CROSS-DOCK SCHEDULING WITH KNOWN SHIPMENT UNLOADING ORDER

Thuy Vu

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

January 2019

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Thuy Vu**

Entitled: **Cross-dock Scheduling with Known Shipment Unloading Order**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. Weiyi Shang ______________________ Chair

Dr. Emad Shihab ______________________ Examiner

Dr. Leila Kosseim ______________________ Examiner

Dr. Brigitte Jaumard ______________________ Supervisor

Approved ______________________________
Dr. Volker Haarslev, Graduate Program Director

__________ 20 _____ ______________________________
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Cross-dock Scheduling with Known Shipment Unloading Order

Thuy Vu

Cross-docking is a logistics strategy which is widely used these days in different industries. Cross-docking takes place in a distribution docking centre and consists of trucks and dock doors on inbound and outbound sides. Products from suppliers get unloaded at inbound doors from incoming trucks, consolidated, transferred and loaded into outgoing trucks at outbound doors, with little or no storing them in between.

We study two scenarios of cross-docking scheduling problem: scheduling inbound side with fixed outbound side scheduling and scheduling both inbound and outbound sides. In the former scenario, we introduce five mixed integer programming models with enhanced pre-processing and extensions to minimize the total number of tardy products. In the later scenario, we proposed new linear mixed integer programming models where transportation time between dock doors are considered. The objective in the second case is to minimize the maximum lateness of outgoing trucks. In both scenarios, we integrate the unloading order of shipments in incoming trucks into our models. Computational results show that taking advantage of that information helps improving the truck scheduling and assessing much more accurately the number of tardy products and lateness.

Acknowledgments

I would like to express my gratitude to my supervisor, Dr. Brigitte Jaumard for her continuous advices, guidance, support during the time of my study. I am also grateful to Dr. Wael Nassief, who I had opportunity to work with and help me lot with my research. I am very lucky and happy to have both of you from the very beginning of my Masters journey.

To my dear wife, Hong Vo, thank you for your love, support, encouragements and sacrifices you give me so I can pursue my dream.

Finally, I am grateful to Clear Destination company for their financial support and funding of my Masters.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Cross-docking is a logistics strategy which is used to speed up the movement of products in supply-chain networks. The basic idea of cross-docking is unloading, consolidating and loading goods from incoming trucks to outgoing trucks with minimal or no storage between. A typical cross-docking facility consists of *strip* doors or inbound doors for unloading goods from incoming trucks, *stack* doors or outbound doors for loading products into outgoing trucks and a sorting space between inbound and outbound sides to gather shipments, which are normally carried out by employees and material handling equipment. The cross-docking strategy presents some advantages in comparison with the traditional warehouse such as: cross-docking reduces the cost of storing, inventory handling and it also help products to reach customer faster while lower transportation costs. This strategy was first used by the US trucking industry in 1930s and followed by the US military in 1950s and Walmart in 1980s (Stalk et al. 1992).

Although cross-docking has a lot of benefits and is used widely by many companies, it should be only used when some conditions are satisfied in order to have maximum of efficiency. Apte et al. 2000 identified the best situation to apply cross-docking strategy as when products have stable and constant demand rate and low unit stock-out cost.

The characteristics of cross-docking can be divided into three main groups, namely, physical characteristics, operational characteristics and characteristics about the flow of goods. The physical characteristics describe the structure of cross-docking facilities such as the shape, number of dock doors or the means of transportation inside the facilities.

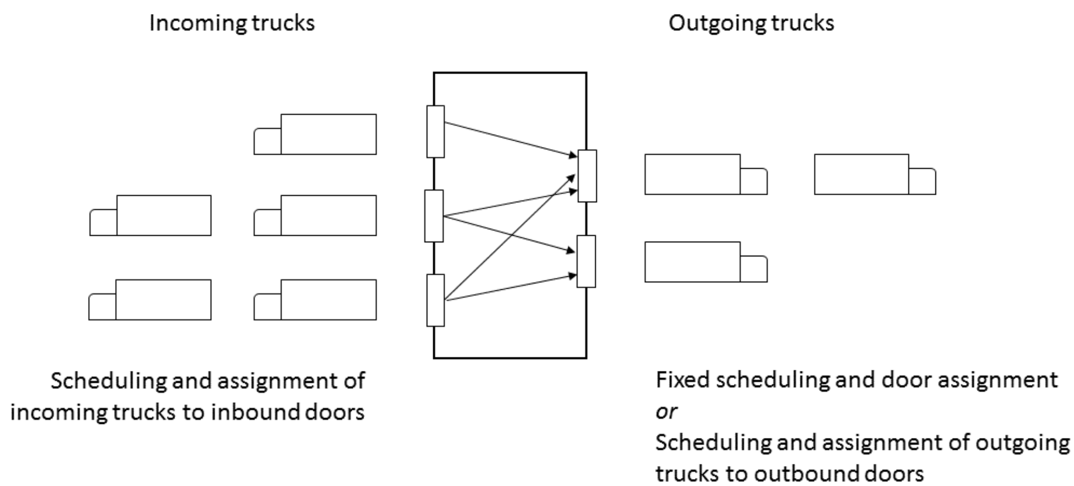


Figure 1: A cross-dock facility

The operational characteristics are influenced by the service mode which decides on dock doors where the inbound and outbound trucks can be assigned to and pre-emption which decides when the unloading and loading processes can be interrupted. The flow characteristics of products that have to be processed by cross-docking such as: arrival times in which all incoming trucks arrive at the same time or in different times of the day, restrictions of departure times of outgoing trucks and the interchangeability of products and temporary storage.

There are many optimization problems which need to be dealt with in cross-docking, they include strategic, tactical and operational decision problems. According to Van Belle et al. 2012, the strategic problems include location and layout design of cross-docking facilities, the tactical decision problems determine the flow of goods through cross-docking networks and the operational decision problems deal with vehicle routing, dock door assignment, truck scheduling and temporary storage.

1.2 Research Projects

The thesis contains two projects to study the problems of trucks scheduling in a cross-dock. The first project is to schedule inbound trucks with fixed outbound scheduling. The second project is to schedule both inbound and bound sides of cross-dock.

The first project deals with the inbound truck scheduling problem to minimize total number of tardy products. In this problem, the scheduling and the departure times of outgoing trucks are assumed to be pre-defined. The products have to be unloaded and then transferred from incoming trucks to their destination, the outgoing trucks, through the cross-docking by workers. At the beginning of the working day, all incoming trucks are available at inbound side and wait to be assigned to dock doors in a sequence. This project extends from the work of Boysen et al. 2013 by introducing a time-indexed model with a pre-processing to speed up the performance of the model. The project also considers two extensions: multiple transfer trip shipment and known goods unload order. The computational results show that our model outperforms that of Boysen et al. 2013 and the two extensions give us better scheduling to lower significantly the number of tardy products.

In the second project, we model the inbound and outbound scheduling problem. We first introduce a time-indexed model with linearized constraints to minimize the lateness of outgoing trucks. After that, we study two different approaches to apply the shipment unloading order. The experimental results show that by applying the order of shipments, we are able to reduce the maximum lateness and have better estimation of the starting times to process outgoing trucks.

1.3 Literature Review

In this section, we will discuss about cross-docking truck scheduling problems which were studied in the literature. We divide this section into two sub-sections that survey the most recent work on Inbound Scheduling problems (Section 1.3.1) and Inbound and Outbound Scheduling problems (Section 1.3.2).

1.3.1 Inbound Scheduling

D. L. McWilliams et al. 2005 study inbound truck scheduling in cross-docking centre for parcel deliver industry in order to minimize the completion time of the last truck. The authors develop a simulation-based scheduling algorithm to solve this problem with the assumption that the unloading times of inbound trucks are all the same. In D. L. McWilliams et al. 2008, they consider the same problem but with unequal unloading time of incoming trucks. D. L. McWilliams 2009 and D. McWilliams 2010 deal with the problem of scheduling inbound trucks to minimize the time-span of the transfer operation inside cross-docking facility for parcel delivery. They propose a combination of time-based and resource-based decomposition. In the former study, the authors use a genetic-based algorithm while in the later they developed an iterative improvement method which consists of Local Search and Simulated Annealing algorithm.

Rosales et al. 2009 study scheduling of inbound trailers to minimize operational cost. The authors provide a balanced workload to all workers and applied successfully for a large cross-docking centre in Georgetown, Kentucky. They introduce a mixed-integer programming model for this problem and used CPLEX to solve the resulting model.

Boysen et al. 2013 consider a truck scheduling problem in cross-docking terminals with fixed departure times of outbound trucks in order to minimize the number of tardy products. The authors propose a mixed-integer programming model and heuristics, namely, decomposition procedures and simulated annealing to solve the problem. A case study is investigated, in which they apply their models and algorithms to a postal company. Liao et al. 2013 study inbound scheduling problem to minimize the total weighted tardiness with the same assumption of fixed outbound truck departure scheduling. They use six types of meta-heuristic algorithms to tackle the problem. Tootkaleh et al. 2016 extends Boysen et al. 2013 to multi-period, multi-product, and substitution condition, i.e. delayed products can be loaded to other outbound trucks, problem. The authors modeled an inbound scheduling problem under substitution condition to minimize the total inventory holding costs.

Boysen et al. 2017 deal with the inbound trucks scheduling problem in postal service industry. In this problem, the cross-docking centre contains a number of inbound segments and the incoming trucks have deadlines to finish their unloading process. The objective is to minimize a total penalty value, i.e. the cost to process incoming truck at a dock door. The authors formed a linear mixed-integer program model and heuristics to solve large instances.

1.3.2 Inbound and Outbound Scheduling

Lim et al. 2006 study a truck scheduling problem which has constraints that force each truck to be unloaded and loaded in fixed time windows and take into account the capacity of the cross-dock centre. The objective is to minimize the shipping distance between dock doors. Miao et al. 2009a extends the work of Lim et al. 2006 by considering the transportation time between inbound and outbound sides. The objective is to minimize the sum of the operational and penalty costs. They introduce two methods namely, Tabu search and Genetic to solve large instances.

Boysen 2010 consider the problem of scheduling truck in the food industry. For this special type of problems, the intermediate storage and product substitution are not allowed. In order to solve real world size instances, the author uses a dynamic programming approach and Simulated Annealing heuristics to minimize different operational objectives. Boysen et al. 2010a take into account different types of products in their scheduling problem to minimize the process makespan, i.e the completion time of last truck at cross-docking facility. They assume that each outbound truck can be assigned when all of its products were unloaded successfully.

F. Chen et al. 2009a consider the inbound and outbound scheduling problem as a two-stage hybrid flow shop scheduling problem. They propose a mixed integer programming model to minimize the makespan and heuristics based on Johnson's rule. Cota et al. 2016 extend this work by introducing a time-based model which has better performance than the network-based model of F. Chen et al. 2009a. Bellanger et al. 2013 study three-stage hybrid flow shop model for cross-docking to minimize the makespan. These three stage models include receiving at inbound side, sorting inside cross-dock and shipping in the outbound side. The authors develop two heuristics, namely list heuristic and dynamic heuristics and a brand-and-bound algorithm to solve the problem.

Many other works such as: Arabani et al. 2011; Ghobadian et al. 2012; Ruiz et al. 2007; Vahdani et al. 2010; Bodnar et al. 2017; Ye et al. 2016; etc. use meta-heuristic approaches to tackle the inbound and outbound scheduling problem.

1.4 Thesis Contributions

The contributions of this thesis are contained in two papers. The first paper has been submitted to Omega journal and when through a first revision in July, 2018. The second paper will be shortly submitted. Below is the brief description of each:

Chapter 2: The contribution is a time-indexed model for the network-based model in Boysen et al. 2013 and introduce an enhanced pre-processing which allows the model to solve larger instances than in the current literature. The objective is to minimize the total number of tardy products. Next, we study two extensions of multiple transfer strips and unloading order of products. The computational results show that the impact of these two modifications are significant and make the models much more realistic.

Chapter 3: This paper models a more complex problem of truck scheduling at cross-dock centre. Instead of assuming that the scheduling of outbound side is fixed and pre-scheduled, we schedule both inbound and outbound of a cross-dock to minimize the maximum lateness of outgoing trucks. We propose three mathematical mixed integer programming models. We introduce new linearized constraints to overcome the current non-linear models of the literature when the transshipment times between dock doors are considered. Next, we study the impact of known shipment unloading order on the qualities of solutions. The experimental results show that the maximum lateness of outgoing trucks decrease significantly when the unloading order of shipments are taken into account.

Chapter 2

Cross-dock Scheduling with Fixed Outbound Departures, Multiple Transfer Trips and Shipment Unloading Order

*Submitted for publication to Omega journal in March, 2018, revised in July, 2018.

2.1 Introduction

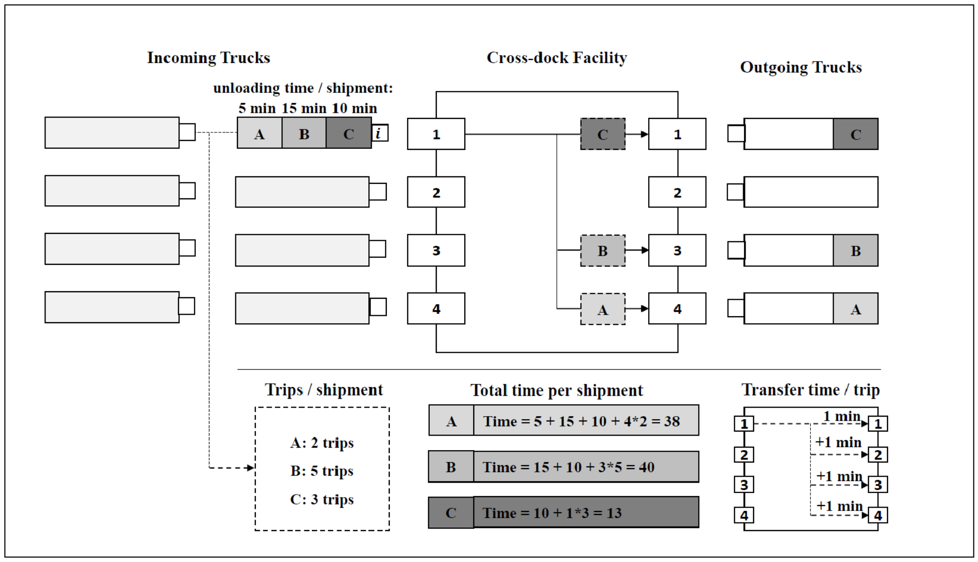
2.1.1 Background and Motivation

Cross-docking is a logistic strategy that speeds up the movement of products across a supply chain network, with little or no storage in between. When demand is constant and stable with low unit stock-out costs, cross-docking is preferred over warehousing and traditional distributions, resulting in a significant reduction in storage and order picking Agustina et al. 2010. Such a strategy has been successfully implemented in the postal, manufacturing and retailing industries. The interested reader may refer to Forger 1995; Kinnear 1997; Witt 1998; Napolitano 2011 for successful examples of cross-dock implementations.

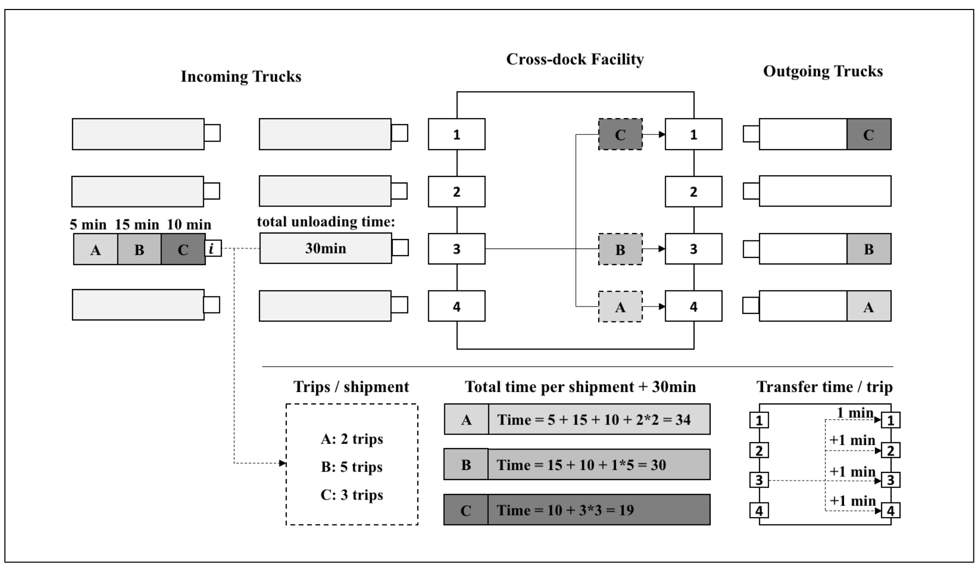
There are strategic, tactical and operational decisions that arise in the context of cross-docking. The location and layout design of cross-docks are strategic decisions e.g., Bartholdi et al. 2004; Ross et al. 2008, while routing flows between multiple cross-docks is considered tactical e.g., Miao et al. 2012. Operational decisions involve routing, assigning, and scheduling trucks over cross-dock facilities, see, e.g., Boysen et al. 2010a; Guignard et al. 2012; Nassief et al. 2016; Nassief et al. 2018a. As cross-docking has become more popular, companies and researchers have been working recently on the integration of several operational decisions, see, e.g., Rosales et al. 2009, Enderer et al. 2017 and Nassief et al. 2018b.

This paper focuses on one important and complex operational problem: cross-dock scheduling. On a daily basis, incoming and outgoing trucks arrive at a cross-dock facility. The incoming trucks get scheduled (i.e., assigned and sequenced concurrently) over inbound doors known as strip doors. Then, their shipments get unloaded and consolidated according to their destinations. Similarly, the outgoing trucks get scheduled over outbound doors known as stack doors, and the consolidated shipments get loaded accordingly. The scheduling decisions of either or both incoming and outgoing trucks over doors are essential for cross-dock managers to speed up the consolidation process inside the facility and meet deadlines. Figure 2 shows an illustration of how scheduling incoming truck i over a strip door can impact the overall unloading and transshipment time of its shipments. In Figure 2(a), the total time of unloading and transferring all shipments inside incoming truck i is 91 minutes whereas 113 minutes in Figure 2(b). The difference is mainly explained by observing that if incoming truck i in Figure 2(b) is scheduled first, it takes 83 minutes to unload and transfer all its shipments, resulting in less time than in Figure 2(a). However, since it is scheduled second, it needs to wait as it takes 30 minutes for the first truck to be fully unloaded.

Moreover, some cross-dock companies pre-assign destinations to outbound doors over a midterm planning horizon, and thus, provide a predetermined fixed schedule of the outgoing trucks whereas they schedule incoming trucks over inbound doors on a daily basis. This practice has been seen in the postal industry where outgoing trucks have fixed departure times, see, e.g., Boysen et al. 2013; Liao et al. 2013; Boysen et al. 2010b.



(a) Scheduling #1



(b) Scheduling #2

Figure 2: Impact of scheduling on the unloading and transshipment time of shipments for a given incoming truck i

2.1.2 Goal and Scope

In this paper, we assume that the outbound operations are indeed predetermined, and hence, all outbound doors are preassigned to their destinations and each outgoing truck is prescheduled over an outbound door. Each outgoing truck is also given a fixed departure time. Therefore, the problem reduces to scheduling incoming trucks over inbound doors with the goal of minimizing the total number of tardy products as seen in Boysen et al. 2013. From now on, this problem is referred to as *Truck Scheduling with Fixed outbound Departures* (TSFD). Since there are always more incoming trucks than inbound doors, there is a combinatorial aspect in the selection of the strip door and the sequence in which an incoming truck is placed, which in turn, has an impact on the total time a shipment, or its products, takes to reach its designated outgoing truck. The TSFD is strongly NP-hard as shown in Boysen et al. 2013.

In order to study the TSFD, a distinction between shipments and products is first clarified. A shipment is a group of products shipped together as part of the same lot, and associated with one incoming and one outgoing truck. There could be several shipments per incoming truck, and also, several (consolidated) shipments per outgoing truck. For example, a crossdock at a WalMart might receive 20 products of Tide detergent and 15 products of Purex detergent without labels for individual stores from a given incoming truck. Workers at the crossdock allocate 3 Tide and 2 Purex products to Store 23, i.e., outbound door 7 (one shipment), 5 Tide and 1 Purex products to Store 14, i.e., outbound door 2 (another shipment), and so on. We next clarify the following important assumptions made in the basic TSFD described in Boysen et al. 2013:

- a shipment is considered tardy if one or more of its products arrive after its outgoing truck has departed. In other words, the whole shipment is considered tardy if part of it is tardy. The model is indeed written as if there is an underlying simplifying assumption of a single pallet jack or forklift transfer trip per shipment regardless of how many products it contains.

- a shipment is considered unloaded from its incoming truck once all shipments inside the same incoming truck have been unloaded too. That is the unloading time of a shipment is assumed equal to the unloading (or processing) time of its incoming truck. Indeed, if the unloading order of shipments inside their incoming trucks is unknown, the worst case is considered for the unloading times of all shipments in a given incoming truck, resulting in an over estimation of the tardy costs.

We first study the basic TSFD with the same assumptions as Boysen et al. 2013, and then propose the following two extensions:

- **multiple transfer trips**: each shipment requires one or several forklift trips, depending on the volume of its products, in order to be transferred across the cross-dock, as well as the capacity of the material handling equipment (e.g., carts for small items, forklifts for pallet loads or large and heavy cartons). We propose to minimize the number of tardy products rather than the number of tardy shipments in order to estimate the item tardiness more accurately.

- **known unloading order**: the order of shipments inside each incoming truck is known. Therefore, we assume that a shipment is considered unloaded from its incoming truck once all its precedent/previous shipments inside the same incoming truck have been unloaded without the need to wait for all its successive/next shipments to be unloaded too. This is not to confuse with the order or sequence of the trucks themselves, which is part of the scheduling decisions. Here, we explicitly refer to the order of unloading shipments/products inside the truck as being known/unknown.

First, multiple transfer trips are inevitable inside any cross-dock facility as there is a limited capacity on the material handling equipment used (e.g., forklifts, carts). Therefore, embedding this practical assumption in our models provide a more accurate information in calculating the tardiness and scheduling decisions as will be demonstrated later. Second, our interaction with a local cross-dock company confirms that knowing the loading order of shipments inside their incoming trucks can be assumed to be known before truck scheduling takes place. Indeed, we can use the reverse loading order, which is usually recorded through product scanning. We will demonstrate in the numerical results in Section 2.7 the economical benefits of taking advantage of the unloading shipment order.

2.1.3 Contributions & Organization

We first introduce a time-indexed formulation and enhanced preprocessing for the Basic TSFD and compares it computationally with the model and solution of Boysen et al. 2013. We then modify our formulation to take into account multiple transfer trips and unloading order of shipments. The impact of these extensions are demonstrated in the computational experiments on two data sets based on the postal and retail industries, respectively.

The remainder of this paper is organized as follows. Section 2.2 provides a literature review for the TSFD problem. In Section 2.3, the Basic TSFD, as defined by Boysen et al. 2013, is recalled. Existing and newly introduced mathematical programming formulations are detailed in Section 2.4 for the Basic TSFD. Extensions are introduced and formulated in Sections 2.5 and 2.6. Section 2.7 provides detailed computational experiments to test our formulations against the existing one as well as providing economical analysis on our extensions. Finally, conclusions with future research directions are provided in Section 2.8.

2.2 Literature Review

The literature of scheduling is abundant, and so, we choose to review and highlight the most relevant work to the TSFD problem in the context of cross-docking, cross-dock scheduling, and its relationship, if any, with parallel machine scheduling.

2.2.1 Cross-docking

Several cross-docking review papers have appeared in the last few years with different classification schemes and scopes. To the best of our knowledge, the most comprehensive review is the one provided by Van Belle et al. 2012 where they cover strategic, tactical and operational decisions arising in cross-docking. Based on this survey, the TSFD problem belongs to the class of *scheduling inbound trucks*. Boysen et al. 2010a provide a classification and review for cross-dock scheduling problems and introduce the TSFD as a new research area that is worth studying. According to their classification scheme, the TSFD problem is denoted by $[E|\delta_{io}, \mathsf{fix}| \sum w_p U_p]$. It assumes an exclusive cross-dock mood where E represents a cross-dock scheduling problem with one side for inbound doors while another side for outbound doors. The travel or transfer time, δ_{io}, is assumed between doors, and fix means fixed outbound departures. Finally, the goal is to minimize the total number of tardy products stated as $\sum\limits_{p \in P} w_p U_p$, where w_p represents the number of products and U_p is a binary variable indicating if they are tardy or not. Ladier et al. 2016 provide a review of assignment and scheduling problems in cross-docking and highlight the major differences between the literature and cross-dock practices in France. For other review papers, the interested reader is referred to Agustina et al. 2010, Stephan et al. 2011 and Buijs et al. 2014.

2.2.2 Cross-dock Scheduling

Rosales et al. 2009 schedule incoming trucks over inbound doors while balancing the workload between employees at a cross-dock company in Georgetown. The authors introduce a mixed-integer programming (MIP) formulation and solve all real life instances (near-)optimally within 1.5 hours of time limit. D. McWilliams 2010 study an inbound scheduling problem to minimize the time-span inside the cross-dock. The authors introduce local search and simulated annealing algorithms, and show that their meta-heuristics outperform an existing genetic algorithm for the same problem. Boysen et al. 2013 present an MIP formulation for the TSFD problem with the goal of minimizing the total number of tardy products. The authors introduce two heuristics to solve the problem with 100 incoming trucks and 30 inbound doors in fractions of seconds. Liao et al. 2013 also study the TSFD problem, but with the goal of minimizing the total weighted tardiness. The authors introduce a new MIP formulation and six different meta-heuristics. However, they only attempt to solve a small size instances with 12 incoming trucks and 5 inbound doors. In the TSFD problem, it is generally assumed that tardy products remain in the cross-dock until the next shift or the next working day, and that the penalty is imposed accordingly. However, Tootkaleh et al. 2016 study a TSFD where urgent late products can be substituted by other products. That is the cross-dock manager can decide to substitute late products with newly arrived ones in order to reduce the delay cost and increase customer's satisfaction. The authors introduce an MIP formulation and a heuristic to solve the problem. Nassief et al. 2018b study a practical inbound scheduling problem where incoming containers must be scheduled over inbound doors, unloaded and emptied, and then returned to the port before their due dates. They introduce two integer programming (IP) formulations for static and dynamic environments and integrate door selection with scheduling decisions to minimize the total tardiness and hiring costs. Finally, Molavi et al. 2018 study a cross-dock scheduling problem with the assumption that the order or position of shipments inside their incoming trucks is known. They introduce a mathematical formulation along with a hybrid meta-heuristic to solve the problem. However, unlike here they consider both inbound and outbound scheduling with arrival times and backup trucks to handle tardy shipments. The largest size of instances they solve approximately (or heuristically) are 30 incoming and 30 outgoing trucks with 12 inbound and 12 outbound doors. For other scheduling problems where inbound operations are rather predetermined, we refer to

Diglio et al. 2017 and Schwerdfeger et al. 2017, and also, to F. Chen et al. 2009a; Cota et al. 2016; Bellanger et al. 2013 where the integration of both inbound and outbound scheduling is studied.

2.2.3 Parallel Machine Scheduling

Finally, it is natural to see the resemblance between the TSFD and parallel machine scheduling. Incoming trucks can be interpreted as jobs while inbound doors can be seen as parallel machines/processors. However, as mentioned in Boysen et al. 2013, the TSFD is a generalization of parallel machine scheduling since each incoming truck may contain several shipments with different destination points, and hence, different due times. In the TSFD any job (or incoming truck) may have multiple due times because of its shipments. Therefore, existing algorithms for parallel machine scheduling provided by Z. Chen et al. 1999; Van Den Akker et al. 1999, 2012; M'Hallah et al. 2005, 2015 cannot be applied.

2.3 TSFD Problem Statement

Given a cross-dock facility, we denote by G^{IN} and G^{OUT} the set of inbound and outbound dock doors, respectively. Let TR^{IN} and TR^{OUT} be the set of incoming and outgoing trucks, respectively. Each shipment is indexed by an incoming truck $i \in \text{TR}^{\text{IN}}$ and an outgoing truck $o \in \text{TR}^{\text{OUT}}$ indicating its pair $(i - o)$ of origin and destination points. We thus denote by w_{io} the number of products of a given shipment coming from incoming truck i and destined to outgoing truck o. Each shipment $i - o$ has an unloading time denoted by u_{io}. We first consider as in all previous papers, that the total processing time to unload an incoming truck depends on the total unloading time of all of its shipments, and so, for each incoming truck $i \in \text{TR}^{\text{IN}}$, the total processing time can be stated as $p_i = \sum_{o \in \text{TR}^{\text{OUT}}} u_{io}$.

On the outbound side of the cross-dock, each outgoing truck $o \in \text{TR}^{\text{OUT}}$ has a fixed departure time denoted by d_o. We also assume that each outgoing truck has a predefined schedule, before the inbound schedule is set. There is a transshipment time inside the cross-dock facility between each inbound and outbound door. We denote by δ_{go} the transshipment time from inbound door $g \in G^{\text{IN}}$ to the outbound door where outgoing truck o is preassigned. Normally, the parameter δ is associated with both inbound and outbound doors. However, in the TSFD, the outbound door's index can be substituted by its outgoing trucks since they are preassigned to doors. We also assume that the transshipment time is per single trip from an inbound to an outbound door.

In the TSFD, each incoming truck must be scheduled over an inbound door such that each of its products is unloaded and transferred to its designated outbound door ideally before the departure time of its outgoing truck. If a product arrives at the outbound door after the departure time of its outgoing truck, it is then considered late. The objective of the TSFD is to minimize the total number of tardy products. Table 1 summarizes the mathematical notations for the TSFD.

Table 1: Mathematical notations for the TSFD

Notation	Definition
G^{IN}	set of inbound (or strip) doors.
G^{OUT}	set of outbound (or stack) doors.
TR^{IN}	set of incoming trucks.
TR^{OUT}	set of outgoing trucks.
w_{io}	number of products for a given shipment $i-o$ going from incoming truck i to outgoing truck o.
u_{io}	unloading time of a shipment $i-o$ coming from incoming truck i and destined to outgoing truck o.
$\bar{u}_{io}$	cumulative unloading time of a shipment $i-o$ and all its precedent shipments in the same incoming truck.
p_i	processing time for incoming truck i (i.e., the total unloading times of all its shipments).
d_o	departure time of outgoing truck o.
δ_{go}	transshipment time from inbound door g to outgoing truck o.

Several assumptions are made for the Basic TSFD. We next list all these assumptions based on the work of Boysen et al. 2013:

- All incoming and outgoing trucks are available at the beginning of the scheduling period.
- Departure times for outgoing trucks are set and given.
- All shipments are known in terms of their types, quantities, weights, volumes of their products, as well as the origin and destination points (or their incoming and outgoing trucks).
- The unloading order of shipments inside their incoming trucks is unknown.
- Outgoing trucks have a predetermined schedule over the outbound doors.
- Incoming trucks are unloaded without interruption (i.e., non-preemption).

- Outgoing trucks will not wait for any late product. Each late product is subcontracted accordingly.

- Without loss of generality, all parameters are assumed to be integers in Boysen et al. 2013. However, with our time-indexed modeling, we can only use integer parameters due to the discretization of the time horizon.

- Each unloaded shipment is available for immediate inner-facility transportation.

- Loading time is constant and hence is omitted. This time can be considered part of the transfer time between doors or by decreasing the departure time of the outgoing truck.

- Congestion inside the cross-dock facility is not taken into account. If it is inevitable, then it can be approximated by increasing the transshipment time accordingly inside the facility.

- The number of tardy products is assumed the same as the number of tardy shipments. Indeed, Boysen et al. 2013 do not distinguish between the two and assume that if a product is tardy, then its shipment, with all its products, is also tardy.

2.4 Basic TSFD

2.4.1 Existing Formulation: Boysen et al. (2013)

We next present the MIP formulation of Boysen et al. 2013 for the TSFD problem as stated in Section 2.3. In their formulation the authors introduce the following decision variables:

$x^g_{0,i} = 1$	if incoming truck i is processed first at inbound door g, 0 otherwise.
$x^g_{i,\text{LAST}} = 1$	if incoming truck i is processed last at inbound g, 0 otherwise.
$x^g_{ii'} = 1$	if incoming truck i' is processed directly after truck i at door g. 0, otherwise.
C_i	completion time of incoming truck i to unload all its shipments.
$U_{io} = 1$	if shipment $i - o$ from truck i is tardy to reach truck o, 0 otherwise.

Moreover, M is defined as a big integer number, and according to Boysen et al. 2013 can be calculated as: $M = \sum_{i \in \text{TR}^{\text{IN}}} p_i + \max\{\delta_{go} : g \in G^{\text{IN}}, o \in \text{TR}^{\text{OUT}}\}$. Their MIP formulation can be stated as:

$$\text{[TSFD]} \quad \min \quad \sum_{i \in \text{TR}^{\text{IN}}} \sum_{o \in \text{TR}^{\text{OUT}}} w_{io} U_{io} \tag{1}$$

subject to:

$$\sum_{g \in G^{\text{IN}}} \sum_{i' \in \text{TR}^{\text{IN}} \cup \{0\} i \neq i'} x^g_{i'i} = 1 \qquad i \in \text{TR}^{\text{IN}} \tag{2}$$

$$\sum_{i \in \text{TR}^{\text{IN}}} x^g_{0,i} \leq 1 \qquad g \in G^{\text{IN}} \tag{3}$$

$$\sum_{\substack{i' \in \text{TR}^{\text{IN}} \cup \{0\} \\ i \neq i'}} x^g_{i'i} = \sum_{i' \in \text{TR}^{\text{IN}} \cup \{\text{LAST}\} i \neq i'} x^g_{ii'} \qquad i \in \text{TR}^{\text{IN}}, g \in G^{\text{IN}} \tag{4}$$

$$C_i \geq C_{i'} + p_i - M \cdot (1 - x^g_{i'i}) \qquad i \in \text{TR}^{\text{IN}}, i' \in \text{TR}^{\text{IN}} \cup \{0\}; g \in G^{\text{IN}} \tag{5}$$

$$U_{io}.M \geq C_i + \sum_{g \in G^{\text{IN}}} \sum_{\substack{i' \in \text{TR}^{\text{IN}} \cup \{0\} \\ i \neq i'}} \delta_{go}.x^g_{i'i} - d_o \qquad i \in \text{TR}^{\text{IN}}, o \in \text{TR}^{\text{OUT}} \tag{6}$$

$$C_0 = 0 \tag{7}$$

$$x^g_{ii'} \in \{0, 1\} \qquad i, j \in I \cup \{0, \text{LAST}\}, g \in G^{\text{IN}} \tag{8}$$

$$C_i \geq 0 \qquad i \in \text{TR}^{\text{IN}} \tag{9}$$

$$U_{io} \in \{0, 1\} \qquad i \in \text{TR}^{\text{IN}}, o \in \text{TR}^{\text{OUT}}. \tag{10}$$

The objective function (1) minimizes the total number of tardy products, or equivalently the number of tardy shipments according to the assumptions of Boysen et al. 2013. Constraints (2) ensure that each incoming truck is scheduled for unloading while constraints (3) make sure that for inbound door, there is at most one sequence of incoming trucks. Constraints (4) ensure that the sequence of incoming trucks at a given inbound door is feasible similar to classical flow conservation constraints. Constraints (5) define completion time C_i for each incoming truck i. Constraints (6) ensure that U_{io} equals one if the shipment delivered by incoming truck i cannot reach its designated outgoing truck o before its departure time, i.e., the sum of C_i and of the transshipment time δ_{go} exceeds the departure time d_o. Constraints (7)-(10) define the nonengativity and binary conditions on the decision variables.

2.4.2 A Time-indexed Formulation

We now introduce a time-indexed formulation for the basic TSFD. This formulation relies on the discretization of the time horizon into blocks of time. We hence define

$t \in T$ the discretized time periods over which the incoming trucks are scheduled. We also introduce new decision variables that are time indexed: $x_{ig}^t = 1$ if incoming truck i starts processing at inbound door g at time t, 0 otherwise. Then, the time-indexed formulation for the TSFD can be stated as:

$$[\text{TSFD}^+] \quad \min \sum_{i \in \text{TR}^{\text{IN}}} \sum_{o \in \text{TR}^{\text{OUT}}} w_{io} U_{io} \tag{1}$$

subject to:

$$\sum_{t \in T} \sum_{g \in G^{\text{IN}}} x_{ig}^t = 1 \qquad i \in \text{TR}^{\text{IN}} \tag{11}$$

$$\sum_{i \in \text{TR}^{\text{IN}}} \sum_{t'=\max\{0, t-p_i\}}^{t-1} x_{ig}^{t'} \leq 1 \qquad g \in G^{\text{IN}}, t \in T \tag{12}$$

$$C_i \geq \sum_{t \in T} \sum_{g \in G^{\text{IN}}} (t + p_i)\, x_{ig}^t \qquad i \in \text{TR}^{\text{IN}} \tag{13}$$

$$U_{io}.M \geq C_i + \sum_{t \in T} \sum_{g \in G^{\text{IN}}} \delta_{go} x_{ig}^t - d_o \qquad i \in \text{TR}^{\text{IN}}, o \in \text{TR}^{\text{OUT}} : w_{io} > 0 \tag{14}$$

$$x_{ig}^t \in \{0, 1\} \qquad i \in \text{TR}^{\text{IN}}, g \in G^{\text{IN}}, t \in T \tag{15}$$

$$C_i \geq 0 \qquad i \in \text{TR}^{\text{IN}} \tag{9}$$

$$U_{io} \in \{0, 1\} \qquad i \in \text{TR}^{\text{IN}}, o \in \text{TR}^{\text{OUT}}. \tag{10}$$

The objective function (1) is the same as in TSFD, where we minimize the total number of tardy products. Constraints (11) make sure that each incoming truck is assigned exactly to one inbound door at a unique time. Inequalities (12) make sure that each inbound door at a given time can handle at most one incoming truck. Constraints (13) define the total completion time c_{io} for each shipment. Inequalities (14) ensure that U_{io} equals one if the shipment delivered by incoming truck i cannot reach its outgoing truck o before its departure time, i.e., if the sum of C_i and of the transshipment time δ_{go} exceeds the departure time d_o. Finally, constraints (9), (10) and (15) define the binary and non-negative restrictions on the decision variables.

2.4.3 An Enhanced Time-indexed Formulation

We next propose a preprocessing of input-parameters in TSFD$^+$ that allows us to reduce the number of constraints and variables. As a result, the convergence speed of reaching the optimal solution is improved.

First, observe that because of constraints (11), constraints (13) can be rewritten as equality constraints, i.e.,

$$C_i = \sum_{t \in T} \sum_{g \in G^{\text{IN}}} (t + p_i)\, x_{ig}^t \qquad i \in \text{TR}^{\text{IN}}. \tag{13$'$}$$

We next substitute them into constraints (14) to obtain:

$$U_{io}.M \geq \sum_{t \in T} \sum_{g \in G^{\text{IN}}} (t + p_i)\, x_{ig}^t + \sum_{t \in T} \sum_{g \in G^{\text{IN}}} \delta_{go} x_{ig}^t - d_o \qquad i \in \text{TR}^{\text{IN}}, o \in \text{TR}^{\text{OUT}} : w_{io} > 0, \tag{14}$$

which in turn can be written as:

$$U_{io}.M \geq \sum_{t \in T} \sum_{g \in G^{\text{IN}}} (t + p_i + \delta_{go})\, x_{ig}^t - d_o \qquad i \in \text{TR}^{\text{IN}}, o \in \text{TR}^{\text{OUT}} : w_{io} > 0. \tag{14-TSFD^{++}}$$

As a result, we reduce the number of constraints in TSFD$^+$ by $|\text{TR}^{\text{IN}}|$ and eliminate variables C_i.

Next, observe that the above modified constraints (14-TSFD^{++}) explicitly state that if an incoming truck i is scheduled to start at time t and at inbound door g, i.e., $\bar{x}_{ig}^t = 1$, then a shipment w_{io} inside the incoming truck will be tardy if $t + p_i + \delta_{go} > d_o$. Knowing this, it is possible to preprocess constraints (14-TSFD^{++}) and embed them into the objective function. We define the following parameters. For every $t \in T$, $i \in \text{TR}^{\text{IN}}$, $o \in \text{TR}^{\text{OUT}}$, and $g \in G^{\text{IN}}$:

$$\alpha_{iog}^t = t + p_i + \delta_{go} - d_o.$$

If $\alpha_{iog}^t > 0$, it implies the products w_{io} of shipment $(i - o)$ assigned at door g at time t will be tardy; otherwise the products are on time. The enhanced time-indexed formulation is denoted by [TSFD^{++}], and can then be stated as:

$$[\text{TSFD}^{++}] \quad \text{minimize} \quad \sum_{t \in T} \sum_{g \in G^{\text{IN}}} \sum_{i \in \text{TR}^{\text{IN}}} \left(\sum_{o \in \text{TR}^{\text{OUT}} : \alpha_{iog}^t > 0} w_{io} \right) x_{ig}^t \tag{16}$$

$$\text{subject to} \quad \sum_{t \in T} \sum_{g \in G^{\text{IN}}} x_{ig}^t = 1 \qquad i \in \text{TR}^{\text{IN}} \tag{11}$$

$$\sum_{i \in \text{TR}^{\text{IN}}} \sum_{t' = \max\{0, t - p_i\}}^{t-1} x_{ig}^{t'} \leq 1 \qquad g \in G^{\text{IN}}, t \in T \tag{12}$$

$$x_{ig}^t \in \{0, 1\} \qquad i \in \text{TR}^{\text{IN}}, g \in G^{\text{IN}}, t \in T \tag{15}$$

In Section 2.7, we computationally compare these three formulations and show the superiority of the TSFD^{++} formulation over the TSFD^{+} and TSFD formulations for the basic TSFD as stated in Section 2.3. In the following two sections, we introduce two important practical extensions and model each using the enhanced time-indexed TSFD^{++} formulation.

2.5 Multiple Transfer Trips

As seen earlier, and following the assumptions of Boysen et al. 2013, the basic TSFD problem along with the three formulations presented in Section 2.4 assume that all products for a given shipment going from an incoming to an outgoing truck will be transferred in a single trip regardless of their volume or quantities. However, in practice, transferring products between inbound and outbound doors within a cross-dock facility may take more than a trip depending on the capacity of the material handling equipment used as well as the number of products needed to be transferred. We assume that each transshipment has a complete capacity, except for the last one to a given outbound door. For that reason, it is convenient to assume that the complete unloading of a truck is carried out before transshipment starts. Indeed, when no information is available on the content or the unloading order of the products of the truck, we cannot know when a transshipment is complete before the truck has been fully unloaded. However, when unloading order is known, transshipment can be carried out concurrently, as it means having the complete information on the products in the truck, and therefore when a transshipment is complete.
We assume that only one worker is assigned to each door to take care of the moves from one inbound door to an outbound door, as adding more workers is not a common practice for cost reasons and for congestion issues in the cross-dock facility due to many workers moving products around.

In this section, we extend the basic TSFD to consider multiple trips instead of one, allowing us to track how many products are actually tardy within a given shipment. Let J_{io} be the number of trips needed to transfer all products of a given shipment w_{io} and CAP be the number of products that can be transferred in each trip:

$$J_{io} = \left\lceil \frac{w_{io}}{\text{CAP}} \right\rceil.$$

Without losing generality, we assume here that the capacity, $\textsc{cap}$, of the material handling equipment is independent of the shipment being transferred. However, this can be easily changed to $\textsc{cap}_{io}$ to allow the capacity to differ based on the type of shipments being transferred without affecting the subsequent models we propose.

In the case when we have more than two types of product in a given shipment w_{io}, the number of trips needed to transfer all these products will be:

$$J_{io} = \left\lceil \frac{\sum\limits_{k \in K} w_{io}^k V^k}{\text{CAP}} \right\rceil$$

where K is the index set of different types of products, w_{io}^k is the number of products of type k, which need to transfer from inbound truck i to outbound truck o and V^k is the weight/volume (depends on which the unit of $\textsc{cap}$ is) of product type k. In this paper, we only use the quantities of products to calculate the number of trips and the $\textsc{cap}$ is the maximum number of products that can be transferred in each trip.

2.5.1 Over Estimation of Tardy Products

We first extend the multiple transfer trips by considering multiple trips per shipment where a shipment is considered tardy if one or more of its trips are tardy, a worst case scenario or an over estimation based on the basic TSFD. In order to do so, we modify constraints (14-TSFD^{++}) to include the number of trips required per shipment $i-o$ as follows:

$$U_{io}.M \geq \sum_{t \in T} \sum_{g \in G^{\text{IN}}} (t + p_i + \boldsymbol{J_{io}} \delta_{go})\, x_{ig}^t - d_o \qquad i \in \text{TR}^{\text{IN}}, o \in \text{TR}^{\text{OUT}} : w_{io} > 0. \tag{14-TSFD$^{++}_{\text{M}}$}$$

These last constraints state that a shipment $i-o$ is considered tardy only if $t + p_i + J_{io}\delta_{go} - d_o > 0$. Hence, following the same preprocessing of the TSFD^{++} in Section 2.4.3, we define $\alpha_{iog}^t = t + p_i + J_{io}\delta_{go} - d_o$ for every $t \in T$, $i \in \text{TR}^{\text{IN}}$, $o \in \text{TR}^{\text{OUT}}$, and $g \in G^{\text{IN}}$. If $\alpha_{iog}^t > 0$, then a shipment is tardy; otherwise the shipment is on time. As a result, we obtain the following modified TSFD^{++} formulation:

$$
\begin{aligned}
&[\mathrm{TSFD}_{\mathrm{M}}^{++}] \quad \text{minimize} \quad && \sum_{t\in T}\sum_{g\in G^{\mathrm{IN}}}\sum_{i\in \mathrm{TR}^{\mathrm{IN}}}\left(\sum_{o\in \mathrm{TR}^{\mathrm{OUT}}:\alpha_{iog}^{t}>0} w_{io}\right) x_{ig}^{t} && && (16)\\
&\text{subject to} && \sum_{t\in T}\sum_{g\in G^{\mathrm{IN}}} x_{ig}^{t} = 1 && i\in \mathrm{TR}^{\mathrm{IN}} && (11)\\
& && \sum_{i\in \mathrm{TR}^{\mathrm{IN}}}\sum_{t'=\max\{0,t-p_i\}}^{t-1} x_{ig}^{t'} \le 1 && g\in G^{\mathrm{IN}}, t\in T && (12)\\
& && x_{ig}^{t}\in\{0,1\} && i\in \mathrm{TR}^{\mathrm{IN}}, g\in G^{\mathrm{IN}}, t\in T && (15)
\end{aligned}
$$

The only difference between the TSFD^{++} and $\mathrm{TSFD}_{\mathrm{M}}^{++}$ formulations is that the latter incorporates the number of trips required per shipment, J_{io}, when a shipment is tardy or not, i.e., some or all of its products are tardy.

2.5.2 Accurate Estimation of Tardy Products

We next extend the basic TSFD to identify the exact number of tardy products within a given shipment, which provides a more accurate tardiness calculation in the newly proposed TSFD formulations. We consequently modify the definition of the variables U_{io} to become U_{io}^{j} indicating that a trip j of a given shipment $i-o$ is tardy or not. The resulting model will be denoted by $\mathrm{TSFD}_{\mathrm{MT}}^{++}$. A tardy trip implies accordingly that all the products transferred in that trip are tardy too, but not necessarily the whole shipment distributed over multiple trips. We introduce the following constraints:

$$
U_{io}^{\boldsymbol{j}}.M \ge \sum_{t\in T}\sum_{g\in G^{\mathrm{IN}}} (t+p_i+\boldsymbol{j}\delta_{go})\, x_{ig}^{t} - d_o \quad i\in \mathrm{TR}^{\mathrm{IN}}, o\in \mathrm{TR}^{\mathrm{OUT}} : w_{io}>0, \boldsymbol{j}\in \boldsymbol{J_{io}}. \qquad (14\text{-}\mathrm{TSFD}_{\mathrm{MT}}^{++})
$$

These last constraints state that for a given shipment $(i-o)$ and all its trips, a trip is considered tardy only if $t+p_i+j\delta_{go}-d_o>0$. When a trip is tardy, we only consider the products in this trip to be tardy. This way a portion of the shipment is considered tardy instead of the whole shipment.

Using the same prepossessing logic of the enhanced time-indexed formulation, we introduce the following parameter. For every $t\in T$, $i\in \mathrm{TR}^{\mathrm{IN}}$, $o\in \mathrm{TR}^{\mathrm{OUT}}$, $g\in G^{\mathrm{IN}}$ and $j\in J_{io}$:

$$
\alpha_{iog}^{tj} = t+p_i+j\delta_{go}-d_o.
$$

For a given scheduled truck i at inbound door g starting at time t, i.e., $x^t_{gi} = 1$, α^{tj}_{iog} calculates the total time it takes part of the shipment $i - o$ for a given trip j to be unloaded and transferred minus the departure time of its outgoing truck. A trip is considered tardy if $\alpha^{tj}_{iog} > 0$; otherwise the trip is on time. Let W^j_{io} be the number of products in a given trip j for shipment $(i - o)$:

$$W^j_{io} = \begin{cases} \text{CAP} & \text{if} \quad j < J_{io} \\ w_{io} - (j-1).\text{CAP} & \text{if} \quad j = J_{io}, \end{cases}$$

where CAP is the transport capacity of a trip. W^j_{io} is the number of products per trip: it is the same in each trip except for the last trip $j < J_{io}$. In the last trip where $j = J_{io}$, the number of products does not necessarily fill the whole material handling equipment's capacity. The multiple trip $\text{TSFD}^{++}_{\text{MT}}$ can then be formulated as:

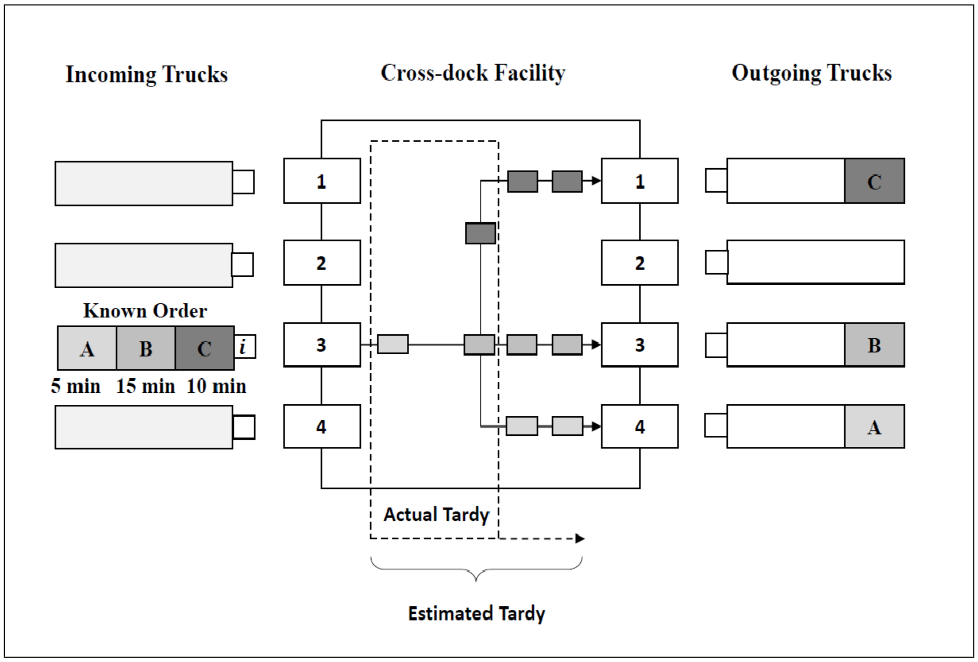
$$[\text{TSFD}^{++}_{\text{MT}}] \quad \text{minimize} \quad \sum_{t \in T} \sum_{g \in G^{\text{IN}}} \sum_{i \in \text{TR}^{\text{IN}}} \left(\sum_{o \in \text{TR}^{\text{OUT}}} \sum_{j \in J_{io} : \alpha^{tj}_{iog} > 0} W^j_{io} \right) x^t_{ig} \tag{17}$$

$$\text{subject to} \quad \sum_{t \in T} \sum_{g \in G^{\text{IN}}} x^t_{ig} = 1 \qquad i \in \text{TR}^{\text{IN}} \tag{11}$$

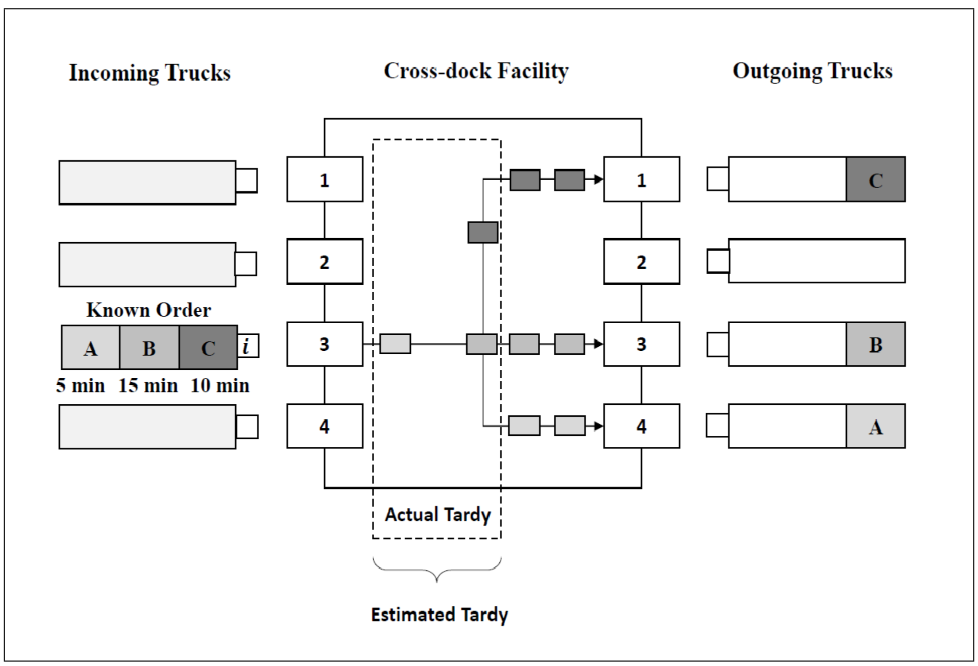
$$\sum_{i \in \text{TR}^{\text{IN}}} \sum_{t' = \max\{0, t - p_i\}}^{t-1} x^{t'}_{ig} \leq 1 \qquad g \in G^{\text{IN}}, t \in T \tag{12}$$

$$x^t_{ig} \in \{0, 1\} \qquad i \in \text{TR}^{\text{IN}}, g \in G^{\text{IN}}, t \in T \tag{15}$$

Figure 3 shows an example of how estimating the total number of tardy shipments can differ between over and accurate estimations in 3(a) and 3(b), respectively. We show in Section 2.7 the percentage of improvement between both scenarios for two sets of instances.



(a) Over estimation using $\mathrm{TSFD}_{\mathrm{M}}^{++}$



(b) Accurate estimation using $\mathrm{TSFD}_{\mathrm{MT}}^{++}$

Figure 3: Difference between over and accurate estimations in the number of tardy products

2.6 Known Unloading Order of Shipments

We next introduce another extension, denoted by $\text{TSFD}^{++}_{\text{MTO}}$, to $\text{TSFD}^{++}_{\text{MT}}$ where the order of shipments inside the incoming trucks is known. For example, an incoming truck contains three shipments A, B and C, and these shipments are ordered as first C to be unloaded, then B, and followed by A where each takes 30 minutes to be unloaded resulting in a total of 90 minutes to empty the truck. In this case, knowing the order will provide the cross-dock managers with more visibility and control over the time a shipment will actually take to be unloaded and transferred to its designated outgoing truck. In other words, the completion time of a shipment (unloading then transferring) now depends on the preceding shipments in the same truck rather than the whole processing time of the truck itself. In order to consider this extension, we next replace p_i with $\bar{u}_{io}$ to calculate accurately the total unloading time of each shipment. To embed this, the following constraints are introduced:

$$U^j_{io}.M \geq \sum_{t \in T} \sum_{g \in G^{\text{IN}}} (t + \bar{\boldsymbol{u}}_{\boldsymbol{io}} + j\delta_{go})\, x^t_{ig} - d_o \quad i \in \text{TR}^{\text{IN}}, o \in \text{TR}^{\text{OUT}} : w_{io} > 0, j \in J_{io}. \qquad (\text{14-TSFD}^{++}_{\text{MTO}})$$

Notice that the above constraints also extend and generalize (14-TSFD^{++}) to multiple trips and known unloading order of shipments. Using the same prepossessing logic of the enhanced time-indexed formulation TSFD^{+}, we introduce the following parameter α. For every $t \in T$, $i \in \text{TR}^{\text{IN}}$, $o \in \text{TR}^{\text{OUT}}$, $g \in G^{\text{IN}}$ and $j \in J_{io}$:

$$\alpha^{tj}_{iog} = t + \bar{u}_{io} + j\delta_{go} - d_o.$$

Similar to the preprocessing in the $\text{TSFD}^{++}_{\text{MT}}$ formulation in Section 2.5, a trip is considered tardy if $\alpha^{tj}_{iog} > 0$; otherwise the trip is on time. The only difference between the $\text{TSFD}^{++}_{\text{MT}}$ and $\text{TSFD}^{++}_{\text{MTO}}$ formulation is that in the latter we consider the unloading time of a shipment to be $\bar{u}_{io}$ instead of what is commonly overestimated by p_i. We show in our computational experiments the economical benefits of doing so. The multiple trip with known order of shipments $\text{TSFD}^{++}_{\text{MTO}}$ can then be formulated as:

$$[\text{TSFD}^{++}_{\text{MTO}}] \quad \text{minimize} \quad \sum_{t \in T} \sum_{g \in G^{\text{IN}}} \sum_{i \in \text{TR}^{\text{IN}}} \left(\sum_{o \in \text{TR}^{\text{OUT}}} \sum_{j \in J_{io} : \alpha^{tj}_{iog} > 0} W^j_{io} \right) x^t_{ig} \qquad (18)$$

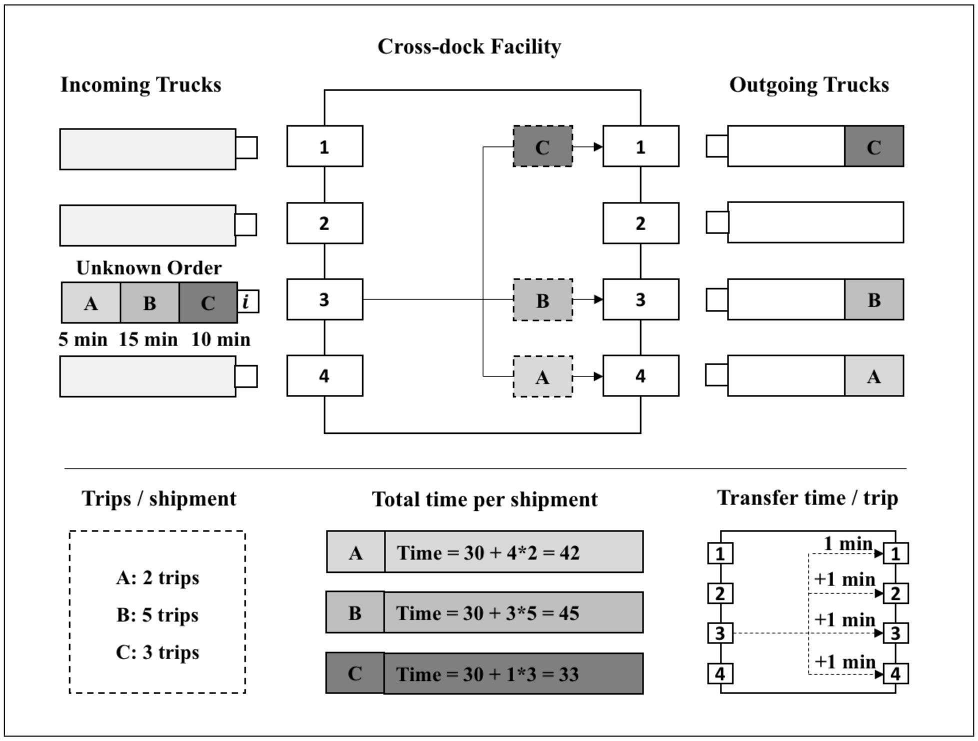
$$\text{subject to} \quad \sum_{t \in T} \sum_{g \in G^{\text{IN}}} x_{ig}^{t} = 1 \qquad i \in \text{TR}^{\text{IN}} \tag{11}$$

$$\sum_{i \in \text{TR}^{\text{IN}}} \sum_{t'=\max\{0, t-p_i\}}^{t-1} x_{ig}^{t'} \leq 1 \qquad g \in G^{\text{IN}}, t \in T \tag{12}$$

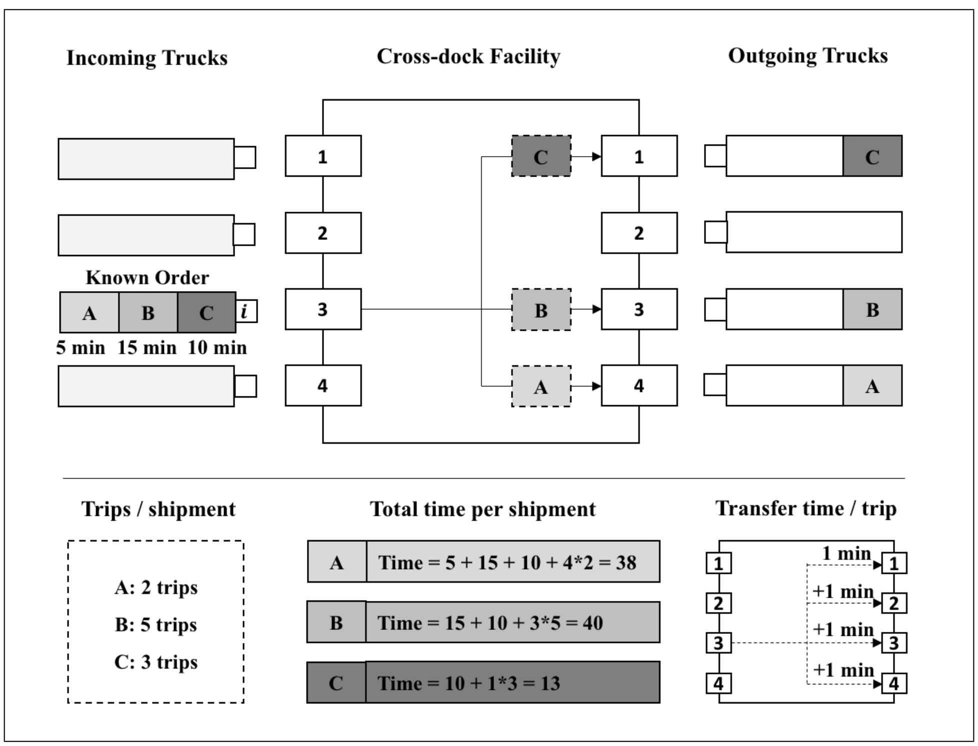
$$x_{ig}^{t} \in \{0, 1\} \qquad i \in \text{TR}^{\text{IN}}, g \in G^{\text{IN}}, t \in T \tag{15}$$

One key aspect of the enhanced time-indexed formulation is that all internal information (e.g., transfer time, completion time) become available once the scheduling decision x is made. Therefore, our proposed prepossessing or enhancement procedures allow us to calculate such information for each given decision variable. As will be seen next, this significantly speeds up the convergence of obtaining the optimal solutions for the TSFD problem, and any of its extensions.

Figure 4 shows an example of how the knowledge of the order of shipments allows us to calculate the transshipment time more accurately, resulting in a better estimate of tardy products per shipment as seen in the known order case in Figure 4(b) versus the unknown order case in Figure 4(a).



(a) Unknown Order using TSFD^{++}



(b) Known Order using TSFD$^{++}_{\mathrm{MTO}}$

Figure 4: Known versus unknown order of shipments inside incoming truck i

2.7 Computational Experiments

We next test the performances of all the TSFD formulations and extensions presented or introduced in this paper. All formulations are implemented in Java using the concert technology library of CPLEX 12.6. CPLEX parameters are set default. We generate two sets of benchmark instances based on cross-dock practices in the postal and retail industries. We first present the data generation mechanism of each set of instances, and then the classical performance measures in the area of operations research in order to compare the performances of these formulations. After this, we report the comparative results along with their analysis on the added precision in the tardiness evaluation. Finally, some graphs are presented to demonstrate the economical impact of considering our proposed extensions.

2.7.1 Data Generation

Two sets of instances are generated. The first set is reported in Boysen et al. 2013. We refer to this first set as P13 where P is used to refer to data based on the postal industry. The second set of instances is based on Nassief et al. 2018b where the authors generate data based on their observations in a cross-dock company in the retailing industry. We refer to this second set as R17 where R is used to refer to data based on the retailing industry. Set P13 consists of three groups of instances: small, medium and large size instances while R17 focuses only on a large set of instances. In what follows, we explain in detail the generation mechanism for each input parameter in both sets P13 and R17.

We introduce two more parameters:

B_{io}: set of all outgoing trucks (o') which have shipments preceding the current one ($i-o$) in incoming truck i

$W(i)$: total number of shipments in incoming truck i.

Table 2: Summary of data generation mechanisms for input parameters

Terminology & notation		Generation Mechanism P13	Generation Mechanism R17
Inbound doors	G^{IN}	$G^{\text{IN}} = \{2, 3, 4, 6, 7, 8, 10, 15, 20\}$	$G^{\text{IN}} = \{2, 3, 4, 6, 7, 8, 10, 15, 20\}$
Outbound doors	G^{OUT}	$G^{\text{OUT}} = \{2, 3, 4, 6, 7, 8, 10, 15, 20\}$	$G^{\text{OUT}} = \{2, 3, 4, 6, 7, 8, 10, 15, 20\}$
Incoming trucks	TR^{IN}	$\text{TR}^{\text{IN}} = \{8, 20, 80\}$	$\text{TR}^{\text{IN}} = \{8, 20, 80\}$
Outgoing trucks	TR^{OUT}	$\text{TR}^{\text{OUT}} = \{8, 20, 80\}$	$\text{TR}^{\text{OUT}} = \{8, 20, 80\}$
Processing time	p_i	$p_i \sim \mathcal{N}(\mu,\, \sigma^2)$: $\mu = 30, \sigma \in \{2, 4, 6, 8\}$	$p_i \sim \log \mathcal{N}(\mu,\, \sigma^2)$: $\mu = 1.25, \sigma = 0.76$
Number of products	w_{io}	$w_{io} \sim \mathcal{U}(1, 10)$ if $\mathcal{U}(0, 1) < 0.5$; 0, otherwise	$w_{io} \sim \dfrac{\exp(\beta, \gamma)}{\lvert G^{\text{OUT}} \rvert}$: $\beta = 73.53, \gamma = 1.22$
Unloading time	u_{io}	$\left(\sum\limits_{o' \in B_{io}} w_{io'} + w_{io} \right) \dfrac{p_i}{W(i)}$	$\left(\sum\limits_{o' \in B_{io}} w_{io'} + w_{io} \right) \dfrac{p_i}{W(i)}$
Departure time	d_o	$d_o \sim \dfrac{\sum\limits_{i \in \text{TR}^{\text{IN}}} p_i}{\lvert G^{\text{IN}} \rvert} \mathcal{U}(0.5, 0.9)$	$d_o \sim q\, \dfrac{\sum\limits_{i \in \text{TR}^{\text{IN}}} p_i}{\lvert G^{\text{IN}} \rvert}\, \mathcal{U}(0.5, 0.9)$: $q \in \{1.2, 1.3, 1.4, 1.5\}$
Transshipment time	δ_{go}	$\delta_{go} \sim \mathcal{U}(1, 10)$	$\delta_{go} \sim \mathcal{U}(10, 15)$

- Sets: we allocate discrete numbers for the four global sets ($G^{\text{IN}}, G^{\text{OUT}}, \text{TR}^{\text{IN}}, \text{TR}^{\text{OUT}}$) that are used throughout all the instances.
  - Inbound and outbound doors: for the small size instances, we consider 2, 3 and 4 inbound doors. For the medium size instances, we consider 6, 7 and 8 inbound doors. For the large size instances, we consider 10, 15 and 20 inbound doors. We keep the same cardinality for both inbound and outbound doors per instance.
  - Incoming and outgoing trucks: for the small size instances, we consider 8 incoming trucks. For the medium size instances, we consider 20 incoming trucks, and for the large size instances, we consider 80 incoming trucks. We keep the same cardinality for both incoming and outgoing trucks per instance.
- Parameters: for each parameter, we provide the distribution function used in each data set.
  - Processing times: in P13, the processing time for each incoming truck is

generated using a normal distribution function with an average processing time of $\mu = 30$ minutes per truck. Boysen et al. 2013 claim that it is the minimum average processing time according to cross-dock managers in the postal industry. In R17, the processing time follows a log normal distribution with an average of $\mu = 1.25$ hours and deviation of $\sigma = 0.76$. These are generated based on real cross-dock data in the retailing industry in the USA see Nassief et al. 2018b.

– Number of products per shipment: in P13, the number of products per shipment is generated with a uniform distribution between one and ten only if a random generator provides 0.5 or more. Otherwise, no shipment is considered between both incoming and outgoing trucks. In R17, we follow the distribution of shipments coming from the retailing industry and observe that it follows a 2-parameter exponential distribution with a scale of $\gamma = 73.53$, and threshold of $\theta = 1.22$.

– Unloading time per shipment: we first generate the order of shipments by shuffling them randomly inside their incoming trucks. Then, the unloading time of each shipment is calculated as

$$u_{io} = \left(\sum_{o' \in B_{io}} w_{io'} + w_{io} \right) \frac{p_i}{W(i)},$$

taking into account the unloading time of the shipment and its precedent ones in the same incoming truck. Notice that the last shipment in the same incoming truck will result in an unloading time that is equivalent to the processing time of the whole truck, i.e.,

$$u_{io_{(\text{LAST})}} = \left(\sum_{o \in \text{TR}^{\text{OUT}}} w_{io} \right) \frac{p_i}{W(i)} = W(i) \frac{p_i}{W(i)} = p_i.$$

– Departure time per outgoing truck: the departure time of each outgoing truck in set P13 is determined by: $d_o \sim \frac{\sum_{i \in \text{TR}^{\text{IN}}} p_i}{|G^{\text{IN}}|} . \mathcal{U}(0.5, 0.9)$. However, in set R17 it is determined by:

$$d_o = \max \left\{ p_{\min} + w_{\min}\delta_{\min}, \sim \mathcal{U}(1, \frac{6|\text{TR}^{\text{IN}}|}{|G^{\text{IN}}|q}) \right\} : q \in \{1.2, 1.3, 1.4, 1.5\},$$

where q ranges from tight to loose departure times.

- Transshipment time between inbound and outbound doors: in both data sets P13 and R17, the transshipment time is generated using a uniform distribution between 1 and 10 as in Boysen et al. 2013.

Using the aforementioned data generation mechanisms, we create two sets of data instances, both of which are identified by the number of incoming and outgoing trucks, inbound and outbound doors. For the set P13, for a given number of trucks, the different instances are characterized with different processing times whereas for the R17 set, they are characterized by different due dates. Table 2 provides a summary of the generation mechanisms for both data sets on all the sets and parameters presented and used in this paper.

2.7.2 Performance Measurements

We use the following measurements in the subsequent tables to compare and to measure the performances of all presented and introduced formulations in this paper. Note that some formulations are ILP models, while other are MILP (Mixed Integer Linear Program) ones.

- %LP: LP deviation. It is calculated as $\%\text{LP} = \frac{|\text{OPT}-\text{LP}|}{\text{OPT}}100$, where OPT is the optimal ILP or MILP value and LP is the optimal value for the LP relaxation of each model. If an optimal ILP or MILP solution is not obtained within the time limit, we use the incumbent value (best known solution) for the tested instance. The LP, here, represents the optimal linear programming relaxation obtained at the root node before branching in CPLEX.

- %GAP: optimality gap upon termination, i.e., $\%\text{GAP} = 100\frac{|\text{OPT}-\text{LB}|}{\text{OPT}}$, where LB is the best lower bound obtained. The LB, here, represents the best lower bound found after the stopping criterion is reached when branching in CPLEX, and not the first LP value at the root of the B&B as in the %LP.

- %UB: percentage deviation of the upper bounds found with respect to the optimal/incumbent value. That is $\%\text{UB} = 100\frac{|\text{OPT}-\text{UB}|}{\text{OPT}}$.

- B&B: number of nodes explored in the branch-and-bound tree.

- CPU: computational time in seconds.

- %TARDY: percentage of tardy products, which is calculated based on the number of tardy products obtained from the optimal (or best) scheduling solution over the total number of products for a given instance.

$$\%\text{TARDY} = \frac{\text{OPT}}{\sum\limits_{i \in \text{TR}^{\text{IN}}} \sum\limits_{o \in \text{TR}^{\text{OUT}}} w_{io}} 100.$$

- %IMPROV: percentage of improvement in the schedules obtained by a given formulation vs. the base one, resulting in (possibly) less number of tardy products.

$$\%\text{IMPROV} = \frac{|\text{OPT}_{\text{base}} - \text{OPT}|}{\text{OPT}_{\text{base}}}.$$

- %DIFF: differences in scheduling decisions between one formulation and another. They are calculated by the number of incoming trucks which have different schedules divided by the number of total incoming trucks. For instance, if the %DIFF is 100% it indicates that the schedule has completely changed in one formulation as opposed to another.

2.7.3 Comparison of Basic TSFD with TSFD$^+$ and TSFD^{++}

We now test the basic TSFD by comparing the performances of the three presented and introduced formulations: TSFD, TSFD$^+$ and TSFD^{++}. We use a length of one unit for the time periods, after rescaling the P13 data sets, i.e., dividing the processing times by 5.

Tables 3 and 4 compare between the three formulations using Sets P13 and R17, respectively. In both tables, TSFD^{++} significantly outperforms the other two formulations in terms of the quality of the solutions and the computational time it takes to prove optimality. We also observe that the %LP is almost zero in all these instances when it comes to the enhanced time-indexed formulation, TSFD^{++}. This indicates that with the preprocessing introduced in Section 2.4.3, we are able to obtain the optimal solution at the root node before branching in most of the tested instances. We also highlight that even though our initially proposed time-indexed formulation TSFD$^+$ under performs the enhanced one, it can still provide better %GAP when terminating within the time limit, as opposed to the TSFD, introduced by Boysen et al. 2013. Indeed, the TSFD, as proposed by Boysen et al. 2013, can only reach the optimal solution for small size instances as opposed to our formulations.

Table 3: Comparing TSFD^{++}, TSFD^{+} and TSFD for data set P13

Set P13				TSFD^{++}					TSFD^{+}					TSFD				
Trucks	G^{IN}	G^{OUT}	σ	%LP	%GAP	%UB	B&B	CPU (seconds)	%LP	%GAP	%UB	B&B	CPU (seconds)	%LP	%GAP	%UB	B&B	CPU (seconds)
8x8	2	2	2	10.29	.	.	.	0.02	99.29	.	.	17	0.45	100	.	.	3.44E+05	5.43
	2	2	4	6.67	.	.	.	0.02	99.83	.	.	0	0.45	100	.	.	1.86E+05	3.14
	2	2	6	0.0	.	.	.	0.01	97.33	.	.	0	0.43	100	.	.	2.71E+05	4.75
	2	2	8	0.0	.	.	.	0.01	95.13	.	.	0	0.23	100	.	.	1.30E+05	2.62
	3	3	2	0.0	.	.	.	0.01	91.91	.	.	0	0.28	100	.	.	93,337	1.99
	3	3	4	0.0	.	.	.	0.01	95.89	.	.	11	0.58	100	.	.	58,878	1.16
	3	3	6	0.0	0.0	0.0	0	0.01	92.95	0.0	0.0	7	0.50	100	0.0	0.0	1.39E+05	3.15
	3	3	8	1.79	.	.	.	0.01	95.66	.	.	2	0.67	100	.	.	57,541	1.23
	4	4	2	0.0	.	.	.	0.01	94.70	.	.	0	0.51	100	.	.	87,856	1.77
	4	4	4	0.0	.	.	.	0.01	94.80	.	.	17	0.63	100	.	.	24,338	0.68
	4	4	6	0.0	.	.	.	0.01	96.74	.	.	0	0.46	100	.	.	82,990	1.62
	4	4	8	0.0	.	.	.	0.01	94.28	.	.	0	0.64	100	.	.	22,622	0.62
Average				**1.56**	**0.0**	**0.0**	**0.0**	**0.01**	**95.71**	**0.0**	**0.0**	**4.50**	**0.49**	**100.00**	**0.0**	**0.0**	**124,796**	**2.35**
20x20	6	6	2	0.13	.	.	.	0.18	97.02	5.33	0.0	4.93E+05	.	100	98.13	1.99	1.77E+07	.
	6	6	4	0.0	.	.	.	0.04	97.33	12.80	0.89	4.60E+05	.	100	97.80	1.34	1.57E+07	.
	6	6	6	0.0	.	.	.	0.14	97.82	4.46	0.0	3.13E+05	.	100	99.77	2.36	1.58E+07	.
	6	6	8	0.0	.	.	.	0.05	97.16	0.36	0.0	4.36E+05	.	100	98.81	0.24	1.49E+07	.
	7	7	2	0.0	.	.	.	0.06	97.67	21.50	0.0	5.65E+05	.	100	98.76	0.75	1.48E+07	.
	7	7	4	0.0	.	.	.	0.05	97.48	9.70	0.0	2.69E+05	.	100	98.01	1.41	1.30E+07	.
	7	7	6	0.31	0.0	0.0	0	0.23	97.88	6.51	0.0	3.23E+05	3600	100	99.48	0.0	1.39E+07	3600
	7	7	8	1.24	.	.	.	0.58	98.54	4.93	0.0	2.88E+05	.	100	100	0.0	1.82E+07	.
	8	8	2	0.0	.	.	.	0.07	97.45	28.33	0.0	4.63E+05	.	100	93.88	0.0	1.25E+07	.
	8	8	4	0.0	.	.	.	0.07	97.98	6.54	0.0	2.93E+05	.	100	96.18	3.67	1.34E+07	.
	8	8	6	0.0	.	.	.	0.17	97.05	9.76	0.0	2.07E+05	.	100	86.98	0.92	1.03E+07	.
	8	8	8	0.0	.	.	.	0.06	97.35	11.64	0.0	2.22E+05	.	100	75.32	0.72	1.08E+07	.
Average				**0.14**	**0.0**	**0.0**	**0.0**	**0.14**	**97.56**	**10.16**	**0.07**	**3.61E+05**	**3600**	**100.00**	**95.26**	**1.12**	**1.42E+07**	**3600**
80x80	10	10	2	0.0	.	.	0	14.92	99.75	99.46	16.43	.	.	100	100	39.07	1.57E+05	.
	10	10	4	0.0	.	.	0	2.74	99.81	99.71	35.58	.	.	100	100	60.67	1.70E+05	.
	10	10	6	0.09	.	.	1,235	95.35	99.70	99.63	36.65	.	.	100	100	52.14	2.34E+05	.
	10	10	8	0.0	.	.	0	31.03	99.85	99.80	34.49	.	.	100	100	53.73	2.81E+05	.
	15	15	2	0.04	.	.	709	61.24	99.60	99.58	17.41	.	.	100	100	37.51	12,906	.
	15	15	4	0.03	0.0	0.0	178	83.14	99.76	99.75	28.26	0	3600	100	100	30.61	3,560	3600
	15	15	6	0.02	.	.	7,930	207.73	99.79	99.78	42.29	.	.	100	100	50.82	13,423	.
	15	15	8	0.06	.	.	208	53.05	99.64	99.63	27.64	.	.	100	100	56.11	42,324	.
	20	20	2	0.0	.	.	0	6.10	99.50	99.46	10.70	.	.	100	100	37.70	13	.
	20	20	4	0.02	.	.	41	69.17	99.65	99.64	21.85	.	.	100	100	44.73	223	.
	20	20	6	0.04	.	.	385	80.48	99.48	99.46	24.56	.	.	100	100	54.84	8	.
	20	20	8	0.01	.	.	0	30.56	99.57	99.55	23.65	.	.	100	100	64.43	3,772	.
Average				**0.03**	**0.0**	**0.0**	**890**	**61.29**	**99.67**	**99.62**	**26.63**	**0.0**	**3600**	**100**	**100**	**48.53**	**76,519**	**36000**

Table 4: Comparing TSFD^{++}, TSFD^{+} and TSFD for data set R17

Set R17				TSFD^{++}					TSFD^{+}					TSFD				
Trucks	G^{IN}	G^{OUT}	q	%LP	%gap	%UB	B&B	CPU (seconds)	%LP	%gap	%UB	B&B	CPU (seconds)	%LP	%gap	%UB	B&B	CPU (seconds)
80x80	10	10	1.2	1.08	.	.	42	23.10	98.81	74.68	51.92	835	.	100	100	48.36	29,346	.
	10	10	1.3	0.93	.	.	3	14.12	97.94	45.21	1.39	24,787	.	100	100	31.67	86,656	.
	10	10	1.4	1.60	.	.	19	20.46	97.62	29.82	7.55	19,422	.	100	100	56.60	61,796	.
	10	10	1.5	0.0	.	.	0	6.72	96.73	0.0	0.0	0	428	100	100	81.50	31,813	.
	15	15	1.2	0.30	.	.	0	14.88	97.92	73.33	33.93	186	.	100	100	58.93	9,744	.
	15	15	1.3	0.49	.	.	0	20.58	96.73	55.56	3.45	4,939	.	100	100	47.58	10,580	.
	15	15	1.4	0.0	0.0	0.0	0	18.03	96.60	48.72	8.33	3,968	3600	100	100	89.72	7,805	3600
	15	15	1.5	0.0	.	.	0	10.06	94.95	14.89	6.82	5,378	.	100	100	1509.09	24,124	.
	20	20	1.2	0.20	.	.	0	21.05	98.51	78.46	30.16	4	.	100	100	364.02	3,287	.
	20	20	1.3	0.72	.	.	0	22.14	97.88	73.91	31.97	5	.	100	100	356.56	25	.
	20	20	1.4	0.0	.	.	0	16.95	96.55	56.99	43.08	1,451	.	100	100	1432.31	5,552	.
	20	20	1.5	2.34	.	.	69	54.34	95.75	44.74	18.75	2,018	.	100	100	729.69	3,779	.
Average				**0.64**	**0.0**	**0.0**	**11.08**	**20.20**	**97.17**	**49.69**	**19.78**	**5,249**	**3335**	**100**	**100**	**711.28**	**22,875**	**3600**

2.7.4 Multiple Transfer Trips: $\text{TSFD}^{++}_{\text{M}}$ and $\text{TSFD}^{++}_{\text{MT}}$

We next test our first extension of considering multiple transfer trips instead of one when calculating the number of tardy products per shipment. This is to demonstrate the impact of accurately estimating the tardy cost using $\text{TSFD}^{++}_{\text{MT}}$ versus over estimating using $\text{TSFD}^{++}_{\text{M}}$. Tables 5 and 6 demonstrate the improvements when considering multiple trips with accurate estimation by $\text{TSFD}^{++}_{\text{MT}}$ as opposed to a multiple trips with over estimation by TSFD^{++}, using Sets P13 and R17, respectively. We first observe that the %TARDY products in the $\text{TSFD}^{++}_{\text{MT}}$ is less than the ones in TSFD^{++} in almost every single tested instance. This behaviour is expected since we are capturing more accurately the time at which a product reaches its destined outgoing truck, allowing us to determine more accurately if it is late or not. This also results in an improvement, %IMPROV, of 18% for P13 and 57% for R17 on average. It should be noted that the changes obtained do not come only from the greater precision with which the cost is calculated, but also result in some major changes in the scheduling decisions. For instance, most of the scheduling decisions (i.e., where trucks are assigned and sequenced) have also changed significantly as shown in %DIFF, ranging from 37% to 96% changes in the schedules between both formulations. Therefore, considering multiple trips, where only products in late trips are considered tardy rather than the whole shipment, is a more practical and encouraging assumption that should be part of modeling cross-dock scheduling problems let alone its beneficial impact on the decisions and tardy costs. Finally, the resulting increment in the CPU time using the $\text{TSFD}^{++}_{\text{MT}}$ instead of the $\text{TSFD}^{++}_{\text{M}}$ is insignificant and negligible, making it still a desirable approach.

Table 5: Comparing $\text{TSFD}^{++}_{\text{MT}}$ to $\text{TSFD}^{++}_{\text{M}}$ for data set P13

Set P13				$\text{TSFD}^{++}_{\text{MT}}$				$\text{TSFD}^{++}_{\text{M}}$			
Truck	G^{IN}	G^{OUT}	σ	%TARDY	%IMPROV	%DIFF	CPU (seconds)	%TARDY	%IMPROV	%DIFF	CPU (seconds)
8x8	2	2	2.0	22.97	0.00	50	0.01	22.97	.	.	0.05
	2	2	4.0	14.86	12.00	25	0.01	16.89	.	.	0.01
	2	2	6.0	25.00	0.00	25	0.01	25.00	.	.	0.01
	2	2	8.0	27.03	0.00	50	0.01	27.03	.	.	0.01
	3	3	2.0	34.46	5.56	37	0.01	36.49	.	.	0.01
	3	3	4.0	33.11	12.50	75	0.01	37.84	0.0	0.0	0.01
	3	3	6.0	49.32	16.09	50	0.01	58.78	.	.	0.01
	3	3	8.0	22.97	27.66	75	0.01	31.76	.	.	0.02
	4	4	2.0	44.59	21.43	37	0.01	56.76	.	.	0.01
	4	4	4.0	54.73	7.95	100	0.01	59.46	.	.	0.01
	4	4	6.0	32.43	31.43	87	0.01	47.30	.	.	0.01
	4	4	8.0	56.08	11.70	25	0.01	63.51	.	.	0.01
Average				**34.80**	**12.19**	**53**	**0.01**	**40.32**	**0.0**	**0.0**	**0.01**
20x20	6	6	2.0	46.44	5.41	60	0.07	49.10	.	.	0.07
	6	6	4.0	43.53	7.80	75	0.06	47.22	.	.	0.08
	6	6	6.0	40.96	6.27	90	0.25	43.70	.	.	0.19
	6	6	8.0	40.79	4.23	80	0.13	42.59	.	.	0.05
	7	7	2.0	40.10	9.13	65	0.07	44.13	.	.	0.14
	7	7	4.0	48.24	9.92	95	0.08	53.56	0.0	0.0	0.06
	7	7	6.0	37.19	7.46	75	0.06	40.19	.	.	0.13
	7	7	8.0	33.25	12.61	90	0.16	38.05	.	.	0.16
	8	8	2.0	52.87	9.13	90	0.09	58.18	.	.	0.09
	8	8	4.0	43.19	11.11	60	0.08	48.59	.	.	0.16
	8	8	6.0	45.33	10.94	90	0.29	50.90	.	.	0.19
	8	8	8.0	41.82	11.27	75	0.33	47.13	.	.	0.14
Average				**42.81**	**8.77**	**78**	**0.14**	**46.95**	**0.0**	**0.0**	**0.12**
80x80	10	10	2.0	32.44	3.28	80	48.47	33.54	.	.	30.17
	10	10	4.0	30.41	3.54	88	56.35	31.52	.	.	43.86
	10	10	6.0	29.42	3.57	87	48.35	30.51	.	.	653.23
	10	10	8.0	26.89	3.25	90	519.85	27.80	.	.	71.09
	15	15	2.0	38.98	4.44	78	51.82	40.79	.	.	28.82
	15	15	4.0	33.28	5.02	85	89.54	35.04	0.0	0.0	55.66
	15	15	6.0	31.12	4.70	91	3600	32.66	.	.	93.19
	15	15	8.0	30.96	5.08	88	69.19	32.62	.	.	231.58
	20	20	2.0	41.72	6.99	70	7.50	44.85	.	.	18.01
	20	20	4.0	38.00	5.89	82	44.30	40.38	.	.	32.39
	20	20	6.0	39.34	6.52	83	48.38	42.09	.	.	7.53
	20	20	8.0	31.89	6.03	83	48.35	33.94	.	.	41.12
Average				**33.70**	**4.86**	**83**	**386.01**	**35.48**	**0.0**	**0.0**	**108.89**

Table 6: Comparison of Models $\text{TSFD}^{++}_{\text{MT}}$ to $\text{TSFD}^{++}_{\text{M}}$

Data set R17											
				$\text{TSFD}^{++}_{\text{MT}}$				$\text{TSFD}^{++}_{\text{M}}$			
Truck	G^{IN}	G^{OUT}	q	%TARDY	%IMPROV	%DIFF	CPU (seconds)	%TARDY	%IMPROV	%DIFF	CPU (seconds)
80x80	10	10	1.2	10.71	5.44	55	7.45	11.33	.	.	8.15
	10	10	1.3	8.69	4.43	91	68.32	9.09	.	.	16.44
	10	10	1.4	8.34	4.04	67	6.74	8.69	.	.	19.34
	10	10	1.5	5.40	5.01	91	15.91	5.69	.	.	13.13
	15	15	1.2	13.90	6.57	65	8.06	14.88	.	.	9.50
	15	15	1.3	13.09	5.35	92	633.94	13.83	0.0	0.0	46.45
	15	15	1.4	9.69	7.40	87	68.19	10.47	.	.	9.59
	15	15	1.5	7.94	5.49	93	67.35	8.41	.	.	8.50
	20	20	1.2	16.83	8.58	67	10.35	18.41	.	.	3.08
	20	20	1.3	15.61	7.92	88	275.54	16.95	.	.	19.86
	20	20	1.4	11.19	6.35	87	10.30	11.95	.	.	21.01
	20	20	1.5	8.17	10.88	88	139.88	9.17	.	.	22.53
Average				**10.80**	**6.45**	**80**	**109.34**	**11.57**	**0.0**	**0.0**	**16.46**

2.7.5 Multiple Trips and Known Order of Shipments $\textbf{TSFD}^{++}_{\textbf{MTO}}$

We next test our second TSFD extension, with the explicit modeling of multiple trips together with the information of the order of shipments inside their incoming trucks. We perform the comparisons in two steps. First, Tables 7 and 8 demonstrate the improvements when considering multiple trips with accurate estimation and known unloading order of $\text{TSFD}^{++}_{\text{MTO}}$ as opposed to multiple trips with over estimation of tardy cost in $\text{TSFD}^{++}_{\text{M}}$, using Sets P13 and R17, respectively. Second, Tables 9 and 10 compare $\text{TSFD}^{++}_{\text{MTO}}$ with $\text{TSFD}^{++}_{\text{MT}}$ to demonstrate the impact of knowing (not knowing) the order of unloading shipments when multiple trips are considered with accurate estimation of tardy products.

We first observe that in Tables 7 and 8, the percentage of tardy products in the $\text{TSFD}^{++}_{\text{MTO}}$ is less than almost half the percentages in $\text{TSFD}^{++}_{\text{M}}$ among all tested instances. This behaviour is expected for two reasons. The first reason is related to considering multiple trips and calculating the actual time at which a product reaches its destined outgoing truck, allowing us to determine accurately if it is tardy or not. The second reason is due to the information on the unloading order of shipments inside the incoming truck: this allows us to know that the latest loaded products by the supplier (the closest to the truck door) will be unloaded earlier than those loaded at the beginning. As a result, the unloading time per product or a shipment with several products can be calculated more accurately and independently of all its successive products. This also results in an improvement, %IMPROV, of 50% for P13 and 80% for R17 on average. Changes in the scheduling decisions are noticeable as well as shown in the column %DIFF, ranging from 50% to even 100% in some instances. Therefore, considering multiple trips and obtaining information on the unloading order of shipments inside the incoming trucks can indeed result in significant savings for cross-dock companies.

We next notice a similar and consistent behaviour in the results shown in Tables 9 and 10 to the aforementioned ones. The only difference is that here the marginal improvements in %TARDY and %IMPROV are less than the ones we see in Tables 7 and 8, respectively. This is due to the fact that in Tables 9 and 10, the comparisons are done on the additional improvement resulted from the second extension: knowing the order of unloading shipments.

Table 7: Comparing $\text{TSFD}^{++}_{\text{MTO}}$ to $\text{TSFD}^{++}_{\text{M}}$ for data set P13

Set P13				$\text{TSFD}^{++}_{\text{MTO}}$				$\text{TSFD}^{++}_{\text{M}}$			
Truck	G^{IN}	G^{OUT}	σ	%TARDY	%IMPROV	%DIFF	CPU (seconds)	%TARDY	%IMPROV	%DIFF	CPU (seconds)
8x8	2	2	2.0	15.54	32.35	62	0.01	22.97	.	.	0.05
	2	2	4.0	11.49	32.00	12	0.01	16.89	.	.	0.01
	2	2	6.0	17.57	29.73	87	0.01	25.00	.	.	0.01
	2	2	8.0	18.24	32.50	87	0.01	27.03	.	.	0.01
	3	3	2.0	26.35	27.78	87	0.01	36.49	.	.	0.01
	3	3	4.0	23.65	37.50	87	0.02	37.84	0.0	0.0	0.01
	3	3	6.0	33.78	42.53	75	0.01	58.78	.	.	0.01
	3	3	8.0	14.19	55.32	75	0.01	31.76	.	.	0.02
	4	4	2.0	27.03	52.38	100	0.01	56.76	.	.	0.01
	4	4	4.0	39.86	32.95	87	0.02	59.46	.	.	0.01
	4	4	6.0	23.65	50.00	75	0.01	47.30	.	.	0.01
	4	4	8.0	35.81	43.62	87	0.01	63.51	.	.	0.01
Average				**23.93**	**39.06**	**76**	**0.01**	**40.32**	**0.0**	**0.0**	**0.01**
20x20	6	6	2.0	32.56	33.68	70	0.06	49.10	.	.	0.07
	6	6	4.0	29.05	38.48	100	0.07	47.22	.	.	0.08
	6	6	6.0	27.16	37.84	85	0.09	43.70	.	.	0.19
	6	6	8.0	27.85	34.61	80	0.22	42.59	.	.	0.05
	7	7	2.0	25.62	41.94	90	0.08	44.13	.	.	0.14
	7	7	4.0	32.13	40.00	95	0.24	53.56	0.0	0.0	0.06
	7	7	6.0	24.94	37.95	95	0.25	40.19	.	.	0.13
	7	7	8.0	20.57	45.95	100	0.33	38.05	.	.	0.16
	8	8	2.0	32.73	43.74	95	0.09	58.18	.	.	0.09
	8	8	4.0	25.79	46.91	100	0.21	48.59	.	.	0.16
	8	8	6.0	27.08	46.80	95	0.32	50.90	.	.	0.19
	8	8	8.0	24.42	48.18	90	0.13	47.13	.	.	0.14
Average				**27.49**	**41.34**	**91**	**0.17**	**46.95**	**0.0**	**0.0**	**0.12**
80x80	10	10	2.0	26.65	20.54	97	50.02	33.54	.	.	30.17
	10	10	4.0	24.94	20.89	92	205.31	31.52	.	.	43.86
	10	10	6.0	23.99	21.38	96	144.05	30.51	.	.	653.23
	10	10	8.0	21.24	23.57	98	582.11	27.80	.	.	71.09
	15	15	2.0	30.31	25.68	97	31.89	40.79	.	.	28.82
	15	15	4.0	24.75	29.38	95	93.03	35.04	0.0	0.0	55.66
	15	15	6.0	22.82	30.11	97	95.48	32.66	.	.	93.19
	15	15	8.0	22.84	29.97	96	31.15	32.62	.	.	231.58
	20	20	2.0	30.76	31.43	97	38.23	44.85	.	.	18.01
	20	20	4.0	26.76	33.72	93	62.97	40.38	.	.	32.39
	20	20	6.0	28.56	32.14	95	82.24	42.09	.	.	7.53
	20	20	8.0	22.19	34.61	95	85.16	33.94	.	.	41.12
Average				**25.48**	**27.78**	**95**	**125.14**	**35.48**	**0.0**	**0.0**	**108.89**

Table 8: Comparing $\text{TSFD}^{++}_{\text{MTO}}$ to $\text{TSFD}^{++}_{\text{M}}$ for data set R17

Set R13				$\text{TSFD}^{++}_{\text{MTO}}$				$\text{TSFD}^{++}_{\text{M}}$			
Truck	G^{IN}	G^{OUT}	q	%TARDY	%IMPROV	%DIFF	CPU (seconds)	%TARDY	%IMPROV	%DIFF	CPU (seconds)
80x80	10	10	1.2	6.06	46.47	100	9.90	11.33	.	.	8.15
	10	10	1.3	4.52	50.34	97	10.94	9.09	.	.	16.44
	10	10	1.4	4.39	49.48	98	11.54	8.69	.	.	19.34
	10	10	1.5	2.22	61.02	98	8.53	5.69	.	.	13.13
	15	15	1.2	7.16	51.86	97	13.69	14.88	.	.	9.50
	15	15	1.3	6.17	55.37	98	8.95	13.83	0.0	0.0	46.45
	15	15	1.4	3.77	63.96	98	24.42	10.47	.	.	9.59
	15	15	1.5	2.34	72.16	93	22.69	8.41	.	.	8.50
	20	20	1.2	7.95	56.81	96	14.66	18.41	.	.	3.08
	20	20	1.3	7.03	58.55	97	14.33	16.95	.	.	19.86
	20	20	1.4	3.75	68.61	100	2.66	11.95	.	.	21.01
	20	20	1.5	2.39	73.94	97	14.37	9.17	.	.	22.53
Average				**4.81**	**59.05**	**97**	**13.06**	**11.57**	**0.0**	**0.0**	**16.46**

Table 9: Comparison of Models $\text{TSFD}^{++}_{\text{MTO}}$ to $\text{TSFD}^{++}_{\text{MT}}$ - Data set P13

Truck	G^{IN}	G^{OUT}	q	$\text{TSFD}^{++}_{\text{MTO}}$ %TARDY	%IMPROV	%DIFF	CPU (seconds)	$\text{TSFD}^{++}_{\text{MT}}$ %TARDY	%IMPROV	%DIFF	CPU (seconds)
8x8	2	2	2.0	15.54	32.35	87	0.01	22.97	.	.	0.01
	2	2	4.0	11.49	22.73	37	0.01	14.86	.	.	0.01
	2	2	6.0	17.57	29.73	87	0.01	25.00	.	.	0.01
	2	2	8.0	18.24	32.50	75	0.01	27.03	.	.	0.01
	3	3	2.0	26.35	23.53	75	0.01	34.46	.	.	0.01
	3	3	4.0	23.65	28.57	87	0.02	33.11	0.0	0.0	0.01
	3	3	6.0	33.78	31.51	75	0.01	49.32	.	.	0.01
	3	3	8.0	14.19	38.24	25	0.01	22.97	.	.	0.01
	4	4	2.0	27.03	39.39	100	0.01	44.59	.	.	0.01
	4	4	4.0	39.86	27.16	100	0.02	54.73	.	.	0.01
	4	4	6.0	23.65	27.08	25	0.01	32.43	.	.	0.01
	4	4	8.0	35.81	36.14	87	0.01	56.08	.	.	0.01
Average				**23.93**	**30.74**	**71**	**0.01**	**34.80**	**0.0**	**0.0**	**0.01**
20x20	6	6	2.0	32.56	29.89	75	0.06	46.44	.	.	0.07
	6	6	4.0	29.05	33.27	90	0.07	43.53	.	.	0.06
	6	6	6.0	27.16	33.68	95	0.09	40.96	.	.	0.25
	6	6	8.0	27.85	31.72	90	0.22	40.79	.	.	0.13
	7	7	2.0	25.62	36.11	95	0.08	40.10	.	.	0.07
	7	7	4.0	32.13	33.39	80	0.24	48.24	0.0	0.0	0.08
	7	7	6.0	24.94	32.95	95	0.25	37.19	.	.	0.06
	7	7	8.0	20.57	38.14	70	0.33	33.25	.	.	0.16
	8	8	2.0	32.73	38.09	80	0.09	52.87	.	.	0.09
	8	8	4.0	25.79	40.28	100	0.21	43.19	.	.	0.08
	8	8	6.0	27.08	40.26	85	0.32	45.33	.	.	0.29
	8	8	8.0	24.42	41.60	80	0.13	41.82	.	.	0.33
Average				**27.49**	**35.78**	**86**	**0.17**	**42.81**	**0.0**	**0.0**	**0.14**
80x80	10	10	2.0	26.65	17.85	92	50.02	32.44	.	.	48.47
	10	10	4.0	24.94	17.99	92	205.31	30.41	.	.	56.35
	10	10	6.0	23.99	18.47	96	144.05	29.42	.	.	48.35
	10	10	8.0	21.24	21.01	100	582.11	26.89	.	.	519.85
	15	15	2.0	30.31	22.23	95	31.89	38.98	.	.	51.82
	15	15	4.0	24.75	25.65	95	93.03	33.28	0.0	0.0	89.54
	15	15	6.0	22.82	26.67	96	95.48	31.12	.	.	3600
	15	15	8.0	22.84	26.22	95	31.15	30.96	.	.	69.19
	20	20	2.0	30.76	26.27	92	38.23	41.72	.	.	7.50
	20	20	4.0	26.76	29.57	95	62.97	38.00	.	.	44.30
	20	20	6.0	28.56	27.40	96	82.24	39.34	.	.	48.38
	20	20	8.0	22.19	30.41	95	85.16	31.89	.	.	48.35
Average				**25.48**	**24.15**	**94**	**125.14**	**33.70**	**0.0**	**0.0**	**386.01**

Table 10: Performance comparison of models $\text{TSFD}^{++}_{\text{MTO}}$ to $\text{TSFD}^{++}_{\text{MT}}$ - Data set R17

Truck	G^{IN}	G^{OUT}	q	$\text{TSFD}^{++}_{\text{MTO}}$ %TARDY	%IMPROV	%DIFF	CPU (seconds)	$\text{TSFD}^{++}_{\text{MT}}$ %TARDY	%IMPROV	%DIFF	CPU (seconds)
80x80	10	10	1.2	6.06	43.39	98	9.90	10.71	.	.	7.45
	10	10	1.3	4.52	48.04	100	10.94	8.69	.	.	68.32
	10	10	1.4	4.39	47.36	98	11.54	8.34	.	.	6.74
	10	10	1.5	2.22	58.97	98	8.53	5.40	.	.	15.91
	15	15	1.2	7.16	48.47	98	13.69	13.90	.	.	8.06
	15	15	1.3	6.17	52.85	98	8.95	13.09	0.0	0.0	633.94
	15	15	1.4	3.77	61.09	98	24.42	9.69	.	.	68.19
	15	15	1.5	2.34	70.54	98	22.69	7.94	.	.	67.35
	20	20	1.2	7.95	52.75	97	14.66	16.83	.	.	10.35
	20	20	1.3	7.03	54.98	97	14.33	15.61	.	.	275.54
	20	20	1.4	3.75	66.49	100	2.66	11.19	.	.	10.30
	20	20	1.5	2.39	70.75	97	14.37	8.17	.	.	139.88
Average				**4.81**	**56.31**	**98**	**13.06**	**10.80**	**0.0**	**0.0**	**109.34**

2.7.6 Economical Analysis

We next provide an economical analysis on the impacts of our proposed extensions over the basic TSFD.

In Figure 5, we show how the percentage of tardy products changes for both unknown and known unloading order of products inside their incoming trucks. This is demonstrated on an instance in Set R17 with different due dates. We observe that taking the unloading order of shipments, along with their products, into account would allow cross-dock managers to produce daily schedules that are less expensive than the ones without information on the unloading order. More visibility on the content of the incoming trucks will definitely have a direct economical impact regardless of the tightness or looseness of the due dates.

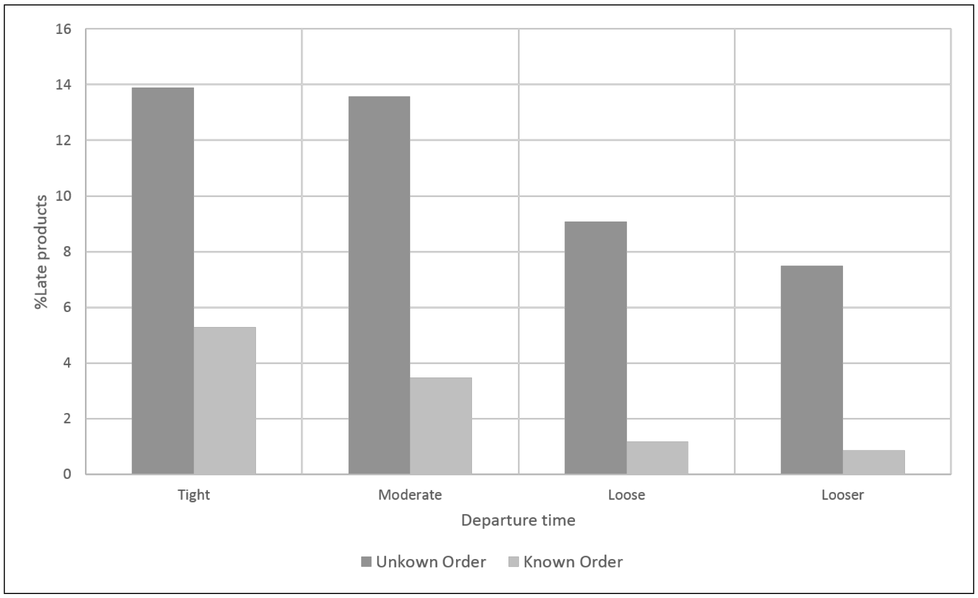


Figure 5: Comparing two scenarios: unknown vs. known unloading order as a function of the percentages of tardy shipments. Data set R13, instance 20x20, gate = 8.

Figures 6 and 7 depict several economical performance measures: unknown and known unloading order of shipments, and their products, respectively, inside the incoming trucks. We show the impact of these assumptions on the outgoing trucks, their delayed departure times if they would wait for the tardy products, the completion times of their last loaded products if they depart as originally planned, and the percentage of on-time loaded products.

We highlight that in Figure 7 the gap between the completion time of the last loaded product in the outgoing trucks and the planned departure times of the outgoing trucks is narrower than in the one in Figure 6. This is simply due to knowing the order of unloading the products at the first place. Both figures also help identifying the additional time needed for the products to be completely loaded in their outgoing trucks. For instance, in Figure 7, outgoing truck 18 would require an additional two hours, i.e., $13 - 11 = 2$, for all its consolidated products to arrive at the designated outbound door and be fully loaded. Hence, a cross-dock manager could make a decision based on this information to whether delay the departure time of this outgoing truck by two hours or make other relevant decisions that affect the associated penalty/fees with its tardy products.

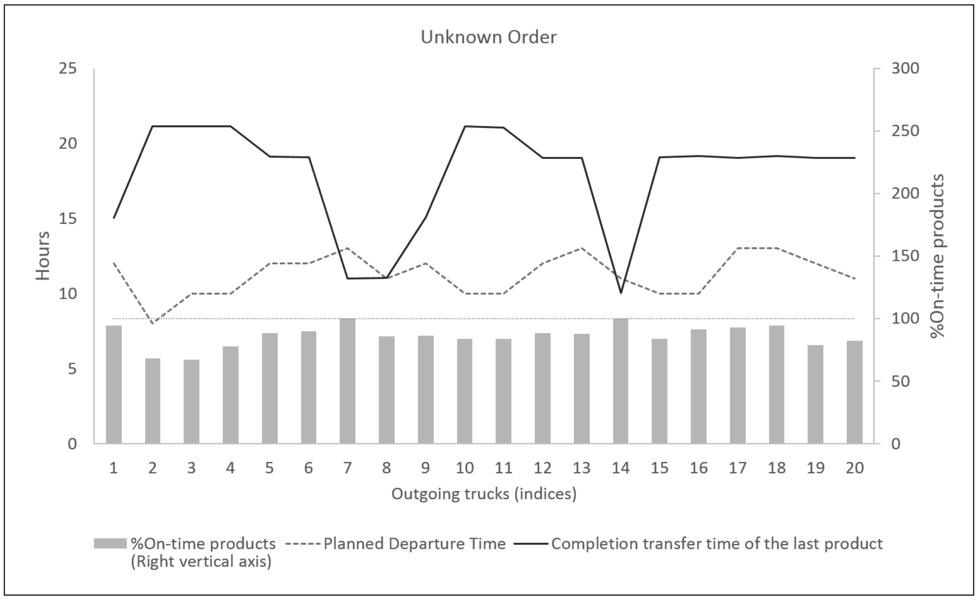


Figure 6: Planned due dates and the completion time of last shipments transferred to outbound doors with percentage of on time shipments in unknown order scenario. Data set R13, instance 20x20, gate = 8.

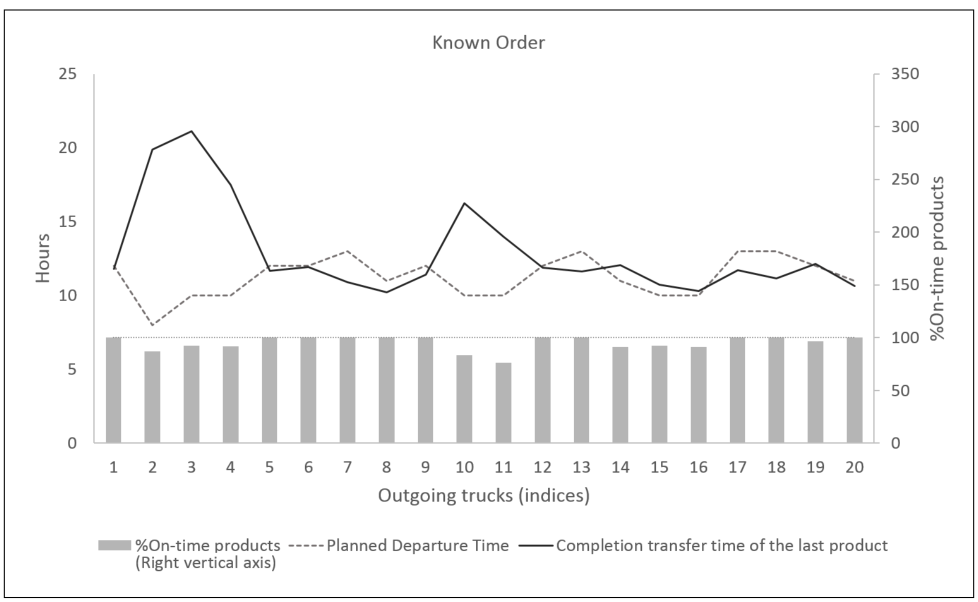


Figure 7: Planned due dates and the completion time of last shipments transferred to outbound doors with percentage of on time shipments in known order scenario. Data set R13, instance 20x20, gate = 8.

Finally, in Figure 8 we depict the impact on the number of tardy products when considering multiple trips with accurate estimation using $\text{TSFD}^{++}_{\text{MT}}$, and with over estimation using $\text{TSFD}^{++}_{\text{M}}$. It is easy to see the consistency in this figure between both scenarios. The decrements are due to the looseness in the departure times of the outgoing trucks, i.e., the looser the departure time is, the less tardy the products are.

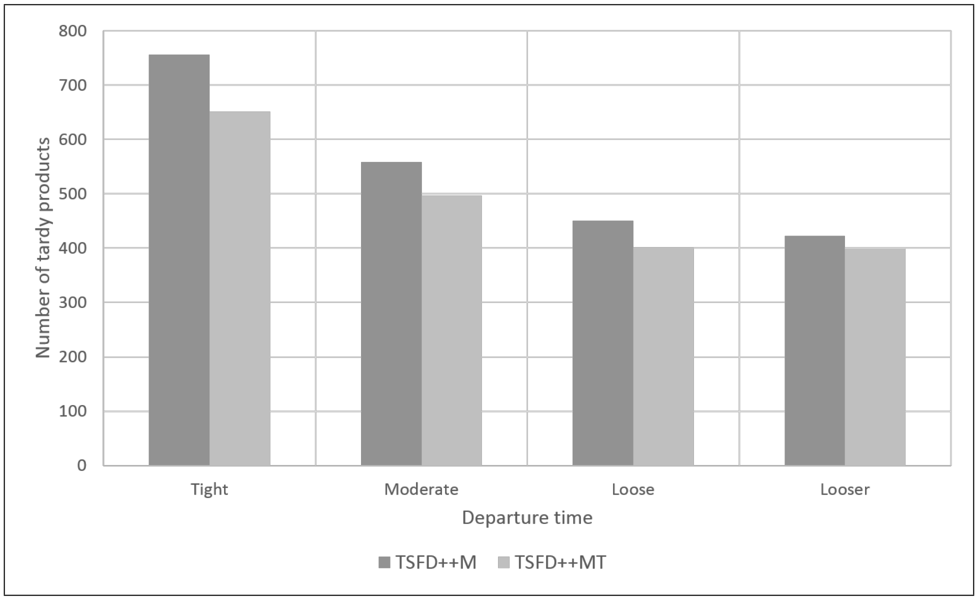


Figure 8: Percentage of tardy shipments when we consider the number of trips to transfer shipments from inbound to outbound doors. Data set R13, instance 20x20, gate = 8.

2.8 Conclusion

In this paper, we solve efficiently a daily complex scheduling problem that arises in the postal and retail industries. After the analysis of a first basic model proposed by Boysen 2010 and Boysen et al. 2013, we introduce two integer programming formulations and compare their performances with the one of Boysen et al. 2013. These new formulations allow the improvement of the modelling of the transshipment operations and their impact on assessing more accurately the number of tardy products, rather than over-estimating them. Furthermore, we investigate the impact of taking advantage of the information on the unloading order of the shipments, along with their associated products, inside the incoming trucks, and adapt our best enhanced time-indexed formulation accordingly.

By doing so, we can then estimate accurately without any over-estimation the minimum number of tardy products.

Extensive computational experiments are run to test all our proposed TSFD formulations with the state-of-the-art as well as with our proposed extended ones. We conclude with an economical analysis that demonstrates an average saving of 50% in the total cost. We encourage cross-dock companies to capture accurately the number of their tardy products by considering the practical extensions proposed here, and to obtain from their suppliers the unloading order of shipments inside their incoming trucks in order to accurately compute the number of tardy jobs and consequently reduce as much as possible the number of their tardy deliveries.

We also highlight that the data instances we tested are representative of average sized cross-docks. For larger cross-docks (the largest ones in the world have 500 cross-dock doors), further developments are needed in order to enhance the scalability of the proposed models with, e.g., decomposition modelling and algorithms.

Acknowledgements

The research of the authors has been supported by Clear Destination Incorporation and The Mathematics of Information Technology and Complex Systems (MITACS): Elevate (W. Nassief) and Accelerate (T. Vu) programs. This support is highly appreciated and acknowledged B. Jaumard has been supported by a Concordia University Research Chair (Tier I) and by an NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

Chapter 3

A linear model for truck scheduling in cross-dock with known unloading order of shipments

3.1 Introduction

In order to transport commodities quickly to satisfy customer needs while reducing the cost of inventory storage, cross docking has emerged as an efficient logistics strategy. Products are carried in incoming trucks, which arrive at cross docking facilities and wait to be assigned to an inbound door before being processed. After goods are unloaded, they are sorted and transferred to outgoing trucks in the cross-dock outbound side without or little of storage.

One critical problem which managers have to deal with in daily working is how to schedule incoming and outgoing trucks to conduct the transfer operation at cross docking facilities as efficiently as possible. Some studies dedicated to the inbound and outbound scheduling problem try to simplify this problem by fixing the transportation time between inbound doors and outbound doors to the same value and make all dock doors be identical on each side such as: F. Chen et al. 2009a; Cota et al. 2016; Boysen et al. 2010b. Other papers are however considering the transportation time between dock doors such as: Miao et al. 2009a; Molavi et al. 2018; Belle et al. 2013.

In this work, we introduce three new mathematical models to schedule incoming and outgoing trucks at a cross docking centre to minimize the maximum lateness of outgoing trucks. In these models, we take into account the transfer time from inbound to outbound doors while still keeping the model linear. In addition, we integrate the unloading order of shipments in incoming trucks and get significant improvement in the lateness value.

The paper is organized as follows: Section 3.2, we review the related works, Section 3.3 is devoted to problem description, Sections 3.4 and 3.5 are for introducing proposed models and Section 3.6 is for computational results.

3.2 Literature Review

In this section, we review some papers that study the problem of scheduling inbound and outbound in a cross-docking. These studies can be classified into two main classes. The first class of papers try to reduce the complexity of the truck scheduling by approximating the transportation times from inbound doors to outbound doors as a given constant. That is, the dock doors on each side become identical. The second class of studies, on the other hand, take into account the transshipment time in side cross-dock. As a consequence, the dock doors become non-identical and the problem become more complex.

3.2.1 Identical Dock Doors

Some papers model the inbound and outbound scheduling as a multiple-stage hybrid machine scheduling problem. F. Chen et al. 2009a consider the cross-docking scheduling problem as a two-stage hybrid machine scheduling problem, which has been proposed by F. Chen et al. 2009b. The first stage is to schedule inbound side and the second stage is to schedule outbound side. In this paper, the authors form a network based mixed integer programming (MIP) model for this problem and some Johnson's rule-based heuristics to minimize the makespan. Their MIP model can only solve small scale problems with 7 trucks and 4 dock doors. Following this work, Cota et al. 2016 formulate a time-indexed model for the same problem. The model is tested with small size problem with the number of trucks and doors are up to 8 and 9, respectively. They show that the time-indexed model outperforms the network-based model in F. Chen et al. 2009a. Fonseca et al. 2017 study a less complex problem of trucks scheduling with only one dock door on each inbound and outbound side to minimize the makespan. In this paper, the author propose a time-indexed model and develop a hybrid Lagrangean metaheuristic. The largest size problem which they attempt to solve has 84 trucks. In Bellanger et al. 2013, the cross-docking operation is split into three stages, namely, receiving, sorting and shipping. The objective is to minimize the completion of the last batch. In order to tackle the problem of scheduling trucks, the authors develop a branch-and-bound algorithm. The results show that their algorithm can generate good schedules with up to 3,000 jobs.

All mentioned papers assume that dock doors are identical, i.e., the transportation times between dock doors are the same and then can be removed from the models. However, this assumption is not fit with real life practice where the travel time between a pair of inbound and outbound doors is different from one pair to the next.

3.2.2 Non-identical Dock Doors

The second class of models in which authors consider the transshipment time can be seen in: Miao et al. 2009a; Molavi et al. 2018; Belle et al. 2013. In Miao et al. 2009a, the authors study problem of dock assignment to minimize the sum of total operation and total of penalty costs. In this paper, they consider the transshipment time of cargoes between dock doors in their constraints to assure products can be transferred to outbound side before the departure time of outgoing trucks. The size of problem they can solve with exact method is 12 to 18 trucks with a number of doors ranging from 4 to 6 doors. Belle et al. 2013 consider the scheduling problem for both inbound and outbound trucks with multiple dock doors to minimize the total travel time and the total tardiness. In this paper, authors take the arrival time of inbound and outbound trucks and the transshipment time between dock door into account. They use a Tabu Search approach to solve the scheduling problem. In their experimental results, they show that their heuristics can solve problem with size up to 30 trucks and the number of dock doors is fixed at 3. Molavi et al. 2018 also consider the arrival times of trucks and transportation time between dock doors in their mixed integer programming model for cross-docking scheduling problem. In addition, they integrate the information of sorting of shipments into incoming trucks in their model. The objective function of proposed model is to minimize the cost of delayed shipments. By using optimizer software, they can solve small size instances with 9 trucks and 5 dock doors. The authors conclude that the sorting of shipments in inbound trucks can improve the scheduling in cross-docking.

When the transshipment times between dock doors are considered, the model become non-linear. This problem is addressed in paper mentioned above by introducing additional decision variables and constraints to keep track of the dock door assignment of each pair of incoming and outgoing trucks. In this paper, we overcome the problem of non-linear constraints without introducing new decision variables and new constraints.

3.3 Problem Statement

3.3.1 Problem Description

In a cross-dock facility, a set of incoming trucks on the inbound side carry products from suppliers which are then unloaded and transferred to the outbound side. On the outbound side, a set of empty outgoing trucks are waiting to be loaded. On each side, there are several dock doors reserved to unload products from incoming trucks and to load products into outgoing trucks. An incoming truck may contain products for several outgoing trucks and an outgoing truck may have its products coming from several incoming trucks. Each outgoing truck has its own planned departure time, i.e., it should leave the cross-dock centre before the given a deadline. However, in some situations, there is no subcontract for late products, therefore, an outgoing truck has to wait until all of its products are loaded successfully. An outgoing truck which cannot leave the cross-dock before its deadline is considered late. In this paper, we schedule incoming and outgoing trucks in order to minimize the maximum lateness of outgoing trucks. At each dock door, we have to decide which truck to assign and the time to start processing that truck. Several assumptions are made for the problem of scheduling inbound and outbound based on Cota et al. 2016 and Molavi et al. 2018:

- All incoming and outgoing trucks are available at the time of scheduling.
- Incoming and outgoing trucks are unloaded and loaded without interruption.
- Outgoing trucks have to wait until all of its products are loaded successfully before it leaves the cross-docking centre.
- Outgoing trucks cannot start loading until all their products unloaded from incoming trucks and transferred to the outbound side.

3.3.2 Notations

Given a cross-dock centre, we denote by TR^{IN} and TR^{OUT} the set of incoming and outgoing trucks, respectively. For each outgoing truck $o \in \text{TR}^{\text{OUT}}$, a departure time d_o is the deadline which outgoing truck o should leave before that. The number of products from incoming truck $i \in \text{TR}^{\text{IN}}$ and outgoing truck $o \in \text{TR}^{\text{OUT}}$ is denoted by w_{io}. Let TR_o^{IN} be the set of incoming trucks which carry products for outgoing truck $o \in \text{TR}^{\text{OUT}}$. On the inbound side of cross docking facility, we have a set of inbound doors G^{IN} and on the outbound side we have a set of outbound doors G^{OUT}. Let p_i be the processing time, i.e., time to being unloaded, of incoming truck $i \in \text{TR}^{\text{IN}}$ and p_o be the processing time, i.e., time to being loaded, outgoing truck $o \in \text{TR}^{\text{OUT}}$. The transportation time between two dock doors, inbound door $g \in G^{\text{IN}}$ and outbound door $g' \in G^{\text{OUT}}$ is $\delta_{gg'}$.

Notations	Descriptions
TR^{IN}	set of incoming trucks
TR^{OUT}	set of outgoing trucks
TR_o^{IN}	set of incoming trucks which contain products for outgoing truck o
G^{IN}	set of dock doors on inbound side
G^{OUT}	set of dock doors on outbound side
p_i, p_o	processing time of incoming truck $i \in \text{TR}^{\text{IN}}$ and outgoing truck $o\text{TR}^{\text{OUT}}$
w_{io}	products from incoming truck i to outgoing truck o
$\delta_{gg'}$	time to transfer a product from inbound door $g \in G^{\text{IN}}$ to outbound door $g' \in G^{\text{OUT}}$

Table 11: Mathematical notations

3.4 Time-indexed Mathematical Model

According to the results in Chapter 2 and in Cota et al. 2016, we can see that the time-indexed models have better bound and performance than network based models. Therefore, we formulate a time indexed model for the problem of scheduling inbound and outbound trucks in cross-docking facilities. This model is based on a time horizon which is divided into equal time blocks. We use one unit of time as the length of each time block. Let $t \in T$ be the discretized time intervals over which incoming and outgoing trucks are scheduled. We define two sets of decision variables to determine the schedule of each incoming truck and outgoing truck:

$$x_{ig}^{t} = \begin{cases} 1 & \text{if incoming truck } i \text{ starts processing, i.e., being unloaded at} \\ & \text{inbound door } g \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_{og'}^{t'} = \begin{cases} 1 & \text{if outgoing truck } o \text{ starts processing, i.e., being loaded at} \\ & \text{outbound door } g' \text{ at time } t' \\ 0 & \text{otherwise} \end{cases}$$

In real life practice, the transshipment time between each pair of dock doors is different from others. It may take less time to transfer products from an inbound door to other close outbound doors and more time to other far outbound doors.

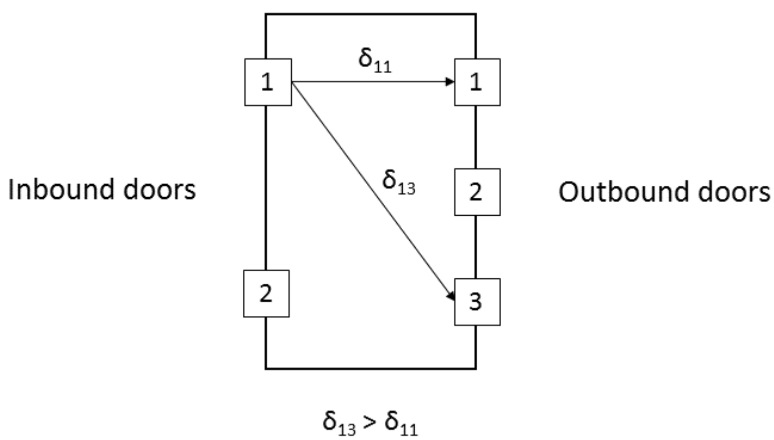


Figure 9: The transshipment times, δ_{11} and δ_{12}, are different between dock doors.

Because of these differences in transportation times, inbound doors are different from each other and the same with outbound doors. This makes the scheduling process become more complex since different dock door assignments for each truck may lead to different values of the objective function. In order to state the relationships between outgoing trucks and incoming trucks, we have to know the assignment of each truck to dock doors. In the literature, researchers try to tackle this problem by defining a new set of decision variables:

$$z_{iogg'}^{tt'} = \begin{cases} 1 & \text{if incoming truck } i \text{ is assigned to inbound door } g \text{ at time } t \\ & \text{and outgoing truck } o \text{ is assigned to outbound door } g' \text{ at time } t' \\ 0 & \text{otherwise} \end{cases}$$

This practice can be seen in Molavi et al. 2018; Belle et al. 2013; Miao et al. 2009b. This is actually a way to linearize the non-linearized term: $z^t_{iogg'} = x^t_{ig} \times y^{t'}_{og'}$. For the scheduling problem in this paper, each outgoing truck has to wait until all of its products are unloaded from incoming trucks before it can start loading. This condition can be stated as follows:

$$\sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} t' y^{t'}_{og'} \geq \sum_{t \in T} \sum_{g \in G^{\text{IN}}} (t + p_i) x^t_{ig} + \sum_{t \in T} \sum_{g \in G^{\text{IN}}} \sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} w_{io} \delta_{gg'} x^t_{ig} y^{t'}_{og'} \qquad o \in \text{TR}^{\text{OUT}}, i \in \text{TR}^{\text{IN}}_o.$$

The reason for these non-linearized constraints is because of the unknown dock door assignment of each truck at the scheduling time. In order to overcome this unknown information, we look in a different way.

We assume that outgoing truck $o \in \text{TR}^{\text{OUT}}$ is assigned to outbound door $g' \in G^{\text{OUT}}$ at time $t' \in T$ and a precedent incoming truck $i \in \text{TR}^{\text{IN}}_o$ is assigned to inbound gate $g \in G^{\text{IN}}$ at time $t \in T$. The precedence condition between outgoing truck o and its precedent incoming truck i can be stated as follows:

$$t' \geq t + p_i + w_{io}\delta_{gg'} \Leftrightarrow t' - w_{io}\delta_{gg'} \geq t + p_i. \tag{19}$$

We then generalize inequality (19) but fix the dock door assignment, i.e., inbound gate g of incoming truck i. inequality (19) can be stated in general as follow:

$$\sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} (t' - w_{io}\delta_{gg'}) y^{t'}_{og'} \geq \sum_{t \in T} (t + p_i) x^t_{ig} \quad o \in \text{TR}^{\text{OUT}}, i \in \text{TR}^{\text{IN}}_o, g \in G^{\text{IN}}. \tag{20}$$

However, if our assumption about the scheduling of incoming truck i does not happen, i.e., $\sum_{t \in T} x^t_{ig} = 0$ at inbound gate g. The inequality (20) will become:

$$t' - w_{io}\delta_{gg'} \geq 0. \tag{21}$$

The inequality (21) is not correct because it creates a false lower bound for the starting time of outgoing truck o. To address this problem, we add a new parameter called $\bar{\delta}$, which is the upper bound of all transportation time $w_{io}\delta_{gg'}$ to transfer a shipment from an incoming truck to another outgoing truck. inequality (20) can be reformulated as follows:

$$\sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} (t' - w_{io}\delta_{gg'}) y^{t'}_{og'} \geq \sum_{t \in T} (t + p_i) x^t_{ig} - \bar{\delta}(1 - \sum_{t \in T} x^t_{ig}) \quad o \in \text{TR}^{\text{OUT}}, i \in \text{TR}^{\text{IN}}_o, g \in G^{\text{IN}}. \tag{28}$$

With constraints (28), inequality (21) become $t' - w_{io}\delta_{gg'} \geq -\bar{\delta}$ or $t' \geq w_{io}\delta_{gg'} - \bar{\delta}(< 0)$. By adding parameter $\bar{\delta}$, we are able to recover the lower bound of starting time for outgoing truck o. We then can model the problem of scheduling truck in cross-docking centre with the new linearized precedence constraints as follows:

$$[\text{LINEAR}] \qquad \min \qquad L_{\max} \tag{22}$$

subject to:

$$L_{\max} \geq \sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} (t' + p_o) y_{og'}^{t'} - d_o \qquad o \in \text{TR}^{\text{OUT}} \tag{23}$$

$$\sum_{t \in T} \sum_{g \in G^{\text{IN}}} x_{ig}^{t} = 1 \qquad i \in \text{TR}^{\text{IN}} \tag{24}$$

$$\sum_{i \in \text{TR}^{\text{IN}}} \sum_{h=\max\{0, t-p_i+1\}}^{t} x_{ig}^{h} \leq 1 \qquad t \in T, g \in G^{\text{IN}} \tag{25}$$

$$\sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} y_{og'}^{t'} = 1 \qquad o \in \text{TR}^{\text{OUT}} \tag{26}$$

$$\sum_{o \in \text{TR}^{\text{OUT}}} \sum_{h=\max\{0, t'-p_o+1\}}^{t'} y_{og'}^{h} \leq 1 \qquad t' \in T, g' \in G^{\text{OUT}} \tag{27}$$

$$\sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} (t' - w_{io}\delta_{gg'}) y_{og'}^{t'} \geq \sum_{t \in T} (t + p_i) x_{ig}^{t} - \bar{\delta}(1 - \sum_{t \in T} x_{ig}^{t}) \qquad o \in \text{TR}^{\text{OUT}}, i \in \text{TR}_o^{\text{IN}}, g \in G^{\text{IN}} \tag{28}$$

$$x_{ig}^{t} \in \{0, 1\} \qquad i \in \text{TR}^{\text{IN}}, g \in G^{\text{IN}}, t \in T \tag{29}$$

$$y_{og'}^{t'} \in \{0, 1\} \qquad o \in \text{TR}^{\text{OUT}}, g' \in G^{\text{OUT}}, t' \in T \tag{30}$$

$$L_{\max} \geq 0. \tag{31}$$

The objective function (22) is to minimize the maximum lateness of outgoing trucks. Constraints (23) define $L_{\max}$ is upper bound value of lateness between completion time of each outgoing truck and its fixed departure time. Constraints (24)-(27) guarantee all incoming trucks and outgoing trucks are well scheduled. Inequalities (24) and (26) force the condition that each incoming and outgoing truck is assigned to one dock door at an unique time. Inequalities (25) and (27) make sure each dock door handles one truck at a given time. Constraints (28) declare that all an outgoing truck can only start being processed when all of its products are unloaded and transferred to the dock where that truck is assigned. The last three constraints (29)-(30) is to define the domain of decision variables. Constraints (31) is to force $L_{\max}$ to have non-negative value.

3.5 Known Unloading Order of Shipments

In this section, we introduce two models with different ways to exploit the knowing order of shipments in incoming trucks. We assume that a shipment is ready to transfer to outbound side as soon as it is unloaded successfully from an incoming truck.

3.5.1 Virtual Incoming Trucks

Inside each incoming truck $i \in \text{TR}^{\text{IN}}$, all the products which have to be transferred to other outgoing trucks are packed as a shipment. These shipment are put side by side inside each incoming truck. We assume that the unloading order of shipments in each incoming truck is Last In - First Out (LIFO). In order to take into account the information about the order of shipments inside each incoming truck, we introduce a new set VTR^{IN} which is the set of virtual incoming trucks. Each of these virtual incoming truck $j \in \text{VTR}^{\text{IN}}$ created from an incoming truck $i \in \text{TR}^{\text{IN}}$ will contain shipment that has to be transferred from incoming truck i to another outgoing truck $o \in \text{TR}^{\text{OUT}}$. Let $\alpha_{jj'}$ = 1 if the virtual incoming truck $j \in \text{VTR}^{\text{IN}}$ and the virtual incoming truck $j' \in \text{VTR}^{\text{IN}}$ are created from the same incoming truck $i \in \text{TR}^{\text{IN}}$ and the shipment in j is unloaded before the shipment in j' in i; 0 otherwise.

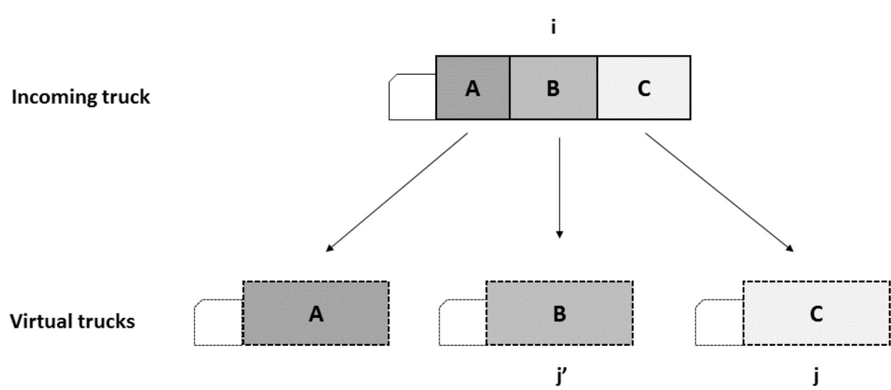


Figure 10: Virtual trucks generation.

In Figure 10, shipment C will be unloaded before shipment B and A and therefore $\alpha_{jj'}$ of the virtual truck j and j' will equal 1. Below is the model which applies virtual trucks extension:

$$\text{[LINEAR-ORDER]} \qquad \min \qquad L_{\max} \tag{32}$$

subject to:

$$L_{\max} \geq \sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} (t' + p_o) y_{og'}^{t'} - d_o \qquad o \in \text{TR}^{\text{OUT}} \tag{33}$$

$$\sum_{t \in T} \sum_{g \in G^{\text{IN}}} x_{jg}^{t} = 1 \qquad j \in \text{VTR}^{\text{IN}} \tag{34}$$

$$\sum_{j \in \text{VTR}^{\text{IN}}} \sum_{h=\max\{0, t-p_j+1\}}^{t} x_{jg}^{h} \leq 1 \qquad t \in T, g \in G^{\text{IN}} \tag{35}$$

$$\sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} y_{og'}^{t'} = 1 \qquad o \in \text{TR}^{\text{OUT}} \tag{36}$$

$$\sum_{o \in \text{TR}^{\text{OUT}}} \sum_{h=\max\{0, t'-p_o+1\}}^{t'} y_{og'}^{h} \leq 1 \qquad t' \in T, g' \in G^{\text{OUT}} \tag{37}$$

$$\sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} (t' - w_j \delta_{gg'}) y_{og'}^{t'} \geq \sum_{t \in T} (t + \boldsymbol{p_j}) x_{jg}^{t} - \bar{\delta}(1 - \sum_{t \in T} x_{jg}^{t}) \qquad o \in \text{TR}^{\text{OUT}}, j \in \text{TR}_o^{\text{IN}}, g \in G^{\text{IN}} \tag{38}$$

$$\sum_{t \in T} x_{jg}^{t} = \sum_{t \in T} x_{j'g}^{t} \qquad j, j' \in \text{VTR}^{\text{IN}} : \alpha_{jj'} = 1, g \in G^{\text{IN}} \tag{39}$$

$$\sum_{t \in T} \sum_{g \in G^{\text{IN}}} (t + p_j) x_{jg}^{t} = \sum_{t \in T} \sum_{g \in G^{\text{IN}}} t x_{j'g}^{t} \qquad j, j' \in \text{VTR}^{\text{IN}} : \alpha_{jj'} = 1 \tag{40}$$

$$x_{jg}^{t} \in \{0, 1\} \qquad j \in \text{VTR}^{\text{IN}}, g \in G^{\text{IN}}, t \in T \tag{41}$$

$$y_{og'}^{t'} \in \{0, 1\} \qquad o \in \text{TR}^{\text{OUT}}, g' \in G^{\text{OUT}}, t' \in T \tag{42}$$

$$L_{\max} \geq 0. \tag{43}$$

Constraints (33)-(43) are the same as in LINEAR, except two new constraints (39) and (40). Constraints (39) force all the virtual incoming trucks from the same incoming truck to be assigned to the same door. Constraints (40) guarantee that all the virtual incoming trucks created from one incoming truck have to be processed in the same unloading order of shipments contained in this incoming truck. In the constraints (38), since each virtual incoming truck now only contains product for one outgoing truck, we replace w_{io} by w_j which is the number of products in virtual incoming truck $j \in \text{VTR}^{\text{IN}}$. The precedent incoming trucks set TR_o^{IN} of outgoing truck o now will contains all the virtual incoming trucks $j \in \text{VTR}^{\text{IN}}$ that carry shipment for o.

3.5.2 Overestimation of the Starting Time of Outgoing Trucks

From the two previous models LINEAR and LINEAR-ORDER, we can observe that the constraints (28) over estimate the starting times to be processed of outgoing trucks in comparison to constraints (38) in the case where shipment is transferred to outbound side right after it is unloaded from incoming trucks. When we do not know the unloading order of shipments in incoming trucks, the safe way to guarantee that each outgoing truck $o \in \text{TR}^{\text{OUT}}$ has to wait until all of its shipments are unloaded is to delay o until all of its precedent incoming trucks unloaded successfully. However, there exists a case in which the shipment for o in its precedent incoming truck i is unloaded before the completion time time of i. In this way, we are over estimating the waiting time of each outgoing truck. If we can calculate the finish time of all shipments destined to o correctly, we can start loading o earlier. In order to accomplish this, we will have to exploit the unloading order of shipments in each incoming truck. Let $\bar{u}_{io}$ be the cumulative unloading time of the shipment from incoming truck i to outgoing truck o. $\bar{u}_{io}$ has to be the summation of the time to unload shipment $(i - o)$ and the waiting time to unload all the previous shipments in i.

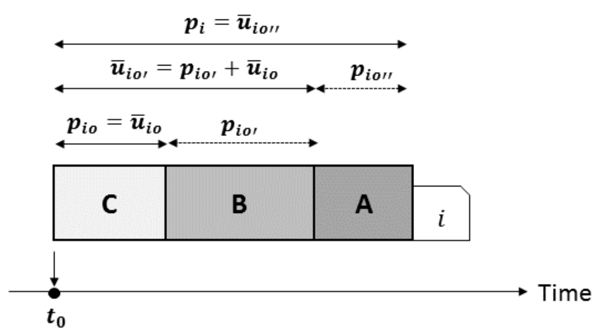


Figure 11: Cumulative unloading time calculation.

In Figure 11, incoming truck i starts unloading at time t_0. Shipment C, B and A belongs to outgoing truck o, o' and o'', respectively. p_{io}, $p_{io'}$ and $p_{io''}$ are the processing times, i.e., the time to unload shipment C, B and A respectively. Shipment C will be unloaded first, then B, and followed by A. Let the processing time of C be p_{io} then it equals $\bar{u}_{io}$. Since shipment B has to wait until shipment C is unloaded successfully, the cumulative unloading time of B will be $\bar{u}_{io'} = \bar{u}_{io} + p_{io'}$. The same calculation is applied to shipment A. Because shipment A is unloaded last, its cumulative unloading time has to equal to the processing time of incoming truck i. The model is formulated as follows:

$$[\text{LINEAR-ORDER}^{+}] \qquad \min \qquad L_{\max} \tag{44}$$

subject to:

$$L_{\max} \geq \sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} (t' + p_o) y_{og'}^{t'} - d_o \qquad o \in \text{TR}^{\text{OUT}} \tag{45}$$

$$\sum_{t \in T} \sum_{g \in G^{\text{IN}}} x_{ig}^{t} = 1 \qquad i \in \text{TR}^{\text{IN}} \tag{46}$$

$$\sum_{i \in \text{TR}^{\text{IN}}} \sum_{h=\max\{0, t-p_i+1\}}^{t} x_{ig}^{h} \leq 1 \qquad t \in T, g \in G^{\text{IN}} \tag{47}$$

$$\sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} y_{og'}^{t'} = 1 \qquad o \in \text{TR}^{\text{OUT}} \tag{48}$$

$$\sum_{o \in \text{TR}^{\text{OUT}}} \sum_{h=\max\{0, t'-p_o+1\}}^{t'} y_{og'}^{h} \leq 1 \qquad t' \in T, g' \in G^{\text{OUT}} \tag{49}$$

$$\sum_{t' \in T} \sum_{g' \in G^{\text{OUT}}} (t' - w_{io}\delta_{gg'}) y_{og'}^{t'} \geq \sum_{t \in T} (t + \bar{\boldsymbol{u}}_{\boldsymbol{io}}) x_{ig}^{t} - \bar{\delta}(1 - \sum_{t \in T} x_{ig}^{t}) \qquad o \in \text{TR}^{\text{OUT}}, i \in \text{TR}_{o}^{\text{IN}}, g \in G^{\text{IN}} \tag{50}$$

$$x_{ig}^{t} \in \{0, 1\} \qquad i \in \text{TR}^{\text{IN}}, g \in G^{\text{IN}}, t \in T \tag{51}$$

$$y_{og'}^{t'} \in \{0, 1\} \qquad o \in \text{TR}^{\text{OUT}}, g' \in G^{\text{OUT}}, t' \in T \tag{52}$$

$$L_{\max} \geq 0. \tag{53}$$

In constraints (50), instead of putting the total processing time of incoming truck i, we replace it by the unloading time of shipment of o in i.

3.6 Experimental Results

We next test the performance of all three newly proposed models. All the models are implemented in Java using the concert technology library of CPLEX 12.6. We first describe the data generation of data sets, and then some performance and quality measurements used in the next tables. After that, we present the computational results with their analysis on the known shipment unloading order.

The number of papers which study the problem of scheduling inbound and outbound sides with transshipment time is very limited. Beside, we cannot able to find other models that have the same inputs and objective function to be comparable with ours. Therefore, in the following tables, we only report the performance of models in this paper to evaluate the instance sizes which these models can able to solve within the time limit.

3.6.1 Data Generation

We use data sets which are generated the same way in Molavi et al. 2018 and Belle et al. 2013.

Problem Parameters	Notations	Value
Number of trucks in each side	$\lvert \mathrm{TR}^{\mathrm{IN}} \rvert, \lvert \mathrm{TR}^{\mathrm{OUT}} \rvert$	4 - 9
Number of dock doors in each side	$\lvert G^{\mathrm{IN}} \rvert, \lvert G^{\mathrm{OUT}} \rvert$	2 - 5
Transshipment time	$\delta_{gg'}$	$U[1,3]$
Percentage of the number of outgoing trucks for which each inbound truck contains	Flow Mix	25% - 50%
Time to unload/load one product	L	2

Table 12: Factors of data generation of D1

*$U[a,b]$ means the uniform distribution of $[a,b]$.

However, this two papers consider the arrival time in their models and generate the departure times base on these arrival times. Therefore we use generation method for departure times or due dates of outgoing trucks of Boysen et al. 2013:

$$d_o = 2 \cdot \frac{\sum\limits_{i \in \mathrm{TR}^{\mathrm{IN}}} p_i}{\lvert G^{\mathrm{IN}} \rvert} \cdot U[0.5, 0.9].$$

In the above formula, we multiply the due date by 2 because we consider the transportation time between dock doors (which has mean value equal the time to unload each product in Table 12). We use a set of small-scale instances, the following table summarizes the instances we use to do experiments.

Test Instances	$\lvert \mathrm{TR}^{\mathrm{IN}} \rvert$	$\lvert \mathrm{TR}^{\mathrm{OUT}} \rvert$	$\lvert G^{\mathrm{IN}} \rvert$	$\lvert G^{\mathrm{OUT}} \rvert$
S01	4	4	2	2
S02	4	4	3	3
S03	4	5	3	3
S04	7	6	3	3
S05	7	8	4	4
S06	8	8	4	4
S07	8	9	4	5
S08	9	9	5	5

Table 13: Small-scale instances

3.6.2 Performance Measurements

In this section, we describe some measurements which are used in the following tables to compare the performances and the quality of solutions of each models presented in this article. These measurements are described as follows:

- %LP: The percentage linear relaxation gap. It is calculated as $\%LP = \frac{|OPT-LP|}{OPT}100$, where OPT is the optimal value for the mixed integer linear programming (MILP) and LP is the optimal value for the linear programming relaxation of each model. If an optimal MILP solution is not obtained within the time limit, we use the best known solution for the tested instance.

- B&B: The number of nodes explored in the branch-and-bound tree.

- CPU (seconds): The computational time in second to solve the MILP model.

- %Improvement: The percentage improvement in lateness value gained when we consider the order of shipments.

- Lateness: The optimal value of the objective function in each model. It is the maximum lateness of outgoing trucks. When a model cannot return an optimal solution within the time limit, we ignore the Lateness and %Improvement values in the subsequent tables.

3.6.3 Computational Results

We now test the performances of three proposed models: LINEAR, LINEAR-ORDER and LINEAR-ORDER$^+$. In all models, we use a length of one time unit for the time periods. Since the computational time of the exact algorithm rises exponentially, we set the time limit of 1,800 seconds for solving time of CPLEX. If a model cannot get the optimal solution within the time limit, we will get the best integer solution that model can reach.

Instance		LINEAR			LINEAR-ORDER			LINEAR-ORDER$^+$		
		%LP	B&B	CPU (seconds)	%LP	B&B	CPU (seconds)	%LP	B&B	CPU (seconds)
S01	Best	62.96	0	0.57	62.96	0	3.32	62.96	0	0.61
	Worst	100	70489	14.09	100	168580	191.29	100	25315	7.18
	Average	91.51	30584	6.59	91.11	92260	46.74	91.06	5959	3.25
	Std	14.21	24547	4.33	14.66	114792	59.86	14.72	8058	1.69
S02	Best	57.51	0	2.32	57.51	6507	8.94	57.51	0	2.43
	Worst	100	8198	6.89	100	36866	22.08	100	163	5.04
	Average	86.76	14796	6.74	86.73	10630	10.91	86.73	981	3.46
	Std	16.22	38402	9.69	16.53	12903	6.04	16.53	2317	1.80
S03	Best	86.21	0	1.43	86.24	408	9.87	86.24	0	4.58
	Worst	100	317555	51.44	100	656284	1800	100	57158	18.52
	Average	96.20	38309	9.95	98.33	95018	207.89	98.33	16651	6.99
	Std	5.95	99940	15.17	4.34	204821	560.73	4.34	29567	6.55
S04	Best	100	31705	26.76	100	236967	1800	100	13679	19.65
	Worst	100	780650	1800	100	429084	1800	100	793511	1800
	Average	100	287757	431.64	100	275158	1800	100	184107	431.77
	Std	0	346614	723.69	0	120465	0	0	236805	724.92
S05	Best	100	12898	33.77	100	92999	1800	100	9164	29.63
	Worst	100	756798	1800	100	98636	1800	100	438253	1800
	Average	100	351346	759.77	100	165399	1800	100	202793	414.13
	Std	0	302056	803	0	62574	0	0	180583	582.44

Table 14: Comparing performance among three models: LINEAR, LINEAR-ORDER and LINEAR-ORDER$^+$

Instance		LINEAR			LINEAR-ORDER			LINEAR-ORDER$^+$		
		%LP	B&B	CPU (seconds)	%LP	B&B	CPU (seconds)	%LP	B&B	CPU (seconds)
S06	Best	100	109570	134.13	100	46926	1800	100	81414	147.67
	Worst	100	536562	1800	100	0	1800	100	522572	1800
	Average	100	446215	1185.86	100	67719	1800	100	492181	932.29
	Std	0	279552	799.09	0	47090	0	0	496224	789.88
S07	Best	100	32352	103.57	100	64319	1800	100	40145	87.13
	Worst	100	310415	1800	100	91892	1800	100	175894	1800
	Average	100	127056	457.71	100	75866	1800	100	109072	382.04
	Std	0	95670	541.69	0	44567	0	0	97138	523.85
S08	Best	100	102303	278.53	100	2839	1800	100	107286	332.48
	Worst	100	133778	1800	100	0	1800	100	215693	1800
	Average	100	347966	1102.72	100	10347	1800	100	299747	1103.87
	Std	0	265966	749.80	0	18402	0	0	149764	735.03

Table 14: (Continued)

Table 14 compares among the three formulations using 8 sets of problem from S01 to S08. For each set of instances, we generate 10 replications and report the Best, Worst, Average, Standard deviation values based on the LP percentage gap, CPU times and the number of branch-and-bound nodes. For the first three set of problem, the %LP of three models do not have much difference. In terms of CPU time, LINEAR-ORDER takes more time to get the optimal solution than LINEAR and LINEAR-ORDER$^+$. The number of branch-and-bound nodes of LINEAR-ORDER$^+$ is less than the one for the other two models. For the remaining data instances, all three models have relative weak linear relaxation gaps. While LINEAR-ORDER and LINEAR-ORDER$^+$ are able to get the optimal solution within the time limit, the model LINEAR-ORDER exceed the time limit for all data instances. This behavior can be predictable because of introduction of virtual incoming trucks, which make the problem more complex.

Problem	LINEAR	LINEAR-ORDER		LINEAR-ORDER$^+$	
	Lateness (time unit, average)	Lateness (time unit, average)	%Improvement (average)	Lateness (time unit, average)	%Improvement (average)
S01	47.00	38.10	22.40	38.10	22.40
S02	64.00	59.70	7.66	59.70	7.66
S03	52.90	44.20	16.69	44.20	16.69
S04	45.88	–	–	33.63	25.48
S05	38.57	–	–	28.14	26.73
S06	45.50	–	–	31.25	32.46
S07	33.67	–	–	22.00	37.87
S08	52.40	–	–	39.60	24.97

Table 15: Comparing maximum lateness between unknown and known order of shipments models.

Table 15 shows the average values of lateness and the improvement percentage in lateness when we take into account the order of shipments in incoming trucks. From Table 15 we can observe that LINEAR-ORDER and LINEAR-ORDER$^+$ help reduce from 20% to 35% in average the time of lateness in comparison with the case of unknown shipment orders. This results are consisted with the analysis we model in the previous section when we claim that the knowledge of shipment unloading order help us to be able to start loading outgoing trucks earlier.

In Table 16, we do experiment on the relationship between the flow mix percentages, the percentages of outgoing trucks that each incoming truck has shipment for, and the improvement we can get when considering shipment unloading order. When the flow mix percentages increase, the number of precedent incoming trucks of each outgoing truck increase or it has to wait more incoming trucks to be unloaded successfully. The results show that the more precedent incoming trucks which each outgoing truck has, the more improvement in lateness we can get. In another words, when the number of precedent incoming trucks of each outgoing truck increases, the over estimation of starting time of outgoing trucks in model LINEAR is worse.

Flow mix (%)	%Improvement							
	S01	S02	S03	S04	S05	S06	S07	S08
25%	0	0	11.52	0	12.74	13.68	13.21	19.26
50%	15.05	11.78	9.38	23.41	21.62	34.46	42.73	25.97
75%	25.23	17.61	20.95	28.93	32.31	28.58	41.85	27.07
100%	31.85	25.16	21.45	33.11	32.37	31.56	45.55	29.86

Table 16: Changes in average percentage improvement of lateness between LINEAR and LINEAR-ORDER$^+$ when the flow mix percentage increases.

3.7 Conclusion

In this paper, we introduce three mixed integer programming models (LINEAR, LINEAR-ORDER and LINEAR-ORDER$^+$) for cross-docking scheduling for inbound side and outbound side problem. We start by formulating a time-indexed model with linearized constraints to tackle the issue of non-linearized when we take into account the transportation time between dock doors. After this, we propose two different ways to integrate the shipments unloading order into our models, namely, virtual incoming trucks and the unloading time of each shipment. The computational results show that applying the shipment unloading order help us to reduce the maximum lateness up to around 35%.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

In this thesis, we study two problems of scheduling trucks at cross-dock facilities. While in Chapter 2, we schedule the inbound side with fixed outbound side scheduling and departure time to minimize the total number of tardy products, Chapter 3 is devoted to the problem of scheduling inbound and outbound side to minimize the maximum lateness of outgoing trucks.

In Chapter 2, our work is initially based on the network based model for truck scheduling with fixed outbound departure from Boysen et al. 2013. From this model, we first introduce a time-indexed model version and then propose an enhanced preprocessing method to make the model more efficient. After that, we investigate two extensions: multiple transfer trips and shipment unloading order. According to an observation that a shipment may need more than one trip to be transfer from inbound to outbound side, we modify the exist models to consider the multiple transfer trip to transfer products of a shipment. In the second extension, we exploit the information about the order of shipments in each incoming trucks by introducing new parameter called cumulative unloading time of each shipment. The cumulative unloading time help us to calculate precisely the time that each shipment is ready to be transfer to outbound side. Through out the experimental results, we show that the preprocessing make our models more scalable, that is they can solve larger size problem and out-performance the model in Boysen et al. 2013. The computational results also show that applying the multiple transfer trip and shipment unloading order help us to calculate more precise the number of tardy product and then to have better truck scheduling.

Chapter 3 cover a more complex problem where we schedule both inbound and outbound side to minimize the maximum lateness of outgoing trucks. In this work, we consider the transportation times between dock doors. However, the differences of transportation times lead to an issue of non-linear in the precedence constraints of outgoing trucks, in which each outgoing truck has to wait until all of its products unloaded and transferred successfully. We address this problem by proposing new linearize constraints. In addition, we study two method to applying the shipment unloading order in the exist model. The first approach is generating virtual trucks from each incoming trucks. The second approach is using the same concept of cumulative unloading time in Chapter 2 to re-formulate the model. The computational results show that the second approach is more scalable than the first approach and to using the unloading order of shipment, we can reduce the maximum lateness of outgoing trucks significantly.

4.2 Future Work

The models in Chapter 2 and Chapter 3 are solved by an optimizer (CPLEX) and will give us exact optimal solution. However, we are unable to solve large size problem with hundred of trucks by this method. In order to solve this real-life problems, other approaches such as heuristics or column generation techniques. are more favorable to approximate or compute the optimal solution while they still satisfy the constraints of time and size of data instances.

Bibliography

Stalk, G., P. Evans, and L. E. Shulman (1992). "Competing on capabilities: the new rules of corporate strategy". In: *Harvard business review* 70, pp. 57–69.

Apte, U. M. and S. Viswanathan (2000). "Effective cross docking for improving distribution efficiencies". In: *International journal of logistics research and applications* 3.3, pp. 291–302.

Van Belle, J., P. Valckenaers, and D. Cattrysse (2012). "Cross-docking: state of the art". In: *Omega* 40.6, pp. 827–846.

Boysen, N., D. Briskorn, and M. Tschöke (2013). "Truck scheduling in cross-docking terminals with fixed outbound departures". In: *Or spectrum* 35.2, pp. 479–504.

McWilliams, D. L., P. M. Stanfield, and C. D. Geiger (2005). "The parcel hub scheduling problem: a simulation-based solution approach". In: *Computers & industrial engineering* 49.3, pp. 393–412.

— (2008). "Minimizing the completion time of the transfer operations in a central parcel consolidation terminal with unequal-batch-size inbound trailers". In: *Computers & industrial engineering* 54.4, pp. 709–720.

McWilliams, D. L. (2009). "Genetic-based scheduling to solve the parcel hub scheduling problem". In: *Computers & industrial engineering* 56.4, pp. 1607–1616.

McWilliams, D. (2010). "Iterative improvement to solve the parcel hub scheduling problem". In: *Computers & industrial engineering* 59.1, pp. 136–144.

Rosales, C., M. Fry, and R. Radhakrishnan (2009). "Transfreight reduces costs and balances workload at georgetown crossdock". In: *Interfaces* 39.4, pp. 316–328.

Liao, T., P. Egbelu, and P. Chang (2013). "Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations". In: *International journal of production economics* 141.1, pp. 212–229.

Tootkaleh, S., S. Ghomi, and M. Sajadieh (2016). “Cross dock scheduling with fixed outbound trucks departure times under substitution condition”. In: *Computers & industrial engineering* 92, pp. 50–56.

Boysen, N., S. Fedtke, and F. Weidinger (2017). “Truck scheduling in the postal service industry”. In: *Transportation science* 51.2, pp. 723–736.

Lim, A., H. Ma, and Z. Miao (2006). “Truck dock assignment problem with time windows and capacity constraint in transshipment network through crossdocks”. In: *Computational science and its applications - iccsa 2006*. Ed. by M. Gavrilova et al. Springer Berlin Heidelberg, pp. 688–697.

Miao, Z., A. Lim, and H. Ma (2009a). “Truck dock assignment problem with operational time constraint within crossdocks”. In: *European journal of operational research* 192.1, pp. 105–115.

Boysen, N. (2010). “Truck scheduling at zero-inventory cross docking terminals”. In: *Computers & operations research* 37.1, pp. 32–41.

Boysen, N., M. Fliedner, and A. Scholl (2010a). “Scheduling inbound and outbound trucks at cross docking terminals”. In: *Or spectrum* 32.1, pp. 135–161.

Chen, F. and K. Song (2009a). “Minimizing makespan in two-stage hybrid cross docking scheduling problem”. In: *Computers & operations research* 36.6, pp. 2066–2073.

Cota, P. et al. (2016). “Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre”. In: *Computers & industrial engineering* 95, pp. 135–143.

Bellanger, A., S. Hanafi, and C. Wilbaut (2013). “Three-stage hybrid-flowshop model for cross-docking”. In: *Computers & operations research* 40.4, pp. 1109–1121.

Arabani, A. B., S. F. Ghomi, and M. Zandieh (2011). “Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage”. In: *Expert systems with applications* 38.3, pp. 1964–1979.

Ghobadian, E. et al. (2012). “Scheduling trucks in cross docking systems with temporary storage and dock repeat truck holding pattern using grasp method”. In: *International journal of industrial engineering computations* 3.

Ruiz, R. and T. Stützle (2007). “A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem”. In: *European journal of operational research* 177.3, pp. 2033–2049.

Vahdani, B. and M. Zandieh (2010). “Scheduling trucks in cross-docking systems: robust meta-heuristics”. In: *Computers & industrial engineering* 58.1, pp. 12–24.

Bodnar, P., R. de Koster, and K. Azadeh (2017). “Scheduling trucks in a cross-dock with mixed service mode dock doors”. In: *Transportation science* 51.1, pp. 112–131.

Ye, Y. et al. (2016). “Scatter search for truck scheduling problem with product loading/unloading constraints in a crossdocking system”. In: *Proceedings of the 22nd international conference on industrial engineering and engineering management 2015*, pp. 335–345.

Agustina, D., C. Lee, and R. Piplani (2010). “A review: mathematical models for cross docking planning”. In: *International journal of engineering business management* 2.2, pp. 47–54.

Forger, G. (1995). “Ups starts world’s premiere cross-docking operation”. In: *Modern material handling* 36.8, pp. 36–38.

Kinnear, E. (1997). “Is there any magic in cross-docking?” In: *Supply chain management: an international journal* 2.2, pp. 49–52.

Witt, C. (1998). “Crossdocking: concepts demand choice”. In: *Material handling engineering* 53.7, pp. 44–49.

Napolitano, M. (2011). “Cross dock fuels growth at dots”. In: *Logistics management* 50.2, pp. 30–34.

Bartholdi, J. and K. Gue (2004). “The best shape for a crossdock”. In: *Transportation science* 38.2, pp. 235–244.

Ross, A. and V. Jayaraman (2008). “An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design”. In: *Computers & industrial engineering* 55.1, pp. 64–79.

Miao, Z. et al. (2012). “Transshipment service through crossdocks with both soft and hard time windows”. In: *Annals of operations research* 192.1, pp. 21–47.

Guignard, M. et al. (2012). “Algorithms for the crossdock door assignment problem”. In: *Proceedings of the fourth international workshop on model-based metaheuristics*. Brazil.

Nassief, W., I. Contreras, and R. As’ ad (2016). “A mixed-integer programming formulation and lagrangean relaxation for the cross-dock door assignment problem”. In: *International journal of production research* 54.2, pp. 494–508.

Nassief, W., I. Contreras, and B. Jaumard (2018a). "A comparison of formulations and relaxations for cross-dock door assignment problems". In: *Computers & operations research* 94, pp. 76–88.

Enderer, F., C. Contardo, and I. Contreras (2017). "Integrating dock-door assignment and vehicle routing with cross-docking". In: *Computers & operations research* 88, pp. 30–43.

Nassief, W. et al. (2018b). "The container scheduling and cross dock-door selection problem". In: *Submitted to the journal of the operational research society.*

Boysen, N. and M. Fliedner (2010b). "Cross dock scheduling: classification, literature review and research agenda". In: *Omega* 38.6, pp. 413–422.

Ladier, A. and G. Alpan (2016). "Cross-docking operations: current research versus industry practice". In: *Omega* 62, pp. 145–162.

Stephan, K. and N. Boysen (2011). "Cross-docking". In: *Journal of management control* 22.1, p. 129.

Buijs, P., I. Vis, and H. Carlo (2014). "Synchronization in cross-docking networks: a research classification and framework". In: *European journal of operational research* 239.3, pp. 593–608.

Molavi, D., A. Shahmardan, and M. Sajadieh (2018). "Truck scheduling in a cross docking systems with fixed due dates and shipment sorting". In: *Computers & industrial engineering* 117, pp. 29–40.

Diglio, A., A. Genovese, and C. Piccolo (2017). "An optimization model for the outbound truck scheduling problem at cross-docking platforms". In: *International conference on optimization and decision science*. Springer, pp. 601–610.

Schwerdfeger, S., N. Boysen, and D. Briskorn (2017). "Just-in-time logistics for far-distant suppliers: scheduling truck departures from an intermediate cross-docking terminal". In: *Or spectrum*, pp. 1–21.

Chen, Z. and W. Powell (1999). "Solving parallel machine scheduling problems by column generation". In: *Informs journal on computing* 11.1, pp. 78–94.

Van Den Akker, J., J. Hoogeveen, and S. van de Velde (1999). "Parallel machine scheduling by column generation". In: *Operations research* 47.6, pp. 862–872.

Van Den Akker, J., J. Hoogeveen, and J. van Kempen (2012). "Using column generation to solve parallel machine scheduling problems with minmax objective functions". In: *Journal of scheduling* 15.6, pp. 801–810.

M'Hallah, R. and R. Bulfin (2005). "Minimizing the weighted number of tardy jobs on parallel processors". In: *European journal of operational research* 160.2, pp. 471–484.

M'Hallah, R. and T. Al-Khamis (2015). "A benders decomposition approach to the weighted number of tardy jobs scheduling problem on unrelated parallel machines with production costs". In: *International journal of production research* 53.19, pp. 5977–5987.

Belle, J. V. et al. (2013). "A tabu search approach to the truck scheduling problem with multiple docks and time windows". In: *Computers & industrial engineering* 66.4, pp. 818–826.

Chen, F. and C.-Y. Lee (2009b). "Minimizing the makespan in a two-machine cross-docking flow shop problem". In: *European journal of operational research* 193.1, pp. 59–72.

Fonseca, G. B., T. H. Nogueira, and M. G. Ravetti (2017). "A hybrid Lagrangean metaheuristic for the two-machine cross-docking flow shop scheduling problem". In: *Arxiv e-prints*.

Miao, Z., A. Lim, and H. Ma (2009b). "Truck dock assignment problem with operational time constraint within crossdocks". In: *European journal of operational research* 192.1, pp. 105–115.