departure times of other vehicles from the warehouse. This class of problems are called routing problems with intertour resource constraints. For a variety of routing problems with intertour resource constraints, Irnich (2008b) presents a unified local search procedure that is based on resource extensions along a *giant route* representation of a solution, where a giant route represents a sequence of individual routes based on relevant time and space information. As an example, the sequence in which trucks depart from the warehouse can be the order in which routes are represented in the giant route.

Van der Zon (2017) and Dabia et al. (2019) investigate the so-called vehicle routing problem with time windows and shifts, which belongs to the family of routing problems with intertour resource constraints. In these papers, a shift defines a continuous duration of time during which the scheduled loading capacity of the warehouse remains constant. The aim of the problem is to generate routes that minimize the total travel distance of routes without violating the loading capacity of each shift (i.e., hard limit) or where these loading capacities can be exceeded with some penalty (i.e., soft limit). Similar to the previously described routing problems, this problem also assumes that the order picking and staging capacities of the warehouse are not constrained. Van der Zon (2017) presents an adaptive large neighborhood search algorithm for the problem with a soft loading capacity limit, and Dabia et al. (2019) present an exact branch-price-and-cut algorithm for the problem with a hard loading capacity limit. In the same nomenclature, we can also classify our problem as a routing problem with intertour resource constraints. However, the crucial difference is that we have three separate resources - order picking, staging and loading capacities - which in turn interact with each other as well as with the vehicle routes.

4.2.2 Capacitated warehouse planning

Numerous processes exist upstream of the warehouse and can be classified into replenishment, order picking, packing, loading, shipping and processing returns (De Koster et al., 2007). The specification of each of these processes can vary across warehouses. However, the largest effort by warehouse managers is usually placed on optimizing the order picking process as this contributes to the majority of the warehousing costs (De Koster et al., 2007). Productivity gains in the order picking activities can be achieved by batching orders (Gademann et al., 2001), generating more efficient routes (Roodbergen & de Koster, 2001), and assigning orders to pickers based on personal traits of the pickers (Matusiak et al., 2017).

In the warehousing literature, the general assumption is that the *ready times* of orders (i.e., the time at which an order is picked and ready for shipment) are not constrained. Some exceptions include Elsayed et al. (1993) and Elsayed & Lee (1996) who consider deadlines of orders in automated warehouses as well as Henn et al. (2012), Scholz et al.

(2017) and Van Gils et al. (2018a) who minimize the tardiness of orders in manual warehouses by assigning and sequencing orders to order pickers. Rijal et al. (2020) propose an order picker scheduling problem that generates workforce schedules for multiple shifts by assigning and sequencing orders to order pickers while ensuring that the staging time windows of orders are respected. The staging time windows in their problem arise from the interaction between the time windows when vehicles have to depart the warehouse and the limited staging space. These staging time windows are determined apriori and serve as input for their warehouse optimization problem.

In practice, we observe that departure deadlines of orders that are generated by solving a vehicle routing problem first and then imposing these deadlines on the warehouse puts a significant strain on warehouse processes. In some cases, vehicles cannot depart before the specified departure time because of limitations in the warehouse processes. Jointly optimizing warehouse operations and routes for vehicles does not only overcome these challenges but also provides an opportunity to achieve a lower system-wide cost of operations. However, none of the studies in the warehousing literature investigate the simultaneous planning of vehicle routing together with warehouse operations.

4.2.3 Integrated production and distribution

When the ready times of shipments from facilities are jointly derived together with the problem that considers production scheduling and vehicle routing, this category of problems are called the *integrated production scheduling with outbound distribution scheduling problem* (Chen & Pundoor, 2006; Chen, 2010). In many applications, the perishable nature of products makes it necessary to determine production and delivery decisions in an integrated manner. For instance, the production schedule and distribution processes are jointly optimized for the delivery of mixed concrete (Naso et al., 2007; García & Lozano, 2004; Garcia et al., 2004; García & Lozano, 2005), fixed life adhesives (Armstrong et al., 2008), newspapers (Hurter & Van Buer, 1996; Van Buer et al., 1999; Chiang et al., 2009) or nuclear medicine (Lee et al., 2014). Because of the complexity of the integrated problem, many studies make context-specific simplifying assumptions such as a single production machine, a single vehicle or vehicles with infinite capacities (Potts, 1980; Hall & Shmoys, 1989; Woeginger, 1998; Zdrzałka, 1995; Lee & Chen, 2001).

Almost all papers in the literature consider industrial applications with a fixed number of machines, either single machine (Garcia et al., 2004; Geismar et al., 2008; Garcia et al., 2004; Low et al., 2013, 2014; Amorim et al., 2013; Belo-Filho et al., 2015) or multiple parallel machines (Lee et al., 2014; Farahani et al., 2012; Amorim et al., 2013; Belo-Filho et al., 2015). These problems are categorized by industrial settings where changing the number of machines is not an operational decision. Moons et al. (2017) is the only work that considers the number of employees as part of the decision problem. The authors present a mixed integer linear programming (MILP) model that integrates the order picking problem with the vehicle routing problem in an offline fashion for an online setting where orders with delivery time windows arrive dynamically. These arrival times are considered the release times of orders for the warehouse. The objective of the problem is to minimize the sum of the transportation cost (which is approximated by the duration) and the cost of order picking such that all orders are delivered in time.

To the best of our knowledge, only Moons et al. (2017, 2019) consider the interaction between order picking and transportation activities in an integrated manner. The choice to only consider order picking is justified as the focus of their study is on B2C online warehouses where shipment sizes are usually small and staging and loading operations are not likely to have an impact on order picking or routing decisions. However, in long distance multi-echelon distribution or retail warehousing, orders have a substantial volume and they constrain the staging space and loading operations. As we will see in the following section, incorporating these processes in the vehicle route planning problem is important and definitely not trivial.

4.3 Problem Description and Model Formulation

In this section, we discuss the details of the problem and we present a model formulation of the problem.

4.3.1 Problem description

The description of the vehicle routing component of the problem is identical to the traditional vehicle routing problem with time windows (Solomon, 1987). We extend this description with warehouse processes that constrain the departure time of trucks from the warehouse (i.e., the start time of each route).

Vehicle Routing

The network of the warehouse and all customer locations is represented by a graph $G = (\mathcal{V}', A)$, where $\mathcal{V}'$ indicates the set of vertices and A the set of arcs. A set of n customer locations is indicated by the vertices $\mathcal{V} := \{1, 2, \ldots, n\}$. The warehouse is represented by node 0 to indicate vehicle departures from the facility, and by node $n+1$ to indicate vehicle returns to the warehouse. The complete set of locations is represented by the set $\mathcal{V}' := \{0\} \cup \mathcal{V} \cup \{n+1\}$. The travel time between two locations i and j is given by τ_{ij}. The service time required at location i is indicated by σ_i. The time window for delivery at customer $i \in \mathcal{V}$ is denoted by $[a_i, b_i]$, where the service (unloading) at that customer has to start within this time window. Deliveries to customers are made with homogeneous vehicles, represented by set $\mathcal{K}$, where each vehicle has a capacity of Q units. Customer $i \in \mathcal{V}$ demands a total of q_i units. Split deliveries are not allowed. The planning horizon starts at time 0 and ends at time T^{day}.

Order Picking

Within the planning horizon of T^{day} time units, the order picking activities are performed by order pickers who can start their shift at different times of the day. The number of available order pickers at the warehouse fluctuates over time due to these shifts as well as their break times. Accordingly, we discretize the planning horizon into a set of time segments $\mathcal{S}$, where a time segment $s \in \mathcal{S}$ indicates a time interval during which the number of available order pickers is constant. In particular, the beginning and ending of a time segment $s \in \mathcal{S}$ is given by T_s^{beg} and T_s^{end} , respectively, and the number of order pickers within this segment is indicated by p_s. For the simplicity of exposition, we define $\mathrm{T}_s^{end} := \mathrm{T}_{s'}^{beg}$ when time segment s' immediately succeeds time segment s.

In many warehouse environments, the number of units in an individual order can be large and the pick routes can be long (e.g., for brick-and-mortar retailers). To determine the warehouse operations associated with such orders, it is usually sufficient to include the total number of products in an order (i.e., the order size) rather than including specific information about individual products (such as the product locations and distances between product locations in the warehouse). For instance, Çeven & Gue (2015) use approximate order picking times to represent the workload of orders in an order picking system. Similarly, we assume that order pickers are identical and are capable of picking orders at a rate of μ units per picker per time unit. In a separate problem (outside the scope of our study), items can be combined to pick routes for order pickers with the use of a batching algorithm.

As a result, the total number of units that can be picked at any given time in the planning horizon can be derived by piece-wise linear functions of the time segments. For a time segment $s \in \mathcal{S}$, the number of units that can be picked per time unit is given by $\beta_s := p_s\mu$. For a planning horizon with four time segments, Figure 4.2 illustrates how the total order picking capacity impacts the order picking output at different time segments. The height where the line of a time segment s starts is given by the overall number of units that can be picked until the beginning of that time segment, which equals $\alpha_s := \sum_{s'=2}^{s} (\mathrm{T}^{beg}_{s'} - \mathrm{T}^{beg}_{s'-1})\beta_{s'}$ for $s \in \mathcal{S} \setminus \{1\}$ and $\alpha_1 := 0$. Note that the cumulative order picking capacity is an upper limit on the order picking output of the warehouse (as illustrated by the solid (blue) line in Figure 4.2), which is only achieved if there are no restrictions on the staging capacity of the warehouse.

Staging

In large warehouses, staging spaces may be sufficiently large so that no restrictions are imposed on the order picking process. However, space is limited in most warehouses. Therefore, staging areas are commonly restricted, and they impose limitations on the order picking output.

The staging space acts as a buffer between the order picking and loading (or dispatching) processes. The warehouse has a limited staging capacity of Z units at any time. If the staging area is fully occupied, order picking activities have to be halted until vehicles are loaded and dispatched to vacate some space in the staging area. For example, the number of units in the staging area reaches the staging capacity in the second time segment of Figure 4.2. As a result, the adjusted order picking output is indicated by the horizontal dashed (red) line. This illustrates the trade off between routing and warehousing operations: the overall order picking output of the warehouse can increase if vehicles are loaded earlier to free the staging space. This is indicated by the dash-dotted (brown) line in Figure 4.2. However, this may incur additional transportation costs.

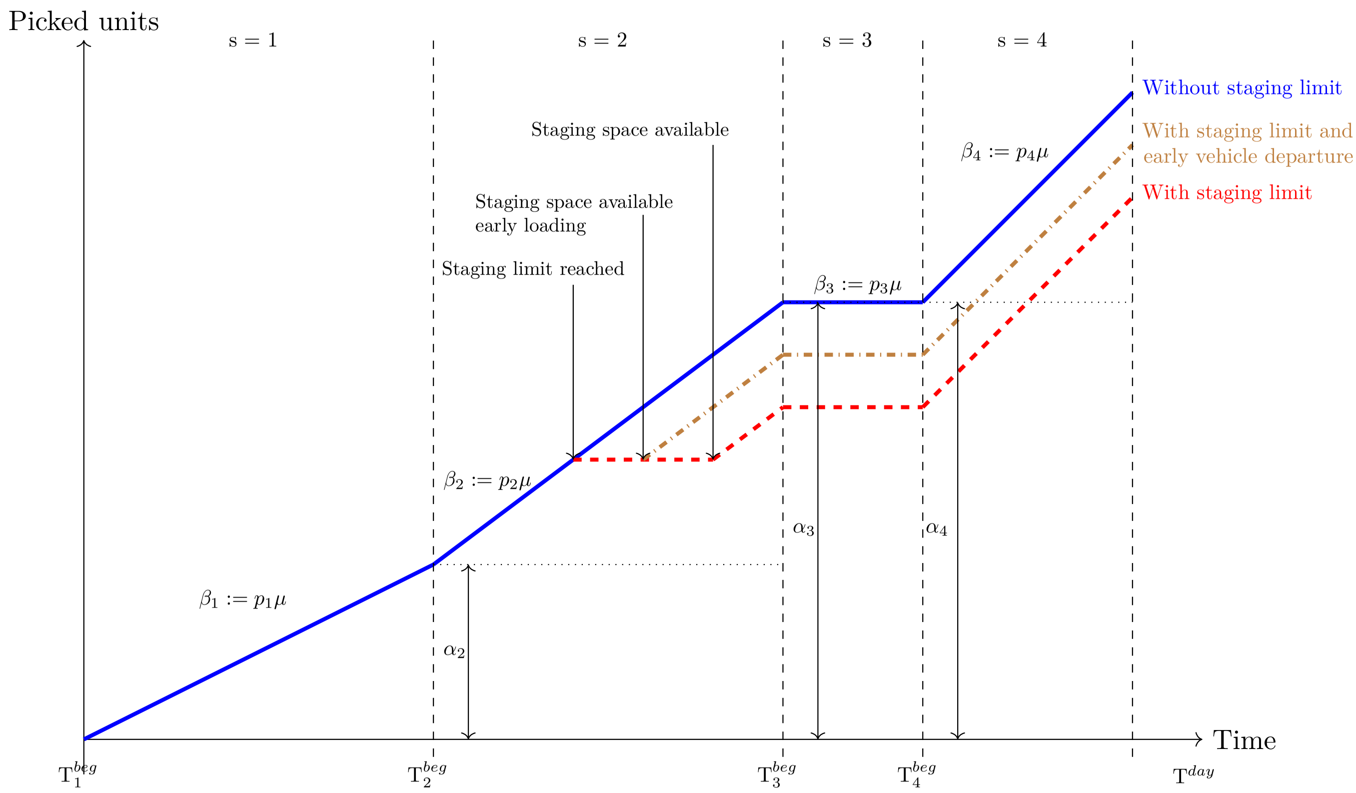


Figure 4.2: Total order picking output given by piece-wise linear functions of the order picking capacity by time segments

Loading

Trucks are loaded at dock doors, where $\mathcal{D}$ represents the set of identical dock doors. The loading speed at each dock door is λ units per unit time. This loading capacity is determined by the type of units handled at the warehouse and the choice of the equipment used for loading the units. We assume that the loading capacity is significantly larger than the order picking capacity of the warehouse (i.e., $\lambda \gg \beta_s$ for all $s \in \mathcal{S}$). This assumption is not restrictive as the order picking process is typically more time consuming than the loading process.

Cost of operation

As illustrated in Figure 4.2, warehouse operations can be more productive when a truck departs early from the warehouse. As a result, a truck might have to wait before it can deliver the shipment within the delivery time window at a certain customer location. The usual objective in the VRPTW (i.e., minimizing total travel distance) does not capture any of these additional waits. Therefore, the objective in our problem is to minimize the time during which vehicles are used, from the instant loading starts until they return to the warehouse after completion of the route, while ensuring that the warehouse operations are feasible. Such an objective function is similar to the objective in the MTDP (Tilk & Irnich, 2017).

4.3.2 Model formulation

The notation for all input parameters of our vehicle routing problem with warehouse considerations is summarized in Table 4.1, whereas all decisions variables (including simplifying notations) are presented in Table 4.2.

In the remainder of this section we discuss the objective function and all constraints associated with the warehousing and transportation activities that are included in our problem formulation.

Objective function

The objective in our problem formulation is to minimize the total duration that vehicles are occupied for loading and transportation (which can include wait times

Table 4.1: Notations for the input parameters of the problem (both warehouse and transportation parameters)

Vehicle routing parameters	
$\mathcal{V}$	set of customer locations
$\mathcal{V}'$	set of customer locations and the warehouse
$\mathcal{A}$	set of arcs
$\mathcal{K}$	set of vehicles
σ_i	service time at location $i \in \mathcal{V}'$
τ_{ij}	travel time between locations i and $j \in \mathcal{V}'$
a_i	start of the service time window at location $i \in \mathcal{V}'$
b_i	end of the service time window at location $i \in \mathcal{V}'$
q_i	number of units demanded by location $i \in \mathcal{V}'$
Q	capacity of a vehicle

Warehouse parameters	
$\mathcal{S}$	set of time segments in the planning horizon
T_s^{beg}	start time of time segment $s \in \mathcal{S}$
T_s^{end}	end time of time segment $s \in \mathcal{S}$
p_s	number of order pickers available in time segment $s \in \mathcal{S}$
μ	order picking rate of one order picker (i.e., number of units picked per unit time)
β_s	order picking rate in time segment $s \in \mathcal{S}$ (equals μp_s)
Z	maximum number of units that can be physically located in the staging area at any time
$\mathcal{D}$	set of outbound dock doors
λ	loading rate (i.e., number of units loaded per hour)

before a vehicle can deliver the products in the designated time windows of the customers to visit):

$$\min \sum_{k \in \mathcal{K}} \left(t_{n+1}^k - l^k \right) \tag{4.1}$$

Routing constraints

The transportation of units to customer locations by vehicles, where the delivery is restricted to a given time interval, is represented by the constraints (4.2) until (4.9).

Table 4.2: Notation for the decision variables of the problem (including simplifying notations)

Decision variables	
x_{ij}^k	is 1 if vehicle k travels along arc $(i, j) \in \mathcal{A}$, else 0
l^k	start time to load truck $k \in \mathcal{K}$
t_i^k	start time of order delivery at node $i \in \mathcal{V}'$ by vehicle $k \in \mathcal{K}$
c^k	number of units available at the staging area when the loading of truck $k \in \mathcal{K}$ starts
o_k^s	total number of units that can be picked until the loading of vehicle $k \in K$ starts if the vehicle is loaded in time segment $s \in \mathcal{S}$, else 0
y_s^k	is 1 if the loading of vehicle $k \in K$ starts in time segment $s \in \mathcal{S}$, else 0
$u_d^{k'k}$	is 1 if the loading of vehicle $k \in \mathcal{K} \cup \{0\}$ starts after vehicle $k' \in \mathcal{K} \cup \{0\}$ is loaded at the same dock door $d \in \mathcal{D}$ (where $k = 0$ indicates a dummy vehicle index)

Simplifying notations	
$\delta^+(i)$	set of arcs emerging from node i
$\delta^-(i)$	set of arcs incoming to node i
$x(\delta^+(i))^k$	$:= \sum_{j \in \mathcal{V}} x_{ij}^k$
$x(\delta^-(i))^k$	$:= \sum_{j \in \mathcal{V}} x_{ji}^k$
$q(k)$	$:= \sum_{i \in \mathcal{V}} q_i x(\delta^+(i))^k$

$$\sum_{k \in K} x(\delta^+(i))^k = 1 \qquad \forall i \in \mathcal{V} \tag{4.2}$$

$$x(\delta^+(0))^k = x(\delta^-(n+1))^k = 1 \qquad \forall k \in \mathcal{K} \tag{4.3}$$

$$x(\delta^+(i))^k = x(\delta^-(i))^k \qquad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{4.4}$$

$$t_i^k + \sigma_i + \tau_{ij} \leq t_j^k + M(1 - x_{ij}^k) \qquad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K} \tag{4.5}$$

$$a_i \leq t_i^k \leq b_i \qquad \forall i \in \mathcal{V}', k \in \mathcal{K} \tag{4.6}$$

$$q(k) \leq Q \qquad \forall k \in \mathcal{K} \tag{4.7}$$

$$x_{i,j}^k \in \{0, 1\} \qquad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K} \tag{4.8}$$

$$0 \leq t_i^k \leq \mathrm{T}^{day} \qquad \forall i \in \mathcal{V}', k \in \mathcal{K} \tag{4.9}$$

Constraints (4.2) ensure that each customer node i is visited exactly once. Constraints (4.3) and (4.4) serve as flow conservation constraints: each vehicle k has to leave and return to the warehouse, and if a vehicle k arrives at node i it also has to leave node i, respectively. A shipment should be delivered at node j after the delivery at node i (i.e., the previous customer) is completed by the same vehicle k plus the travel time between the two locations by constraints (4.5), and the delivery should occur within the specified time window according to constraints (4.6). The number of units delivered in a route cannot exceed the loading capacity of a vehicle k based on constraints (4.7). Constraints (4.8) and (4.9) indicate the domain of the decision variables for the routing component of the problem.

Order picking and staging constraints

Constraints (4.10) until (4.21) capture the interaction between the order picking and staging activities and the time when the loading process of a vehicle can start.

$$l^{k-1} \leq l^k \qquad \forall k \in \mathcal{K} \setminus \{1\} \tag{4.10}$$

$$l^k \geq \sum_{s \in \mathcal{S}} T_s^{beg} y_s^k \qquad \forall k \in \mathcal{K} \tag{4.11}$$

$$l^k \leq \sum_{s \in \mathcal{S}} T_s^{end} y_s^k \qquad \forall k \in \mathcal{K} \tag{4.12}$$

$$\sum_{s \in \mathcal{S}} y_s^k = 1 \qquad \forall k \in K \tag{4.13}$$

$$o_s^k = \big(\alpha_s + \beta_s(l_k - T_s^{beg})\big) y_s^k \qquad \forall s \in \mathcal{S}, k \in \mathcal{K} \tag{4.14}$$

$$c^1 \leq \sum_{s \in \mathcal{S}} o_s^1 \tag{4.15}$$

$$c^k \leq c^{k-1} - q(k-1) + \sum_{s \in \mathcal{S}} o_s^k - \sum_{s \in \mathcal{S}} o_s^{k-1} \qquad \forall k \in \mathcal{K} \setminus \{1\} \tag{4.16}$$

$$c^k \leq Z \qquad \forall k \in \mathcal{K} \tag{4.17}$$

$$q(k) \leq c^k \qquad \forall k \in \mathcal{K} \tag{4.18}$$

$$0 \leq c^k, l^k \qquad \forall k \in \mathcal{K} \tag{4.19}$$

$$0 \leq o_s^k \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{4.20}$$

$$y_s^k \in \{0, 1\} \qquad \forall k \in \mathcal{K}, s \in \mathcal{S} \tag{4.21}$$

Constraints (4.10) determine the sequence in which the loading of vehicles start. Constraints (4.11) and (4.12) determine the corresponding time segment in which the loading of vehicle k starts. Note that the optimal time to start the loading of a vehicle is also affected by the availability of dock doors, which is explained later. Constraints (4.13) force each vehicle k to be loaded in one of the time segments $s \in \mathcal{S}$. Even unused vehicles are scheduled to be loaded at one of the time segments but with zero units. The total number of units that can be picked until the loading of vehicle k starts in time segment s is given by constraints (4.14). Together with constraints (4.14), constraints (4.15) and (4.16) determine the number of units at the staging area when the loading for vehicle k starts, and constraints (4.17) limit the number of picked units that can be staged at any time. Finally, constraints (4.18) ensure that vehicle k can only be loaded with the number of units that are available at the staging lanes. Constraints (4.19) to (4.21) define the domain of the decision variables associated with the order picking and staging processes.

Dock-door specific loading capacity constraints

The sequence in which the loading of vehicles starts at each dock door $d \in \mathcal{D}$ is recorded by the decision variables $u_d^{k'k}$. The following constraints represent dock-door specific loading activities.

$$t_0^k \geq l^k + \frac{q(k)}{\lambda} \qquad \forall k \in \mathcal{K} \tag{4.22}$$

$$l^k \geq t_0^{k'} - M(1 - u_d^{k'k}) \qquad \forall k, k' \in \mathcal{K}, d \in \mathcal{D} \tag{4.23}$$

$$\sum_{k \in \mathcal{K}} u_d^{0k} \leq 1 \qquad \forall d \in \mathcal{D} \tag{4.24}$$

$$\sum_{d \in \mathcal{D}} \sum_{k' \in \{0\} \cup \mathcal{K}} u_d^{k'k} = 1 \qquad \forall k \in \mathcal{K} \tag{4.25}$$

$$\sum_{k' \in \{0\} \cup \mathcal{K}} u_d^{k'k} = \sum_{k' \in \{0\} \cup \mathcal{K}} u_d^{kk'} \qquad \forall k \in \mathcal{K}, d \in \mathcal{D} \tag{4.26}$$

$$u_d^{k'k} \in \{0, 1\} \qquad \forall k, k' \in \mathcal{K} \cup \{0\}, d \in \mathcal{D} \tag{4.27}$$

Constraints (4.22) ensure that vehicle k can only depart from the warehouse after it is fully loaded. Constraints (4.23) allow for a subsequent vehicle k' to start loading at the same dock door d when the preceding vehicle k has departed. Constraints (4.24) ensure that at most one vehicle can be labeled as the first to be processed at dock door d. Finally, constraints (4.25) and (4.26) preserve that each vehicle k is assigned to once to a dock door and there is another (dummy) vehicle assigned before and after vehicle k at the assigned dock door d, respectively. These constraints are comparable to the flow conservation constraints in constraints (4.4). Constraints (4.27) define the domain of the additional decision variables.

The loading operations modelled by constraints (4.22) until (4.27) make the assumption that the loading resources (e.g., people, pallet trucks) are not shared between dock doors. This is typical for warehouses where truck drivers load the vehicle themselves. This implies that the loading operations at a dock door are not constrained by similar operations at other doors. However, if the same set of loading resources is shared by all dock doors, this can be easily modeled by assuming that the warehouse has only one dock door with a loading capacity equal to λ and the constraints (4.23) and (4.24) as well as constraints (4.26) and (4.27) have to be adjusted accordingly.

Linearization of order picking constraints

Note that the order picking constraints (4.14) are non-linear as both l_k and y_s^k are decision variables. In order to rewrite the problem formulation as a linear model, these constraints can be replaced by the following inequalities:

$$o_s^k \leq \big(\alpha_s + \beta_s(\mathrm{T}_s^{end} - \mathrm{T}_s^{beg})\big) y_s^k \qquad \forall s \in \mathcal{S}, k \in \mathcal{K} \qquad (4.28)$$

$$o_s^k \geq \alpha_s y_s^k \qquad \forall s \in \mathcal{S}, k \in \mathcal{K} \qquad (4.29)$$

$$o_s^k \leq \alpha_s y_s^k + \beta_s(l_k - \mathrm{T}_s^{beg}) \qquad \forall s \in \mathcal{S}, k \in \mathcal{K} \qquad (4.30)$$

$$o_s^k \geq \alpha_s y_s^k + \beta_s(l_k - \mathrm{T}_s^{beg}) - \mathrm{T}^{day} \max_{s \in \mathcal{S}}\{\beta_s\}(1 - y_s^k) \qquad \forall s \in \mathcal{S}, k \in \mathcal{K} \qquad (4.31)$$

Symmetry breaking constraints

The aforementioned problem formulation still suffers from issues of symmetry that can negatively impact the performance of commercial solvers when solving the problem. The routing component of the problem has less symmetry since the start times to load a vehicle has an impact on the objective function value. Additionally, vehicles are sequenced in an increasing order of the start times when they are loading, see constraints (4.10). Symmetry in the assignment of trucks to dock doors can be partially eliminated by incorporating the following equalities into the model formulation:

$$u_k^{0,k} = 1 \qquad \forall k \in \{1, \dots, |\min\{|\mathcal{D}|, |\mathcal{K}|\}\} \tag{4.32}$$

Constraints (4.32) ensure that the first series of vehicles to start loading are assigned to dock doors in an ascending order (i.e., the first vehicle is assigned to the first dock door, the second vehicle to the second dock door, and so on) until all vehicles are assigned or all dock doors have been exhausted. This pattern of dock-door assignments can only be used as valid equalities for the first vehicles to be loaded at each dock door.

The proposed problem is NP-complete as the VRPTW is a special case of our problem. When we only consider the warehousing operations, the problem is comparable to a two-stage flow shop problem with a finite buffer. The first stage corresponds to the order picking operations, staging is the limited buffer, and the loading operations is the second stage of the problem.

4.4 Solution Procedure

We propose to solve the problem formulated in the previous section as a linear programming with dynamic programming and the giant route representation introduced by Irnich (2008b). Unlike other routing problems, where the cost of an individual route is not affected by the start time when other routes are loaded, the solution approach to our problem requires a solution representation that not only captures information on individual routes but also the sequence in which the vehicles are loaded at the warehouse. When multiple vehicles *compete* for the same resources to execute distinct routes, the start times when vehicles are loaded and the associated use of limited resources can be readily represented by and extracted from a giant route (Irnich, 2008b). The giant route representation that we propose enables us to readily encode the loading times and departure times of vehicles and to evaluate the cost and feasibility of the solution. The solution procedure relies on a number of key concepts that we outline first to build intuition for the approach.

For any given route R, the *minimal duration* represents the shortest time required to load the vehicle with the units it needs to deliver, to visit the customer nodes and to return to the warehouse. This includes any waiting time at customer locations if necessary to adhere to the start of the delivery time windows. Additionally, a route incurs only the minimal duration if the loading of the vehicle starts within a time window between the so-called earliest-minimal-duration-loading-time (ELT_R) and the latest-minimal-duration-loading-time (LLT_R). A vehicle cannot execute route R without violating the delivery time window at one or more customer locations visited in route R if the loading of the vehicle starts after LLT_R. In contrast, it is feasible for a vehicle to start the loading process before ELT_R, but this incurs an additional cost as the vehicle has to wait before it can deliver the items at one or more customer locations. Figure 4.3 illustrates how the cost of executing a route R can be determined as a function of ELT_R and LLT_R. The values of ELT_R and LLT_R depend on the customer locations visited in route R. Therefore, these values are determined dynamically in the solution approach by adapting the methodology proposed by Tilk & Irnich (2017) for the MTDP.

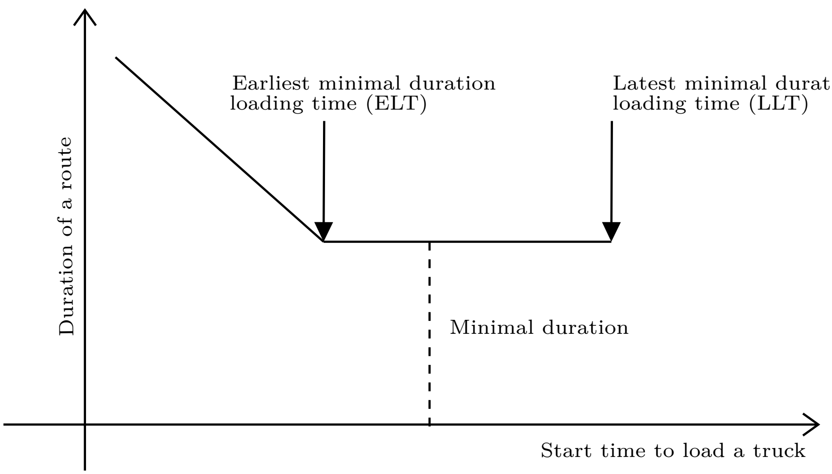


Figure 4.3: Duration cost of a route as a function of the time when the truck is loaded

The solution procedure develops a complete solution by progressively adding customer locations to routes and routes to the giant route representation. The sequencing of the routes is determined in descending order of the start times when a vehicle is loaded for the route, i.e, the first route added to the giant route is loaded last at the warehouse. In general, it is better for a vehicle to start loading as late as possible (i.e., at LLT_R) as this leaves the most time for the order picking process without additional costs. However, postponing the loading of all vehicles can capacitate warehouse operations and render the entire operations infeasible. The proposed solution approach aims to avoid this outcome by checking and calculating the optimal times when the loading of vehicles start. This verification mechanism requires to evaluate the utilization of loading, staging and order picking resources. To explain the dynamic programming procedure, we first present the augmented graph, labels, extension functions and acceleration techniques that are used in the dynamic programming algorithm. An overview how these components are combined in the DP algorithm is provided in Section 4.7.1.

4.4.1 Augmented graph

The graph introduced in Section 4.3 requires a number of adjustments to use it in our solution approach. Consider the set of vertices $\mathcal{V}'$ that specifies the set of customer nodes as well as the start and end location of each route. Additionally, vertex $\tilde{0}$ represents the start of the planning horizon and is added to $\mathcal{V}'$. All customer nodes in $\mathcal{V}$ are connected to each other by arcs. Furthermore, node $n+1$ has outgoing arcs to each vertex in $\mathcal{V}$ and one incoming arc originating from node 0. Similarly, node 0 has incoming arcs originating from each vertex in $\mathcal{V}$, and two outgoing arcs: one to node $n+1$ to indicate the start of a new route and one to node $\tilde{0}$ to indicate the beginning of the operations. An illustration of this augmented graph with three customer nodes is represented in Figure 4.4. The solution in the figure has one giant route where route $(0, 3, n+1)$ is loaded at the warehouse before route $(0, 1, 2, n+1)$ is loaded. Note that a vehicle completes only one route and routes are added to the giant route representation from the right-to-left direction. The associated giant route representation of the final solution is represented by $|\tilde{0}|0|3|n+1|0|1|2|n+1|$, where "|" separates the sequence of the nodes that are visited.

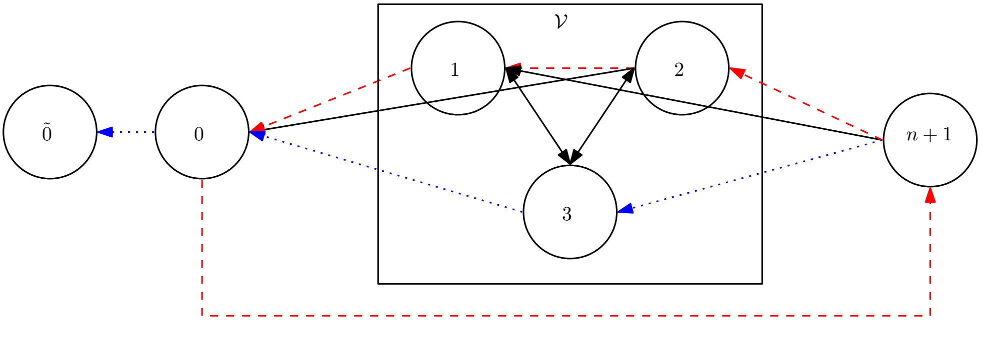


Figure 4.4: An example of an augmented graph for the dynamic programming procedure

4.4.2 Labels

The dynamic program (DP) uses labels to store information on partial solutions. The h^{th} label with node i as the first visited node in the partial solution is indicated by the label $\mathcal{L}_h^i = (i, R, q_h^R, l_h^i, l_h^0, c_h^i, c_h^0, \tilde{m}_h^i, T_i^{bw}, \kappa_h, T_h^{Av}, (V_h^1, \ldots, V_h^{|\mathcal{V}|}))$, where

- i is the first vertex (or customer node) visited in the partial solution
- R is the incomplete route indicated by the set of vertices in the partial solution, i.e., a route starting at node $n+1$ which has not yet reached node 0
- q_h^R is the number of units delivered by a vehicle in route R
- l_h^i is the time when the loading process starts for route R if node i is the first node to visit in route R
- l_h^0 is the time when the loading process starts for the route that succeeds the incomplete route R (determined with the DP)
- c_h^i is the number of units at the staging area when the loading process starts for route R at time l_h^i if node i is the first customer to visit on route R
- c_h^0 is the number of units at the staging area when the loading process starts for the route that succeeds route R
- $\tilde{m}_h^i$is the total duration required for loading and transportation of routes added to the partial solution so far, where node i is the first customer to visit in route R
- $T_i^{bw} = (T_i^{bw,time}, T_i^{bw,dur}, T_i^{bw,help})$ represents the time resources for node i, where
  - $T_i^{bw,time}$ is the latest start time to visit node i in route R
  - $T_i^{bw,dur}$ is T_{day} minus the minimal duration required for route R
  - $T_i^{bw,help}$ is T_{day} minus the earliest possible completion time of the route with the minimal duration
- κ_h is the number of vehicles used so far in the partial solution
- $T_h^{Av} := (T_{h,1}^{Av}, \ldots, T_{h,|\mathcal{D}|}^{Av})$ is a vector of size $|\mathcal{D}|$ where $T_{h,d}^{Av}$ represents the latest time dock door $d \in \mathcal{D}$ is available
- V_h^v is 1 if node $v \in \mathcal{V}$ is visited in the partial solution

A label represents a feasible solution if the time resources at the nodes are within the following bounds: $T_i^{bw,time} \in [a_i, b_i]$, $T_i^{bw,dur} \in [0, \mathrm{T}^{day}]$, $T_i^{bw,help} \in [0, \mathrm{T}^{day}]$, $q_h^R \leq Q$, and $\kappa_h \leq |\mathcal{K}|$. The first label is initialized at node $n+1$ as $\mathcal{L}_h^{n+1} := (n+1, \emptyset, 0, \mathrm{T}^{day}, \mathrm{T}^{day}, 0, 0, T_i^{bw}, 1, T_h^{Av}, (V_h^1, \ldots, V_h^{|\mathcal{V}|}))$, where $T_i^{bw} = (\mathrm{T}^{day}, \mathrm{T}^{day}, \mathrm{T}^{day})$, and $T_{h,d}^{Av} := \mathrm{T}^{day}$ for all $d \in \mathcal{D}$.

4.4.3 Resource extension functions

Labels are created and updated with the use of resource extension functions along four different types of nodes in the augmented graph, where each type represents a different dimension of the solution: (i) times when vehicles are loaded and departure from the warehouse, (ii) times when customer locations are visited, (iii) times when vehicles return to the warehouse, and (iv) the time when all vehicles have completed the operations for the planning horizon. Unique extension functions (and procedures) of resources are specified to transition from each of these nodes in the augmented graph, which are discussed in the remainder of this subsection.

4.4.3.1 Extension to nodes in $\mathcal{V}$

The extension to a node $v \in \mathcal{V}$ represents a visit to the customer location. This extension procedure is the most involved and consists of several subroutines: (i) check the feasibility of a given route R without considering the warehousing operations, (ii) determine the latest-minimal-duration-loading-time LLT_R, (iii) calculate the minimal cost of route R and (iv) determine the optimal start time to load the vehicle for route R in the partial solution.

Feasibility check of route R ignoring warehouse constraints: To determine the feasibility of route R without considering the warehouse constraints, the extension function $f_{bw}(T_j, i)$ extends the time resources T_j^{bw} from node j to node i along the arc (i, j) for nodes $i \in \mathcal{V}$:

$$T_i^{bw,time} = f_{bw}^{time}(T_j, i) := \min\{T_j^{time} - \tau_{ij} - \sigma_i, b_i\} \tag{4.33}$$

$$T_i^{bw,dur} = f_{bw}^{dur}(T_j, i) := \min\{T_j^{bw,dur} - \tau_{ij} - \sigma_i, T_j^{bw,help} - \tau_{ij} - \sigma_i + b_j\} \tag{4.34}$$

$$T_i^{bw,help} = f_{bw}^{help}(T_j, i) := \min\{T_j^{bw,dur} - a_j, T_j^{bw,help}\} \tag{4.35}$$

If the time resources T_i^{bw} are within the feasibility bounds for node i, then the route can be completed by the vehicle when all warehouse constraints are completely ignored. The new customer location i is added to $R := \{i\} \cup R$ and $q_h^R := \sum_{i \in R} q_i$. Similarly, $V_h^i := 1$. However, the feasibility and cost of the solution also depend on the latest time that the loading of the vehicle can start for route R (i.e., l_h^i) and the number of units required at the staging lane when the loading starts (i.e., c_h^i). To determine l_h^i, we first need to determine LLT_R.

Calculation of LLT_R: To determine LLT_R, the resources in an auxiliary straight line graph need to be extended with the use of a forward propagation function. This is based on the premise that node i is the first node to visit in the route that is being created. Let $\vartheta_R(\tilde{n})$ denote the last node to visit in route R that consists of $\tilde{n}$ nodes to visit. The auxiliary graph contains two additional nodes: node 0 and node $\tilde{o}$. Node $\tilde{o}$ indicates the start time to load the vehicle for route R, and node 0 indicates the departures of the vehicle from the warehouse (i.e., when the loading is completed). For the dummy node $\tilde{o}$, the earliest and latest arrival times are 0 and T_{day}, respectively. The arc travel time $\tau_{\tilde{o}0} := q_h^R/\lambda$ indicates the time required to load the vehicle for route R. Finally, the service time at node $\tilde{o}$ equals $\sigma_{\tilde{o}} := 0$. An example of the auxiliary graph is visualized in Figure 4.5. Note that the auxiliary graph is a straight line graph that is generated with every extension, and it has a different interpretation and it is significantly smaller than the augmented graph that has been explained earlier.

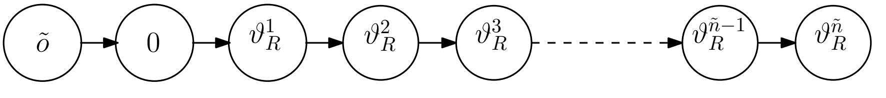


Figure 4.5: Auxiliary graph to determine the latest minimal duration loading time LLT_R

We can determine the minimal duration loading time and LLT_R for a route R with either the forward slack time method (Savelsbergh, 1992), the time warp method (Vidal et al., 2013) or the forward propagation function (Tilk & Irnich, 2017). We use the latter approach as it is symmetric to the backward propagation that has been explained earlier. The resources vector in the forward propagation function is represented by $T_i^{fw} = (T_i^{fw,time}, T_i^{fw,dur}, T_i^{fw,help})$. The semantics of the resources are as follows:

- $T_i^{fw,time}$ is the start time to visit (or service) node i

- $T_i^{fw,dur}$ is the minimum duration from the start time a vehicle for route R is loaded until the vehicle returns to the warehouse
- $T_i^{fw,help}$ is the latest possible start time a vehicle for route R can be loaded while ensuring that it can visit all customer locations within their respective time windows

The resource vector is initialized at node $\tilde{o}$ as $(0, 0, T_{day})$. The extension function $f_{fw}(T_j, i)$ extends the resources from node i to node j along the arc (i, j) in a forward manner until it reaches node $\vartheta_R(\tilde{n})$:

$$T_j^{time} = f_{fw}^{time}(T_i, j) := \max\{T_i^{time} + \sigma_i + \tau_{ij}, a_j\} \tag{4.36}$$

$$T_j^{dur} = f_{fw}^{dur}(T_i, j) := \max\{T_i^{dur} + \sigma_i + \tau_{ij}, a_j - T_i^{help}\} \tag{4.37}$$

$$T_j^{help} = f_{fw}^{help}(T_i, j) := \min\{b_j - (T_i^{dur} + \sigma_i + \tau_{ij}), T_i^{help}\} \tag{4.38}$$

The latest time a vehicle can be loaded without violating the delivery time windows of the customers in route R is given by $LLT_R := T_{\vartheta_R(\tilde{n})}^{help}$.

So far, the procedure has determined if the incomplete route R is feasible without considering the warehouse operations, it calculated LLT_R and determined the minimum duration of the route without considering other routes that are already added to the partial solution. Next, the start time to load a vehicle (i.e., l_h^i) is adjusted (if necessary) to ensure that the warehouse operations are optimized and that it is feasible to execute route R as well as the other routes that are already added to the partial solution.

Determine and optimize l_h^i: The latest start time to load a vehicle (i.e., l_h^i) is determined in two stages. First, only the loading resources are considered to determine l_h^i. Next, the order picking and staging resources are considered and the value of l_h^i is adjusted if necessary to satisfy all capacity constraints related to the warehousing processes.

The value of l_h^i is restricted by LLT_R as the vehicle cannot be loaded later than this value. In addition, a vehicle for route R cannot be loaded after any other route that is already included to the giant route (i.e., not after l_h^0) as the solution is constructed in a descending order when the loading of vehicles starts. Finally, the availability of dock doors also limits when the loading of the vehicle for a route can start. Consequently, the latest start time to load a vehicle is given by

$$l_h^i = \min\{LLT_R, l_h^0, T_{h,\tilde{d}}^{Av} - \frac{q_h^R}{\lambda}\}, \tag{4.39}$$

where $\tilde{d} := \arg\max_{d \in \mathcal{D}}\{T^{Av}_{h,d}\}$ is the dock door that facilitates the latest start time to load the vehicle at the warehouse.

Next, we add the staging and order picking constraints at the warehouse to optimize the start time when to load the vehicle for route R. Let the function $o(t)$ denote the cumulative order picking capacity of the warehouse from the beginning of the planning horizon until time t. The optimal value of l^i_h is then adjusted based on the following three rules:

Rule 1: If $o(l^i_h) < q^R_h + \sum\limits_{i \in \{i|V^i_h=0, \forall i \in V\}} q_i + c^0_h - (o(l^0_h) - o(l^i_h))$, then the incomplete route R is infeasible. The order picking operations has time until l^i_h to pick the units for the customers in route R and the units for the customers not yet considered in the partial solution (i.e., , q^R_h and $\sum\limits_{i \in \{i|V^i_h=0, \forall i \in V\}} q_i$ units, respectively). Additionally, at the start time to load the vehicle for route R, $c^0_h - (o(l^0_h) - o(l^i_h))$ units are required to be present at the staging area for the routes already added to the partial solution. If the cumulative order picking capacity $o(l^i_h)$ is smaller than this required number of units, the solution is not feasible at any cost and the label can be discarded. Otherwise, rules 2 and 3 are applied to the solution.

Rule 2: If $q^R_h + c^0_h - (o(l^0_h) - o(l^i_h)) \leq Z$, then the number of units that are necessary to be present at the staging area at the time when loading the vehicle for route R is less than the staging limit. Consequently, the value of l^i_h is feasible and the number of units at the staging lane can be set to $c^i_h := q^R_h + c^0_h - (o(l^0_h) - o(l^i_h))$.

Rule 3: If $q^R_h + c^0_h - (o(l^0_h) - o(l^i_h)) > Z$, then the staging capacity is violated. Consequently, the start time to load the vehicle for route R has to be moved forward to the latest time that ensures that the vehicles for the routes that are loaded after the vehicle for route R is loaded have sufficient time for order picking without violating the staging limit. The new start time to load the vehicle for route R is then given by $l^i_h := \max_t\{t|o(t) \leq o(l^0_h) + Z - q^R_h - c^0_h\}$ and $c^i_h := \min\{Z, o(l^i_h)\}$. Rule 1 needs to be reapplied to check the feasibility whether the new value of l^i_h satisfies the order picking capacity.

The worst-case complexity of the extension functions for all nodes in $\mathcal{V}$ is given by $O(n + log(s))$, where n is the number of customer nodes in a vehicle route and s is the number of time segments in the order picking planning horizon.

4.4.3.2 Extension to node 0

An extension to node 0 indicates that an individual route in the partial solution has been finalized. As a result, the associated costs and start time to load the vehicle for the route can be calculated. We can set $l_h^0 := l_h^j$ and $c_h^0 := c_h^j$. Additionally, we need to determine the cost of the partial solution.

$$T_i^{bw,time} = g_{bw}^{time}(T_j, i) := \min\{T_j^{time} - \tau_{ij} - \frac{q_h^R}{\lambda}, l_i^h\} \tag{4.40}$$

$$T_i^{bw,dur} = g_{bw}^{dur}(T_j, i) := \min\{T_j^{bw,dur} - \tau_{ij} - \frac{q_h^R}{\lambda}, T_j^{bw,help} - \tau_{ij} - \frac{q_h^R}{\lambda} + b_j\} \tag{4.41}$$

$$T_i^{bw,help} = g_{bw}^{help}(T_j, i) := \min\{T_j^{bw,dur} - a_j, T_j^{bw,help}\} \tag{4.42}$$

The earliest start time to load the vehicle for route R while respecting the minimal duration is given by $ELT_R := \max\{0, T_i^{bw,dur} - T_i^{bw,help}\}$. If the start time to load the vehicle is between ELT_R and LLT_R, the cost to execute the new routes (including route R) equals $\mathrm{T}^{day} - T_i^{bw,dur}$. However, if the start time to load the vehicle is earlier than ELT_R, then the duration required by the vehicle to execute route R is given by $\mathrm{T}^{day} - T_i^{bw,help} - l_i^h$. Therefore, the cost of adding route R to the partial solution is $\tilde{m}_i^h := \tilde{m}_i^h + \max\{\mathrm{T}^{day} - T_i^{bw,dur}, \mathrm{T}^{day} - T_i^{bw,help} - l_i^h\}$.

Furthermore, dock door $\tilde{d}$ facilitates the latest start time to load the vehicle for route R, such that $\tilde{d} := \arg\max_{d\in\mathcal{D}}\{T_{h,d}^{Av}\}$. The latest time that dock door $\tilde{d}$ becomes available is adjusted to $T_{h,\tilde{d}}^{Av} := l_h^i$. As the solution is developed in a descending order of the start time to load the vehicles, the latest time that the remaining dock doors $d \in \mathcal{D} \setminus \{\tilde{d}\}$ are adjusted to $T_{h,d}^{Av} := \min\{T_{h,d}^{Av}, l_h^i + Q/\lambda\}$. Note that these updates are not necessary to determine the optimal start times to load vehicles, but they assist in performaning dominance checks between solutions, which is explained in the next subsection.

4.4.3.3 Extension to node $n+1$

A visit to node $n+1$ in the solution representation signifies that a new route is to be executed by a different vehicle. This requires resetting the following elements of the label: the vector of time resources is reset to $T_i^{bw} = (\mathrm{T}^{day}, \mathrm{T}^{day}, \mathrm{T}^{day})$, $R := \{n+1\}$ to indicate a new route, and $\kappa := \kappa + 1$ to indicate the use of an additional vehicle.

4.4.3.4 Extension to node $\tilde{0}$

Finally, once the label reaches node $\tilde{0}$, all customer nodes are visited in the solution. All elements of the labels are unchanged. Note that extending a label h to node $\tilde{0}$ is not done unless all customer nodes are visited in the solution, i.e., $V_h^v = 1$ for all $v \in \mathcal{V}$.

4.4.4 Acceleration techniques

To speed up the labeling algorithm and find feasible solutions rapidly, we apply two acceleration techniques.

4.4.4.1 Dominance rules

If the following conditions hold for two labels $\mathcal{L}_h^i$ and $\mathcal{L}_{h'}^i$, then any possible extension of $\mathcal{L}_{h'}^i$ is also possible as extension of label $\mathcal{L}_h^i$ with no higher costs:

i. $q_h^R \leq q_{h'}^R$

ii. $l_h^i \geq l_{h'}^i$

iii. $c_h^i \leq c_{h'}^i$

iv. $c_h^0 \leq c_{h'}^0$

v. $\tilde{m}_h^i \leq \tilde{m}_{h'}^i$

vi. $T_{i,h}^{bw} \geq T_{i,h'}^{bw}$ for each element of the vectors

vii. $T_{h,d}^{Av} \geq T_{h',d}^{Av}$ for all $d \in \mathcal{D}$

viii. $V_h^v \geq V_{h'}^v$ for all $v \in \mathcal{V}$

When all dominance rules (i) until (viii) apply, this is called *complete dominance*. In that case, label $\mathcal{L}_{h'}^i$ can be removed from the solution space as it is guaranteed to result in sub-optimal solutions. When only the dominance rules (i) until (vi) apply, the problem becomes comparable to the shortest path problem with resource constraints (Desaulniers et al., 2008). Removing the last two elementary conditions between labels is called *relaxed dominance*. In that case, label $\mathcal{L}_{h'}^i$ is reserved in a pool of labels that are only considered for extension if no label remains that is not dominated by the complete dominance check.

4.4.4.2 Precedence rule

Besides the dominance rules to discard labels, we also use the precedence check introduced in the following proposition.

Proposition 4.1. *Given two routes R_1 and R_2, where $ELT_{R_1} \leq ELT_{R_2}$, $LLT_{R_1} \leq LLT_{R_2}$ and $q^{R_1} = q^{R_2}$, if there exists a feasible solution to the problem, there exists an optimal solution where the start time to load the vehicles asociated with route R_2 is at or after the start time to load the vehicles for route R_1.*

We refer to Section 4.7.2 for the proof of this proposition.

Proposition 4.1 is directly used to verify whether the routes in a (partial) solution can be optimal. If that is not the case, the labels associated with those (partial) solutions are immediately discarded. To further speed up the algorithm, we tentatively relax the equality condition between the start times to load the vehicles. Consequently we also eliminate labels where the routes satisfy the three conditions of Proposition 4.1, but the start times to load the vehicles associated with the routes are equal. Finally, if a partial solution has two routes R_1 and R_2 for which $ELT_{R_1} \leq ELT_{R_2}$, $LLT_{R_1} \leq LLT_{R_2}$ and $q^{R_1} \leq q^{R_2}$, and the start time to load the vehicle for route R_1 is after the start time load the vehicle for route R_2, then the extension of the label associated with the (partial) solution is postponed by adding it to same pool of labels that are dominated based on the *relaxed dominance* rules.

The algorithm uses these acceleration strategies dynamically in the following manner. First, in order to find feasible solutions rapidly, acceleration strategies are used and the discarded labels are reserved for later extensions. If no improved solution is found within 10,000 extensions since the last improvement, acceleration strategies are not used until an improved feasible solution is found again. This dynamic use of the acceleration strategies help to find good solutions more rapidly while also paying attention to improve the lower bounds (see Section 4.7.1 for more details).

4.5 Case study

This section presents a numerical investigation on the performance of the dynamic programming algorithm proposed in Section 4.4 to solve the problem as formulated in Section 4.3. The numerical instances are generated based on a case study presented by one of the largest grocery retailers in The Netherlands. Using a case study from practice provides an opportunity to apply realistic values from industry for the parameters in the problem (such as order picking rate, staging limit, loading rates and so on). These values are not available from other instances commonly used in the vehicle routing literature. Another motivation to use a case study is that the results generated from the experiments provide managerial insights that can be translated into real-world operations. The solution procedure is implemented in *C++* on an i7 3.6 GHz machine with 16 GB of RAM. This section only reports the results for the dynamic programming algorithm and does not include the performance of commercial solvers such as *Gurobi* or *CPLEX*, as such solvers are not able to generate optimal solutions or even feasible solutions within a reasonable amount of computation time (we used a stopping criterion of 1,800 seconds in the experiments that we conducted).

4.5.1 Generation of instances

The instances that we generate have two distinct components: parameters associated with the network of retail stores (such as location, travel duration and demand) and the parameters associated with the warehouse operations (such as shift structures of order pickers, size of the staging area and number of dock doors). A description how we derived the values for all parameters of the problem is presented in this subsection.

4.5.1.1 Retail store information

The retailer operates four warehouses, where each facility serves four non-overlapping regions in The Netherlands. In this case study, we focus on the regional warehouse that supplies the retail stores in the northwest of the country. The region contains 277 store locations of different sizes. Smaller stores tend to be closer toward city centers while larger stores are further away from a city center. The retailer provided us with the travel durations between the warehouse and retail stores for predefined routes. The complete matrix with all travel durations was accessed from Google Geolocation API, which shows that the closest store is 10 minutes away from the warehouse and the furthest store is 120 minutes away. On average, it takes 40 minutes to drive between the warehouse and a retail store location.

The number of deliveries to retail stores ranges between 163 and 317 locations per day. The daily operations start at 11pm and the earliest arrival time at a customer location is at least 240 minutes and at most 1,260 minutes after that (with an average of 850 minutes after the start of the daily operations). The length of the delivery time window to start the service at a store location is on average 60 minutes, but can be as tight as 5 minutes or as large as 3 hours. The average service time (or unloading time) at a retail store location is 100 minutes.

Deliveries are made to retail stores with trucks that can hold at most 54 roll cages, which have a standard unit size and are used to consolidate products. The size of the order delivery to retail stores is measured in terms of the number of roll cages. The minimum and maximum order size is 3 and 54 roll cages, respectively, with an average of 27.8 roll cages.

To solve the instances in our experiments within a reasonable amount of time (i.e., within 1,800 seconds as stopping criterion), we consider instances with only 10 retail locations that are sampled from the retailer's network. To generate instances with a similar diverse set of characteristics as observed by the reatiler, we develop a number of scenarios. The *store locations* in each scenario are based on a clustered or random sampling strategy (Solomon, 1987). Besides store locations, we use *travel time*, *drop size*, *unloading time* and *time window length* to differentiate between retail store characteristics as proposed by Quak & de Koster (2007).

To select store locations with a clustered sampling strategy, we first create clusters based on travel distances between the warehouse location and the retail stores. A clustering technique with complete linkage (Friedman et al., 2001) suggests that there are three clusters of store locations in the data set. A K-means clustering technique is applied to allocate each retail store to one of the three clusters (Arthur & Vassilvitskii, 2007). The store locations in each of the three disjoint clusters and the warehouse location are shown in Figure 4.6: one cluster contains stores in and around Amsterdam (cluster 2), whereas the other two clusters contain stores east and north of the Amsterdam region (cluster 1 and 3, respectively).

The *travel time* (or distance) between the warehouse and a retail store defines the structure of the delivery network and the delivery cost. Additionally, the *drop size* (or the ratio of the order size relative to the vehicle capacity) determines the utilization of vehicles as well as the complexity of any vehicle routing solution. A drop size closer to one indicates a full truck load delivery, and the routing decisions are not necessary challenging. However, routing decisions become more important and complex when drop sizes are smaller and multiple customer locations are visited by a vehicle in a single route. The *unloading time* (or service time) required to unload a vehicle at a customer location also has an impact on the usage of the vehicles and the possible routes that can be constructed. Finally, the *time window length* to start the unloading

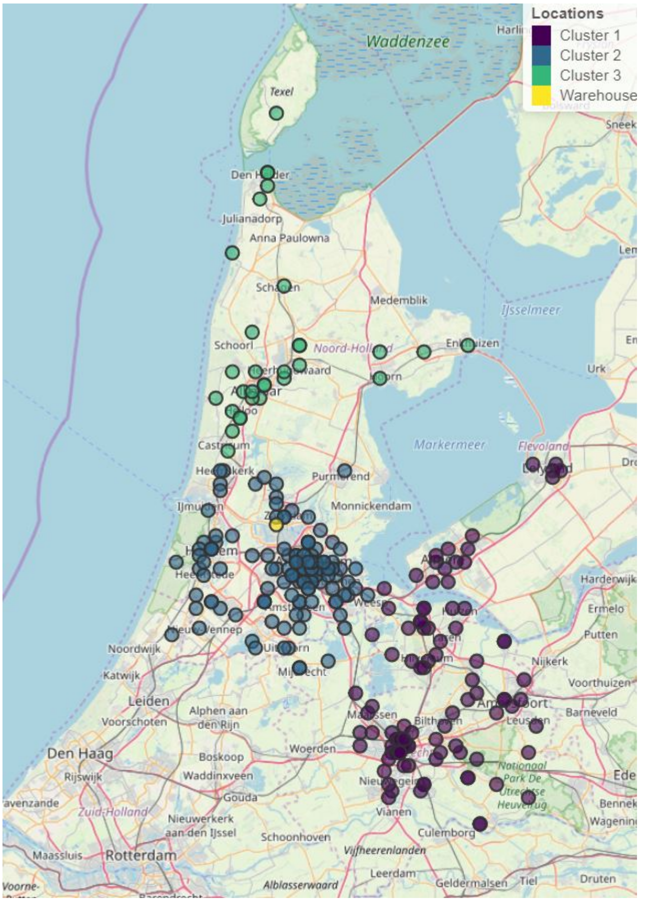
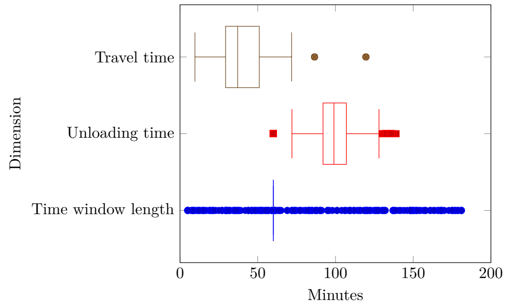


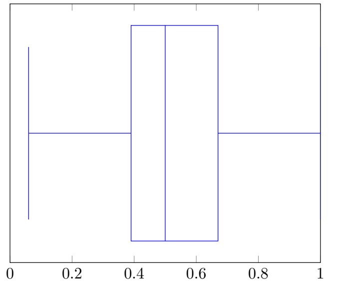
Figure 4.6: Store and warehouse locations, and clusters of stores based on the travel time between locations

at retail stores determines the time pressure in our instances. With larger time windows, the delivery planning becomes more flexible. Figure 4.7a summarizes the distribution of the travel time, unloading time and arrival time window length as observed in the retailer's data with box-and-whisker plots. Figure 4.7b illustrates the distribution of the drop sizes in the retailer's data.

In addition to the aformentioned dimensions, Quak & de Koster (2007) also propose *percentage of stores in shopping areas* and *delivery frequency* as other dimensions to consider when constructing numerical instances for vehicle routing problems. We do not include these dimensions as they are not relevant to our problem.



(a) Travel time, unloading time and time window length



(b) Drop sizes

Figure 4.7: Box-and-whisker plot of retail store characteristics in our case study

For each retail location we classify the four delivery dimensions of Quak & de Koster (2007) as *low*, *medium* or *high*. If the value of a dimension for a store location falls within the first tercile of the observations across all store locations, it is classified as low; if the value falls within the first and second tercile, it is classified as medium; and otherwise it is classified as high. This approach is taken for all dimensions except for the travel time as travel time and store locations are related. To classify the travel time of a particular store location as one of the three levels, we use the terciles of the observations across all store locations of the same cluster that the store location that we are classifying belongs to. Otherwise, most store locations that are classified at the low level for the travel time would belong to cluster 2. This would result in a bias towards the clusters when store locations are randomly sampled from the same level of the travel time dimension. The threshold values for the three levels of each dimension are presented in Table 4.3. This results in 162 combinations for the store characteristics.

4.5.1.2 Warehouse information

Outbound trucks are loaded at 29 dock doors which are used with different intensity levels across the week. The demand for dock-door availability is reflected by the number of vehicles (or routes) departing from the warehouse. However, the number of vehicle departures is determined when routes are constructed. A reasonable estimator for the potential utilization of dock doors before routes are constructed is the number of customer orders per dock door per day. This value indicates the highest possible utilization of dock doors, because at the worst case a unique vehicle will be required for each customer. In the data set, the number of customers per dock door ranges between 5 and 10 orders, with an average of 8.5 customer orders per dock door. In our instances, the customer-to-dock-door ratio is set at 10 for scenarios with a high utilization and at 5 for scenarios with a low utilization.

Truck drivers load vehicles at a rate of 15 minutes for a full truck load of 54 roll cages. Dabia et al. (2019) set the value of the loading rate by dividing the planning horizon into three non-overlapping shifts, where an equal loading capacity is used in each shift. The authors use a loading rate equal to 105%, 120% or 150% of one third of the daily loading demand. Moons et al. (2019) use a fixed loading time of 20 minutes per vehicle regardless of the quantity loaded into the vehicle. We believe that a constant loading time per roll cage is a more accurate representation than a constant loading time per truck. Therefore, we use a loading rate of 54 roll cages in 15 minutes as observed in the retailer's data.

Data regarding the utilization of the staging area was not shared by the retailer. However, we relate the total size of the staging area to the daily outbound volume. On the day with the lowest demand (i.e., sum of the order sizes), the staging area can hold 59.5% of the daily output at once. On the most challenging day, the staging space can hold only 31.2% of the daily output at any given time. To mimic the impact of a limited staging area, the size of the staging space is set to 30%, 45% and 60% of the daily outbound volume in our numerical instances (indicated by low, medium and high levels, respectively).

The order pickers work in three 9-hour shifts starting at hour 0, 8 and 9 (where hour 0 represents the starts of the day at 11pm). They have a 15-minute coffee break after two hours of the start of their shift, a 30-minute lunch break after 3.5 hours and another 15-minute coffee break after 6 hours. Figure 4.8 shows a Gantt chart of the three order picking shifts. The orders are picked in batches of 5 roll cages at a time, and each batch requires two hours to complete. In Gong & De Koster (2008) and Moons et al. (2019), the time to pick an order ranges between 10 and 27 minutes for an e-commerce environment, where orders have an average size of three lines. We decided to use the same order picking rate as observed by the retailer to maintain consistency, and any deviation can unintentionally impact the feasibility of our numerical instances. The difference in order picking rate can be explained by the fact that our case study is from an offline retailer instead of an online retailer.

It is a complicated process to determine the appropriate number of order pickers for each shift. Currently, the company's warehouse managers do this based on experience

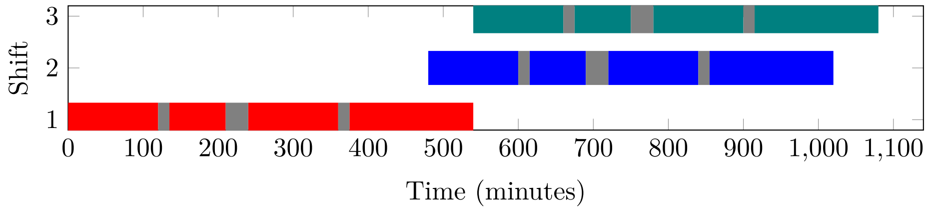


Figure 4.8: Shift patterns (gray indicate break periods)

and intuition. Rijal et al. (2020) present a methodology to schedule order pickers when the staging of orders have time windows that are determined after the start times to load the vehicles are established. We cannot use the same procedure to solve the problem in this paper as the start times to load the vehicles are part of the decision variables. However, the procedure presented in Section 4.4 can be adapted to derive the minimum number of order pickers for each shift in the following manner. Let the latest start time to load the vehicle for retail store location $j \in \mathcal{V}$ be defined as $\hat{l}_j := (b_j - \tau_{0j} - q_j/\lambda)$. However, because of the staging limit and the number of available dock doors, this latest start time $\hat{l}_j$ may not be feasible for all customer locations $j \in \mathcal{V}$ and may have to be adjusted accordingly. The adjustment is determined by applying a relaxed version of the dynamic programming algorithm proposed in Section 4.4. The new latest start time to load the vehicle is denoted by $\tilde{l}_j$ and has to adhere to the ascending order of the start times to load the vehicles at the warehouse (i.e., in the ascending order of $\hat{l}_j$). Initially, the size of the staging area and the number of available dock doors are set at low levels (i.e., 30% of the daily output and one dock door, respectively), whereas the number of available pickers is set to infinity. The actual number of order pickers can be determined once the tentative start times to load the vehicles are determined. Let ω_i indicate the minimum number of order pickers in shift $i \in \{1, 2, 3\}$. The values for these variables can be found by minimizing the value of $\sum_{i \in \{1,2,3\}} \omega_i$ subject to the order picking requirements associated with the start times $\tilde{l}_j$ to load the vehicles for all customer orders $j \in \mathcal{V}$. This requires us to solve a simple linear programming model (see Section 4.7.3 for the details of this problem formulation). The values of ω_i are directly used as the number of order pickers for each shift when the order picker availability is set to a scenario with a low level in the numerical instance. When the order picker availability is set to a medium or high level, the number of order pickers is increased by 25% or 50%, respectively.

A summary of the dimensions for the warehouse operations and its threshold values are presented in Table 4.3. With 162 combinations for the store characteristics in the distribution network and another 18 combinations for the warehousing characteristics, this reslts in a numerical test bed that contains 2,916 instances.

4.5.2 Performance of DP algorithm

Table 4.4 summaries the results on the performance of the algorithm developed in Section 4.4, where we distinguish between instances with clustered and random store locations. *Instances solved* indicates the number of instances that is solved to optimality as well as the total number of instances considered. *Optimality gap %* indicates the average percentage gap between the upper and lower bound as determined within the run time for the instances that are not solved to optimality. The average increase in the total routing costs (i.e., the total duration that vehicles are occupied) due to the warehouse limitations is calculated as $\%\Delta(\text{cost}) = (z(C) - z(U))/z(U) \times 100$, where $z(C)$ is the objective function value of the best solution found by the dynamic programming algorithm for the instance with capacitated warehouse parameters, and $z(U)$ is the objective function value of the best solution if warehouse resources are unconstrained (i.e., the number of dock doors, the number of order pickers and the size of the staging area are infinite). Finally, CPU reports the average computation time required to find the best (or optimal) solution for the instances.

The results show that the proposed solution procedure generates near-optimal solutions. For all except for 16 of the 2,916 instances, an optimal solution is found within the time limit of 1,800 seconds, with an average computation time of 50.4 seconds. For the instances not solved to optimality, the average gap between the best found solution and the lower bound as determined within 1,800 seconds is 4.9% (with a maximum of 10.0%). However, when the algorithm runs for two hours instead of 1,800 seconds, the same solution is proven to be optimal for all of these 16 instances. Consequently, we conclude that an optimal solution is found for all instances within 1,800 seconds, only the lower bound might not have converged within that time restriction.

The required computation time to solve the instances depends on the level of the dimensions that specify the instances. Concerning the store characteristics in the vehicle routing network, the direct travel time between the warehouse and the retail stores ($F = 26.02, p < 0.01$), the drop size ($F = 62.92, p < 0.01$) and the length of the delivery time windows ($F = 97.35, p < 0.01$) have a significant impact on the computation time. In the warehouse characteristics, only the number of available

Table 4.3: Summary of the characteristics that specify the numerical instances

Component	Dimension	Levels	Threshold Values
Vehicle Routing	Store locations	Clustered	C1, C2, C3
		Random	-
Network	Travel time (minutes) ⋆	Low	C1: (9.6,27.5], C2: (29.7,36.1], C3: (31.4,48.8]
		Medium	C1: (27.5,32.2], C2: (36.1,42.8], C3: (48.8,57.8]
		High	C1: (32.2,42.8], C2: (42.8,119.6], C3: (57.8,86.6]
	Drop size	Low	[0.06,0.43]
		Medium	[0.43,0.59]
		High	[0.59,1]
	Unloading time (minutes)	Low	[60,95]
		Medium	[95,105]
		High	[105,139]
	Time window length (minutes)	Low	[5,60]
		Medium	[60,60]
		High	[60,181]
Warehouse Operations	Dock door utilization (customers per dock door)	Low	5
		High	10
	Size of staging area (% of daily output)	Low	30%
		Medium	45%
		High	60%
	Order picker availability (% of the estimated pickers)	Low	100%
		Medium	125%
		High	150%

⋆ C1, C2 and C3 indicate cluster 1, 2 and 3, respectively.

Table 4.4: Summary of the results on the performance of our proposed solution procedure

Instance information		**Random store locations**				**Clustered store locations**			
Dimension	Levels	Instances solved	Optimality gap %	%Δ(cost)	*CPU* (sec.)	Instances solved	Optimality gap %	%Δ(cost)	*CPU* (sec.)
Travel time	Low	486 /486	-	1.7%	20.0	486 /486	-	2.6%	30.8
	Medium	484 /486	8.2%	2.6%	79.2	476 /486	5.4%	3.5%	88.5
	High	482 /486	1.9%	1.7%	53.2	486 /486	-	1.5%	30.9
Drop size	Low	480 /486	4.0%	1.5%	126.2	486 /486	-	2.1%	67.8
	Medium	486 /486	-	1.9%	22.3	476 /486	5.4%	2.9%	77.4
	High	486 /486	-	2.5%	4.0	486 /486	-	2.6%	5.0
Unloading time	Low	486 /486	-	2.3%	52.6	486 /486	-	2.2%	48.3
	Medium	486 /486	-	1.6%	11.7	476 /486	5.4%	4.2%	81.6
	High	480 /486	4.0%	2.1%	88.1	486 /486	-	1.2%	20.3
Time window length	Low	486 /486	-	3.2%	9.3	486 /486	-	3.2%	28.3
	Medium	486 /486	-	1.4%	10.2	486 /486	-	1.0%	22.6
	High	480 /486	4.0%	1.3%	132.9	476 /486	5.4%	3.3%	99.2
Dock door utilization	Low	727 /729	4.0%	2.0%	41.8	725 /729	3.8%	2.5%	43.0
	High	725 /729	4.0%	2.0%	59.8	723 /729	6.5%	2.5%	57.1
Staging space	Low	481 /486	4.6%	4.9%	67.0	484 /486	7.2%	5.4%	61.2
	Medium	485 /486	0.7%	0.8%	44.2	482 /486	4.5%	1.5%	44.5
	High	486 /486	-	0.2%	41.2	482 /486	5.4%	0.6%	44.5
Order picker availability	Low	481 /486	4.5%	3.5%	52.8	486 /486	-	4.2%	29.0
	Medium	485 /486	1.6%	1.5%	49.3	481 /486	5.3%	2.0%	55.1
	High	486 /486	-	0.9%	50.3	481 /486	5.6%	1.4%	66.0

Note: Instances solved indicates the number of instances solved to optimality out of the total number of instances in the instance set; Optimality gap indicates the average optimality gap among the instances not solved to optimality. %Δ(cost) shows the percentage increase in cost compared to warehouse without capacity constraints.

dock doors ($F = 3.98, p < 0.03$) and the size of the staging area ($F = 5.43, p < 0.03$) have a significant impact on the run time. The other dimensions that specify the characteristics of the instances have no statistically significant impact on the computation time.

4.5.3 Impact of limited warehousing resources

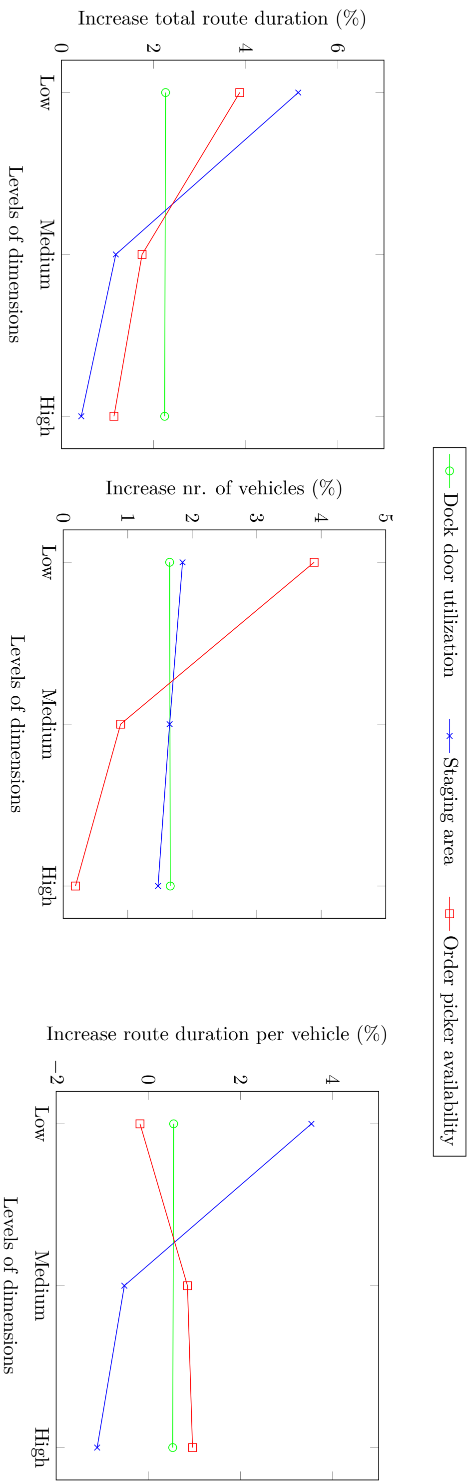
Figure 4.9 presents a comparison between the routing solutions with limited warehousing resources and routing solutions that are not constrained by warehouse operations. It shows the relative increase in the total route duration (i.e., the objective function value), the relative increase in the total number of vehicles (or routes) and the absolute increase in route duration per vehicle when averaging the results over one level for the dock door utilization, size of the staging area or availability of the order pickers. From the results in Figure 4.9a it becomes clear that the limited availability of warehousing resources has a significant impact on the total duration of the optimal routes. In particular, the objective function value increases on average by 2.3% when warehouse resources are constrained. The size of the staging area has the largest impact: when it is restricted to 30%, 45% or 60% of the daily output, this leads to an increase of the total duration for the optimal routes by 5.14%, 1.18% or 0.43%, respectively. When the availability level for the order pickers is set to low, medium or high, the total duration of the optimal routes increases by 3.87%, 1.75% or 1.14%, respectively. The levels of utilization for the dock doors do not have a significantly different impact on the total duration of the optimal routes. This is because the loading time for each vehicle is small compared to the picking time of orders.

The availability of the order pickers has the largest impact on the number of vehicles (or routes) that are needed. Compared to the instances with no constrained warehouse operations, the average number of vehicles increases from 7.6 to 7.9 over all instances (i.e., an increase of 3.89%) when the order picker availability is low. See Figure 4.9b for more details. When we consider the change of the route duration for individual vehicles, Figure 4.9c illustrates that the size of the staging areas has again the most significant impact. In particular, the total route duration over all vehicles increases more than the increase in the number of vehicles needed when the staging area is constrained to low levels, which results in an average increase of 3.54% for the route duration of a vehicle. In contrast, when the staging area is hardly restricted (i.e., it has a high level), there is a small increase in the total route duration over all vehicles (on average only 0.43%) but the number of required vehicles to perform these routes increases by 1.47% on average, such that the route duration per vehicle decreases by 1.11% on average.

Based on the previous results, it is concluded that the size of the staging area and the availability of the order pickers have the largest impact. When the restrictions on the two warehouse resources are combined, this has an even larger effect on the increase of the total duration of the optimal routes ($F = 14.79, p < 0.01$ after controlling for the main effects). The impact when restrictions on the staging area and the order picker availability are combined is illustrated in Figure 4.10 (similar to Figure 4.9). Besides the average impact, this figure also includes the 95% confidence interval for each of the three performance measures. It shows that the effect of additional staging area space is larger when the order picking workforce is limited and vice versa. When the warehouse is severely constrained in the number of order pickers and the size of the staging area, the average increase in total route duration is 8.3% compared to the solutions where the warehouse resources are not constrained (see Figure 4.10a), and it can be as high as 91.5%. Figure 4.10b illustrates that the impact of the staging area on the number of routes (or vehicles needed for the transportation) is minor, regardless of the order picker availability. This corresponds to our conclusion from Figure 4.9b. However, the size of the staging area and the availability of order pickers have a significant combined impact on the route duration per vehicle ($F = 7.01, p < 0.01$ after controlling for the main effects). Similar to the impact on the total route duration, the staging space has a larger impact on the route duration per vehicle when the order picker availability is low (see Figure 4.10c). Based on these figures, we conclude that limited warehouse resources are compensated by the deployment of more vehicles and some of the vehicles have to be loaded at the warehouse earlier to free space in the staging area compared to routing plans where warehouse resources are not constrained.

4.5.4 Managerial Insights

The trade-offs between the utilization of order pickers, the utilization of the staging area and the total duration of the optimal routes become clear from Figure 4.10. When the available space for staging is high, the savings in the total vehicle duration from employing additional order pickers are only marginal. In warehouses with severely limited staging space, if transportation managers are willing to accept an average increase of the total duration of routes by 8.4%, the warehouse managers can operate the warehouse with 50% fewer order pickers (when comparing the order picker

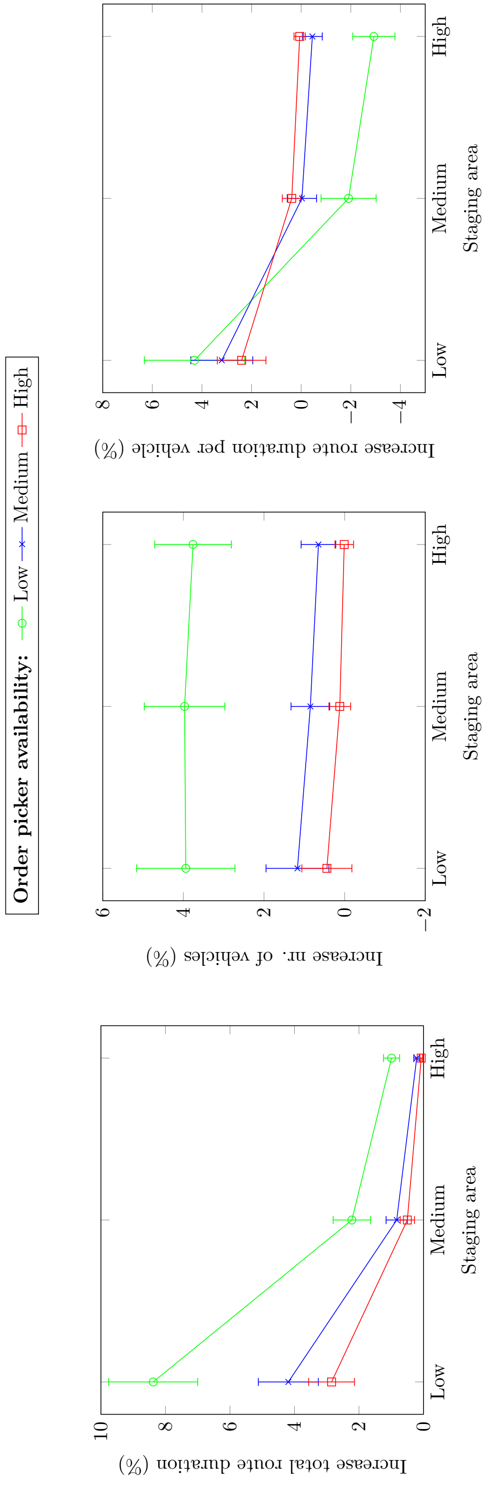


(a) Impact on total route duration

(b) Impact on the number of vehicles

(c) Impact on route duration per vehicle

Figure 4.9: Impact over all instances when one dimension of the warehouse resources is constrained



(a) Impact on total route duration

(b) Impact on the number of vehicles

(c) Impact on route duration per vehicle

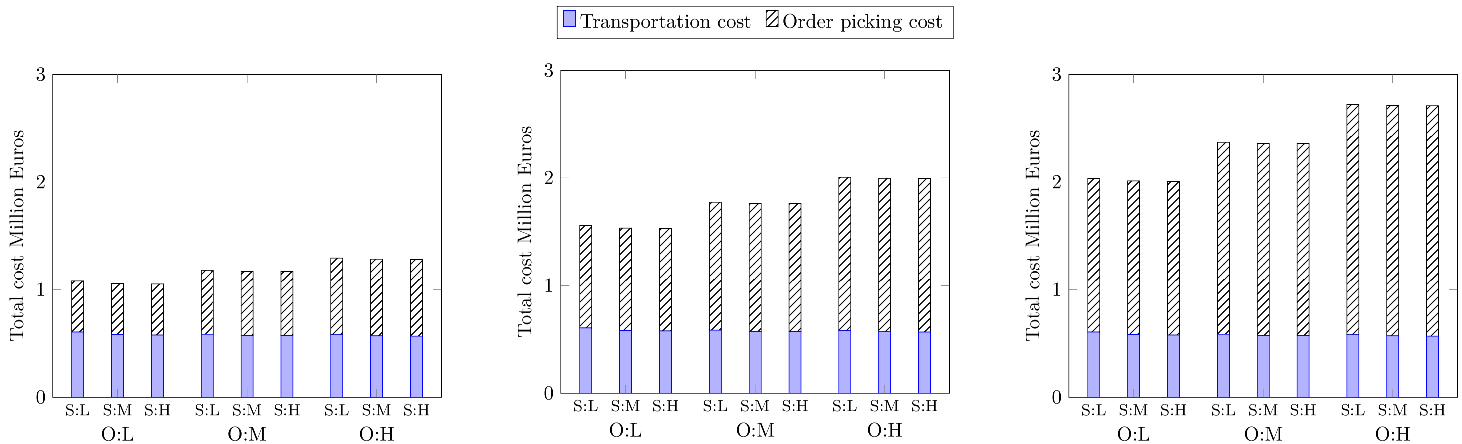
Figure 4.10: Combined impact of the size of the staging area and the order picker availability

availability from high to low). However, the validity of this decision relies on the relationship between the cost for order pickers, vehicles and truck drivers. To better understand these trade-offs between the objectives of warehouse and transportation managers, the system-wide cost needs to be studied. In this case study, we use an hourly cost rate to use a large rigid truck in The Netherlands of €20 and the hourly wage for a truck driver is €21 (Quak & de Koster, 2009). The hourly wage for an order picker is set as 50%, 100% or 150% of the hourly wage for a truck driver (corresponding to low, medium or high levels for the cost of the order picking operations). Furthermore, the cost of a truck-trailer combination is approximately €160,000, which depreciates on average over 7 years (Janssen et al., 2015). We assume that a working week consists of 7 days and there are 48 working weeks in a year.

Figure 4.11 shows the average total transportation costs and the average total costs of order picking for different levels of the hourly wage for order pickers, the availability of order pickers and the size of the staging area. Details on these costs are provided in Section 4.7.4. It becomes apparent that the cost of additional order pickers cannot be offset by the savings that can be generated in transportation costs when more order pickers are available. This observation is consistent in all instances regardless of their characteristics for the store locations, travel time, drop sizes, unloading time or time window length. This finding suggests that logistics managers should re-evaluate the conventional focus to prioritize the optimization of routing costs over warehouse costs when warehouses have significant operational constraints. Particularly when the number of available order pickers or the size of the staging area is limited, supply chain management should give precedence to the objectives of warehouse managers over the isolated optimization of vehicle routes.

4.6 Conclusion

In this study, we investigate the joint planning of warehouse operations and transportation for warehouses with limited capacities for order picking, staging and loading when deliveries are dictated by time windows. Most work in the vehicle routing literature does not include any warehouse limitations, and only a handful of papers have incorporated just one warehousing aspect when deciding on the delivery planning. This paper contributes to the vehicle routing and the warehouse planning literature by presenting an integrated model that captures the interactions between the three most important warehouse processes (i.e., order picking, staging and loading) and the



(a) Low cost of order pickers **(b)** Medium cost of order pickers **(c)** High cost of order pickers

Figure 4.11: Average total annual cost (in million Euros) for order picking and distribution operations
Note: *O:L*, *O:M* and *O:H* indicate low, medium and high level for the order picker availability, respectively. *S:L*, *S:M* and *S:H* indicate low, medium and high level for the size of the staging area, respectively.

planning of delivery routes with time windows. We develop a dynamic programming algorithm to solve the problem exactly.

To gain managerial insights into the impact of warehouse restrictions on routing costs, we have performed extensive numerical experiments on instances that are generated from industry data. The results indicate that when warehouse resources are restricted (particularly the availability of the order picking workforce and the size of the staging area), the delivery plans have to be adapted to consider these warehouse restrictions. When the cost for order picking and transportation are analyzed, it becomes clear that the costs associated with additional order pickers to alleviate the order picking restrictions are significantly higher than the savings in the transportation costs (both trucker driver and vehicle costs). This statement remains true even when the hourly wage for an order picker is only 50% of the hourly wage for a truck driver. For instance, when the order picker availability is reduced by 33%, this restricts the vehicle routing but the total transportation costs would increase by only 2-5% (depending on whether the staging area is also restricted). In other words, it is more cost effective to focus on reducing the number of order pickers rather than optimizing the routing of vehicles. These findings contradict the traditional approaches in the literature and practice to focus on the optimization of distribution processes rather than the warehouse operations. Therefore, when warehouses have a limited number of order pickers available, it is crucial for logistics managers to consider warehouse resources when performing route planning. This paper proposes a solution method that can support the decision-making process by warehouse and transportation managers to accomplish this.

A number of assumptions regarding the available warehouse capacities are made in our problem formulation. In particular, we assume fixed capacities. The number of available dock doors or the size of the staging area cannot be changed on a daily basis. However, the number of available order pickers can vary. Another assumption is that the order picking capacity is known before the route planning is decided. This is a reasonable assumption for many warehouses where the number of employees is a decision that is made at a tactical level. However, the size of the workforce can be determined operationally on a day-to-day basis when a warehouse has many flexible order pickers (Rijal et al., 2020). A joint optimization of the staffing levels for order picking together with the route planning of vehicles can be a valuable research direction for such warehouses. Finally, in this paper we assume that the order picking rate is continuous. This assumption is not restrictive for an environment with retail warehouses where orders are large and picking routes are long. However, for warehouses where order sizes are small (such as e-commerce warehouses), it might be more suitable to also consider the size of each customer order.

4.7 Appendix

4.7.1 Pseudocode of dynamic programming algorithm

This appendix includes an overview how the labels $\mathcal{L}_h^i = (i, R, q_h^R, l_h^i, l_h^0, c_h^i, c_h^0, \tilde{m}_h^i, T_i^{bw}, \kappa_h, T_h^{Av}, (V_h^1, \dots, V_h^{|\mathcal{V}|}))$ on the vertices $i \in \mathcal{V}'$ for route R in the augmented graph are updated with resource extension functions, where $f_{bw}(T_j, i)$ and $g_{bw}(T_j, i)$ extend the time resources T_j^{bw} from node j to node i along the arc (i, j) for nodes $i \in \mathcal{V}$ and nodes $i \in \{0\}$, respectively. If the time resources in the new label are feasible, then the new label is added to the label set $\mathscr{L}_j$. After all labels are extended, the accelaration techniques are applied to eliminate labels from the label set $\mathscr{L}_j$ that are known to be sub-optimal. This is repeated until all customers are visited and the vehicles have returned to the warehouse.

Algorithm 3 Dynamic programming algorithm

$\mathcal{L}_h^{n+1} := (n+1, \emptyset, 0, \mathrm{T}^{day}, \mathrm{T}^{day}, 0, 0, T_i^{bw}, 1, T_h^{Av}, (V_h^1, \dots, V_h^{|\mathcal{V}|}))$ ▷ Initialization at route completion node, $n+1$
$\mathscr{L}_{n+1} := \mathscr{L}'_{n+1} := \{\mathcal{L}_h^{n+1}\}$
for $i \in V' \setminus \{n+1\}$ **do**
    $\mathscr{L}_i := \emptyset$, $\mathscr{L}'_i := \emptyset$ ▷ Initialization at remaining nodes
$\Delta \leftarrow \{n+1\}$ ▷ Vertices with labels to be processed
$R \leftarrow \{n+1\}$ ▷ Route under consideration
$\eta := 0$ ▷ Count of extensions since last improved solution
while $\Delta \neq \emptyset$ **do**
    for $j \in \Delta$ **do**
        for $i \in \mathcal{V}' \setminus \{j\}$ **do** ▷ Feasible nodes only
            for $L_h^j \in \mathscr{L}'_j$ **do**
                $\eta := \eta + 1$
                if $V_{L_h^i}^i = 0$ **then** ▷ If i is not already visited
                    if $i \in \mathcal{V}$ **then** ▷ Extension to customer nodes
                        $T_i := f_{bw}(T_j, i)$, $R \leftarrow R \cup \{i\}$, Adjust other resources
                    if $i \in \{0\}$ **then**

$T_i := g_{bw}(T_j, i)$, $R \leftarrow R \cup \{i\}$, Adjust other resources
if $i \in \{n+1\}$ **then**
$R \leftarrow \{n+1\}$, $T_i^{bw} := (\mathrm{T}^{day}, \mathrm{T}^{day}, \mathrm{T}^{day})$, $\kappa := \kappa + 1$
if $i \in \{\tilde{0}\}$ **then**
if $V_h^{j'} = 1 \quad \forall j' \in \mathcal{V}$ **then**
if $\tilde{m}_h^i < \tilde{m}_{h'}^{\tilde{0}}, \forall \mathcal{L}_{h'}^{\tilde{0}} \in \mathscr{L}_{\tilde{0}}$ **then** ▷ Improved solution found
$\eta := 0$
if T_i is feasible **then**
ADD $\mathcal{L}_h^i := (i, R, q_h^R, l_h^i, l_h^0, c_h^i, c_h^0, \tilde{m}_h^i, T_i^{bw}, \kappa_h, T_h^{Av}, (V_h^1, \ldots, V_h^{|\mathcal{V}|}))$ to $\mathscr{L}_j$
if $\mathscr{L}_j$ changed **then**
$\mathscr{L}_j \leftarrow DOMINANCE(\mathscr{L}_j)$ ▷ Eliminate dominated labels
$\mathscr{L}'_j \leftarrow PARTIALDOMINANCE(\mathscr{L}_j)$ ▷ Eliminate partially dominated labels
if $\eta > 10,000$ **then** $\mathscr{L}'_j \leftarrow \mathscr{L}_j$
$\Delta \leftarrow \Delta \cup \{j\}$
$\Delta \leftarrow \Delta \setminus \{j\}$
return $\arg\min_{L \in \mathscr{L}_{\tilde{0}}} \{m\}$

4.7.2 Proof of Proposition 4.1

Proof: Consider that the routes R_1 and R_2 are loaded at time l_{R_1} and l_{R_2}, respectively, where $l_{R_2} \leq l_{R_1}$. To prove the proposition, it suffices to show that the order in which the vehicles are loaded for these routes can be interchanged without increasing the the total costs and without violating the feasibility conditions. Let the new start times to load the routes R_1 and R_2 be l'_{R_1} and l'_{R_2}, respectively. Consequently, we can switch the start times to load the routes, i.e., $l'_{R_1} := l_{R_2}$ and $l'_{R_2} := l_{R_1}$. As $q^{R_1} = q^{R_2}$, the staging and loading plans remain identical.

Let us denote the total cost in addition to the minimal duration cost by $m(R_1, R_r)$. Since $l_{R_2} \leq l_{R_1}$, we obtain

$$m(R_1, R_2) := \max\{ELT_{R_1} - l_{R_1} + ELT_{R_2} - l_{R_2}, ELT_{R_1} - l_{R_1}, ELT_{R_2} - l_{R_2}, 0\}.$$

Because $ELT_{R_1} - l_{R_1} \leq ELT_{R_2} - l_{R_2}$, we can rewrite this as

$$m(R_1, R_2) := \max\{ELT_{R_1} - l_{R_1} + ELT_{R_2} - l_{R_2}, ELT_{R_2} - l_{R_2}, 0\}.$$

When the order in which the vehicles for the routes are loaded is interchanged, the new cost m'_{R_1,R_r} equals

$$\begin{aligned} m'(R_1, R_2) &:= \max\{ELT_{R_1} - l'_{R_1} + ELT_{R_2} - l'_{R_2}, ELT_{R_1} - l'_{R_1}, ELT_{R_2} - l'_{R_2}, 0\} \\ &:= \max\{ELT_{R_1} - l_{R_2} + ELT_{R_2} - l_{R_1}, ELT_{R_1} - l_{R_1}, ELT_{R_2} - l'_{R_2}, 0\}. \end{aligned}$$

Since $ELT_{R_2} - l_{R_2} \geq ELT_{R_1} - l_{R_2}$ and $ELT_{R_2} - l_{R_2} \geq ELT_{R_2} - l_{R_1}$, we obtain $m(R_1, R_2) \geq m'(R_1, R_2)$. This completes the proof.

4.7.3 Determining the minimum number of order pickers in each shift

Before we determine the number of order pickers, we determine the latest start time to load the vehicle for customer location $j \in \mathcal{V}$ based on a relaxed version of the dynamic programming algorithm presented in Section 4.4, where we assume that each vehicle visits only one customer location before returning to the warehouse. The order in which the vehicles are loaded corresponds to the increasing order of $b_j - \tau_{0j} - q_j/\lambda$, and the order picking capacity in the warehouse is assumed to be infinite (see Section 4.5.1.2 for more details). Let us denote the resulting latest start time to load the vehicle for customer location $j \in \mathcal{V}$ by $\tilde{l}_j$. The time segment corresponding to $\tilde{l}_j$ is denoted by $\tilde{y}_j$ and the order in which the vehicles for the customers are loaded at the warehouse is indicated by the function $v(\cdot)$. Table 4.5 provides an overview of all notations for the input parameters as well as the decision variables for this problem.

To determine the minimum number of order pickers for each of the three shift types in our case study, we solve the following linear programming model:

$$\min \sum_{i \in \{1,2,3\}} \omega_i \tag{4.43}$$

subject to

Table 4.5: Notations for the problem to determine the minimum number of order pickers for each shift

Parameters	
$\tilde{l}_j$	latest start time to load the units for customer location $j \in \mathcal{V}$ determined by the relaxed dynamic program
$\tilde{y}_j$	$:= \{s \,|\, \mathrm{T}_s^{beg} < l_j \leq \mathrm{T}_s^{end}\}$
$v(n)$	n^{th} customer in ascending order of $\tilde{l}_j \;\; \forall j \in \mathcal{V}$
q_j	the number of units demanded by customer location $j \in \mathcal{V}$

Decision variables	
ω_i	number of order pickers scheduled in shift $i \in \{1, 2, 3\}$
$\tilde{o}_j$	total number of units that can be picked until the loading of the vehicle for customer $j \in \mathcal{V}$ starts
$\tilde{c}_j$	number of units available at the staging area when the loading of the vehicle for customer $j \in \mathcal{V}$ starts

Simplifying notations	
Ω_s	$:= \{\omega_i \, \forall \, i \in \{1, 2, 3\} \vert \text{order pickers in shift } i \text{ work in time segment } s \, \forall s \in \mathcal{S}\}$

$$\tilde{o}_j \leq \mu \sum_{s'=2}^{\tilde{y}_j - 1} \sum_{i \in \Omega_{s'}} \omega_i \Big(\mathrm{T}_{s'}^{beg} - \mathrm{T}_{s'-1}^{beg}\Big) + \mu \sum_{i \in \Omega_{\tilde{y}_j}} \omega_i \Big(\tilde{l}_j - \mathrm{T}_{\tilde{y}_j}^{beg}\Big) \qquad \forall j \in \mathcal{V} \tag{4.44}$$

$$\tilde{c}_{v(1)} \leq \tilde{o}_{v(1)} \tag{4.45}$$

$$\tilde{c}_{v(n)} \leq \tilde{c}_{v(n-1)} - q_{v(n-1)} + \big(\tilde{o}_{v(n)} - \tilde{o}_{v(n-1)}\big) \qquad \forall n \in \{2, 3, \ldots |\mathcal{V}|\} \tag{4.46}$$

$$\tilde{c}_j \geq q_j \qquad \forall j \in \mathcal{V} \tag{4.47}$$

$$\omega_i \in \mathbb{Z}_0^+ \qquad \forall i \in \{1, 2, 3\} \tag{4.48}$$

The objective in (4.43) is to minimize the total number of order pickers over the three shifts. Constraints (4.44) determine the overall picking capacity at the start time to load the vehicle associated with the customer in node j. The first term in the right-hand side of the constraint represents the number of units that can be picked until the beginning of the time segment in which the order for customer location

$j \in V$ starts to be loaded in a vehicle. The second term represents the number of units that can be picked from the time segment in which the loading of the vehicle for customer j is started until the vehicle is loaded with all units for that customer. Constraints (4.45) and (4.46) ensure that the necessary items are picked and become available at the staging area when the vehicles are loaded. Constraints (4.47) ensure that sufficient units are available to load in the vehicle for each customer location $j \in \mathcal{V}$ at the time when the loading starts. The final constraints indicate that the decision variables are either zero or positive integers.

4.7.4 Detailed result for Figure 4.11

To calculate the average annual cost for the transportation (both vehicle and truck driver), we use an hourly cost rate of €20 to use a vehicle and an hourly wage of €21 for a truck driver (Quak & de Koster, 2009). The cost of a truck-trailer combination is approximately €160,000, which depreciates on average over 7 years (Janssen et al., 2015). The hourly wage for an order picker is set as 50%, 100% or 150% of the hourly wage for a truck driver (corresponding to low, medium or high levels for the cost of the order picking operations). Furthermore, we assume that a working week consists of 7 days and there are 48 working weeks in a year.

Table 4.6 provides a summary of the average total costs. The first row indicates the results when the warehouse operations are not constrained. The fixed transportation costs consist of the costs associated with the depreciation of the vehicles. The variable transportation costs consist of the hourly wages paid for the truck drivers as well as the hourly cost for using the vehicle (both are proportional to the travel duration). The order picking costs are determined based on the availability of the order pickers (i.e., proportional to $\sum_s p_s$).

Table 4.6: Average annual cost of the solutions based on our proposed dynamic programming algorithm

Order picker availability	Staging space	Total route duration (hrs)	Number vehicles needed	Transportation cost		Order picking cost		
				Fixed	Variable	Low rate	Medium rate	High rate
-	-	9,608	7.64	174,674	393,540	-	-	-
Low	Low	10,389	7.91	180,882	425,963	475,627	951,253	1,426,880
Low	Medium	9,818	7.91	180,741	402,541	475,627	951,253	1,426,880
Low	High	9,704	7.90	180,600	397,850	475,627	951,253	1,426,880
Medium	Low	10,002	7.73	176,579	410,075	594,533	1,189,067	1,783,600
Medium	Medium	9,687	7.69	175,873	397,149	594,533	1,189,067	1,783,600
Medium	High	9,628	7.68	175,591	394,743	594,533	1,189,067	1,783,600
High	Low	9,876	7.69	175,661	404,916	713,440	1,426,880	2,140,320
High	Medium	9,654	7.65	174,885	395,826	713,440	1,426,880	2,140,320
High	High	9,615	7.65	174,744	394,199	713,440	1,426,880	2,140,320

5 Conclusions and future outlook

In this dissertation, we investigate the impact of external *temporal* constraints from upstream and downstream facilities on warehouses with an emphasis on warehouses with *manual* workforce. Warehouses often face external temporal constraints such as time windows to unload and process vehicles from suppliers or delivery time windows to process and (un)load vehicles for customers. In warehouses that are operated manually, managers have to plan the use of warehouse workers and other resources to ensure that a high service level is maintained while considering the temporal constraints. Complicating factors such as labor laws and union agreements make the planning of manual workers particularly challenging. However, most of the warehousing literature focuses on the use of machines in warehouses and not humans. Even when the use of human workers is considered, they do not consider any of the specifics that pertain to the use of manual workers. Additionally, the literature largely assumes that warehouses are isolated entities whose operational planning can be done independently without consideration for upstream or downstream temporal constraints. This dissertation contributes to the literature by focusing on manually operated warehouses as part of larger supply chain networks and investigates the impact of external constraints on the planning of resources in the warehouse.

We focus on three different planning problems in manual warehouses in this dissertation. The first study in the dissertation investigates the planning of cross-dock operations constrained by *time windows* of inbound and outbound trucks. The cross-dock facility uses *cross-dock workers* to process shipments using a *temporary storage area* and a limited number of *dock doors*, some of which can operate in mixed mode. In a second study, a warehouse with limited *staging lanes* has to plan the *flexible order pickers* to ensure that orders are picked and delivered to the staging lanes within order dependent *delivery due time windows*. The last study investigates the usage of *order pickers* and truck *loaders* in warehouses within limited availability of the *staging area* and *dock doors*. Together with warehouse planning, this study plans delivery routes for vehicles with delivery time windows at customer locations. We formulate mathematical models and develop exact and heuristic solution approaches to solve the problems. These models and algorithms can provide valuable assistance as decision support to warehouse managers. In addition, experiments are performed with the models and algorithms based on industry data to provide practical insights on warehouse operations.

5.1 Conclusions

In Chapter 2 of this dissertation, we investigate the integrated scheduling and assignment of trucks to dock doors in a unit-load cross-dock facility where a subset of dock doors can operate in mixed mode. Temporally, processing of inbound and outbound trucks is constrained by time windows. Inbound time windows cannot be violated whereas outbound trucks can depart from the warehouse with penalized delays. Most of the literature as well as practice only consider either the scheduling or assignment of trucks to dock doors in cross-docks facilities with mixed-mode dock doors. When assignment and scheduling decisions are integrated, they are only considered for either inbound or outbound operations. However, the use of sequential solution approaches can lead to sub-optimal solutions and even to infeasible solutions. In this chapter, we develop a model that allows simultaneous scheduling and assignment of trucks to dock doors. To solve realistic instances of the problem, an adaptive large neighborhood search (ALNS) algorithm is developed. The solution of the integrated model is validated against optimal sequential solutions as well as solutions generated by metaheuristics from the literature that solve the problem sequentially (schedule-first-assign-second). Extensive experiments on randomized data used in the literature demonstrate that the integrated solution approach can reduce operational costs on average by 12% compared to the optimal sequential solutions and as much as 20-30% compared to the heuristic solutions to address the problems sequentially.

Additionally, the integrated solution approach is used to investigate the impact of the position and the proportion of mixed-mode dock doors on operational costs. The findings suggests that for U-shaped cross-dock terminals, the utilization of dock doors impacts the value of mixed-mode dock doors. When the dock-door utilization is low, the positioning of mixed-mode dock doors is more important, and the strategy of placing mixed-mode dock doors at the center of the cross-dock facility provides the largest benefit. However, when utilization is high, the proportion of mixed-mode dock doors becomes more important. In either of the low or high utilization cases, the results suggest that at least 40% of the dock doors should be used in the mixed mode, and they should be positioned at the center of the cross-dock facility. This study also makes a contribution by providing a framework that can be used to make layout decisions, such as determining the proportion and position of the mixed-mode dock doors. A simplified version of the solution approach developed in this chapter is currently used by a Dutch retailer to plan their cross-dock operations.

In Chapter 3, we study the scheduling of flexible manual order pickers in warehouses with delivery due time windows. The manual workers are hired on a short notice and are only paid from the start until the end of their shift. The start and end times of the shift are variable and part of the warehouse manager's decision. Using these flexible order pickers, managers have to ensure that orders are picked and delivered to the staging area of the warehouse within order-dependent delivery due time windows. In this environment, workforce scheduling (of order pickers) has to consider traditional personnel scheduling decisions such as determining the start and end times of shifts and break times together with the scheduling and assignment of order-picking activities. Two mathematical models are formulated for the problem. To solve the problem, we develop an exact procedure based on a brand-and-price framework. Additionally, to generate quality solutions within limited time, we develop a heuristic based on the large neighborhood search (LNS) algorithm. Computational experiments illustrate that the LNS is able to generate quality solutions within reasonable computation times. To further test the applicability of the proposed solution approach and generate practical insights, we conduct a case study at the warehouse operations of the largest grocery retailer in The Netherlands. The case study compares a variety of shift structures and their impact on the order picking cost at the warehouse. The results show the value of flexibility in break scheduling for workers. Without additional costs to the warehouse, the managers can ensure that the order pickers are compensated for shifts of at least 8 hours (instead of the current minimum compensation of 6 hours) in return for flexible breaks (i.e., breaks with 15 minutes of flexibility in their starting time compared to the current case). This study contributes to the warehousing literature by introducing and demonstrating the importance of incorporating personnel scheduling decisions into traditional order picking problems. The models and solution approaches developed in this chapter can be valuable for further extensions and adaptations. Managerially, this work shows that workforce scheduling, and in particular the idea of flexible breaks, can be a lever for gaining efficiency in warehouses. This study has been translated into a decision support tool that is currently used by a Dutch retailer.

The fourth chapter of this dissertation studies the impact of hard time windows at customer delivery locations on the operations of warehouses with limited resources. This problem is inspired by the operational challenges faced by the warehouse operation of a large Dutch grocery retailer. The warehouse under consideration has a limited time-dependent order picking capacity which arises from the use of manual order pickers in multiple shifts with breaks. Additionally, the warehouse has limited staging space, which serves as a buffer between the order picking and loading operations. The warehouse managers have to translate the schedule of outbound delivery trucks into plans for the usage of warehouse resources. When staging space is limited in comparison to the daily output from the warehouse, warehouse may not be able to ensure that vehicle can leave with the required shipments in the departure times determined by distribution plans. A potential solution to this problem lies in the planning of the delivery routes while jointly taking the limited warehouse resources into consideration. However, the literature mostly considers a decoupled approach to the problem where the warehouse and transportation problems are solved in isolation. When they are integrated, the problems consider only one of warehouse processes, either order picking or loading. We formulate a model for the holistic problem that can develop routes for vehicles to deliver orders to customers with hard time window constraints while simultaneously considering limited order picking, staging and loading capacities. A dynamic programming algorithm is developed to solve the problem exactly. Computational experiments are performed on instances that are generated from industry data provided by a large Dutch retailer. The results suggest that a limited staging area and a limited order picker availability have a significant and multiplicative impact on the cost of distribution routes. When the staging area and order picker availability are low, the duration of routes increases on average by 8.3% compared to optimal routes not constrained by any warehousing resources. Furthermore, when the costs of order pickers, trucks and truck drivers are analyzed, the results suggest that even when the cost of order pickers is 50% less than the cost of truck drivers, it is better for the system to incur additional costs on distribution (costs of trucks and truck drivers) rather than at the warehouse as it is done traditionally. This study contributes to the warehousing literature by developing a holistic problem which can capture the interaction of limited resources in warehouse processes and the planning of distribution routes. For practitioners, this work illustrates that when warehouse resource are limited and heavily utilized, the traditional approach of generating distribution plans first and imposing them on warehouse operations needs to be reconsidered.

5.2 Future outlook

The planning problems studied in this dissertation consider warehouses with manual operations and some external temporal constraints. Our case examples are predominantly based on the offline retail warehouse operations. While this warehouse setting is important, these operational contexts represent only a small sample of warehouse environments in practice. Given the trends of adoption of automation and the rise of e-commerce, several opportunities exist in extending the problems studied in this dissertation to consider different and richer warehouse operations.

In Chapter 2, the experiments primarily consider U-shaped cross-dock facilities and present limited results for I-shaped cross-dock facilities. Existing results on the best shape of cross-dock facilities (Bartholdi & Gue, 2004) only consider exclusive-mode dock doors. A new analysis on the shapes and layouts with mixed-mode dock doors could be valuable for practitioners. Additionally, positioning and determining the size of the temporary storage area have an impact on the operational cost of cross-dock facilities. A problem that can incorporate the size and position of the temporary storage area into the design decision could enable a more informed cross-dock design. Achieving these objectives requires the formulation of new cross-dock problems and developing new solution approaches.

The second study in this dissertation assumes that the batches of orders have been determined before workforce scheduling. Extending the problem to consider batching of orders can provide larger operational benefits. An added advantage of this approach enables incorporating the order picker location into the decision problem. Using this information, decision support systems can generate batches in such a way that order pickers will be at pick locations closer to the break areas when they need to go for breaks. This will shorten the time wasted by order pickers traveling to and from break areas. An efficient solution approach that can solve a problem with these details will be a valuable addition to the literature. It will also be of value to practice as higher cost savings may be achieved.

The last study in the dissertation assumes that customers have hard delivery time windows. This is true for most retailer stores. However, in some cases, a subset of customers may have soft time window constraints which can be violated with a penalty. The study in Chapter 2 considers this scenario. Extending soft time windows into the problem can make the study more general. Similarly, a straight forward extension of the work would be to generate warehouse plans with fixed routes but variable times for the start of loading for the routes. This extension retains the sequential decision making process as is done in practice, but provides managers a planning support to identify which routes need to be loaded earlier or later to ensure that the warehouse operations are feasible.

This study also assumes fluid order picking work rates which assumes that an order can be picked by multiple order pickers at the same time. This assumption is a good approximation for the order picking process in retail warehouses that have a large output volume where orders are large and pick routes are long. However, when orders are small (as in warehouses for e-commerce), the assignment of individual orders to order pickers can be necessary. Assigning orders to order pickers while considering multiple warehouse processes and routing of vehicles is likely to be a prohibitively complex problem. It will not allow for a scalable exact solution approach. Nevertheless, even a heuristic approach that can handle real-sized instances would be a valuable contribution to the warehouse management literature.

An overlapping theme in all of the studies in this dissertation is offline planning where all the orders are known before the planning is performed. This is not a restrictive assumption for retail warehouses where orders from retailers are known in advance. However, when orders arrive during the planning horizon, incorporating them into existing personnel scheduling decisions can be challenging and will require a new framework for formulating and solving the problems. We envision that the operational planning models will increasingly consider an online problem setting. The operating environment in warehouses also invites challenges to mitigate uncertainties. Order pickers may take a longer time to pick orders or take breaks. Trucks may arrive late at the warehouse or some of the dock doors may become nonfunctional. A paradigm shift in warehouse management that considers dynamic disruption management, similar to those in aircraft and railway operations, could be a challenging but worthwhile direction of study. This goal can be achieved as is done in practice by decomposing holistic problems into smaller sub-problems such as batching orders and assigning them to order pickers in real time. However, this approach can be suboptimal and provides opportunities for improvement. Using problem formulations that incorporate uncertainties, in particular a robust optimization framework, can help achieve these goals. We believe that this will be a new and exciting direction for the research on operational planning of warehouse processes.

Bibliography

Acar, K., Yalcin, A., & Yankov, D. (2012). Robust door assignment in less-than-truckload terminals. *Computers & Industrial Engineering*, 63(4), 729–738.

Agatz, N., Campbell, A., Fleischmann, M., & Savelsbergh, M. (2011). Time slot management in attended home delivery. *Transportation Science*, 45(3), 435–449.

Agustina, D., Lee, C., & Piplani, R. (2014). Vehicle scheduling and routing at a cross docking center for food supply chains. *International Journal of Production Economics*, 152, 29–41.

Akyol, D. E. & De Koster, R. (2013). Non-dominated time-window policies in city distribution. *Production and Operations Management*, 22(3), 739–751.

Akyol, E. & De Koster, R. (2018). Determining time windows in urban freight transport: A city cooperative approach. *Transportation Research Part E: Logistics and Transportation Review*, 118.

Alicke, K. & Lösch, M. (2010). Lean and mean: How does your supply chain shape up? *McKinsey & Company 2010*.

Alsheddy, A. & Tsang, E. P. (2011). Empowerment scheduling for a field workforce. *Journal of Scheduling*, 14(6), 639–654.

Amorim, P., Belo-Filho, M., Toledo, F., Almeder, C., & Almada-Lobo, B. (2013). Lot sizing versus batching in the production and distribution planning of perishable goods. *International Journal of Production Economics*, 146(1), 208–218.

Armstrong, R., Gao, S., & Lei, L. (2008). A zero-inventory production and distribution problem with a fixed customer sequence. *Annals of Operations Research*, 159(1), 395–414.

Arthur, D. & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027–1035).: Society for Industrial and Applied Mathematics.

Assadi, M. T. & Bagheri, M. (2016). Differential evolution and population-based simulated annealing for truck scheduling problem in multiple door cross-docking system. *Computers & Industrial Engineering*, 96, 149–161.

Aykin, T. (1996). Optimal shift scheduling with multiple break windows. *Management Science*, 42(4), 591–602.

Aykin, T. (2000). A comparative evaluation of modeling approaches to the labor shift scheduling problem. *European Journal of Operational Research*, 125(2), 381–397.

Azadeh, K., De Koster, R., & Roy, D. (2019). Robotized and automated warehouse systems: Review and recent developments. *Transportation Science*, 53(4), 917–945.

Baldacci, R., Mingozzi, A., & Roberti, R. (2011). New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5), 1269–1283.

Bard, J. F., Morton, D. P., & Wang, Y. M. (2007). Workforce planning at usps mail processing and distribution centers using stochastic optimization. *Annals of Operations Research*, 155(1), 51.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., & Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3), 316–329.

Bartholdi, J. J. & Gue, K. R. (2004). The best shape for a crossdock. *Transportation Science*, 38(2), 235–244.

Bartholdi, J. J. & Hackman, S. T. (2008). Allocating space in a forward pick area of a distribution center for small parts. *IIE Transactions*, 40(11), 1046–1053.

Bartholdi, J. J. & Hackman, S. T. (2019). *Warehouse & Distribution Science: Release 0.98.1*. Supply Chain and Logistics Institute.

Bartholdi III, J. J. & Gue, K. R. (2000). Reducing labor costs in an ltl crossdocking terminal. *Operations Research*, 48(6), 823–832.

Baumol, W. J. & Wolfe, P. (1958). A warehouse-location problem. *Operations Research*, 6(2), 252–263.

Bechtold, S. E. & Jacobs, L. W. (1990). Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science*, 36(11), 1339–1351.

Beliën, J. & Demeulemeester, E. (2008). A branch-and-price approach for integrating nurse and surgery scheduling. *European Journal of Operational Research*, 189(3), 652–668.

Bellanger, A., Hanafi, S., & Wilbaut, C. (2013). Three-stage hybrid-flowshop model for cross-docking. *Computers & Operations Research*, 40(4), 1109–1121.

Belo-Filho, M., Amorim, P., & Almada-Lobo, B. (2015). An adaptive large neighbourhood search for the operational integrated production and distribution problem of perishable products. *International Journal of Production Research*, 53(20), 6040–6058.

Benjamini, Y. & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, (pp. 1165–1188).

Berghman, L., Briand, C., Leus, R., & Lopez, P. (2015). The truck scheduling problem at crossdocking terminals-exclusive versus mixed mode. In *4th International Conference on Operations Research and Enterprise Systems (ICORES 2015)* (pp. pp–247).

Bertsimas, D., Chang, A., Mišić, V. V., & Mundru, N. (2019). The airlift planning problem. *Transportation Science*, 53(3), 773–795.

Bhandari, A., Scheller-Wolf, A., & Harchol-Balter, M. (2008). An exact and efficient algorithm for the constrained dynamic operator staffing problem for call centers. *Management Science*, 54(2), 339–353.

Bodnar, P., de Koster, R., & Azadeh, K. (2017). Scheduling trucks in a cross-dock with mixed service mode dock doors. *Transportation Science*, 51(1), 112–131.

Boysen, N. (2010). Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research*, 37(1), 32–41.

Boysen, N., Briskorn, D., & Emde, S. (2017). Sequencing of picking orders in mobile rack warehouses. *European Journal of Operational Research*, 259(1), 293–307.

Boysen, N., Briskorn, D., & Tschöke, M. (2013). Truck scheduling in cross-docking terminals with fixed outbound departures. *OR Spectrum*, 35(2), 479–504.

Boysen, N. & Fliedner, M. (2010). Cross dock scheduling: Classification, literature review and research agenda. *Omega*, 38(6), 413–422.

Boysen, N., Fliedner, M., & Scholl, A. (2010). Scheduling inbound and outbound trucks at cross docking terminals. *OR Spectrum*, 32(1), 135–161.

Bozer, Y. A. & Carlo, H. J. (2008). Optimizing inbound and outbound door assignments in less-than-truckload crossdocks. *IIE Transactions*, 40(11), 1007–1018.

Brown, A. M. (2003). Improving the efficiency of hub operations in a less-than-truckload distribution network. Master's thesis, Virginia Polytechnic and State University.

Brusco, M. J. & Jacobs, L. W. (1998). Personnel tour scheduling when starting-time restrictions are present. *Management Science*, 44(4), 534–547.

Brusco, M. J. & Jacobs, L. W. (2000). Optimal models for meal-break and start-time flexibility in continuous tour scheduling. *Management Science*, 46(12), 1630–1641.

Buijs, P., Vis, I. F., & Carlo, H. J. (2014). Synchronization in cross-docking networks: A research classification and framework. *European Journal of Operational Research*, 239(3), 593–608.

Cattaruzza, D., Absi, N., & Feillet, D. (2016). The multi-trip vehicle routing problem with time windows and release dates. *Transportation Science*, 50(2), 676–693.

Çeven, E. & Gue, K. R. (2015). Optimal wave release times for order fulfillment systems with deadlines. *Transportation Science*, 51(1), 52–66.

Chen, F. & Song, K. (2009). Minimizing makespan in two-stage hybrid cross docking scheduling problem. *Computers & Operations Research*, 36(6), 2066–2073.

Chen, T.-L., Cheng, C.-Y., Chen, Y.-Y., & Chan, L.-K. (2015). An efficient hybrid algorithm for integrated order batching, sequencing and routing problem. *International Journal of Production Economics*, 159, 158–167.

Chen, Z.-L. (2010). Integrated production and outbound distribution scheduling: review and extensions. *Operations Research*, 58(1), 130–148.

Chen, Z.-L. & Pundoor, G. (2006). Order assignment and scheduling in a supply chain. *Operations Research*, 54(3), 555–572.

Chiang, W.-C., Russell, R., Xu, X., & Zepeda, D. (2009). A simulation/metaheuristic approach to newspaper production and distribution supply chain problems. *International Journal of Production Economics*, 121(2), 752–767.

Chmielewski, A., Naujoks, B., Janas, M., & Clausen, U. (2009). Optimizing the door assignment in ltl-terminals. *Transportation Science*, 43(2), 198–210.

Clarke, G. & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568–581.

Contardo, C., Desaulniers, G., & Lessard, F. (2015). Reaching the elementary lower bound in the vehicle routing problem with time windows. *Networks*, 65(1), 88–99.

Cordeau, J.-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52(8), 928–936.

Cordeau, J.-F., Laporte, G., Pasin, F., & Ropke, S. (2010). Scheduling technicians and tasks in a telecommunications company. *Journal of Scheduling*, 13(4), 393–409.

Côté, M.-C., Gendron, B., & Rousseau, L.-M. (2011). Grammar-based integer programming models for multiactivity shift scheduling. *Management Science*, 57(1), 151–163.

Dabia, S., Ropke, S., & Van Woensel, T. (2019). Cover inequalities for a vehicle routing problem with time windows and shifts. *Transportation Science*, 53(5), 1354–1371.

Dahmen, S., Rekik, R., & Soumis, F. (2018). An implicit model for multi-activity shift scheduling problems. *Journal of Scheduling*, 21(3), 285–304.

Dallari, F., Marchet, G., & Melacini, M. (2009). Design of order picking system. *The International Journal of Advanced Manufacturing Technology*, 42(1-2), 1–12.

Danna, E. & Le Pape, C. (2005). Branch-and-price heuristics: A case study on the vehicle routing problem with time windows. In *Column generation* (pp. 99–129). Springer.

Dantzig, G. B. (1954). Letter to the editor—a comment on edie's "traffic delays at toll booths". *Journal of the Operations Research Society of America*, 2(3), 339–341.

De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2), 481–501.

Dellaert, N., Dashty Saridarq, F., Van Woensel, T., & Crainic, T. G. (2018). Branch-and-price–based algorithms for the two-echelon vehicle routing problem with time windows. *Transportation Science*, 53(2), 463–479.

Desaulniers, G. (2010). Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research*, 58(1), 179–192.

Desaulniers, G., Desrosiers, J., & Solomon, M. M. (2002). Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems. In *Essays and surveys in metaheuristics* (pp. 309–324). Springer.

Desaulniers, G., Lessard, F., & Hadjar, A. (2008). Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3), 387–404.

Desaulniers, G. & Villeneuve, D. (2000). The shortest path problem with time windows and linear waiting costs. *Transportation Science*, 34(3), 312–319.

Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2), 342–354.

Dondo, R. & Cerdá, J. (2014). A monolithic approach to vehicle routing and operations scheduling of a cross-dock system with multiple dock doors. *Computers & Chemical Engineering*, 63, 184–205.

Dondo, R. & Cerdá, J. (2015). The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system. *Computers & Chemical Engineering*, 76, 42–62.

Drexl, M. (2012). Synchronization in vehicle routing—a survey of vrps with multiple synchronization constraints. *Transportation Science*, 46(3), 297–316.

Dror, M. (1994). Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5), 977–978.

Edie, L. C. (1954). Traffic delays at toll booths. *Journal of the Operations Research Society of America*, 2(2), 107–138.

Elahipanah, M., Desaulniers, G., & Lacasse-Guay, E. (2013). A two-phase mathematical-programming heuristic for flexible assignment of activities and tasks to work shifts. *Journal of Scheduling*, 16(5), 443–460.

Elsayed, E. & Lee, M.-K. (1996). Order processing in automated storage/retrieval systems with due dates. *IIE Transactions*, 28(7), 567–577.

Elsayed, E., Lee, M.-K., Kim, S., & Scherer, E. (1993). Sequencing and batching procedures for minimizing earliness and tardiness penalty of order retrievals. *International Journal of Production Research*, 31(3), 727–738.

Enderer, F., Contardo, C., & Contreras, I. (2017). Integrating dock-door assignment and vehicle routing with cross-docking. *Computers & Operations Research*, 88, 30–43.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., & Sier, D. (2004a). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127(1–4), 21–144.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004b). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1), 3–27.

European Parliament, Council of the European Union (1991). Council Directive 91/533/EEC of 14 October 1991 on an employer's obligation to inform employees of the conditions applicable to the contract or employment relationship. `https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A31991L0533`.

European Parliament, Council of the European Union (2003). Directive 2003/88/ec of the european parliament and of the council of 4 november 2003 concerning certain aspects of the organisation of working time. `http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32003L0088`.

European Parliament, Council of the European Union (2019). Directive (EU) 2019/1152 of the European Parliament and of the Council of 20 June 2019 on transparent and predictable working conditions in the European Union. `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32019L1152`.

Farahani, P., Grunow, M., & Günther, H.-O. (2012). Integrated production and distribution planning for perishable food products. *Flexible Services and Manufacturing Journal*, 24(1), 28–51.

Feillet, D., Dejax, P., Gendreau, M., & Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3), 216–229.

Feillet, D., Gendreau, M., & Rousseau, L.-M. (2007). New refinements for the solution of vehicle routing problems with branch and price. *INFOR: Information Systems and Operational Research*, 45(4), 239–256.

Field, A. (2013). *Discovering statistics using IBM SPSS statistics*. Sage.

Fırat, M. & Hurkens, C. (2012). An improved mip-based approach for a multi-skill workforce scheduling problem. *Journal of Scheduling*, 15(3), 363–380.

Fischetti, M., Martello, S., & Toth, P. (1987). The fixed job schedule problem with spread-time constraints. *Operations Research*, 35(6), 849–858.

Fischetti, M., Martello, S., & Toth, P. (1989). The fixed job schedule problem with working-time constraints. *Operations Research*, 37(3), 395–403.

Fonseca, G. B., Nogueira, T. H., & Ravetti, M. G. (2019). A hybrid lagrangian metaheuristic for the cross-docking flow shop scheduling problem. *European Journal of Operational Research*, 275(1), 139–154.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Gademann, A., Van Den Berg, J. P., & Van Der Hoff, H. H. (2001). An order batching algorithm for wave picking in a parallel-aisle warehouse. *IIE Transactions*, 33(5), 385–398.

Gademann, N. & Velde, S. (2005). Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE Transactions*, 37(1), 63–75.

Gallien, J. & Weber, T. (2010). To wave or not to wave? order release policies for warehouses with an automated sorter. *Manufacturing & Service Operations Management*, 12(4), 642–662.

Garcia, J., Lozano, S., & Canca, D. (2004). Coordinated scheduling of production and delivery from multiple plants. *Robotics and Computer-Integrated Manufacturing*, 20(3), 191–198.

García, J. M. & Lozano, S. (2004). Production and vehicle scheduling for ready-mix operations. *Computers & Industrial Engineering*, 46(4), 803–816.

García, J. M. & Lozano, S. (2005). Production and delivery scheduling problem with time windows. *Computers & Industrial Engineering*, 48(4), 733–742.

Garey, M. R. & Johnson, D. S. (1979). Computers and intractability: a guide to the theory of np-completeness. 1979. *San Francisco, LA: Freeman*, 58.

Geismar, H. N., Laporte, G., Lei, L., & Sriskandarajah, C. (2008). The integrated production and transportation scheduling problem for a product with a short lifespan. *INFORMS Journal on Computing*, 20(1), 21–33.

Gérard, M., Clautiaux, F., & Sadykov, R. (2016). Column generation based approaches for a tour scheduling problem with a multi-skill heterogeneous workforce. *European Journal of Operational Research*, 252(3), 1019–1030.

Goel, A. (2010). Truck driver scheduling in the european union. *Transportation Science*, 44(4), 429–441.

Goel, A. & Irnich, S. (2017). An exact method for vehicle routing and truck driver scheduling problems. *Transportation Science*, 51(2), 737–754.

Goel, A. & Kok, L. (2012). Truck driver scheduling in the united states. *Transportation Science*, 46(3), 317–326.

Gong, Y. & De Koster, R. (2008). A polling-based dynamic order picking system for online retailers. *IIE Transactions*, 40(11), 1070–1082.

Gose, J. (2018). Growth in E-commerce Driving Larger Warehouse Designs. `https://urbanland.uli.org/planning-design/growth-e-commerce-driving-larger-warehouse-designs/`, Accessed: 10-02-2018.

Grangier, P., Gendreau, M., Rousseau, L.-M., & Lehuédé, F. (2017). A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers and Operations Research*, 84, 116–126.

Gu, J., Goetschalckx, M., & McGinnis, L. F. (2007). Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177(1), 1–21.

Gue, K. R. (1999). The effects of trailer scheduling on the layout of freight terminals. *Transportation Science*, 33(4), 419–428.

Gurobi Optimization, L. (2020). Gurobi optimizer reference manual, version 9.0.

Hall, L. A. & Shmoys, D. B. (1989). Approximation schemes for constrained scheduling problems. In *Foundations of Computer Science, 1989., 30th Annual Symposium on* (pp. 134–139).: IEEE.

Heidari, F., Zegordi, S. H., & Tavakkoli-Moghaddam, R. (2018). Modeling truck scheduling problem at a cross-dock facility through a bi-objective bi-level optimization approach. *Journal of Intelligent Manufacturing*, 29(5), 1155–1170.

Hempsch, C. & Irnich, S. (2008). Vehicle routing problems with inter-tour resource constraints. In *The Vehicle Routing Problem: Latest Advances and New Challenges* (pp. 421–444). Springer.

Henn, S. (2012). Algorithms for on-line order batching in an order picking warehouse. *Computers & Operations Research*, 39(11), 2549–2563.

Henn, S. (2015). Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses. *Flexible Services and Manufacturing Journal*, 27(1), 86–114.

Henn, S., Koch, S., Doerner, K. F., Strauss, C., & Wäscher, G. (2010). Metaheuristics for the order batching problem in manual order picking systems. *BuR-Business Research*, 3(1), 82–105.

Henn, S., Koch, S., & Wäscher, G. (2012). Order batching in order picking warehouses: a survey of solution approaches. In *Warehousing in the Global Supply Chain* (pp. 105–137). Springer.

Henn, S. & Schmid, V. (2013). Metaheuristics for order batching and sequencing in manual order picking systems. *Computers & Industrial Engineering*, 66(2), 338–351.

Heragu, S. S., Du, L., Mantel, R. J., & Schuur, P. C. (2005). Mathematical model for warehouse design and product allocation. *International Journal of Production Research*, 43(2), 327–338.

Hermel, D., Hasheminia, H., Adler, N., & Fry, M. J. (2016). A solution framework for the multi-mode resource-constrained cross-dock scheduling problem. *Omega*, 59, 157–170.

Holder, A. (2005). Navy personnel planning and the optimal partition. *Operations Research*, 53(1), 77–89.

Hurter, A. P. & Van Buer, M. G. (1996). The newspaper production/distribution problem. *Journal of Business Logistics*, 17(1), 85.

Irnich, S. (2008a). Resource extension functions: Properties, inversion, and generalization to segments. *OR Spectrum*, 30(1), 113–148.

Irnich, S. (2008b). A unified modeling and solution framework for vehicle routing and local search-based metaheuristics. *INFORMS Journal on Computing*, 20(2), 270–287.

Irnich, S. & Desaulniers, G. (2005). Shortest path problems with resource constraints. In *Column generation* (pp. 33–65). Springer.

Jans, R. (2009). Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. *INFORMS Journal on Computing*, 21(1), 123–136.

Janssen, R., Zwijnenberg, H., Blankers, I., & de Kruijff, J. (2015). Truck platooning. *Driving The Future of Transportation, TNO.*

Kallehauge, B., Larsen, J., Madsen, O. B., & Solomon, M. M. (2005). Vehicle routing problem with time windows. In *Column generation* (pp. 67–98). Springer.

Kniffin, K. M., Wansink, B., Devine, C. M., & Sobal, J. (2015). Eating together at the firehouse: how workplace commensality relates to the performance of firefighters. *Human Performance*, 28(4), 281–306.

Kohl, N., Desrosiers, J., Madsen, O. B., Solomon, M. M., & Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33(1), 101–116.

Kolen, A. W., Lenstra, J. K., Papadimitriou, C. H., & Spieksma, F. C. (2007). Interval scheduling: A survey. *Naval Research Logistics*, 54(5), 530–543.

Konur, D. & Golias, M. M. (2013a). Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty. *Computers & Industrial Engineering*, 65(4), 663–672.

Konur, D. & Golias, M. M. (2013b). Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach. *Transportation Research Part E: Logistics and Transportation Review*, 49(1), 71–91.

Krishnamoorthy, M., Ernst, A. T., & Baatar, D. (2012). Algorithms for large scale shift minimisation personnel task scheduling problems. *European Journal of Operational Research*, 219(1), 34–48.

Kroon, L. G., Salomon, M., & Van Wassenhove, L. N. (1995). Exact and approximation algorithms for the operational fixed interval scheduling problem. *European Journal of Operational Research*, 82(1), 190–205.

Kuijt, I. & Finlayson, B. (2009). Evidence for food storage and predomestication granaries 11,000 years ago in the jordan valley. *Proceedings of the National Academy of Sciences*, 106(27), 10966–10970.

Kuo, Y. (2013). Optimizing truck sequencing and truck dock assignment in a cross docking system. *Expert Systems with Applications*, 40, 5532–5541.

Ladier, A.-L. & Alpan, G. (2016). Cross-docking operations: Current research versus industry practice. *Omega*, 62, 145–162.

Lambert, D. M., Stock, J. R., & Ellram, L. M. (1998). *Fundamentals of logistics management*. McGraw-Hill/Irwin.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4), 408–416.

Lee, C.-Y. & Chen, Z.-L. (2001). Machine scheduling with transportation considerations. *Journal of Scheduling*, 4(1), 3–24.

Lee, J., Kim, B.-I., Johnson, A. L., & Lee, K. (2014). The nuclear medicine production and delivery problem. *European Journal of Operational Research*, 236(2), 461–472.

Lee, K., Kim, B. S., & Joo, C. M. (2012). Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance. *Expert Systems with Applications*, 39, 12975–12983.

Levene, H. et al. (1960). Robust tests for equality of variances. *Contributions to Probability and Statistics*, 1, 278–292.

Liao, T., Egbelu, P., & Chang, P.-C. (2013). Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations. *International Journal of Production Economics*, 141(1), 212–229.

Low, C., Chang, C.-M., Li, R.-K., & Huang, C.-L. (2014). Coordination of production scheduling and delivery problems with heterogeneous fleet. *International Journal of Production Economics*, 153, 139–148.

Low, C., Li, R.-K., & Chang, C.-M. (2013). Integrated scheduling of production and delivery with time windows. *International Journal of Production Research*, 51(3), 897–909.

Matusiak, M., de Koster, R., Kroon, L., & Saarinen, J. (2014). A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. *European Journal of Operational Research*, 236(3), 968–977.

Matusiak, M., de Koster, R., & Saarinen, J. (2017). Utilizing individual picker skills to improve order batching in a warehouse. *European Journal of Operational Research*, 263(3), 888–899.

Miao, Z., Lim, A., & Ma, H. (2009). Truck dock assignment problem with operational time constraint within crossdocks. *European Journal of Operational Research*, 192(1), 105–115.

Michel, R. (2016). 2016 warehouse/dc operations survey: ready to confront complexity. *Logistics management (Highlands Ranch, Colo.: 2002)*.

Ministry of Social Affairs and Employment (2010). The working hours act. `https://www.rijksoverheid.nl/binaries/rijksoverheid/documenten/brochures/2010/05/10/de-arbeidstijdenwet-engels/brochure-de-arbeidstijdenwet-engels.pdf`.

Molavi, D., Shahmardan, A., & Sajadieh, M. S. (2018). Truck scheduling in a cross docking systems with fixed due dates and shipment sorting. *Computers & Industrial Engineering*, 117, 29–40.

Moons, S., Braekers, K., Ramaekers, K., Caris, A., & Arda, Y. (2019). The value of integrating order picking and vehicle routing decisions in a b2c e-commerce environment. *International Journal of Production Research*, (pp. 1–19).

Moons, S., Ramaekers, K., Caris, A., & Arda, Y. (2017). Integration of order picking and vehicle routing in a b2c e-commerce context. *Flexible Services and Manufacturing Journal*, (pp. 1–31).

Muñuzuri, J., Larrañeta, J., Onieva, L., & Cortés, P. (2005). Solutions applicable by local administrations for urban logistics improvement. *Cities*, 22(1), 15–28.

Naso, D., Surico, M., Turchiano, B., & Kaymak, U. (2007). Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. *European Journal of Operational Research*, 177(3), 2069–2099.

Nassief, W., Contreras, I., Guignard, M., Hahn, P., & Jaumard, B. (2017). The container scheduling and cross-dock door selection problem.

Optimization, G. (2016). Gurobi optimizer reference manual version 6.5, 2016. `https://www.gurobi.com/documentation/6.5/refman.pdf`. Accessed on 21 June 2017.

Pisinger, D. & Ropke, S. (2010). Large neighborhood search. In *Handbook of metaheuristics* (pp. 399–419). Springer.

Potts, C. N. (1980). Analysis of a heuristic for one machine sequencing with release dates and delivery times. *Operations Research*, 28(6), 1436–1441.

Quak, H. & de Koster, R. (2009). Delivering goods in urban areas: how to deal with urban policy restrictions and the environment. *Transportation Science*, 43(2), 211–227.

Quak, H. J. & de Koster, R. (2007). Exploring retailers' sensitivity to local sustainability policies. *Journal of Operations Management*, 25(6), 1103–1122.

Rasmussen, M. S., Justesen, T., Dohn, A., & Larsen, J. (2012). The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research*, 219(3), 598–610.

Rickman, G. (2002). Rome, ostia and portus: the problem of storage. *Mélanges de l'école française de Rome*, 114(1), 353–362.

Rijal, A., Bijvank, M., & de Koster, R. (2019). Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors. *European Journal of Operational Research*, 278(3), 752–771.

Rijal, A., Bijvank, M., Goel, A. S., & De Koster, R. (2020). Workforce scheduling with order-picking assignments in distribution facilities. *Working paper*.

Roodbergen, K. J. & de Koster, R. (2001). Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research*, 133(1), 32–43.

Roodbergen, K. J. & Vis, I. F. (2009). A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research*, 194(2), 343–362.

Ropke, S. & Pisinger, D. (2006a). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4), 455–472.

Ropke, S. & Pisinger, D. (2006b). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3), 750–775.

Rouwenhorst, B., Reuter, B., Stockrahm, V., van Houtum, G.-J., Mantel, R., & Zijm, W. H. (2000). Warehouse design and control: Framework and literature review. *European Journal of Operational Research*, 122(3), 515–533.

Saddle Creek Corporation (2011). 2011 cross-docking trends report. `http://www.distributiongroup.com/articles/070111DCMwe.pdf`. Accessed on 21 November 2016.

Savelsbergh, M. W. (1985). Local search in routing problems with time windows. *Annals of Operations research*, 4(1), 285–305.

Savelsbergh, M. W. (1992). The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4(2), 146–154.

Schiffer, M., Schneider, M., Walther, G., & Laporte, G. (2019). Vehicle routing and location routing with intermediate stops: a review. *Transportation Science*, 53(2), 319–343.

Scholz, A., Schubert, D., & Wäscher, G. (2017). Order picking with multiple pickers and due dates – simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems. *European Journal of Operational Research*, 263(2), 461–478.

Shakeri, M., Low, M. Y. H., Turner, S. J., & Lee, E. W. (2012). A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. *Computers & Operations Research*, 39(11), 2564–2577.

Sherali, H. D. & Smith, J. C. (2001). Improving discrete model representations via symmetry considerations. *Management Science*, 47(10), 1396–1407.

Smet, P., Ernst, A. T., & Berghe, G. V. (2016). Heuristic decomposition approaches for an integrated task scheduling and personnel rostering problem. *Computers & Operations Research*, 76, 60–72.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2), 254–265.

Spliet, R., Dabia, S., & Van Woensel, T. (2018). The time window assignment vehicle routing problem with time-dependent travel times. *Transportation Science*, 52(2), 261–276.

Spliet, R. & Gabor, A. F. (2014). The time window assignment vehicle routing problem. *Transportation Science*, 49(4), 721–731.

Stenger, A., Vigo, D., Enz, S., & Schwind, M. (2013). An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science*, 47(1), 64–80.

Sungur, B., Özgüven, C., & Kariper, Y. (2017). Shift scheduling with break windows, ideal break periods, and ideal waiting times. *Flexible Services and Manufacturing Journal*, 29(2), 203–222.

Thompson, G. M. (1988). *A comparison of techniques for scheduling non-homogeneous employees in a service environment subject to non-cyclical demand volume I, chapters l-7*. PhD thesis.

Thompson, G. M. (1995). Improved implicit optimal modeling of the labor shift scheduling problem. *Management Science*, 41(4), 595–607.

Thompson, G. M. & Pullman, M. E. (2007). Scheduling workforce relief breaks in advance versus in real-time. *European Journal of Operational Research*, 181(1), 139–155.

Tilk, C. & Goel, A. (2020). Bidirectional labeling for solving vehicle routing and truck driver scheduling problems. *European Journal of Operational Research*, 283(1), 108–124.

Tilk, C. & Irnich, S. (2017). Dynamic programming for the minimum tour duration problem. *Transportation Science*, 51(2), 549–565.

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. (2010). *Facilities Planning*. John Wiley & Sons, Hoboken, NJ.

Tootkaleh, S. R., Ghomi, S. F., & Sajadieh, M. S. (2016). Cross dock scheduling with fixed outbound trucks departure times under substitution condition. *Computers & Industrial Engineering*, 92, 50–56.

Tsai, C.-Y., Liou, J. J., & Huang, T.-M. (2008). Using a multiple-ga method to solve the batch picking problem: Considering travel distance and order due time. *International Journal of Production Research*, 46(22), 6533–6555.

Van Belle, J., Valckenaers, P., Berghe, G. V., & Cattrysse, D. (2013). A tabu search approach to the truck scheduling problem with multiple docks and time windows. *Computers & Industrial Engineering*, 66(4), 818–826.

Van Belle, J., Valckenaers, P., & Cattrysse, D. (2012). Cross-docking: State of the art. *Omega*, 40(6), 827–846.

Van Buer, M. G., Woodruff, D. L., & Olson, R. T. (1999). Solving the medium newspaper production/distribution problem. *European Journal of Operational Research*, 115(2), 237–253.

van den Bergh, J., Beliën, J., de Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3), 367–385.

Van der Zon, N. (2017). Vehicle routing with departure smoothing. Master's thesis, Erasmus University Rotterdam.

Van Gils, T., Ramaekers, K., Braekers, K., Depaire, B., & Caris, A. (2018a). Increasing order picking efficiency by integrating storage, batching, zone picking, and routing policy decisions. *International Journal of Production Economics*, 197, 243–261.

Van Gils, T., Ramaekers, K., Caris, A., & de Koster, R. B. (2018b). Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research*, 267(1), 1–15.

Vanderbeck, F. (2000). On dantzig-wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48(1), 111–128.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., & Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3), 611–624.

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). 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.

Vis, I. F. & Roodbergen, K. J. (2008). Positioning of goods in a cross-docking environment. *Computers & Industrial Engineering*, 54(3), 677–689.

Wisittipanich, W. & Hengmeechai, P. (2017). Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization. *Computers & Industrial Engineering*, 113, 793–802.

Woeginger, G. J. (1998). A polynomial-time approximation scheme for single-machine sequencing with delivery times and sequence-independent batch set-up times. *Journal of Scheduling*, 1(2), 79–87.

Yu, W. & Egbelu, P. J. (2008). Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research*, 184(1), 377–396.

Zdrzałka, S. (1995). Analysis of approximation algorithms for single-machine scheduling with delivery times and sequence independent batch setup times. *European Journal of Operational Research*, 80(2), 371–380.

Zhang, T., Saharidis, G., Theofanis, S., & Boile, M. (2010). Scheduling of inbound and outbound trucks at cross-docks. *Transportation Research Record*, 2162, 9–16.

About the author

Arpan Rijal was born in Sarlahi, Nepal on May 12, 1989. He holds a dual bachelor's degree in Economics and Mathematics, summa cum laude, from St. Peter's University, USA and master's degree in International Logistics Engineering and Management from Jacobs University, Bremen, Germany. In January 2015, he joined the Technology & Operations Department at Rotterdam School of Management for his doctoral studies.

His research interests revolve around optimization of warehousing and transportation processes. His research findings have been presented at international conferences such as INFORMS Annual Meeting, POMS Annual Meeting, International Conference on Computational Logistics and International Symposium on Scientific Logistics.

Currently, Arpan is a Postdoctoral Fellow at the Department of Operations at University of Groningen.

Portfolio

Publications

Publications in journals:

Rijal, A., Bijvank, M., and De Koster, R. (2019). Integrated Scheduling and Assignment of Trucks at Cross-dock Terminals with Mixed Service Mode Dock Doors, *European Journal of Operational Research*, 278(3), 752-771.

Rosca, E., Möllering, G., Rijal, A., and Bendul, J. (2019). Supply Chain Inclusion in Base of the Pyramid Markets, *International Journal of Physical Distribution & Logistics Management*, 49(5), 575-598.

Book chapters:

Rijal, A., Bijvank, M., and De Koster, R. (2016). "Integrated versus Sequential Scheduling and Assignment at a Unit Load Cross-dock", in *Progress in Material Handling Research: 2016.*

Working papers:

Rijal, A., Bijvank, M., Goel, A., and De Koster, R. (2019). Workforce Scheduling with Order-picking Assignments in Distribution Facilities.

Rijal, A., Bijvank, M., and De Koster, R. (2020). Impact of Warehouse Capacities on Vehicle Routing.

PhD Courses

Advanced Inventory Theory
Algorithms and Complexity
Data Visualization, Web Scraping, and Text Analysis in R
Deep Learning Demystified
English (CPE) certificate
Interior Point Methods
OR-Games
Programming
Publishing Strategy
Randomized Algorithms
Robust Optimization

Facility Logistics Management
Freight Transport Management
Integer Programming
Interaction Performance Training / Coaching
School on Column Generation
Scientific Integrity
Statistical Methods

Teaching

Lecturer:

Research Training and Bachelor Thesis, Rotterdam School of Management, 2019.
Distribution & E-commerce, Jacobs University, 2018-2019.

Tutorial lecturer:

Facility Logistics Management, Rotterdam School of Management, 2015-2019.
Global Supply Chain Management, Rotterdam School of Management, 2015, 2018.
Operations Management, Rotterdam School of Management, 2016-2017.

Guest lecturer:

Distribution Logistics, Tilburg University, 2018-2020.
Supply Chain Management, Tilburg University, 2018-2020.

Conferences attended

ICCL 2015, Delft, The Netherlands
ISSL 2016, Karlsruhe, Germany
ICCL 2016, Lisbon, Portugal
ELA 2016, Vienna, Austria
TRAIL Congress 2016, Utrecht, The Netherlands
POMS International Conference 2017, Tel Aviv, Israel
INFORMS Annual Meeting 2018, Phoenix, USA
POMS Annual Meeting 2019, Washington D.C., USA
INFORMS Annual Meeting 2019, Seattle, USA

Summary

Warehouse management is critical to the success of supply chains. But, to achieve larger supply chain goals such as on-time and cost efficient delivery can be difficult because of the limited resources available at the warehouse. Planning the human workers in a warehouse is arguably complex, because managers have to consider issues such as start and end times of shifts, breaks, and incentive payment schemes. Obviously, these types of restrictions are less important in automated warehouses. When warehouses have temporal restrictions on the processing time of inbound trucks or when they have deadlines for outbound orders, these constraints impact the schedules and cost of employing human workers. In this environment, warehouse management has to consider the temporal restrictions from external entities as well as the limited resources available at the warehouse. This thesis contributes to the warehousing literature by focusing on manual warehouse operations with such external temporal constraints. We develop mathematical models for three operational warehouse planning problems that need to plan human workers with consideration for the limited warehouse resources and with external temporal constraints. Additionally, we propose solution approaches for these problems and perform computational experiments to derive insights.

In Chapter 2, we study integrated scheduling and assignment of trucks to dock doors in a unit-load cross-dock facility with mixed-mode dock doors. The cross-dock operation is constrained by processing time windows for inbound and outbound vehicles. The aim of the operational problem is to synchronize the processing time and dock location of inbound and outbound trucks such that outbound trucks depart from the cross-dock with minimal delays and such that shipments can be moved from inbound to outbound trucks directly without being moved to a temporary storage area. Additionally, the objective is to have cross-dock workers travel as little as possible within the cross-dock facility to move shipments between dock doors and between the dock doors and temporary storage area. We formulate a mathematical model for the problem and develop an adaptive large neighborhood search algorithm to solve it. The solution approach is benchmarked against state-of-the-art sequential approaches that solve the

problem sequentially by scheduling trucks first and assigning them second. The results suggest that the integrated approach can reduce operational cost by 12% on average compared to the sequential approach. Furthermore, computational experiments show that the percentage as well as the positions of mixed-mode dock doors are important for U-shaped cross-dock terminals with low dock-door utilization. However, only the percentage of mixed-mode dock doors is important for cross-dock facilities that are heavily utilized. The solution approach is also applied to a case study investigating cross-dock operations of a large Dutch retailer. The retailer currently uses a simplified version of our solution approach to plan the dock-door layouts in its cross-dock facilities.

The third chapter studies a shift scheduling problem for distribution centers with limited staging space. The warehouse has limited space to buffer orders compared to the number of orders that leave the warehouse, and each customer order has to be picked and brought to the staging area within a predefined order-dependent delivery due time window. The warehouse uses flexible order pickers who work in multiple, potentially overlapping, shifts to pick and deliver the orders. To ensure that the orders are delivered to the staging area within their due time window, scheduling of the flexible order pickers has to consider the start and end times of shifts, break periods of workers as well as the order to picker assignment and sequencing of orders. We present two formulations for the problem and develop two solution approaches. The first solution approach is exact, based on a branch-and-price framework, while the second one is a heuristic based on large neighborhood search. Extensive computational experiments are conduced to validate the solution approaches. Additionally, a case study is done to investigate the impact of flexible break times for a large Dutch retailer's distribution center. The results show that the retailer can make substantial savings by adopting flexible breaks that have 15 minutes flexibility in the time they start. The solution approach developed in this study also has been translated to a decision support tool for the retailer.

In Chapter 4 of this thesis, we investigate planning of operations in warehouses with limited order picking, staging and loading resources. The warehouse uses manual workers who work in multiple shifts to pick orders. Once an order is picked it is brought to a staging area before it is loaded in a truck via a dock door. The staging space acts as a buffer between the order picking and dispatching of trucks and presents unique challenges to managers. For example, if many vehicles need to depart simultaneously, the staging space can be full and the order picking may have to stop. In the traditional approach, the deliveries to customers (or retail stores) is planned first and the outbound schedule of orders is then imposed on the warehouse. This sequential approach is not only suboptimal but can lead to infeasible plans because of the limited resource availability in the warehouse. In this study, we present a model of the problem that can generate routes with consideration for the limited resources in the warehouse. We develop a dynamic programming algorithm to solve the problem. Computational experiments on instances generated from real-life data show that the proposed solution approach can generate good solutions. Furthermore, the results suggest that when warehouse resources are severely limited, it is cheaper to focus on hiring fewer order pickers at the warehouse than to adopt the traditional approach of optimizing transportation resources first and warehouse resources second. This observation holds true even when the cost of order pickers is 50% less than the cost of truck drivers.

Samenvatting (Summary in Dutch)

Magazijnbeheer is cruciaal voor het succes van supply chains. Echter, om grotere supply chain doelen te bereiken, zoals: een tijdige en kostenefficiënte levering, kan een uitdaging zijn vanwege de beperkte middelen die beschikbaar zijn in de magazijnen. Wat het plannen van de werknemers in een magazijn complex maakt, is dat managers rekening moeten houden met zaken als begin- en eindtijden van (ploegen)diensten, pauzes en beloningsregelingen. Dit soort beperkingen zijn uiteraard minder belangrijk in geautomatiseerde magazijnen. Wanneer magazijnen tijd gerelateerde restricties hebben op de verwerkingstijd van inkomende vrachtwagens of deadlines hebben voor uitgaande bestellingen, hebben deze restricties invloed op de schema's en kosten van de ingezette werknemers. In deze omgeving moeten magazijnbeheerders rekening houden met de tijd gerelateerde beperkingen die veroorzaakt worden door externe factoren en rekening houden met de beperkte middelen die beschikbaar zijn in het magazijn. Dit proefschrift draagt bij aan de wetenschappelijke literatuur op het gebied van magazijnen en distributiecentra door zich te concentreren op handmatige magazijn operaties met dergelijke externe tijdsbeperkingen. We ontwikkelen wiskundige modellen voor drie operationele magazijnplanningsproblemen die werknemers moeten plannen met inachtneming van beperkte magazijnmiddelen en met externe tijdsbeperkingen. Daarnaast stellen we oplossingsbenaderingen voor deze problemen voor en voeren we computationele experimenten uit om inzichten te verkrijgen.

In hoofdstuk 2 bestuderen we de geïntegreerde planning en toewijzing van vrachtwagens voor het docken van deuren in een unit-load crossdock-faciliteit met mixed-mode dockdeuren. De crossdock-operatie is gebonden aan restricties met betrekking tot tijdvensters voor inkomende en uitgaande voertuigen. Het doel van het operationele probleem is om de verwerkingstijd en de docklocatie van inkomende en uitgaande vrachtwagens zo te synchroniseren dat uitgaande vrachtwagens met minimale vertragingen van het crossdock vertrekken en zo te synchroniseren dat zendingen rechtstreeks van inkomende naar uitgaande vrachtwagens kunnen worden verplaatst zonder naar een tijdelijke opslagruimte te hoeven gaan. Daarnaast is het doel om crossdock-medewerkers zo weinig mogelijk binnen de crossdock-faciliteit te laten rondlopen om zendingen tussen dockdeuren en tijdelijke opslagruimte te verplaatsen. We formuleren een wiskundig model voor het probleem en ontwikkelen een variant op het Adaptive Large Neighborhood Search algoritme om het op te lossen. De oplossingsbenadering wordt vergeleken met de modernste sequentiële benaderingen die het probleem achtereenvolgens oplossen door eerst vrachtwagens te plannen en ze vervolgens toe te wijzen. De resultaten suggereren dat de geïntegreerde aanpak de operationele kosten gemiddeld met 12% kan verlagen in vergelijking met de sequentiële aanpak. Bovendien laten computationele experimenten zien dat zowel het percentage als de posities van mixed-mode dockdeuren belangrijk zijn voor U-vormige crossdockterminals met een laag gebruik van dockdeuren. Echter, alleen het percentage mixed-mode dockdeuren is belangrijk voor crossdock-faciliteiten die intensief worden gebruikt. De oplossingsaanpak wordt ook toegepast op een case study naar crossdock-activiteiten van een grote Nederlandse detailhandelaar. De detailhandelaar gebruikt momenteel een vereenvoudigde versie van onze oplossingsbenadering om de lay-out van de dockdeur in zijn crossdock-faciliteiten te plannen.

Het derde hoofdstuk bestudeert een probleem met shiftplanning voor distributiecentra met beperkte orderverzamelingsruimte. Het magazijn heeft beperkte ruimte om bestellingen te bufferen in vergelijking met het aantal bestellingen dat het magazijn verlaat, en elke klantorder moet worden gepickt en naar het verzamelgebied worden gebracht binnen een vooraf gedefinieerde orderafhankelijke leveringstijdsvenster. Het magazijn maakt gebruik van flexibele orderpickers die in meerdere, mogelijk overlappende, ploegen werken om de orders te picken en af te leveren. Om ervoor te zorgen dat de bestellingen binnen het gewenste tijdsbestek worden afgeleverd bij het verzamelgebied, moet bij het plannen van de flexibele orderpickers rekening worden gehouden met de begin- en eindtijden van diensten, met pauzeperiodes van werknemers, evenals met de toewijzing van bestellingen aan pickers en de volgorde van die bestellingen. We presenteren twee formuleringen voor het probleem en ontwikkelen twee oplossingsrichtingen. De eerste oplossingsaanpak is exact, gebaseerd op een branch-and-price framework, terwijl de tweede een heuristiek is gebaseerd op het Large Neighborhood Search algoritme. Er worden uitgebreide computationele experimenten uitgevoerd om de oplossingsbenaderingen te evalueren. Daarnaast wordt er een case study gedaan om de impact van flexibele pauzetijden voor een groot distributiecentrum van een Nederlandse detailhandelaar te onderzoeken. De resultaten tonen aan dat de detailhandelaar aanzienlijke besparingen kan realiseren door flexibele pauzes in te voeren die 15 minuten flexibiliteit hebben in hun aanvangstijd. De oplossingsaanpak die in dit onderzoek is ontwikkeld, is ook vertaald naar een beslissingsondersteunende tool voor de detailhandelaar.

In Hoofdstuk 4 van dit proefschrift onderzoeken we de planning van operaties in magazijnen met beperkte order picking- en laadmiddelen en beperkingen gerelateerd aan de orderverzamelplaats. Het magazijn maakt gebruik van werknemers die in meerdere ploegen werken om orders te picken. Nadat een bestelling is gepickt, wordt deze naar een verzamelplaats gebracht voordat deze via een dockdeur in een vrachtwagen wordt geladen. De verzamelplaats fungeert als buffer tussen het order verzamelen en verzenden van vrachtwagens en stelt managers voor unieke uitdagingen. Als er bijvoorbeeld veel voertuigen tegelijk moeten vertrekken, kan de verzamelplaats vol zijn en moet het order verzamelen mogelijk stoppen. Bij de traditionele aanpak worden de leveringen aan klanten (of winkels) eerst gepland en vervolgens wordt het uitgaande orderschema opgelegd aan het magazijn. Deze sequentiële aanpak is niet alleen suboptimaal, maar kan tot onhaalbare plannen leiden vanwege de beperkte beschikbaarheid van middelen in het magazijn. In deze studie presenteren we een model van het probleem dat routes kan genereren met inachtneming van de beperkte middelen in het magazijn. We ontwikkelen een dynamic programming algoritme om het probleem op te lossen. Computationele experimenten met instanties die zijn gegenereerd op basis van real-life data laten zien dat de voorgestelde oplossingsaanpak goede oplossingen kan opleveren. Bovendien suggereren de resultaten dat wanneer de magazijnmiddelen zeer beperkt zijn, het goedkoper is om te focussen op het inhuren van minder orderpickers in het magazijn dan om de traditionele aanpak te volgen, waarbij eerst de transportmiddelen en daarna de magazijnmiddelen worden geoptimaliseerd. Deze waarneming geldt zelfs wanneer de kosten van orderpickers 50% lager zijn dan die van vrachtwagenchauffeurs.

ERIM Ph.D. Series Research in Management

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: http://repub.eur.nl/pub. ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics at the Erasmus University Rotterdam (EUR).

Dissertations last five years

Ahmadi, S.,*A motivational perspective to decision-making and behavior in organizations*, Promotors: Prof. J.J.P. Jansen & Dr T.J.M. Mom, EPS-2019-477-S&E, https://repub.eur.nl/pub/116727

Akemu, O.,*Corporate Responses to Social Issues: Essays in Social Entrepreneurship and Corporate Social Responsibility*, Promotors: Prof. G.M. Whiteman & Dr S.P. Kennedy, EPS-2017-392-ORG, https://repub.eur.nl/pub/95768

Albuquerque de Sousa, J.A.,*International stock markets: Essays on the determinants and consequences of financial market development*, Promotors: Prof. M.A. van Dijk & Prof. P.A.G. van Bergeijk, EPS-2019-465-F&A, https://repub.eur.nl/pub/115988

Alserda, G.A.G.,*Choices in Pension Management*, Promotors: Prof. S.G. van der Lecq & Dr O.W. Steenbeek, EPS-2017-432-F&A, https://repub.eur.nl/pub/103496

Arampatzi, E.,*Subjective Well-Being in Times of Crises: Evidence on the Wider Impact of Economic Crises and Turmoil on Subjective Well-Being,* Promotors: Prof.

H.R. Commandeur, Prof. F. van Oort & Dr. M.J. Burger, EPS-2018-459-S&E, https://repub.eur.nl/pub/111830

Avci, E.,*Surveillance of Complex Auction Markets: a Market Policy Analytics Approach*, Promotors: Prof. W. Ketter, Prof. H.W.G.M. van Heck & Prof. D.W. Bunn, EPS-2018-426-LIS, https://repub.eur.nl/pub/106286

Balen, T.H. van, *Challenges of Early Stage Entrepreneurs: the Roles of Vision Communication and Team Membership Change*, Promotors: Prof. J.C.M. van den Ende & Dr M. Tarakci, EPS-2019-468-LIS, https://repub.eur.nl/pub/115654

Bernoster, I.,*Essays at the Intersection of Psychology, Biology, and Entrepreneurship*, Promotors: Prof. A.R. Thurik, Prof. I.H.A. Franken & Prof. P.J.F Groenen, EPS-2018-463-S&E, https://repub.eur.nl/pub/113907

Bouman, P.,*Passengers, Crowding and Complexity: Models for Passenger Oriented Public Transport*, Prof. L.G. Kroon, Prof. A. Schöbel & Prof. P.H.M. Vervest, EPS-2017-420-LIS, https://repub.eur.nl/pub/100767

Breugem, T.,*Crew Planning at Netherlands Railways: Improving Fairness, Attractiveness, and Efficiency*, Promotors: Prof. D. Huisman & Dr T.A.B. Dollevoet, EPS-2020494-LIS, https://repub.eur.nl/pub/124016

Bunderen, L. van, Tug-of-War: *Why and when teams get embroiled in power struggles*, Promotors: Prof. D.L. van Knippenberg & Dr. L. Greer, EPS-2018-446-ORG, https://repub.eur.nl/pub/105346

Burg, G.J.J. van den, *Algorithms for Multiclass Classification and Regularized Regression*, Promotors: Prof. P.J.F. Groenen & Dr. A. Alfons, EPS-2018-442-MKT, https://repub.eur.nl/pub/103929

Chammas, G.,*Portfolio concentration*, Promotor: Prof. J. Spronk, EPS-2017-410-F&E, https://repub.eur.nl/pub/94975

Chan, H.Y.,*Decoding the consumer's brain: Neural representations of consumer experience*, Promotors: Prof. A. Smidts & Dr M. A.S. Boksem, EPS-2019-493-MKT, https://repub.eur.nl/pub/124931

Dalmeijer, K.,*Time Window Assignment in Distribution Networks*, Promotors: Prof A.P.M. Wagelmans & Dr R. Spliet, EPS-2019-486-LIS, https://repub.eur.nl/pub/120773

Dennerlein, T., *Empowering Leadership and Employees' Achievement Motivations: the Role of Self-Efficacy and Goal Orientations in the Empowering Leadership Process*,

Promotors: Prof. D.L. van Knippenberg & Dr J. Dietz, EPS-2017-414-ORG, https://repub.eur.nl/pub/98438

Dolgova, E.,*On Getting Along and Getting Ahead: How Personality Affects Social Network Dynamics*, Promotors: Prof. P.P.M.A.R Heugens & Prof. M.C. Schippers, EPS-2019-455-S&E, https://repub.eur.nl/pub/119150

Duijzer, L.E.,*Mathematical Optimization in Vaccine Allocation,* Promotors: Prof. R. Dekker & Dr W.L. van Jaarsveld, EPS-2017-430-LIS, https://repub.eur.nl/pub/101487

Eijlers, E.,*Emotional Experience and Advertising Effectiveness: on the use of EEG in marketing*, Prof. A. Smidts & Prof. M.A.S. Boksem, Eps-2019-487-MKT, https://repub.eur.nl/pub/124053

El Nayal, O.S.A.N.,*Firms and the State: An Examination of Corporate Political Activity and the Business-Government Interface*, Promotor: Prof. J. van Oosterhout & Dr. M. van Essen, EPS-2018-469-S&E, https://repub.eur.nl/pub/114683

Feng, Y.,*The Effectiveness of Corporate Governance Mechanisms and Leadership Structure: Impacts on strategic change and firm performance*, Promotors: Prof. F.A.J. van den Bosch, Prof. H.W. Volberda & Dr J.S. Sidhu, EPS-2017-389-S&E, https://repub.eur.nl/pub/98470

Frick, T.W.,*The Implications of Advertising Personalization for Firms, Consumer, and Ad Platfroms*, Promotors: Prof. T. Li & Prof. H.W.G.M. van Heck, EPS-2018-452-LIS, https://repub.eur.nl/pub/110314

Fytraki, A.T.,*Behavioral Effects in Consumer Evaluations of Recommendation Systems*, Promotors: Prof. B.G.C. Dellaert & Prof. T. Li, EPS-2018-427-MKT, https://repub.eur.nl/pub/110457

Ghazizadeh, P. *Empirical Studies on the Role of Financial Information in Asset and Capital Markets*, Promotors: Prof. A. de Jong & Prof. E. Peek,EPS-2019-470-F&A, https://repub.eur.nl/pub/114023

Giurge, L.,*A Test of Time; A temporal and dynamic approach to power and ethics*, Promotors: Prof. M.H. van Dijke & Prof. D. De Cremer, EPS-2017-412-ORG, https://repub.eur.nl/pub/98451

Gobena, L.,*Towards Integrating Antecedents of Voluntary Tax Compliance,* Promotors: Prof. M.H. van Dijke & Dr P. Verboon, EPS-2017-436-ORG, https://repub.eur.nl/pub/103276

Groot, W.A.,*Assessing Asset Pricing Anomalies*, Promotors: Prof. M.J.C.M. Verbeek & Prof. J.H. van Binsbergen, EPS-2017-437-F&A, https://repub.eur.nl/pub/103490

Hanselaar, R.M.,*Raising Capital: On pricing, liquidity and incentives,* Promotors: Prof. M.A. van Dijk & Prof. P.G.J. Roosenboom, EPS-2018-429-F&A, https://repub.eur.nl/pub/113274

Harms, J. A.,*Essays on the Behavioral Economics of Social Preferences and Bounded Rationality*, Prof. H.R. Commandeur & Dr K.E.H. Maas, EPS-2018-457-S&E, https://repub.eur.nl/pub/108831

Hendriks, G.,*Multinational Enterprises and Limits to International Growth: Links between Domestic and Foreign Activities in a Firm's Portfolio*, Promotors: Prof. P.P.M.A.R. Heugens & Dr. A.H.L Slangen, EPS-2019-464-S&E, https://repub.eur.nl/pub/114981

Hengelaar, G.A.,*The Proactive Incumbent: Holy grail or hidden gem? Investigating whether the Dutch electricity sector can overcome the incumbent's curse and lead the sustainability transition*, Promotors: Prof. R.J. M. van Tulder & Dr K. Dittrich, EPS-2018-438-ORG, https://repub.eur.nl/pub/102953

Jacobs, B.J.D.,*Marketing Analytics for High-Dimensional Assortments,* Promotors: Prof. A.C.D. Donkers & Prof. D. Fok, EPS-2017-445-MKT, https://repub.eur.nl/pub/103497

Jia, F.,*The Value of Happiness in Entrepreneurship,* Promotors: Prof. D.L. van Knippenberg & Dr Y. Zhang, EPS-2019-479-ORG, https://repub.eur.nl/pub/115990

Kahlen, M. T.,*Virtual Power Plants of Electric Vehicles in Sustainable Smart Electricity Markets*, Promotors: Prof. W. Ketter & Prof. A. Gupta, EPS-2017-431-LIS, https://repub.eur.nl/pub/100844

Kampen, S. van, *The Cross-sectional and Time-series Dynamics of Corporate Finance: Empirical evidence from financially constrained firms*, Promotors: Prof. L. Norden & Prof. P.G.J. Roosenboom, EPS-2018-440-F&A, https://repub.eur.nl/pub/105245

Karali, E.,*Investigating Routines and Dynamic Capabilities for Change and Innovation*, Promotors: Prof. H.W. Volberda, Prof. H.R. Commandeur & Dr J.S. Sidhu, EPS-2018-454-S&E, https://repub.eur.nl/pub/106274

Keko. E, *Essays on Innovation Generation in Incumbent Firms,* Promotors: Prof. S. Stremersch & Dr N.M.A. Camacho, EPS-2017-419-MKT, https://repub.eur.nl/pub/100841

Kerkkamp, R.B.O.,*Optimisation Models for Supply Chain Coordination under Information Asymmetry*, Promotors: Prof. A.P.M. Wagelmans & Dr. W. van den Heuvel, EPS-2018-462-LIS, https://repub.eur.nl/pub/109770

Khattab, J.,*Make Minorities Great Again: a contribution to workplace equity by identifying and addressing constraints and privileges*, Promotors: Prof. D.L. van Knippenberg & Dr A. Nederveen Pieterse, EPS-2017-421-ORG, https://repub.eur.nl/pub/99311

Kim, T. Y.,*Data-driven Warehouse Management in Global Supply Chains,* Promotors: Prof. R. Dekker & Dr C. Heij, EPS-2018-449-LIS, https://repub.eur.nl/pub/109103

Klitsie, E.J.,*Strategic Renewal in Institutional Contexts: The paradox of embedded agency*, Promotors: Prof. H.W. Volberda & Dr. S. Ansari, EPS-2018-444-S&E, https://repub.eur.nl/pub/106275

Koolen, D.,*Market Risks and Strategies in Power Systems Integrating Renewable Energy*, Promotors: Prof. W. Ketter & Prof. R. Huisman, EPS-2019-467-LIS, https://repub.eur.nl/pub/115655

Kong, L., *Essays on Financial Coordination*, Promotors: Prof. M.J.C.M. Verbeek, Dr. D.G.J. Bongaerts & Dr. M.A. van Achter. EPS-2019-433-F&A, https://repub.eur.nl/pub/114516

Kyosev, G.S.,*Essays on Factor Investing*, Promotors: Prof. M.J.C.M. Verbeek & Dr J.J. Huij, EPS-2019-474-F&A, https://repub.eur.nl/pub/116463

Lamballais Tessensohn, T.,*Optimizing the Performance of Robotic Mobile Fulfillment Systems*, Promotors: Prof. M.B.M de Koster, Prof. R. Dekker & Dr D. Roy, EPS-2019-411-LIS, https://repub.eur.nl/pub/116477

Leung, W.L.,*How Technology Shapes Consumption: Implications for Identity and Judgement*, Promotors: Prof. S. Puntoni & Dr G Paolacci, EPS-2019-485-MKT, https://repub.eur.nl/pub/117432

Li, X., *Dynamic Decision Making under Supply Chain Competition*, Promotors: Prof. M.B.M de Koster, Prof. R. Dekker & Prof. R. Zuidwijk, EPS-2018-466-LIS, https://repub.eur.nl/pub/114028

Liu, N.,*Behavioral Biases in Interpersonal Contexts*, Promotors: Prof. A. Baillon & Prof. H. Bleichrodt, EPS-2017-408-MKT, https://repub.eur.nl/pub/95487

Maas, A.J.J.,*Organizations and their external context: Impressions across time and space*, Promotors: Prof. P.P.M.A.R Heugens & Prof. T.H. Reus, EPS-2019-478-S&E, https://repub.eur.nl/pub/116480

Maira, E.,*Consumers and Producers*, Promotors: Prof. S. Puntoni & Prof. C. Fuchs, EPS-2018-439-MKT, https://repub.eur.nl/pub/104387

Mirzaei, M.,*Advanced Storage and Retrieval Policies in Automated Warehouses*, Promotors: Prof. M.B.M. de Koster & Dr N. Zaerpour, EPS-2020-490-LIS, https://repub.eur.nl/pub/125975

Nair, K.P.,*Strengthening Corporate Leadership Research: The relevance of biological explanations*, Promotors: Prof. J. van Oosterhout & Prof. P.P.M.A.R Heugens, EPS-2019-480-S&E, https://repub.eur.nl/pub/120023

Nullmeier, F.M.E.,*Effective contracting of uncertain performance outcomes: Allocating responsibility for performance outcomes to align goals across supply chain actors*, Promotors: Prof. J.Y.F.Wynstra & Prof. E.M. van Raaij, EPS-2019-484-LIS, https://repub.eur.nl/pub/118723

Okbay, A.,*Essays on Genetics and the Social Sciences*, Promotors: Prof. A.R. Thurik, Prof. Ph.D. Koellinger & Prof. P.J.F. Groenen, EPS-2017-413-S&E, https://repub.eur.nl/pub/95489

Peng, X.,*Innovation, Member Sorting, and Evaluation of Agricultural Cooperatives*, Promotor: Prof. G.W.J. Hendriks, EPS-2017-409-ORG, https://repub.eur.nl/pub/94976

Petruchenya, A.,*Essays on Cooperatives: Emergence, Retained Earnings, and Market Shares*, Promotors: Prof. G.W.J. Hendriks & Dr Y. Zhang, EPS-2018-447-ORG, https://repub.eur.nl/pub/105243

Plessis, C. du, *Influencers: The Role of Social Influence in Marketing*, Promotors: Prof. S. Puntoni & Prof. S.T.L.R. Sweldens, EPS-2017-425-MKT, https://repub.eur.nl/pub/103265

Pocock, M.,*Status Inequalities in Business Exchange Relations in Luxury Markets*, Promotors: Prof. C.B.M. van Riel & Dr G.A.J.M. Berens, EPS-2017-346-ORG, https://repub.eur.nl/pub/98647

Pozharliev, R.,*Social Neuromarketing: The role of social context in measuring advertising effectiveness*, Promotors: Prof. W.J.M.I. Verbeke & Prof. J.W. van Strien, EPS-2017-402-MKT, https://repub.eur.nl/pub/95528

Reh, S.G.,*A Temporal Perspective on Social Comparisons in Organizations*, Promotors: Prof. S.R. Giessner, Prof. N. van Quaquebeke & Dr. C. Troster, EPS-2018-471-ORG, https://repub.eur.nl/pub/114522

Riessen, B. van, *Optimal Transportation Plans and Portfolios for Synchromodal Container Networks*, Promotors: Prof. R. Dekker & Prof. R.R. Negenborn, EPS-2018-448-LIS, https://repub.eur.nl/pub/105248

Schie, R. J. G. van, *Planning for Retirement: Save More or Retire Later?* Promotors: Prof. B. G. C. Dellaert & Prof. A.C.D. Donkers, EOS-2017-415-MKT, https://repub.eur.nl/pub/100846

Schouten, K.I.M., *Semantics-driven Aspect-based Sentiment Analysis*, Promotors: Prof. F.M.G. de Jong, Prof. R. Dekker & Dr. F. Frasincar, EPS-2018-453-LIS, https://repub.eur.nl/pub/112161

Sihag, V.,*The Effectiveness of Organizational Controls: A meta-analytic review and an investigation in NPD outsourcing*, Promotors: Prof. J.C.M. van den Ende & Dr S.A. Rijsdijk, EPS-2019-476-LIS, https://repub.eur.nl/pub/115931

Smolka, K.M.,*Essays on Entrepreneurial Cognition, Institution Building and Industry Emergence*, Promotors: P.P.M.A.R. Heugens, & Prof. J.P. Cornelissen, Eps-2019-483-S&E, https://repub.eur.nl/pub/118760

Straeter, L.M.,*Interpersonal Consumer Decision Making*, Promotors: Prof. S.M.J. van Osselaer & Dr I.E. de Hooge, EPS-2017-423-MKT, https://repub.eur.nl/pub/100819

Stuppy, A.,*Essays on Product Quality*, Promotors: Prof. S.M.J. van Osselaer & Dr N.L. Mead. EPS-2018-461-MKT, https://repub.eur.nl/pub/111375

Subaşi, B.,*Demographic Dissimilarity, Information Access and Individual Performance*, Promotors: Prof. D.L. van Knippenberg & Dr W.P. van Ginkel, EPS-2017-422-ORG, https://repub.eur.nl/pub/103495

Suurmond, R.,*In Pursuit of Supplier Knowledge: Leveraging capabilities and dividing responsibilities in product and service contexts*, Promotors: Prof. J.Y.F Wynstra & Prof. J. Dul. EPS-2018-475-LIS, https://repub.eur.nl/pub/115138

Toxopeus, H.S., *Financing sustainable innovation: From a principal-agent to a collective action perspective*, Promotors: Prof. H.R. Commandeur & Dr. K.E.H. Maas. EPS-2019-458-S&E, https://repub.eur.nl/pub/114018

Turturea, R.,*Overcoming Resource Constraints: The Role of Creative Resourcing and Equity Crowdfunding in Financing Entrepreneurial Ventures*, Promotors: Prof. P.P.M.A.R Heugens, Prof. J.J.P. Jansen & Dr. I. Verheuil, EPS-2019-472-S&E, https://repub.eur.nl/pub/112859

Valboni, R., *Building Organizational (Dis-)Abilities: The impact of learning on the performance of mergers and acquisitions*,Promotors: Prof. T.H. Reus & Dr A.H.L. Slangen, EPS-2020-407-S&E, https://repub.eur.nl/pub/125226

Vandic, D.,*Intelligent Information Systems for Web Product Search*, Promotors: Prof. U. Kaymak & Dr Frasincar, EPS-2017-405-LIS, https://repub.eur.nl/pub/95490

Verbeek, R.W.M.,*Essays on Empirical Asset Pricing*, Promotors: Prof. M.A. van Dijk & Dr M. Szymanowska, EPS-2017-441-F&A, https://repub.eur.nl/pub/102977

Visser, T.R. *Vehicle Routing and Time Slot Management in Online Retailing*, Promotors: Prof. A.P.M. Wagelmans & Dr R. Spliet, EPS-2019-482-LIS, https://repub.eur.nl/pub/120772

Vlaming, R. de.,*Linear Mixed Models in Statistical Genetics*, Prof. A.R. Thurik, Prof. P.J.F. Groenen & Prof. Ph.D. Koellinger, EPS-2017-416-S&E, https://repub.eur.nl/pub/100428

Vries, H. de, *Evidence-Based Optimization in Humanitarian Logistics*, Promotors: Prof. A.P.M. Wagelmans & Prof. J.J. van de Klundert, EPS-2017-435-LIS, https://repub.eur.nl/pub/102771

Wang, R.,*Corporate Environmentalism in China*, Promotors: Prof. P.P.M.A.R Heugens & Dr F. Wijen, EPS-2017-417-S&E, https://repub.eur.nl/pub/99987

Wasesa, M.,*Agent-based inter-organizational systems in advanced logistics operations*, Promotors: Prof. H.W.G.M van Heck, Prof. R.A. Zuidwijk & Dr A. W. Stam, EPS-2017-LIS-424, https://repub.eur.nl/pub/100527

Wessels, C.,*Flexible Working Practices: How Employees Can Reap the Benefits for Engagement and Performance*, Promotors: Prof. H.W.G.M. van Heck, Prof. P.J. van Baalen & Prof. M.C. Schippers, EPS-2017-418-LIS, https://repub.eur.nl/

Wiegmann, P.M.,*Setting the Stage for Innovation: Balancing Diverse Interests through Standardisation*, Promotors: Prof. H.J. de Vries & Prof. K. Blind, EPS-2019-473-LIS, https://repub.eur.nl/pub/114519

Wijaya, H.R.,*Praise the Lord!: Infusing Values and Emotions into Neo-Institutional Theory*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. J.P. Cornelissen, EPS-2019-450-S&E, https://repub.eur.nl/pub/115973

Williams, A.N.,*Make Our Planet Great Again: A Systems Perspective of Corporate Sustainability*, Promotors: Prof. G.M. Whiteman & Dr. S. Kennedy, EPS-2018-456-ORG, https://repub.eur.nl/pub/111032

Witte, C.T.,*Bloody Business: Multinational investment in an increasingly conflict-afflicted world*, Promotors: Prof. H.P.G. Pennings, Prof. H.R. Commandeur & Dr M.J. Burger, EPS-2018-443-S&E, https://repub.eur.nl/pub/104027

Ye, Q.C.,*Multi-objective Optimization Methods for Allocation and Prediction*, Promotors: Prof. R. Dekker & Dr Y. Zhang, EPS-2019-460-LIS, https://repub.eur.nl/pub/116462

Yuan, Y.,*The Emergence of Team Creativity: a social network perspective,* Promotors: Prof. D. L. van Knippenberg & Dr D. A. Stam, EPS-2017-434-ORG, https://repub.eur.nl/pub/100847

Zhang, Q.,*Financing and Regulatory Frictions in Mergers and Acquisitions,* Promotors: Prof. P.G.J. Roosenboom & Prof. A. de Jong, EPS-2018-428-F&A, https://repub.eur.nl/pub/103871

Most of the warehouse operations are still performed manually despite the increasing development and adoption of automated warehouse solutions. Planning human workers in a warehouse is a complex task because managers have to consider issues such as start and end times of shifts, breaks, and incentive payment schemes. When warehouses have temporal restrictions on the processing time of inbound trucks or when they have deadlines for outbound orders, these constraints impact the schedules and cost of employing human workers. In this environment, warehouse management has to consider the temporal restrictions from external entities as well as the limited resources available at the warehouse. In this thesis, we study the impact of external temporal constraints in three operational planning problems at manual warehouses. We develop mathematical models for the problems, propose solution approaches for them and conduct computational experiments to derive insights.

The first study in the dissertation explores integrated scheduling and assignment of trucks to dock doors in unit-load cross-dock facilities with mixed mode dock doors. The processing time of both inbound and outbound trucks at the cross-dock are constrained by time windows. In the second study, we investigate order picker scheduling problem in distribution centers where order picking operations are constrained temporally by predefined time windows for delivery of orders to the staging area of the warehouse. In the final study, we consider the impact of delivery time windows at customers on the capacity requirements of three warehouse processes – order picking, staging and loading.

ERIM

ERIM PhD Series
Research in Management

Erasmus University Rotterdam (EUR)
Erasmus Research Institute of Management
Mandeville (T) Building
Burgemeester Oudlaan 50
3062 PA Rotterdam, The Netherlands

P.O. Box 1738
3000 DR Rotterdam, The Netherlands
T +31 10 408 1182
E info@erim.eur.nl
W www.erim.eur.nl