Resource Planning in Cross-Docking Platforms – Models and Algorithms

Zur Erlangung des Grades eines Doktors der Politikwissenschaften (Dr. rer. pol.) im Rahmen des Doppel-Promotions-Programms der Technischen Universität Darmstadt und der Tongji University
Genehmigte Dissertation von Pascal Wolff aus Kirchheimbolanden

1. Gutachten: Prof. Dr. Dr. h.c. mult. Hans-Christian Pfohl
2. Gutachten: Prof. Dr. Jiazhen Huo
3. Gutachten: Prof. Dr. Christoph Glock

Darmstadt, 2021

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Rechts- und Wirtschaftswissenschaften
Supply Chain- und Netzwerkmanagement

Resource Planning in Cross-Docking Platforms – Models and Algorithms

Accepted doctoral thesis by Pascal Wolff

1. Review: Prof. Dr. Dr. h.c. mult. Hans-Christian Pfohl
2. Review: Prof. Dr. Jiazhen Huo
3. Review: Prof. Dr. Christoph Glock

Date of submission: September 23, 2020
Date of thesis defense: March 22, 2021

Darmstadt, 2021

Parts of this dissertation work were published in the following peer-reviewed journal article:

Wolff, P., Emde, S.* & Pfohl, H.-Chr. (2021): "Internal resource requirements: The better performance metric fur truck scheduling?", *Omega*, DOI: https://doi.org/10.1016/j.omega.2021.102431.

* External advisor who provided helpful ideas and feedback regarding the conceptualization, methodology, and writing.

URN: urn:nbn:de:tuda-tuprints-185010
URL: https://tuprints.ulb.tu-darmstadt.de/id/eprint/18501

To my grandmother Charlotte, in loving memory.

Danksagung

Die vorliegende Arbeit enstand im Rahmen des Doppelpromotionsprogramms der Technischen Universität Darmstadt und der Tongji Universität Shanghai. Durch die Teilnahme am Doppelpromotionsprogramm konnte ich zwei recht unterschiedliche Forschungsumfelder kennenlernen. Ich blicke insgesamt auf eine sehr spannende und lehrreiche Zeit mit vielen tollen Erinnerungen zurück, die ich auf keinen Fall missen möchte. An dieser Stelle möchte ich gerne einer Reihe von Personen danken die mich in den vergangenen Jahren unterstützt haben.

Zuerst möchte ich meinen akademischen Betreuern Prof. Dr. Dr. h.c. mult. Hans-Christian Pfohl und Prof. Dr. Jiazhen Huo für ihr großes Vertrauen in meine Arbeit herzlich danken. Sie haben mich in fachlicher und auch in persönlicher Hinsicht stets gefördert und mir während meiner Zeit an der Tongji Universität Shanghai die Möglichkeit gegeben, sowohl Lehrerfahrung in diversen Master- und MBA-Programmen als auch Projekterfahrung zu sammeln. Mein ausdrücklicher Dank gilt zudem Prof. Dr. Christoph Glock für die Übernahme des Zweitgutachtens, den weiteren Teilnehmerinnen und Teilnehmern der Prüfungskommission, Prof. Dr. Michael Neugart, Prof. Dr. Ralf Elbert, Prof. Dr. Yongrui Duan und Jun.-Prof. Dr. Eric Grosse, sowie Dr. Giorgi Tadumadze für die Unterstützung bei der Organisation der Disputation. Bei Prof. Dr. Simon Emde, den ich während eines Doktorandenworkshops der European Logistics Association (ELA) in Neapel kennengelernt habe, bedanke ich mich für die konstruktiven Diskussionen zu meinem Dissertationsvorhaben und die anschließende Unterstützung bei einer gemeinsamen Veröffentlichung.

Während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Bosch-Lehrstuhl für Global Supply Chain Management des Chinesisch-Deutschen Hochschulkollegs (CDHK) an der Tongji Universität Shanghai hatte ich zudem das große Glück, mit einer Vielzahl exzellenter Kolleginnen und Kollegen zusammenarbeiten zu dürfen. Mein Dank gilt hier insbesondere Dr. Johannes Kern, Dr. Nikos Moraitakis, Dr. Weidi Wang und Limeng Chai (Ph.D.), die ich als gute Freundinnen und Freunde gewinnen konnte. Besonders danke ich Johannes, der die zugrundeliegenden Modellprämissen und deren praktische Tauglichkeit stets kritisch hinterfragt hat. Meinen "Vorgängern"

Nikos und Weidi danke ich für die Unterstützung in meiner Anfangszeit an der Tongji Universität Shanghai. Durch ihre Hilfe konnte ich mich schnell in der chinesischen Forschungslandschaft einleben. Allen Masterandinnen, Masteranden und studentischen Hilfskräften des Lehrstuhls möchte ich an dieser Stelle ebenfalls für den unermüdlichen Einsatz danken. Des Weiteren danke ich dem Deutschen Akademischen Austauschdienst (DAAD) und dem Chinese Scholarship Council (CSC) für die finanzielle Unterstützung meines Forschungsaufenthaltes an der Tongji Universität Shanghai.

Abschließend möchte ich einen besonderen Dank meiner Familie und meiner Freundin Meichen Pan zuteilwerden lassen. Allen voran meinen Eltern, Jürgen und Roswitha Wolff, die mich in jeder Hinsicht bedingungslos unterstützt haben. Ihr seid die Besten!

Pascal Wolff

Ningbo, 5. April 2021

Abstract

Cross-docking, a relatively new warehouse strategy that has its roots in the industry, can improve the efficiency of a company's logistics and distribution processes. Specifically, it can minimize the costly storage and order picking function of traditional warehouses by efficiently coordinating (i.e., synchronizing) incoming freight flows and outgoing freight flows. Companies from various industries such as the retailing industry, the less-than-truckload logistics service industry, the express and small parcel delivery industry, and the automotive industry operate cross-docking terminals in their transportation networks and benefit from improved service levels, reduced transportation costs, reduced inventory holding costs, reduced handling costs, etc.

Besides its practical relevance, cross-docking has also received a lot of academic attention in the last 30 years. Many academic studies have addressed a wide range of strategic, tactical, and operational cross-docking decision problems. Most studies, however, have neglected resource planning aspects and hence failed to address two major concerns of cross-docking practitioners:

- Determining the number of resources needed;
- Scheduling internal resources in an efficient way.

This thesis sets out to bridge this theory-practice gap in the cross-docking domain by proposing new models that combine two interdependent operational problems faced by cross-docking practitioners, namely the scheduling of internal resources and the scheduling of trucks.

Three novel problems are introduced in this thesis. First, the resource and truck scheduling problem, denoted as **TSFD-RC-F**, is proposed. It allows scheduling both resources and trucks when the resource requirements of trucks are given and known in advance. The **TSFD-RC-F** aims to determine a truck schedule that can be executed with a minimum number of resources. Then, the multi-mode resource and truck scheduling problem (**TSFD-RC-V**) is proposed. It is a model extension of the **TSFD-RC-F** and offers the additional flexibility of adapting the number of resources for processing trucks. While deploying more operators accelerates truck processing,

deploying fewer operators prolongs the processing time. The model aims to determine how many resources should be deployed for truck processing and at what time trucks should be serviced in order to minimize the maximum number of required resources. Lastly, the shift and truck scheduling problem (**ISTSFD**) is proposed. It considers different operator types (e.g., temporary and regular workers), shift patterns, and work breaks. The **ISTSFD** seeks to find a truck schedule and employee timetable with minimum labor costs. Two variants of the **ISTSFD** are presented: a single-mode problem (**ISTSFD-F**) and a multi-mode problem (**ISTSFD-V**).

As the proposed models' complexity statuses make it challenging to solve large-sized instances with a default solver, tailored column generation-based solution procedures for all three problems are developed.

Extensive computational experiments are conducted in order to assess the computational performance of both the mixed-integer programs and the proposed solution procedures. In addition, managerial insights are derived by benchmarking the proposed models against frequently used truck scheduling models. It is shown that the proposed discrete-time MIP formulations clearly outperform the proposed continuous-time MIP formulations in terms of both solution quality and computational time. Moreover, the solution time can be reduced by using the proposed preprocessing parameters for calculating the number of delayed freight units and compelling the service level. While a default solver can solve the discrete-time MIPs for small and medium-sized instances in a reasonable time, it often fails to provide good solutions for very large problem instances with a fine time granularity. The proposed heuristics solution procedures, on the other hand, can provide high-quality solutions for very large problem instances in a short time and clearly outperform commercial solvers. In addition, the following key take-home managerial insights could be derived:

- By using the internal resource requirements instead of the frequently used makespan or processing time as the primary performance metrics, the cross-docking platform's operational efficiency can be significantly increased.
- By integrating the decision of how many resources should be deployed for truck processing (i.e., considering multi-mode processing), further operational efficiency gains can be realized.
- The defined service level has a significant impact on the operator demand. Lowering the required service level can be a reasonable means to improve a cross-docking facility's operational efficiency further.
- The work break patterns have a significant impact on the operator requirements. Too low

a number of work break patterns may result in a strong surge in operator demand.

Zusammenfassung

Cross-Docking ist eine besondere Form des Warenumschlags. Im Gegensatz zur traditionellen Lagerhaltung zielt das Cross-Docking auf einen bestandslosen Warenumschlag ab. Die beim Cross-Docking angestrebte zeitliche und mengenmäßige Koordination von ankommenden und ausgehenden Warenlieferung ermöglicht unter anderem eine Reduzierung der Lagerhaltungs- und Kommissionierkosten. Zudem können im Idealfall kürzere Durchlaufzeiten und eine bessere Auslastung von Transportkapazitäten realisiert werden. Cross-Docking-Zentren haben sich in der Praxis vielfach bewährt. Sie sind z. B. elementarer Bestandteil in Distributionsnetzwerken von Groß- und Einzelhandelsunternehmen, Kurier-, Express-, und Paketdienstleistern, Automobilunternehmen sowie Transportdienstleistern.

Cross-Docking und die damit einhergehenden strategischen, taktischen und operativen Planungsprobleme wurden auch in einer Vielzahl von wissenschaftlichen Beiträgen untersucht. Bei der Analyse der Literatur lässt sich überwiegend jedoch die Vernachlässigung von Ressourcen- und Personalbedarfen zur Durchführung der internen Transport- und Kommissionierprozesse konstatieren. Die wissenschaftliche Literatur hat es bis heute weitestgehend versäumt, Entscheidungsträger bei der Ermittlung von Personal- und Ressourcenbedarfen in Cross-Docking-Zentren zu unterstützen.

Die vorliegende Arbeit leistet einen Beitrag zur Schließung dieser Forschungslücke, indem neue operative Planungsansätze entwickelt werden, die explizit die Personal- und Ressourcenbedarfe berücksichtigen.

Hierfür wird zunächst ein Basismodell zur integrierten Ressourcen- und Torbelegungsplanung (**TSFD-RC-F**) in Cross-Docking-Zentren entwickelt. Das **TSFD-RC-F** zielt auf die Ermittlung eines Torbelegungsplans ab, der mit einer minimalen Anzahl an Ressourcen ausgeführt werden kann. Dabei wird die Annahme getroffen, dass der Ressourcenbedarf zur Bearbeitung eines jeden LKWs bekannt ist. In der anschließend entwickelten Modellerweiterung (**TSFD-RC-V**) wird diese Annahme des Basismodells verworfen. Das **TSFD-RC-V** trifft hingegen die Annahme, dass

die Ressourcenanzahl zur Bearbeitung von LKWs durch den Entscheidungsträger variiert werden kann. Das **TSFD-RC-V** ermittelt demnach für jeden LKW den optimalen Ressourceneinsatz. Des Weiteren wird ein mathematisches Modell zur integrierten Schicht- und Torbelegungsplanung (**ISTSFD**) entwickelt. Im **ISTSFD** können verschiedene Personalarten und Schichtmuster berücksichtigt werden.

Die im Rahmen dieser Arbeit entwickelten gemischt-ganzzahligen Optimierungsmodelle sind nachweislich NP-schwer. Es kann deshalb nicht garantiert werden, dass große Probleminstanzen mit Hilfe von Standardsolvern (z. B. CPLEX oder Gurobi) gelöst werden können. Zur Lösung großer Probleminstanzen werden deshalb Spaltengenerierungsverfahren entwickelt.

Die Eignung der vorgestellten Modelle und Lösungsverfahren wird durch umfangreiche Tests bewertet. Es wird beispielsweise gezeigt, dass die zeitdiskreten Modellformulierungen den zeitkontinuierlichen Modellformulierungen im Hinblick auf die Lösungszeit und Lösungsqualität überlegen sind. Mit Hilfe von Standardsolvern können kleine und mittelgroße Probleminstanzen effizient gelöst werden. Bei der Lösung großer Probleminstanzen stoßen kommerzielle Standardsolver allerdings oftmals an ihre Grenzen. Die entwickelten heuristischen Lösungsverfahren sind hingegen in der Lage sehr gute – oftmals sogar optimale – Lösungen für große Probleminstanzen zu ermitteln. Des Weiteren werden durch die numerischen Tests eine Vielzahl von betriebswirtschaftlichen Erkenntnissen gewonnen, z. B.:

- Verglichen mit herkömmlichen Modellen zur Torbelegungsplanung generiert das Basismodell zur integrierten Ressourcen- und Torbelegungsplanung (**TSFD-RC-F**) Arbeitspläne die sich durch einen deutlich geringeren Personalbedarf auszeichnen.

- Der im **TSFD-RC-V** integrierte zusätzliche Freiheitsgrad zur Auswahl des optimalen Ressourceneinsatzes für jeden LKW ermöglicht weitere Effizienzsteigerungen von ca. 15%.

- Der Personalbedarf wird maßgeblich durch das vorgegebene Lieferserviceniveau beeinflusst. Bereits geringfügige Senkungen des Lieferserviceniveaus können zu signifikanten Personaleinsparungen führen.

- Die Pausenregelungen haben einen erheblichen Einfluss auf den untertägigen Personalbedarf. Gestaffelte Mittagspausen sind gemeinsamen Mittagspausen vorzuziehen, da letztgenannte mit einem erhöhten Mitarbeiterbedarf einhergehen können.

Abstract

越库作业是一种源于产业、较为新颖的仓库策略，可以提高公司物流和配送流程的效率。具体地说，它可以通过有效地协调(即同步)进站货物流和出站货物流来最大限度地减少传统仓库高昂的储存费率，并提高仓库的订单分拣能力。来自各种行业如零售业、零担货运物流服务行业,快递和小包裹递送行业和汽车行业的公司，在它们的运输体系中运行着越库终端，并且受益于改善的服务水平,减少运输成本,减少库存持有成本,降低处理成本等。

在过去的30年里，越库除了具有现实意义外，也受到了学术界的广泛关注。许多学术研究已经解决了广泛的战略性、策略性和运营性的越库决策问题。然而，大多数研究都忽视了资源规划方面的问题，因此未能解决越库实践者的两个主要关注点:

- 确定所需资源的数量;
- 高效调度内部资源。

本文提出了一个新的模型，将越库实践者所面临的两个相互依赖的操作问题，即内部资源的调度和车辆的调度结合起来，以弥合越库领域的理论与实践的鸿沟。

本文介绍了三个新问题。首先，提出资源与车辆调度问题，记为TSFD-RC-F。在卡车资源要求已知时，它允许同时调度资源和卡车。TSFD-RC-F的目标是确定一个用最少资源执行的卡车调度。然后，提出了多模式资源与货车调度问题(TSFD-RC-V)。它是TSFD-RC-F的一个模型扩展，并提供了调节处理卡车的资源数量的额外灵活性。配置更多的操作者可以加快卡车处理速度，配置更少的操作者会延长处理时间。该模型的目的是确定应该为卡车处理配置多少资源，以及应该在什么时间对卡车进行服务，以最小化所需资源的最大数量。最后，提出了轮班和车辆调度问题(ISTSFD)。它考虑了不同的操作者类型(例如，临时和正式工人)、轮班模式和工作休息时间。ISTSFD寻求找到最低的劳动力成本的卡车时间表和员工时间表。它提出了ISTSFD的两种变体，即单模问题(ISTSFD- F)和多模问题(ISTSFD- V)。

由于所提出模型的复杂性状态使得使用默认求解器难以求解大型实例，因此针对这三个问题开发了基于列生成的定制求解过程。

此外，为了评估混合整数程序和所提出的解决程序的计算性能，进行了大量的计算实验。管理洞见是通过对所提出的模型与经常使用的卡车调度模型进行基准测试而得到的。

结果表明，在求解质量和计算时间上，所提出的离散MIP公式明显优于连续MIP公式。此外，利用所提出的预处理参数计算延迟货运单元数和强制服务水平可以缩短求解时间。虽然默认求解器可以在合理的时间内解决小型和中型实例的离散时间MIPs，但它通常无法为具有良好时间粒度的大型问题实例提供良好的解决方案。另一方面，提出的启发式解决程序可以在短时间内为非常大的问题实例提供高质量的解决方案，并且明显优于商业解决方案。此外，可以得出以下关键管理洞见:

- 通过使用内部资源需求而不是经常使用的最大完工时间或处理时间作为主要性能指标，可以显著越库平台的运行效率。
- 通过整合卡车处理需要配置多少资源的决策(即考虑多模式处理)，可以进一步提高运营效率。
- 定义的服务水平对运营商的需求有重大影响。降低所需的服务水平是进一步提高越库运行效率的合理手段。
- 工作休息模式对操作人员的要求有重大影响。过低的工作休息模式可能会导致操作者需求的强劲增长。

Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

AGV	Automated guided vehicle
AS/RS	Automated storage and retrieval system
BIP	Binary integer programs
CG	Column generation
CT	Continuous-time model formulation
DT	Discrete-time model formulation
d-R(ISTSFD-V-DT2)	Dual of the **R(ISTSFD-V-DT2)**
d-R(TSFD-RC-F-DT2)	Dual of the **R(TSFD-RC-F-DT2)**
d-R(TSFD-RC-V-DT2)	Dual of the **R(TSFD-RC-V-DT2)**
GRASP	Greedy randomized adaptive search procedure
h	Hours
ISTSFD	Shift and truck scheduling problem
ISTSFD-F	Single-mode shift and truck scheduling problem
ISTSFD-F-DT1	First discrete-time model formulation for the single-mode shift and truck scheduling problem
ISTSFD-F-DT2	Second discrete-time model formulation for the single-mode shift and truck scheduling problem
ISTSFD-U-V	MIP for identifying initial columns for the multi-mode shift and truck scheduling problem
ISTSFD-V	Multi-mode shift and truck scheduling problem
ISTSFD-V-CG	Column generation-based solution procedure for the multi-mode shift and truck scheduling problem
ISTSFD-V-DT1	First discrete-time model formulation for the multi-mode shift and truck scheduling problem
ISTSFD-V-DT2	Second discrete-time model formulation for the multi-mode shift and truck scheduling problem
ISTSFD-V-PP-S(g)	The shift pricing subproblem for operator group g and the multi-mode shift and truck scheduling problem

KPI	Key performance indicator
LP	Linear programming
LTL	Less-than-truckload
min	Minutes
MIP	Mixed-integer program
NP	Non-deterministic polynomial time
OEM	Original equipment manufacturer
PMSP	Parallel machine scheduling problem
RCPSP	Resource-constrained project scheduling problem
RIP	Resource investment problem
R(ISTSFD-V-DT2)	Restricted version of the **ISTSFD-V-DT2**
R(TSFD-RC-F-DT2)	Restricted version of the **TSFD-RC-F-DT2**
R(TSFD-RC-V-DT2)	Restricted version of the **TSFD-RC-V-DT2**
RO	Research objective
SMSP-TTW	Single machine scheduling problem to minimize tardy task weights
SP	Shift planning model
TSFD-MS-F	Single-mode truck scheduling model for minimizing the makespan
TSFD-MS-F/SP	Sequential approach consisting of the single-mode truck scheduling model for minimizing the makespan (**TSFD-MS-F**) and the shift planning model (**SP**)
TSFD-PT-F	Single-mode truck scheduling model for minimizing the total processing time
TSFD-PT-F/SP	Sequential approach consisting of the single-mode truck scheduling model for minimizing the total processing time (**TSFD-PT-F**) and the shift planning model (**SP**)
TSFD-RC-F	Resource and truck scheduling problem
TSFD-RC-F-CG	Column generation-based solution procedure for the resource and truck scheduling problem
TSFD-RC-F-CT1	First continuous-time model formulation for the resource and truck scheduling problem
TSFD-RC-F-CT2	Second continuous-time model formulation for the resource and truck scheduling problem
TSFD-RC-F-DT1	First discrete-time model formulation for the resource and truck scheduling problem

TSFD-RC-F-DT2	Second discrete-time model formulation for the resource and truck scheduling problem
TSFD-RC-F-PP(i,d)	Pricing subproblem for the truck-door pair (i, d) and the resource and truck scheduling problem
TSFD-RC-V	Multi-mode resource and truck scheduling problem
TSFD-RC-V-CG	Column generation-based solution procedure for the multi-mode resource and truck scheduling problem
TSFD-RC-V-CT1	First continuous-time model formulation for the multi-mode resource and truck scheduling problem
TSFD-RC-V-CT2	Second continuous-time model formulation for the multi-mode resource and truck scheduling problem
TSFD-RC-V-DT1	First discrete-time model formulation for the multi-mode resource and truck scheduling problem
TSFD-RC-V-DT2	Second discrete-time model formulation for the multi-mode resource and truck scheduling problem
TSFD-RC-V-PP(i,d)	The pricing subproblem for the truck-door pair (i, d) and the multi-mode resource and truck scheduling problem
TSFD-U-F	MIP for identifying initial columns for the resource and truck scheduling problem
TSFD-U-V	MIP for identifying initial columns for the multi-mode resource and truck scheduling problem

1 Introduction

1.1 Research background and objective

Cross-docking is a relatively new warehouse strategy that can improve the efficiency of a company's logistics and distribution processes. The basic idea of cross-docking is to transfer incoming cargo directly to outgoing trailers. Through this synchronization of in- and outbound flows, the costly storing and order picking functions of traditional warehouses can be minimized[1]. Since the cargo usually spends less than 24h in the cross-docking terminal, reduced inventory holding costs, required storage space, and handling costs, as well as faster inventory turnover, can be realized[2]. Companies from various industries such as retailing[3], express and small parcel delivery[4], automotive industry[5], and less-than-truckload (LTL) logistics service industry[6] operate cross-docking terminals in their distributions networks.

Besides its increasing practical relevance, cross-docking also received a lot of academic attention, especially in the last two decades. Scholars have studied manifold strategic (e.g., location and design of a cross-dock), tactical (e.g., material flow through cross-docking distribution networks), and operational (e.g., assigning trucks to dock-doors, determining times at which trucks are processed) decision problems related to cross-docking terminals. A vast number of publications have dealt with the so-called **truck scheduling problem**[7]. It aims to compute feasible and appropriate truck schedules by determining **where**, i.e., at which dock-door, and

[1] Vahdani and Zandieh (2010, p. 12), Wen et al. (2009, p. 1708).

[2] Apte and Viswanathan (2000, p. 292), Cook et al. (2005, p. 55).

[3] E.g., Wal-Mart (Stalk et al., 1992).

[4] E.g., DHL (Boysen et al., 2013) and UPS (Forger, 1995).

[5] E.g., Renault (Serrano et al., 2017) and Toyota (Witt, 1998).

[6] Gue (1999).

[7] Ladier and Alpan use the term "truck-to-door scheduling problem", cf. Ladier and Alpan (2016).

when trucks should be processed.

When comparing truck scheduling in academia and industry, it can be seen that ongoing research is detached from industry practice. LADIER AND ALPAN recently observed that academic research on truck scheduling often incorporates unrealistic assumptions[1]. For instance, the majority of reviewed truck scheduling models assume that an infinite number of internal resources, such as workers and material handling equipment, is available. This assumption clearly does not reflect reality, where both workforce and material handling equipment are usually limited. By neglecting resource scarcity, these models are not suitable to address two major concerns of cross-docking practitioners: (i) determining the number of resources needed, and (ii) scheduling the resources in an efficient way[2]. In addition, a discrepancy in terms of applied performance measures can be observed between truck scheduling in practice and theory. Practitioners usually strive for efficient truck schedules, i.e., plans which deploy a minimum number of resources and finish the workload "in time". Therefore, practitioners usually use performance measures directly related to the resource requirements and utilization to steer cross-docking operations. These metrics, however, are rarely used by academic researchers. Instead, the **makespan**, defined as the time that elapses from the start of the first operation until the completion of the last operation, followed by the **travel distance**, defined as the total distance traveled by the cargo inside the facility, are the two most frequently used performance metrics in truck scheduling models[3]. It is often argued that minimizing the travel distance results in a minimum workload and ultimately minimizes the working time since it requires a shorter time for a worker to complete the task[4]. The following simplified example, however, shows that neither of the two metrics necessarily leads to a truck schedule that deploys a minimum amount of resources. For the sake of simplicity, the total processing time is used as a surrogate objective function[5] for the total travel distance.

Example 1: Consider a situation with five inbound trucks. The trucks' arrival and departure times are given in Table 1.1 and must not be violated. It is further assumed that loading operations of outbound trucks start at 10:00. In order to ensure a smooth loading process, all inbound trucks must be processed by 10:00. A total of three dock-doors can be used for unloading operations. Travel distance differences between them are reflected by door-dependent

[1] Ladier and Alpan (2016).

[2] Ladier and Alpan (2016, p. 156).

[3] Ladier and Alpan (2016, p. 158).

[4] Chmielewski et al. (2009, p. 202).

[5] A surrogate objective is defined as "a function that is correlated to the true objective, but is less computationally demanding", cf. Gendreau and Potvin (2019, p. 47).

processing times. Unloading an inbound truck requires exactly one worker equipped with a forklift (denoted as an **operator** in the following).

			Truck processing times at different dock-doors		
Truck	Arrival time	Departure time	$d = 1$	$d = 2$	$d = 3$
1	08:00	09:00	30	35	35
2	08:20	09:20	40	45	50
3	08:20	09:20	60	60	50
4	08:50	09:50	35	30	35
5	09:15	10:15	30	25	30

Processing times in minutes.

Table 1.1 Truck information for the example.
Source: Own table.

Based on the truck information for the example, Figure 1.1 shows truck schedules for various performance measures. When aiming to minimize the makespan, it can be observed from Figure 1.1a that unloading operations start relatively late and finish early, that is, before 10:00. This leads to a compact truck schedule and expedites parallel processing on all available dock-doors. Therefore, three operators are necessary for executing the plan. When applying the total travel distance (or total processing time) as the objective (or surrogate objective) in the example, each truck is assigned to the dock-door with the shortest processing time. Different from the makespan minimization, the whole 2h horizon is used for unloading operations. Figure 1.1b shows that for most of the time, two operators are sufficient to handle the workload. Due to the 5 minutes overlap of trucks 2, 3, and 4 from 09:05-09:10, however, three operators must be deployed to execute the plan. Lastly, Figure 1.1c shows a feasible truck schedule that can be executed with a minimum number of operators. The workload is distributed evenly over the planning horizon in order to avoid large peak workloads. It requires at most two operators at a time to handle the workload, which is ca. 33% less than when applying the makespan or travel distance as the key performance measure. Since the "manpower is very often the first cost center of a logistic platform where the operations are done manually"[1)], this decrease helps to reduce the operational costs and increase the efficiency significantly. The example suggests that when using the makespan or travel distance as the performance metrics in a truck scheduling model, the generated plan is not inevitably efficient. The most frequently used performance measures, hence, seem to fail to support cross-docking practitioners in efficiently scheduling the internal resources.

1) Ladier and Alpan (2016, p. 147).

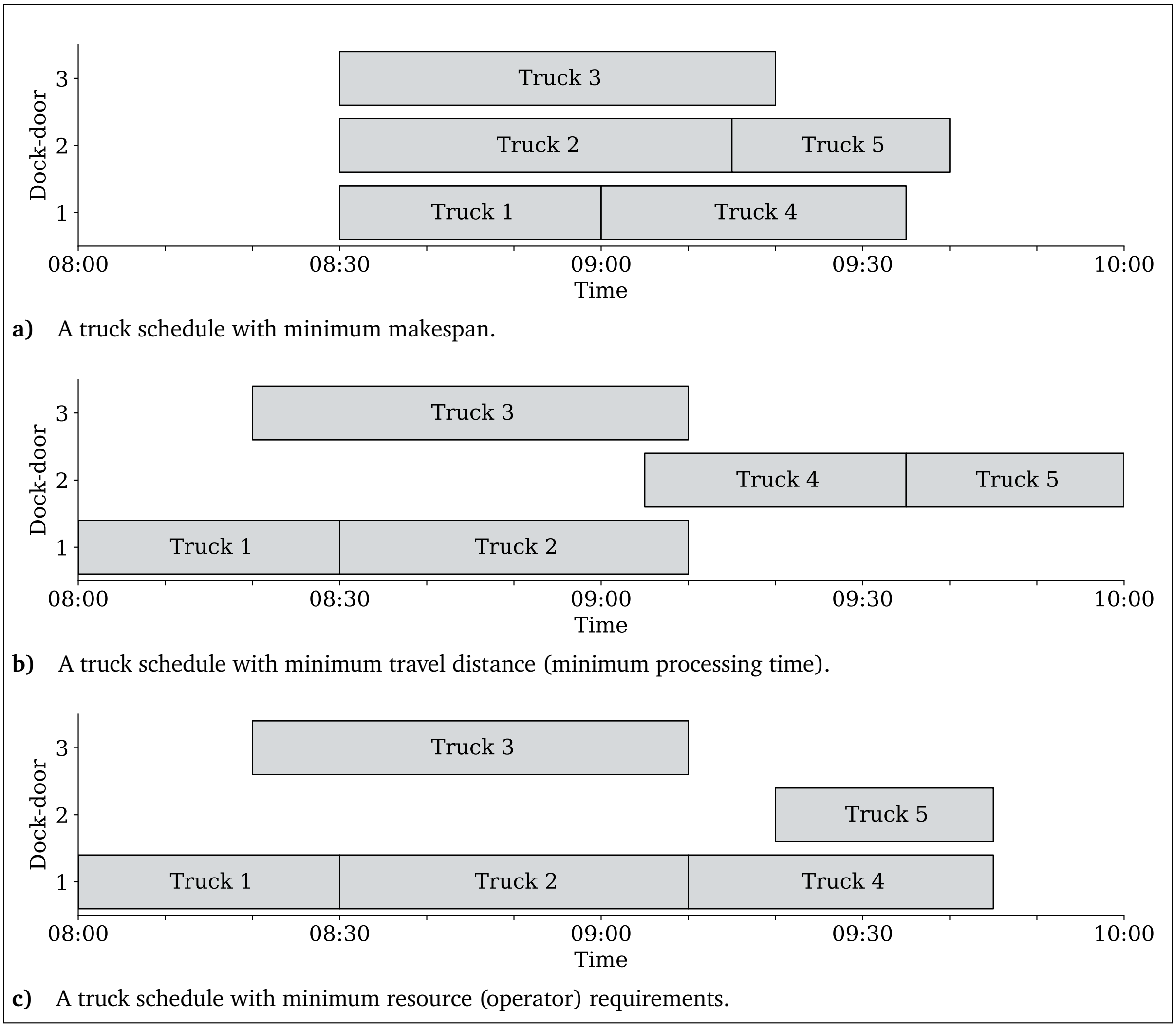


Figure 1.1 Comparison of truck schedules for different objective functions.
Source: Own figure.

This thesis sets out to bridge the identified theory-practice gap in the cross-docking operations planning domain. Its overall research objective can be summarized as follows:

> *Develop novel planning tools that allow cross-docking managers to allocate and schedule internal resources more efficiently.*

More specifically, this study proposes new models that combine two interdependent operational problems faced by cross-docking managers, namely the scheduling of internal resources and the scheduling of trucks. Van Belle et al. also identified the integration of both planning problems as an important research task, which may improve cross-docking operations[1)].

The course of the study will be outlined in the next section.

1.2 Thesis structure

This section describes the general structure of this thesis. It is also illustrated in Figure 1.2.

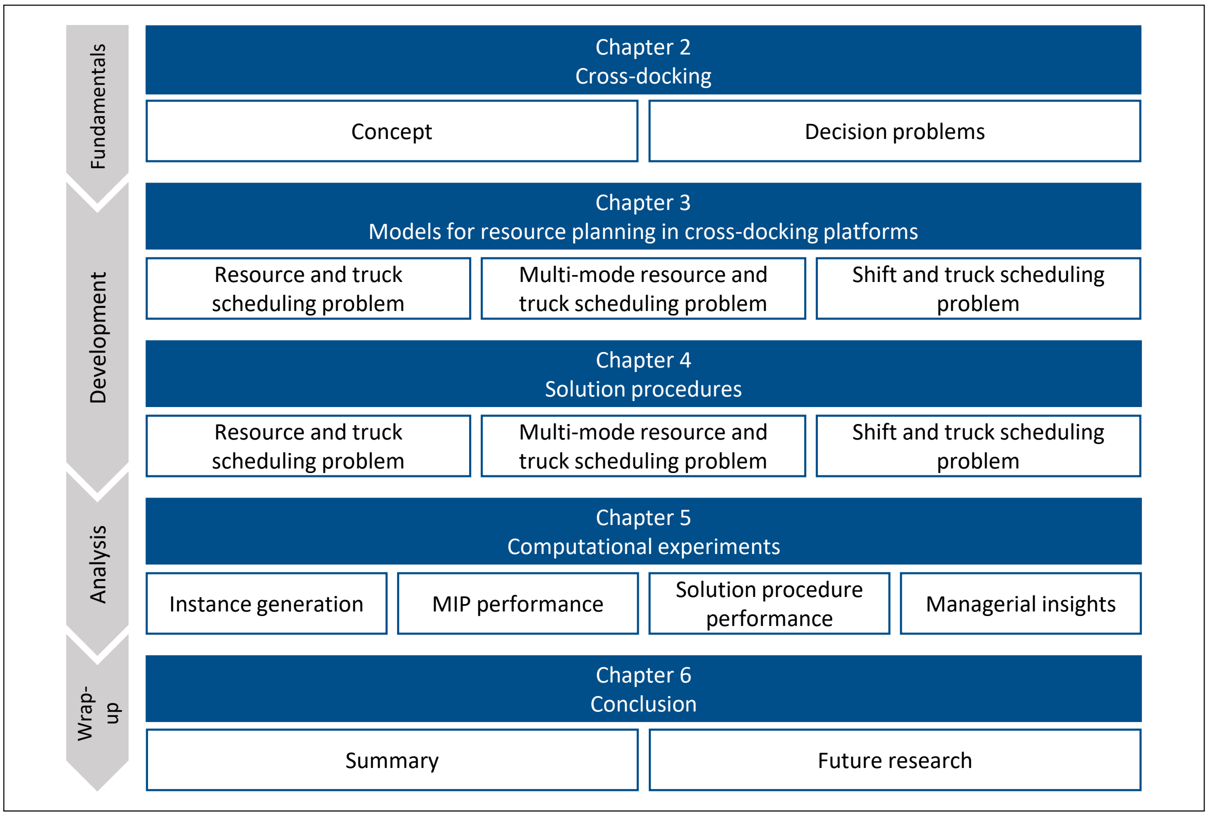


Figure 1.2 Study outline.
Source: Own figure.

This introduction is followed by Chapter 2, establishing the research context and setting of the thesis. Based on the explanation of the cross-docking concept and its practical relevance, different types of cross-docking facilities are described. Furthermore, the literature on the major decision problems faced by managers of cross-docking terminals is summarized.

1) Van Belle et al. (2012, p. 844).

Chapter 3 presents different mathematical models which aim to support decision-makers in scheduling resources in cross-docking facilities. First, a basic model for scheduling resources and trucks in cross-docking platforms is proposed. The model can help to identify a truck schedule which requires a minimum number of resources. Subsequently, two extensions of the basic model are presented. The first extension, which considers variable resource requirements for truck processing, can determine how many resources should be deployed for truck processing and at what time each truck should be processed in order to minimize the total number of required resources. Lastly, a model for integrated shift and truck scheduling is described. It can distinguish between different operator types (e.g., regular employees and temporary employees) and can be used to develop shift plans for workers and timetables for truck processing.

Unfortunately, the complexity status of the proposed models may hinder large-sized instances from being solvable to optimality by simply feeding the mixed-integer programs (MIPs) into an off-the-shelf solver. Therefore, Chapter 4 presents heuristic solution procedures that can be used to tackle instances of real-world size.

Chapter 5 contains the computational experiments. First, the performance of the different mixed-integer programming formulations from Chapter 3 is benchmarked. Then, the performance of the heuristic solution procedures is evaluated. In addition to exploring the computational performance of the mathematical programs and solution procedures, the chapter also includes a large numerical study for deriving managerial insights.

Finally, in Chapter 6, the results of this whole thesis are summarized, and an outlook on future research opportunities is presented.

2 Cross-docking

2.1 Concept

Manufacturers of consumer goods, which often produce their products in big batches, usually prefer to ship their products in full truckloads to benefit from economies of scale in transportation. Retailers, however, usually need considerably smaller quantities of a single product. In order to overcome this problem, consumer goods can be transshipped through a storage location, e.g., a warehouse. This allows manufacturers to push the production quantities to the storage location and to realize economies of scale in transportation, while the retailers can pull the quantities they really need from the storage location. This approach, however, may result in high inventory levels and hence high inventory costs at the storage location.[1] Cross-docking, a distribution system that has its roots in the industry, can help to reduce the inventory levels at the storage location by synchronizing (full truckload) inbound flows and (less-than-truckload) outbound flows. When implemented correctly, the costly storing and order picking functions of traditional warehouse locations can be significantly reduced.[2] Figure 2.1 depicts the schematic representation of a cross-docking terminal with three inbound trucks and four outbound trucks.

Wal-mart is a famous example of how cross-docking may help to increase a company's profit. Stalk et al., who analyzed Wal-Mart's success, noticed that Wal-Mart ran ca. 85% of its goods through cross-docking terminals in the early 1990s. By doing so, Wal-Mart was able to reduce the costs of sales by ca. 2-3% compared to the industry average. This, in turn, contributed to Wal-Mart's success, which became the highest profit retailer at that time.[3] Another example for the successful transition from a traditional stockholding supply chain

[1] Ladier (2014, p. 6).

[2] Vahdani et al. (2010, p. 12), Wen et al. (2009, p. 1708).

[3] Stalk et al. (1992, p. 58), Ladier and Alpan (2016, p. 146).

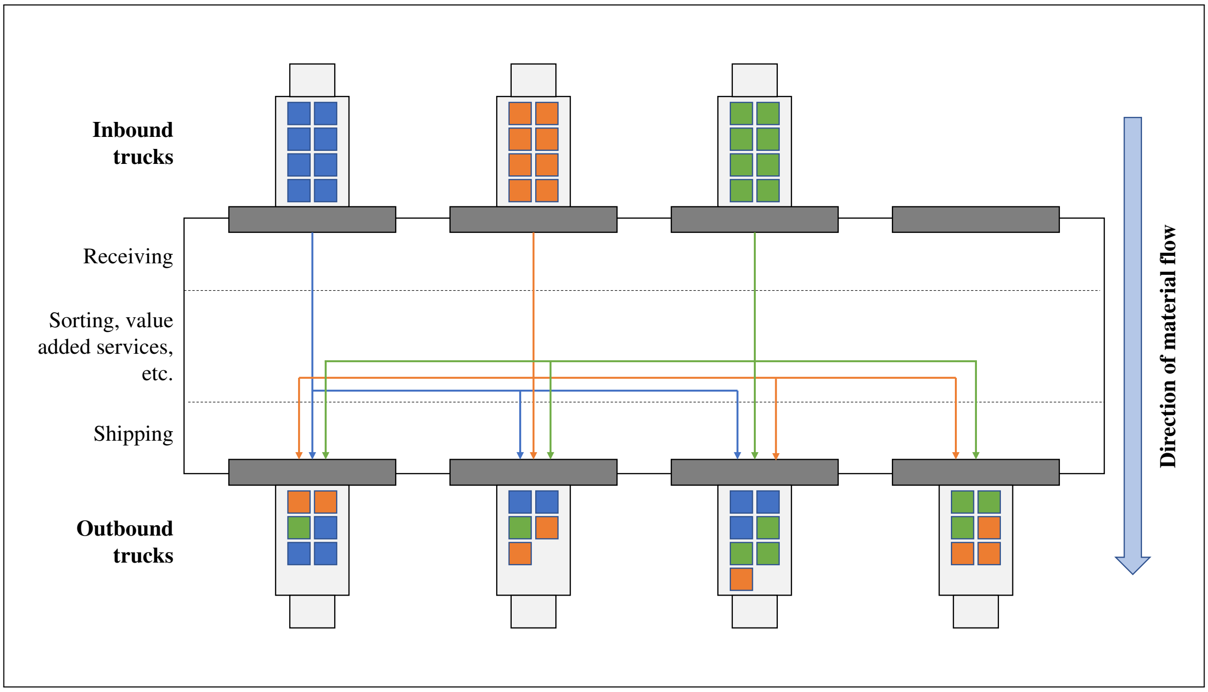


Figure 2.1 Schematic representation of a cross-docking terminal.
Source: Own figure after Boysen and Fliedner (2010, p. 414).

operation to a cross-docking system is Goodyear GB Ltd. By implementing a cross-docking system, Goodyear achieved an inventory reduction of 16%, released more than 12,500 square meters of warehousing space, improved the next day delivery in the United Kingdom from 87% to 96%, and reduced the operating costs by more than 12%[1). Many of these benefits were also identified in a survey conducted among 219 practitioners with a logistics and supply chain management background. Table 2.1 reports the major benefits of cross-docking from the respondents' perspective. According to the survey respondents, an improved service level and reduced transportation costs are the two most important benefits of cross-docking.

1) Kinnear (1997, pp. 51-52).

Description	% of respondents
Improved service level	19.4%
Reduced transportation costs	14.3%
Consolidated shipments to destination	13.1%
Get Products to market more quickly	10.2%
Reduced need for warehouse space	8.5%
Improved inventory management	8.0%
Savings from reduced inventory carrying costs	5.7%
Increased demand for just-in-time service	4.5%
Shipments/consignee customization	4.0%
Reduced labor costs	4.0%
Other	8.3%

Table 2.1 Main benefits of cross-docking.
Source: Own table after Saddle Creek Corporation (2011, p. 5).

According to Van Belle et al., cross-docking has the following advantages compared to traditional distribution centers:[1)]

- Reduced warehousing, inventory holding, handling, and labor costs;
- Shorter delivery lead times from suppliers to customers;
- Improved customer service;
- Reduced storage space;
- Faster turnover of the inventory;
- Fewer overstocks;
- Reduced risk for damage and loss.

In light of all the potential benefits that can be achieved through cross-docking, it is not surprising that cross-docking terminals can be found in many supply chains. In the survey conducted by Saddle Creek Corporation, 68.5% of the respondents already used cross-docking, and another 15.1% of the respondents had plans to use cross-docking within the next two years[2)]. Stephan and Boysen propose a classification scheme to structure the diversity of cross-docking applications that can be found in practice. Figure 2.2 depicts the classification scheme.

1) Van Belle et al. (2012, p. 828).

2) Saddle Creek Corporation (2011, p. 4), Rijal et al. (2019, p. 752).

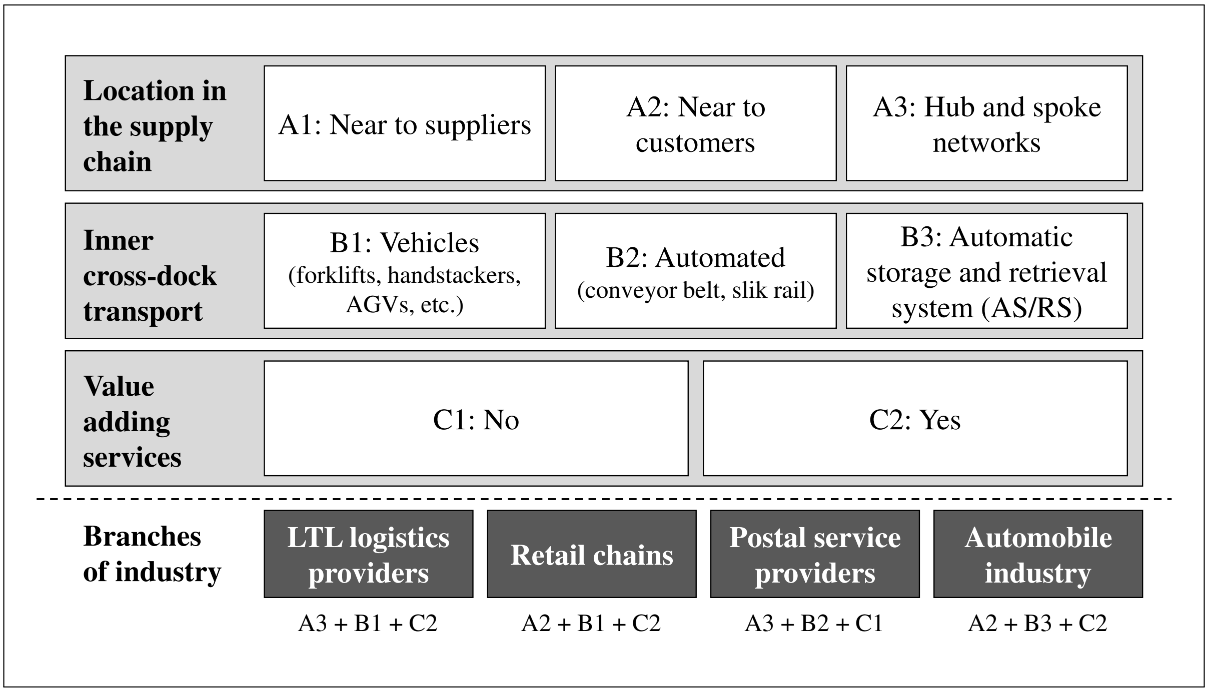


Figure 2.2 Classification of cross-docking terminals and typical cross-docking settings in different industries.
Source: Own figure after Stephan and Boysen (2011, p. 131).

The scheme uses the following three attributes to distinguish cross-docking platforms:

- Location of the cross-docking platform in the supply chain;
- Mode of product movement within the cross-docking platform;
- Value adding services within the cross-docking platform.

Based on that, Stephan and Boysen identify four important cross-docking settings in different industries:[1)]

- **LTL logistics providers:** LTL logistics providers transport shipments between 30 kilograms and 2 tons for many shippers and recipients. They are usually dependent on an efficient freight consolidation in the hub-and-spoke network of multiple cross-docking terminals. Inside the cross-docking terminals, forklifts are usually used to move (palletized) cargo. LTL logistics providers often offer various value-added services such as relabeling or final assembly processes.

1) Stephan and Boysen (2011, p. 132).

- **Retail chains:** In retail supply chains, a cross-docking terminal often serves multiple retail stores. Hence, deliveries received by large suppliers can be broken down at the cross-docking terminal according to the actual demand of the store. Sometimes value adding services such as price labeling services can be found.

- **Postal service providers:** Postal service providers consolidate freight units of up to 30 kilograms within a multi-stage hub-and-spoke system. The relatively small freight units, which do not require any additional services, can be moved through the cross-docking facility with a conveyor belt system.

- **Automobile industry:** Original equipment manufacturers (OEMs) in the automotive industry store parts delivered by remote suppliers in cross-docking terminals located close to multiple final assembly lines. The parts are usually of small size and temporarily stored in an automated storage and retrieval system (AS/RS) until the exact demand for parts is known. Oftentimes, value adding services such as arranging the parts in bins just-in-sequence are provided.

Other classifications to structure the wide range of cross-docking applications can be found in the literature. For instance, Van Belle et al. distinguish cross-docking terminals based on **physical characteristics** (shape of the cross-docking platform, number of dock-doors, and mode of internal transportation), **operational characteristics** (service mode and pre-emption), and **flow characteristics** (arrival and departure patterns of cargo, product interchangeability, and temporary storage).[1]

2.2 Decision problems

In order to successfully implement a cross-docking system, multiple interdependent decision problems need to be solved[2]. A vast number of academic studies have addressed a wide range of cross-docking related problems in the last three decades. Moreover, various surveys and classifications of the existing literature have been proposed. Van Belle et al., for instance, provide a broad overview of the cross-docking literature. They use a rather general classification and categorize the literature based on the problem type into strategic decisions (e.g., location of cross-docks in the distribution network and layout of cross-docks), tactical decisions (e.g.,

1) Van Belle et al. (2012, pp. 831-832).

2) Stephan and Boysen (2011, p. 132).

material flows in distribution networks with cross-docking terminals), and operational decisions (e.g., dock-door assignment, truck scheduling, and size of temporary storage locations)[1]. Buijs et al. provide another recent overview of the cross-docking literature. The authors identify 24 individual decision problems, cluster them based on their decision-making level (strategic, tactical, and operational problems) and problem scope (whether the problem originates locally at the cross-docking terminal or elsewhere in the cross-docking network), and identify interdependencies between the individual problem classes[2]. Other surveys have a tighter scope. Ladier and Alpan review the literature on the most common operational decision problems in cross-docking facilities, namely the problem of assigning trucks to dock-doors and the problem of scheduling/sequencing trucks[3]. The scheduling of trucks is a common problem at many platforms and has received a lot of academic attention. Boysen and Fliedner provide an in-depth survey on the truck scheduling problem and classify the literature based on the **door environment** (number of dock-doors and service mode), **operational characteristics** (e.g., processing times, arrival times, deadlines, availability of intermediate storage space, interchangeability of products, etc.), and the **objective function**.

This section sets out to provide a brief overview of the existing literature dealing with decision problems in the life cycle of a cross-docking terminal. It is structured into strategic, tactical, and operational planning problems. Decision problems related to the first two areas are briefly described. As the study at hand aims to propose new models for short-term resource planning, the operational planning problems will be reviewed in more detail.

2.2.1 Strategic decisions

On a strategic level, cross-docking practitioners are mainly confronted with two major decisions:

1. Determine the location of a cross-docking terminal in the supply or distribution network;
2. Determine the layout of the cross-docking terminal.

The majority of studies that aim to identify the optimal locations for cross-docking terminals optimize the freight flows through this network of facilities simultaneously[4]. Many scholars have

[1] Van Belle et al. (2012, p. 832).
[2] Buijs et al. (2014, p. 595).
[3] Ladier and Alpan (2016).
[4] Buijs et al. (2014, p. 599).

studied this so-called location-allocation problem over the past three decades[1]. Bhaskaran was among the first authors who investigated the problem. The paper presents a heuristic approach which can be used to determine the appropriate number and location of the cross-docking terminals. Moreover, a case study from General Motors is presented.[2]

A few studies have dealt with the design of a cross-docking terminal. The facility design, which has a significant impact on the operations, aims to enable a fast transshipment of cargo and to ensure sufficient capacity. In this context, the physical characteristics of the cross-docking terminal, such as the facility shape, number of dock-doors, the size and layout of the temporary storage area, and internal transportation mode are fixed. Bartholdi and Gue, for instance, investigate the optimal shape of a cross-docking terminal for different settings[3]. They consider various cross-docking shapes, including I-, L-, T-, H-, and X-shape. According to their study, the optimal shape depends on the size of the facility and the freight flow patterns inside the facility. For smaller facilities, an I-shaped terminal is beneficial, whereas, for facilities with ca. 150 dock-doors, a T-shaped terminal is beneficial. If more than 200 dock-doors are required, the authors recommend an X-shaped facility.[4] For rectangular-shaped cross-docking facilities, Carlo and Bozer show that if a cross-docking terminal's perimeter (i.e., the number of dock-doors) is fixed, a narrow-shaped terminal minimizes the expected travel distance of forklifts or operators in the facility. However, if a cross-docking terminal's required area is fixed, a square ground plan is the best shape[5]. Other papers deal with the design of the temporary storage area in a cross-docking platform. Vis and Roodbergen, for instance, propose an algorithm that helps to compute the optimal number of storage rows and their lengths[6].

2.2.2 Tactical decisions

Once the decisions on the locations and layouts of cross-docking facilities have been made, decision-makers must decide how to route the cargo through the transportation network such that the costs are minimized[7]. On a tactical level, decision-makers are dealing with the following

[1] E.g., Bhaskaran (1992), Gümüş and Bookbinder (2004), Mousavi and Tavakkoli-Moghaddam (2013), Ross and Jayaraman (2008), Sung and Song (2003), and Sung and Yang (2008).

[2] Bhaskaran (1992, p. 141).

[3] Bartholdi and Gue (2004).

[4] Bartholdi and Gue (2004, p. 243).

[5] Carlo and Bozer (2011, p. 162).

[6] Vis and Roodbergen (2008).

[7] Van Belle et al. (2012, p. 832).

network planning problems:[1)]

1. Capacity planning for network routes;
2. Freight flow allocation;
3. Shipment to destination assignment.

Only a few papers that address cross-docking network planning aspects have been published. For instance, Musa et al. propose a model for assigning capacity to network routes and allocating freight flows to these routes[2)]. Furthermore, the truck-to-door assignment problem can be tackled on a tactical level. However, due to its operational characteristics, and since many authors classify this problem as an operational cross-docking problem, it will be discussed in the next section.

2.2.3 Operational decisions

On the operational level, scholars have studied various cross-docking related problems. Most operational cross-docking problems aim to determine **where** (i.e., at which dock-doors) and/or **when** (i.e., at which time), inbound trucks should be unloaded and outbound trucks should be loaded in a cross-docking terminal. Models that solely address the **spatial dimension** and ignore time aspects, decide to which dock-doors arriving inbound and outbound trucks should be assigned[3)]. Models that solely address the **temporal dimension** and ignore the spatial dimension (i.e., the location of dock-doors), on the other hand, aim to either sequence trucks or schedule trucks. Sequencing models aim to determine the optimal order in which trucks should be processed. These models take the time aspect implicitly into account. Scheduling models, on the other hand, explicitly consider the temporal dimension. Instead of determining a truck sequence in time, these models explicitly determine the exact start and end time for processing each truck.

Decisions on the spatial and temporal dimensions could be further made one after another. For instance, a decision-maker could first determine where to (un-)load trucks by solving the assignment problem. Based on the obtained truck-to-door assignment, the decision-maker could then solve a sequencing or scheduling model to compute at what time each truck should be

1) Buijs et al. (2014, p. 603).
2) Musa et al. (2010).
3) Van Belle et al. (2012, p. 835)

processed[1]. However, this sequential two-stage approach does not necessarily result in an optimal solution for the integrated problem[2]. Therefore, scholars have considered to integrating both decisions, i.e., they have aimed to assign a set of trucks to dock-doors and determine the times at which every truck should be processed simultaneously. Table 2.2 summarizes the main operational cross-docking planning problems that can be found in the literature. Each problem category and its related literature will be described in more detail below.

	Spatial dimension	Temporal dimension	
Problem description	Which door?	What time?	In which order?
Assignment of trucks	✓		
Sequencing of trucks			✓
Scheduling of trucks		✓	
Assignment and sequencing of trucks	✓		✓
Assignment and scheduling of trucks	✓	✓	

Table 2.2 Overview of different operational cross-docking decision problems.
Source: Own table after Ladier and Alpan (2016, p. 149).

Assignment of trucks

Once inbound and outbound trucks arrive at the cross-dock, it must be decided at which dock-doors inbound trucks and outbound trucks should be unloaded and loaded, respectively. These assignment patterns can have a huge impact on the cross-dock performance since they affect the traveling distance of material handling equipment and traveling time to transship inbound trucks' cargo from inbound dock-doors to outbound dock-doors[3]. A good truck-to-door assignment can help to increase productivity and reduce the (handling) costs[4]. The assignment of trucks to dock-doors can be done on a mid-term or short-term horizon[5]. In the former case, inbound trucks from the same origin (or outbound trucks with the same destination, respectively) are clustered and assigned to dock-doors. In other words, every dock-door is used to serve a group of trucks coming from the same origin or going to the same destination. Once the assignment has been made, it is used for a longer time period (e.g., 4-6 months). In the latter case, instead

[1] It may also be possible to make the temporal decision before making the spatial decision.

[2] Ladier and Alpan (2016, p. 148)

[3] Nassief et al. (2016, p. 495)

[4] Van Belle et al. (2012, p. 834).

[5] Boysen and Fliedner (2010, p. 415).

of having a fixed assignment based on a truck's origin or destination, an individual truck is assigned to a dock-door based on its actual freight volume on a short-term horizon (e.g., daily).

Peck and Tsui and Chang are among the first authors investigating the assignment of trucks to dock-doors. Peck tries to minimize the total travel time, i.e., the time required for transshipping cargo from inbound dock-doors to outbound dock-doors, in a less-than-truckload terminal with the help of a simulation model[1]. Tsui and Chang consider the assignment problem on a mid-term horizon and formulate a mathematical program to assign origins and destinations to dock-doors. They minimize the travel distance of forklifts in the cross-docking terminal and propose a heuristic solution method[2] and a branch-and-bound procedure[3] to solve the problem. Other authors such as Bermúdez and Cole and Cohen and Keren build on the early work of Tsui and Chang and develop model extensions as well as solution methods[4]. Bozer and Carlo study the assignment of inbound and outbound trucks to dock-doors in a rectangular cross-docking platform and develop a simulated annealing algorithm to minimize the total travel distance[5]. In a recent study, Nassief et al. develop a mixed-integer programming formulation for the truck-to-door assignment problem and embed it into a Lagrangian relaxation. The efficiency of the algorithm is proven through a numerical study[6]. Gelareh et al. recently proposed a total of eight new mixed-integer programming formulations for the assignment of trucks to dock-doors[7]. In a large computational study on benchmark instances from literature, the authors show that their best model formulation outperforms existing model formulations for the truck-to-door assignment problem[8].

Other studies focus on the assignment of outbound trucks (or destinations) to outbound dock-doors and ignore the assignment of inbound trucks (or origins) to inbound dock-doors. Bartholdi and Gue, for instance, aim to find an optimal assignment of destinations to outbound doors and to specify for each dock-door whether it is used for unloading inbound trucks or loading outbound trucks. They do not deal with the assignment of inbound trucks to inbound doors, as this is usually done in real-time according to their argumentation[9]. Jarrah et al.

[1] Peck (1983).

[2] Tsui and Chang (1990).

[3] Tsui and Chang (1992).

[4] Bermúdez and Cole (2001), Cohen and Keren (2009).

[5] Bozer and Carlo (2008).

[6] Nassief et al. (2016).

[7] Gelareh et al. use the term "cross-dock door assignment problem", cf. Gelareh et al. (2020).

[8] Gelareh et al. (2020, p. 1).

[9] Bartholdi and Gue (2000, p. 825).

study the door assignment problem for the configuration of an automated package sorting center. The authors develop a multi-objective model that hierarchically deals with the following objectives:[1)]

1. Minimization of the number of changes in destination-to-door assignments between shifts;
2. Minimization of the number of required workers;
3. Balancing the workload among workers.

Oh et al. consider a Korean mail distribution center and develop a mathematical model for improving the operations in the facility. Their model helps to cluster destinations and to assign the clusters to outbound doors[2)].

Sequencing of trucks

Models that focus on sequencing a set of trucks neglect the spatial dimension. That is, the location of dock-doors and the distance between pairs of dock-doors is not considered. Hence, a truck is not assigned to a specific door, but any door as long as the total number of doors is not exceeded[3)]. Such sequencing models aim to determine an optimal order in which inbound trucks should be unloaded, and outbound trucks should be loaded.

A considerably large number of papers investigate a simplified version of a cross-docking terminal with one inbound and one outbound dock-door. The overall goal of these papers can be seen in the identification of structural characteristics that could be transferred to the generalized case with multiple inbound and outbound dock-doors. Many of these papers try to minimize the makespan[4)]. Further optimization objectives are:

- Minimization of the required storage area;[5)]
- Minimization of the mean completion time of outbound trucks;[6)]

1) Jarrah et al. (2016, p. 1323).

2) Oh et al. (2006, p. 288).

3) Ladier and Alpan (2016, p. 150).

4) E.g., Chen and Lee (2009), Chiarello et al. (2018), Ghobadian et al. (2012), Liao et al. (2012), Romanova (2015), Yu and Egbelu (2008).

5) E.g., Sadykov (2012).

6) E.g., Amini et al. (2014).

- Minimization of the maximum lateness;[1)]
- Minimization of the total penalized tardiness, earliness, and the number of preemption for the outbound trucks;[2)]
- Minimization of product movements.[3)]

Larbi et al. study the value of order arrival and truck loading information and develop exact and heuristic solution methods to solve the model[4)]. Other studies focus on developing solution methods for the truck sequencing problem in a cross-docking platform with one inbound and one outbound dock-door[5)].

In addition, various authors study the truck sequencing problem in a more generalized case with multiple inbound and outbound dock-doors. The makespan has also been widely applied as the performance indicator in this setting[6)]. Moreover, sequencing models for minimizing the inventory level[7)], for maximizing the total number of directly transferred product units[8)], and for minimizing the earliness and tardiness of trucks[9)] have been proposed.

Scheduling of trucks

While models for sequencing of trucks implicitly consider time aspects, models for the scheduling of trucks explicitly consider the temporal dimension. Similar to the sequencing literature, both minimizing the makespan and minimizing the earliness and tardiness are among the most popular objectives in studies for scheduling trucks.

Some authors research a simplified scheduling problem with one inbound and one outbound door. Boysen et al., for instance, aim to derive insights into the underlying problem structure and therefore study the impact of truck arrival times and due dates, truck-specific processing

1) E.g., Briskorn et al. (2010).
2) E.g., Fazel Zarandi et al. (2016).
3) E.g., Forouharfard and Zandieh (2010), Maknoon and Baptiste (2009).
4) Larbi et al. (2011).
5) E.g., Shiguemoto et al. (2014), Soltani and Sadjadi (2010), Vahdani and Zandieh (2010), Vahdani et al. (2010).
6) E.g., Chen and Song (2009), Joo and Kim (2013), Yazdani et al. (2015).
7) E.g., Alpan et al. (2011a), Alpan et al. (2011b).
8) E.g., Maknoon et al. (2014).
9) E.g., Li et al. (2004).

times, and service modes in a setting with one inbound and one outbound dock-door[1]. Boloori Arabani, together with changing co-authors, also considers a setting with one inbound and one outbound dock-door and proposes different heuristic solution methods[2].

Furthermore, several studies focus on scheduling trucks in multi-door cross-docking platforms. Álvarez-Pérez et al., for instance, use a machine scheduling notation and formulate a model for scheduling trucks such that the transit storage time of cargo is minimized. As the problem turns out to be NP-hard, the authors develop a metaheuristic that combines a Greedy Randomized Adaptive Search Procedure (GRASP) with a Tabu Search.[3] Ladier and Alpan propose a model for planning the material handling in the cross-docking terminal and inbound/outbound trucks' arrival and departure times. For this purpose, the authors seek to balance two objectives: (i) maximize the number of pallets directly transshipped from inbound doors to outbound doors, and (ii) minimize the dissatisfaction of the transportation providers[4]. Fanti et al. schedule the internal operations (namely deconsolidation, sortation, and consolidation of cargo units) in a post-distribution cross-docking terminal. The authors seek to determine a schedule for internal operations that has a minimum makespan.[5] Bellanger et al. also consider sorting operations and aim to find a schedule with a minimum makespan[6]. The makespan minimization is once again among the most frequently used objective functions for scheduling trucks in multi-door cross-docking terminals.

Assignment and sequencing of trucks

With the increasing computational power, more and more scholars have studied operational cross-docking problems which integrate the spatial and temporal dimension. A vast number of studies simultaneously determine the processing sequence of trucks and the assignment of trucks to dock-doors. As the case with one inbound and one outbound dock-door would reduce the problem to a pure sequencing problem and make the door allocation trivial, only multi-door cross-docking terminals have been considered. In order to simplify the problem, some authors only schedule inbound trucks and assume that the outbound truck departures are known and

[1] Boysen et al. (2010).

[2] Boloori Arabani et al. (2010), Boloori Arabani et al. (2011a), Boloori Arabani et al. (2011b), Boloori Arabani et al. (2012).

[3] Álvarez-Pérez et al. (2009, p. 554).

[4] Ladier and Alpan (2018, p. 569).

[5] Fanti et al. (2014, p. 1).

[6] Bellanger et al. (2013, p. 1109).

given in advance.[1)]

When dealing with the assignment and sequencing of trucks, the makespan is again a frequently used performance measure. For instance, it is used by Madani-Isfahani et al., who study the problem of assigning and sequencing trucks in a multi-door cross-docking facility. Moreover, the authors develop two metaheuristics to solve the problem[2)]. McWilliams and McWilliams et al., who study the assignment and sequencing of inbound trucks in parcel hubs, also apply the makespan as the performance metric[3)]. Moreover, objective functions such as the minimization of inbound trucks' unloading times[4)], the minimization of the total number of delayed product units[5)], the minimization of the total travel distance within the cross-docking facility[6)], and the minimization of the inventory holding costs[7)] can also be found in studies that deal with the assignment and sequencing of trucks.

Assignment and scheduling of trucks

Similar to assigning and sequencing trucks, the problem of assigning and scheduling trucks also determines at which dock-doors and at what times trucks should be processed. However, it explicitly determines the exact start and end time for processing each truck instead of deriving this information from the processing order.

Existing research either assigns and schedules both in- and outbound trucks or inbound trucks only. The majority of existing papers assume that outbound trucks can only leave the cross-docking terminal once all freight units have been loaded. Acar et al., for instance, propose a mixed-integer program that aims to generate schedules that equally distribute idle times at dock-doors. By doing so, the authors seek to increase the robustness of the obtained schedules in order to encounter delayed truck arrivals.[8)] Other studies use the makespan[9)], the total travel

[1)] Ladier and Alpan (2016, p. 149).

[2)] Madani-Isfahani et al. (2014).

[3)] McWilliams (2009), McWilliams et al. (2005).

[4)] E.g., Golias et al. (2013), Golias et al. (2012).

[5)] E.g., Boysen et al. (2013), Boysen and Fliedner (2010) Liao et al. (2013).

[6)] E.g., Zhang et al. (2010). Moreover, Rosales et al. (2009) include the forklift operation costs, which are proportional to the forklifts' total travel distance, in their objective function.

[7)] E.g., Rahmanzadeh Tootkaleh et al. (2016).

[8)] Acar et al. (2012, p. 729).

[9)] E.g., Shakeri et al. (2012).

distance within the facility[1], and the waiting time of trucks[2] as performance measures.

These constraint-free truck departures, however, are often not applicable in practice. Instead, fixed outbound departures are used in many cross-docking platforms[3]. For instance, express and small parcel delivery and less-than-truckload logistics service companies, which usually strive for reliable and steady material flows in their transportation networks, often apply fixed outbound schedules[4]. Defining the departure times of outbound trucks prior to the scheduling task reduces the problem to scheduling inbound trucks only. CHMIELEWSKI ET AL., for instance, consider fixed departure times for both inbound and outbound trucks. The authors study the problem of assigning and scheduling inbound trucks and also include the assignment of destinations to outbound doors on a mid-term horizon. They aim to find a truck schedule that leads to a minimum total travel distance and minimum truck waiting times.[5] Other authors also apply weighted objective functions. VAN BELLE ET AL., for instance, propose an assignment and scheduling model which aims to minimize the weighted sum of the total travel time and the total tardiness[6]. BODNAR ET AL. also consider the tardiness of trucks in their objective function. Specifically, they aim to minimize the sum of handling and tardiness costs[7]. RIJAL ET AL. consider an objective function with three cost components: (i) The transportation costs for direct shipments between inbound and outbound dock-doors, (ii) the costs for temporarily storing freight units in the terminal, and (iii) the total tardiness costs of outbound trucks[8].

2.3 Chapter summary

This chapter introduced the cross-docking system, a widely used logistics concept that can improve the efficiency of logistics and distribution processes. Section 2.1 described the cross-docking concept. Furthermore, the wide range of potential benefits of cross-docking was presented. For instance, cross-docking can result in an improved service level and lower transportation, inventory holding, and handling costs if appropriately implemented. Finally,

1) E.g., Hermel et al. (2016), who propose a hierarchical framework in which they consider the makespan.

2) E.g., Bartz-Beielstein et al. (2006), who propose a multi-objective approach which also includes the total travel distance.

3) Ladier and Alpan (2016, p. 158).

4) Boysen et al. (2013, p. 480).

5) Chmielewski et al. (2009, pp. 200-202).

6) Van Belle et al. (2013, p. 820).

7) Bodnar et al. (2017, p. 117).

8) Rijal et al. (2019, p. 758).

classifications for structuring the wide range of cross-docking applications were presented.

Moreover, Section 2.2 provided an overview of the existing literature dealing with decision problems in the life cycle of a cross-docking terminal. While the strategic and tactical decision problems were briefly explained, the operational decision problems were described in more detail. Specifically, the operational cross-docking problems of assigning trucks to dock-doors and scheduling/sequencing trucks were explained. The most challenging decision problems are those which integrate both the spatial and temporal dimension. In the following, these decision problems are referred to as the **truck scheduling problem**, a commonly used term for decision problems that aim to determine both at which dock-door and at which time trucks should be processed.

3 Models for resource planning in cross-docking platforms

This chapter sets out to develop mathematical models for short-term resource planning in cross-docking platforms. Section 3.1 presents a basic model which allows to schedule both operators and trucks when the operator requirements of each truck are given and known in advance. Moreover, Section 3.2 proposes a model that allows to adapt the number of deployed operators for truck processing instead of assuming that the operator requirements are given. Finally, Section 3.3 presents a mathematical model which allows to consider different operator types and shift patterns.

3.1 Resource and truck scheduling problem

3.1.1 Introduction

3.1.1.1 Problem description

In the **resource and truck scheduling problem** (**TSFD-RC-F**), a cross-docking terminal with multiple inbound and outbound dock-doors and an **exclusive service mode** is considered. That is, in-/outbound trucks can only be processed at in-/outbound doors, respectively. It is assumed that the set of outbound trucks has already been scheduled, i.e., the truck-to-door assignment and the start times of outbound trucks is known. Hence, the problem reduces to a scheduling problem for inbound trucks which carry cargo to be loaded into various outbound trucks. Unloading of inbound trucks must start within their presupposed time windows. The time window of an inbound truck is defined through its **release time** and **due time**. It is assumed that the cargo is shipped in standardized freight units, e.g., pallets, and that a **sort-at-receiving protocol** is applied. Under this protocol, which requires incoming freight units to be labelled with the final destination, operators unload the cargo from inbound trucks and directly transfer it to the associated outbound dock-doors[1]. Hence, the **processing time** of an inbound truck, which includes the required time for unloading all freight units and transshipping them to the associated outbound dock-doors, is directly proportional to the number of freight units and depends on the distance between unloading and loading docks[2]. At most one truck can be processed at a dock-door at a time and preemption is not allowed, that is a truck cannot leave the dock-door before it has been processed completely. Upon arrival, the cargo can temporarily be stored in front of the outbound dock-door until loading of the outbound truck starts. Moreover, it is assumed that the intermediate storage space in front of each outbound dock-door is infinite. Cargo that arrives at the outbound dock-door after loading of the outbound truck starts is regarded as delayed freight units and postponed until the next departure to the same destination[3]. In addition, it is assumed that the required **service level** and hence the **allowed number of delayed freight units** are defined and known.

In this setting, inbound trucks are processed by **operators**, i.e., workers equipped with material handling equipment such as forklifts or pallets jacks. It is presupposed that an operator cannot process multiple trucks at the same time and must not start a new task prior to finishing her current task. The required number of operators for processing a truck is given and known in advance. Specifically, potential differences in operator requirements due to different truck sizes, are permitted. A couple of simplifying assumptions are made with respect to the availability of operators. A shift with a shift length equal to the planning horizon length and without breaks is considered as the standard shift type for operators. Therefore, it follows that operators on duty are available throughout the entire planning horizon (e.g., eight hours shift).

The described operating mode is widely used in practice, e.g., in unit-load cross-docking platforms of logistics service providers and retail companies. Recall that the labor cost and manpower, “very often the first cost center of a logistic platform”[4], are a main component of the total operational cost in such facilities and appear to be the dominant key performance indicator. The number of operators, which is correlated to the labor costs in the described setting with one standard shift type, is used as a surrogate objective function for the **TSFD-RC- F**. Hence, the **TSFD-RC-F** problem boils down to finding an efficient and feasible truck schedule, that is, a schedule with minimum operator requirements that complies with the truck time windows

[1] Bartholdi et al. (2008, p. 8).

[2] Van Belle et al. (2013, p. 819).

[3] Van Belle et al. (2012, p. 831).

[4] Ladier and Alpan (2016, p. 147).

and service level, and prevents potential conflicts regarding the usage of the shared resources (inbound dock-doors and operators). The following section presents different ways to formulate the **TSFD-RC-F** in optimization models.

3.1.1.2 Related literature

An overview of the relevant truck scheduling literature for the **TSFD-RC-F** is provided in Table 3.1.

Less than a dozen research papers consider internal resources for (un-)loading and transferring cargo. This gap in the literature is also identified in recent survey papers of Ladier and Alpan and Van Belle et al., who consider the integration of resource scheduling as an important future research task[1)]. The limited number of papers that attempt to consider resource scarcity usually do that by simply introducing an upper bound. Scarcity of material handling equipment[2)], workforce[3)], or unspecified internal resources[4)] has been recently considered. In this context, the makespan and travel distance – the two dominant performance measures in current academic truck scheduling models[5)] – are frequently applied performance measures. Furthermore, objectives such as minimizing the number of delayed product shipments and the earliness/tardiness of shipments have been discussed. To the best of the author's knowledge, Rosales et al., who aim to minimize the operational costs consisting of the cost of forklift operations and labor cost, are the only authors who apply an objective function that is directly related to the internal resource requirements[6)]. The study, however, assumes that the unloading start times for each trailer are given in advance and hence is rather an assignment problem than a scheduling problem.

Another research stream closely related to the **TSFD-RC-F** is the parallel machine scheduling problem (PMSP)[7)]. In this case, inbound trucks correspond to jobs and dock-doors correspond to machines. Specifically, the PMSP with unrelated machines and additional resources comes closest to the **TSFD-RC-F**. Unrelated parallel machines consider machine-dependent job processing

1) Ladier and Alpan (2016, p. 159), Van Belle et al. (2012, p. 844).
2) E.g., Shakeri et al. (2012), McWilliams (2009).
3) E.g., Tadumadze et al. (2019), Serrano et al. (2017), Li et al. (2004), Ladier and Alpan (2015).
4) E.g., Hermel et al. (2016), Chmielewski et al. (2009).
5) Ladier and Alpan (2016, p. 158).
6) Rosales et al. (2009).
7) For an overview on machine scheduling in general, see Pinedo (2016).

Publications	Problem characteristics						Objective function
	Oper. focus	Arrival times	Departure times	Processing times	Internal resources	Service level target	
Boysen et al. (2013)	I	-	O	DD	∞	No	Delayed product shipments
Tadumadze et al. (2019)	I	I	I+O	DD	lim.	No	Delayed product shipments
Rosales et al. (2009)	I	-	-	DD	∞	No	Labor cost, forklift operations cost
McWilliams (2009)	I	-	-	NDD	lim.	No	Makespan
Rahmanzadeh Tootkaleh et al. (2016)	I	-	O	NDD	∞	No	Inventory holding cost
Shakeri et al. (2012)	I+O	-	-	DD	lim.	No	Makespan
Li et al. (2004)	I+O	I	I+O	NDD	lim.	No	Earliness, tardiness
Serrano et al. (2017)	I+O	I	-	NDD	lim.	No	Time window violations
Chmielewski et al. (2009)	I+O	I+O	I+O	DD	lim.	No	Travel distance, truck waiting times
Hermel et al. (2016)	I+O	-	-	NDD	lim.	No	Travel distance, makespan
Ladier and Alpan (2015)	I+O	I	I+O	NDD	lim.	No	LSP satisfaction, temporary storage, shift changes, etc.)
Molavi et al. (2018)	I+O	I	I+O	DD	∞	No	Delayed product shipments
TSFD-RC-F	I	I+O	I+O	DD	lim.	Yes	Internal resource requirements

I: Inbound; O: Outbound; I+O: In- and outbound.

DD: Door-dependent processing times; NDD: Door-independent processing times.

∞: Unlimited internal resources; lim.: Limited internal resources.

Table 3.1 Summary of the related literature on truck scheduling in cross-docking platforms.
Source: Own table.

times. This is an essential characteristic of the **TSFD-RC-F** in which the truck processing times depend on the internal travel distances between the inbound and outbound dock-doors. A vast number of publications also include renewable or non-renewable resources in the PMSP. Since the **TSFD-RC-F** includes a discrete and renewable internal resource (e.g., workers equipped with material handling equipment), that is a resource that cannot be used up and is available throughout the entire planning horizon, the PMSP with renewable resources is relevant to this study. Edis et al. provide an in-depth survey on this field of research[1]. More recently, Fanjul-Peyro et al. analyze a PMSP with scarce renewable resources and propose matheuristics to tackle the problem[2]. Fleszar and Hindi propose different mixed-integer programming formulations and a constraint programming formulation for a PMSP with a resource constraint[3]. These machine scheduling models, however, are not immediately applicable to the **TSFD-RC-F** as they aim to minimize the makespan of the truck schedule, whereas the problem at hand utilizes a resource-leveling objective. Moreover, they do not consider the synchronization between inbound and outbound doors.

The **TSFD-RC-F** is also somewhat similar to the resource-constrained project scheduling problem (RCPSP). In this case, inbound trucks correspond to activities, which have a certain processing time and consume various renewable resources, namely dock-doors and internal resources. Brucker et al. and Hartmann and Briskorn provide a survey of the RCPSP literature[4]. The traditional RCPSP usually aims to minimize the makespan while a certain capacity level of each resource must be observed. Hence, it is not immediately applicable to this study. Moreover, the RCPSP typically considers precedence relations between activities. In a cross-docking facility, tasks such as unloading operations are rather constrained by time windows instead. The so-called resource investment problem (RIP), a “dual” version of the RCPSP, has been studied extensively in recent years[5]. It aims to minimize the costs for providing the resource capacity level while a project deadline must be met[6]. Even though it applies a similar objective as the **TSFD-RC-F**, it generally fails to consider essential truck scheduling particularities, such as travel distances between inbound and outbound doors. Emde et al. study the problem of scheduling personnel for the build-up of unit load devices in an air cargo terminal with space restrictions[7]. The authors formulate the model as a special type of multi-mode RCPSP with two renewable resources, namely workers and space for building up the unit load devices. The model considers the possibility to speed up processing by assigning multiple workers to a job. However, it is not capable to model door-dependent truck processing times, which are essential for the **TSFD-RC-F**.

---

[1] Edis et al. (2013).

[2] Fanjul-Peyro et al. (2017).

[3] Fleszar and Hindi (2018).

[4] Brucker et al. (1999), Hartmann and Briskorn (2010).

[5] E.g., Drexl and Kimms (2001), Neumann and Zimmermann (2000)

[6] Hartmann and Briskorn (2010, p. 7)

[7] Emde et al. (2020).

It can be concluded that existing research cannot directly be applied to the problem at hand and that a new solution concept, which integrates the scheduling of internal resources and trucks, needs to be developed.

3.1.2 Model formulations

Truck scheduling problems, amongst other scheduling problems such as the above-mentioned machine scheduling and resource constrained project scheduling problems, are often formulated as binary integer programs (BIP) or linear mixed-integer programs (MIP). Existing models can be classified into continuous-time[1)] and discrete-time[2)] model formulations. **Continuous-time models** (**CT**) use a set of continuous decision variables to specify when trucks are processed. In these models, truck processing can start at any time within a truck's time window. Furthermore, they often rely on disjunctive (precedence) constraints in combination with precedence-based (binary) decision variables to express the processing sequence between pairs of trucks that are assigned to the same dock-door. In order to obtain linear formulations for the disjunctive constraints, big-M formulations are often used. While big-M formulations are usually fairly compact, they tend to produce rather weak relaxations and hence large search trees[3)]. **Discrete-time model formulations** (**DT**) were introduced to overcome this major drawback. These models divide the planning horizon into a finite set of time intervals $t \in \mathcal{T} := \{0, ..., |\mathcal{T}|\}$ of equal length, and apply time-indexed binary decision variables to specify at which dock-door and in which time periods trucks are being processed. Note that processing a truck can only start at the beginning of a time interval. Discrete-time formulations typically produce a much stronger relaxation and lower bound than the continuous-time formulations. However, the time-indexation frequently results in a huge number of decision variables, especially for problems with long planning horizons and a fine time granularity[4)]. Since the degree of difficulty of solving an integer program increases exponentially as the number of integer variables increases, it is often challenging to solve large problem instances with a discrete-time model[5)]. In this section, both discrete-time and continuous-time model formulations for the **TSFD-RC-F** are presented.

---

1) E.g., Boysen et al. (2017), Van Belle et al. (2013), Shakeri et al. (2012), Li et al. (2004).

2) E.g., Rijal et al. (2019), Shahram fard and Vahdani (2019), Tadumadze et al. (2019), Ladier and Alpan (2018), Bodnar et al. (2017), Serrano et al. (2017).

3) Lamorgese and Mannino (2019, p. 1586-1587).

4) Lamorgese and Mannino (2019, p. 1587).

5) Chen et al. (2010, p. 80).

3.1.2.1 Discrete-time model formulations

The first discrete-time formulation for the resource and truck scheduling problem, denoted as **TSFD-RC-F-DT1**, applies a set of binary decision variables x_{idt} indexed by truck, dock-door, and time. This set of decision variables expresses at which dock-doors and times inbound trucks are unloaded. It is defined so that $x_{idt} = 1$ if truck i is assigned to dock-door d and processing starts in time interval t. Applying the notation listed in Tab. 3.2, the **TSFD-RC-F-DT1** can be formulated with the objective function (3.1) and the constraints (3.2) to (3.9).

TSFD-RC-F-DT1:

$$\text{Minimize} \quad W \tag{3.1}$$

$$\text{subject to} \quad \sum_{i \in \mathcal{I}} \sum_{d \in \mathcal{D}} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} \kappa_i x_{idt'} \leq W \qquad \forall t \in \mathcal{T} \tag{3.2}$$

$$\sum_{d \in \mathcal{D}} \sum_{t=r_i}^{d_i} x_{idt} = 1 \qquad \forall i \in \mathcal{I} \tag{3.3}$$

$$\sum_{i \in \mathcal{I}} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} x_{idt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{3.4}$$

$$\sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} (t + p_{id} - 1) x_{idt} - d_o \leq \Lambda y_{io} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.5}$$

$$\sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} y_{io} \leq (1 - \alpha) \sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} \tag{3.6}$$

$$W \in \mathbb{R}^+ \tag{3.7}$$

$$x_{idt} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, t \in \mathcal{T} \tag{3.8}$$

$$y_{io} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.9}$$

The objective is to find a feasible truck schedule that can be executed with a minimum number of operators W (3.1). Constraints (3.2) set a lower bound on the number of operators and ensure that in each time interval a sufficient number of operators is available. Hence, due to the combination of (3.1) and (3.2), the objective function is of the **minmax** type. Every inbound truck must be processed at exactly one dock-door and processing must start within its time window (3.3). Note that this set of constraints may be further relaxed and expressed as $\sum_{d \in \mathcal{D}} \sum_{t=r_i}^{d_i} x_{idt} \geq 1$ for all $i \in \mathcal{I}$. Inequalities (3.4) prevent overlaps, i.e., that multiple trucks are processed in parallel at a dock-door. The value of the binary variable y_{io}, which express

Sets:	
$\mathcal{I}$	Set of inbound trucks.
$\mathcal{O}$	Set of outbound trucks.
$\mathcal{D}$	Set of inbound doors.
$\mathcal{T}$	Set of time intervals.
Indices:	
i	Index used for inbound trucks, $i \in \mathcal{I}$.
o	Index used for outbound trucks, $o \in \mathcal{O}$.
d	Index used for inbound doors, $d \in \mathcal{D}$.
t, t'	Indices used for time intervals, $t, t' \in \mathcal{T}$.
Input parameters:	
r_i	Release time, i.e., earliest possible time to start processing inbound truck $i \in \mathcal{I}$; $r_i \in \mathcal{T}$.
d_i	Due time, i.e., latest possible time to start processing inbound truck $i \in \mathcal{I}$; $d_i \in \mathcal{T}$.
κ_i	Number of operators required to process inbound truck $i \in \mathcal{I}$; $\kappa_i \in \mathbb{Z}^+$.
d_o	Time when processing of outbound truck $o \in \mathcal{O}$ starts; $d_o \in \mathcal{T}$.
p_{id}	Time for processing inbound truck $i \in \mathcal{I}$ at inbound dock-door $d \in \mathcal{D}$; $p_{id} \in \mathbb{Z}^+$.
f_{io}	Number of product units to be transferred from inbound truck $i \in \mathcal{I}$ to outbound truck $o \in \mathcal{O}$; $f_{io} \in \mathbb{Z}_0^+$.
α	Required minimum service level; $\alpha \in [0, 1]$.
Λ	Big number.
Decision variables:	
W	Continuous variable: Number of operators.
x_{idt}	Binary variable: 1, if inbound truck $i \in \mathcal{I}$ is processed at inbound door $d \in \mathcal{D}$ and starts processing in time interval $t \in \mathcal{T}$; 0, otherwise.
y_{io}	Binary variable: 1, if processing inbound truck $i \in \mathcal{I}$ is finished before processing outbound truck $o \in \mathcal{O}$ starts; 0, otherwise.

Table 3.2 Notations for the discrete-time model formulations of the **TSFD-RC-F**.
Source: Own table.

whether the cargo of inbound truck i arrives in the outbound area before loading operations for outbound truck o start, is determined by inequalities (3.5). The variable y_{io} is set to 1, if processing of inbound truck i finishes late and its cargo could not arrive at the outbound dock-door before outbound truck o's planned departure time. In contrast, $y_{io} = 0$, if inbound truck i's cargo arrives at the outbound dock-door before o's planned departure time and hence can be loaded. Furthermore, constraint (3.6) compels that the predefined service level is met, that is, the total number of delayed product units may not exceed the permitted number of delayed product units. For this purpose, the permitted number of delayed product units is calculated through the term $(1-\alpha) \cdot \sum_{i \in I} \sum_{o \in O} f_{io}$. Lastly, the decision variables are defined in (3.7) to (3.9). Note that the decision variable W is defined as a continuous variable. Since κ_i can only take positive integer values, it is guaranteed that $W \in \mathbb{Z}^+$.

The **TSFD-RC-F-DT1** is characterized by a small number of constraints. However, note that it contains big-M formulation in the form of constraints (3.5). These formulations are required for determining the values of the decision variables y_{io} and could potentially produce a weak relaxation and lower bound. Therefore, an alternative discrete-time formulation, denoted as **TSFD-RC-F-DT2**, that spares both the decision variables y_{io} and big-M formulations, is proposed subsequently. Instead, it involves preprocessing parameters a_{idt} that express the number of delayed product units associated with truck i if it is assigned to dock-door d and processing starts in time interval t. Given that both the possible truck-to-door assignments and the number of possible truck start times are finite, the computation of a_{idt} is straightforward. More formally, the parameters a_{idt} can be calculated as follows:

$$a_{idt} = \sum_{o \in \mathcal{O}: t+p_{id}-1 > d_o} f_{io} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, t \in \mathcal{T} \tag{3.10}$$

The **TSFD-RC-F-DT2** can be formulated with the objective function (3.11) and the constraints (3.12) to (3.17).

TSFD-RC-F-DT2:

$$\text{Minimize} \quad W \tag{3.11}$$

$$\text{subject to} \quad \sum_{i \in \mathcal{I}} \sum_{d \in \mathcal{D}} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} \kappa_i x_{idt'} \leq W \qquad \forall t \in \mathcal{T} \tag{3.12}$$

$$\sum_{d \in \mathcal{D}} \sum_{t=r_i}^{d_i} x_{idt} = 1 \qquad \forall i \in \mathcal{I} \tag{3.13}$$

$$\sum_{i \in \mathcal{I}} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} x_{idt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{3.14}$$

$$\sum_{i \in \mathcal{I}} \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} a_{idt} x_{idt} \leq (1-\alpha) \sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} \tag{3.15}$$

$$W \in \mathbb{R}^+ \tag{3.16}$$

$$x_{idt} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, t \in \mathcal{T} \tag{3.17}$$

The **TSFD-RC-F-DT2's**, objective function (3.11), lower bound on the number of operators (3.12), truck-to-door assignment constraints (3.13), and no overlap constraints (3.14) remain unchanged compared with the **TSFD-RC-F-DT1**. Though, the logic to compel that the required service level is met, is different. It incorporates the preprocessing parameters a_{idt} and does not require the decision variables y_{io} and the associated big-M formulations. Hence, the **TSFD-RC-F-DT2** gets along with up to $|\mathcal{I}| \cdot |\mathcal{O}|$ fewer decision variables and constraints than the **TSFD-RC-F-DT1**.

3.1.2.2 Continuous-time model formulations

A major drawback of the presented time-discrete model formulations is the time-indexation, which inevitably results in a huge number of decision variables for big problem instances, especially when long planning horizons with many time intervals must be considered. To overcome this disadvantage, continuous-time model formulations are proposed in this section. Since time is not modeled explicitly in these formulations, they allow to significantly reduce the number of decision variables.

The first continuous-time model formulation, referred to as **TSFD-RC-F-CT1**, applies a set of binary decision variables x_{id} for the assignment of inbound trucks $i \in \mathcal{I}$ to dock-doors $d \in \mathcal{D}$ and a set of continuous variables s_i to indicate the associated start times of the inbound trucks. Note that the truck-to-door assignment and the start time of a truck is encoded in two separate variables. This is a distinguishing characteristic compared to the discrete-time model formulations, which encoded both information in a single binary variable x_{idt}. Since the planning horizon is not discretized, continuous-time models must deal with an infinite number of possible start times for inbound trucks. This makes it impossible to adapt some of the discrete-time models' constraints, specifically the calculation of required operators W and preventing multiple trucks from being processed in parallel at a dock-door. While formulating these constraints in the discrete-time setting with a finite set of time intervals is straightforward, it becomes more challenging in a continuous-time setting. Moreover, two additional sets of binary decision variables, ϕ_{ij} and ω_{ji}, are introduced for the sake of determining the truck sequence at a dock-door and the operator requirements. Both variables are defined for every truck pair $(i, j) \in \mathcal{I}^2$. While ϕ_{ij} signals whether truck i starts before truck j (i.e., $s_i < s_j \Rightarrow \phi_{ij} = 1$) or truck j starts before truck i (i.e., $s_j < s_i \Rightarrow \phi_{ij} = 0$), variable ω_{ji} indicates if processing truck j finishes before processing of truck i starts ($\omega_{ji} = 1$) or not ($\omega_{ji} = 0$). When applying the notation summarized in Table 3.3, the **TSFD-RC-F-CT1** can be formulated with the objective function (3.18) and the constraints (3.19) to (3.33).

Indices:	
j, l	Indices used for inbound trucks, $j, l \in \mathcal{I}$.
Decision variables:	
x_{id}	Binary variable: 1, if inbound truck $i \in \mathcal{I}$ is processed at inbound door $d \in \mathcal{D}$; 0, otherwise.
s_i	Continuous variable: Start time of inbound truck $i \in \mathcal{I}$.
ϕ_{ij}	Binary variable: 1, if processing of inbound truck i starts before processing of inbound truck j starts; 0, otherwise.
ω_{ji}	Binary variable: 1, if processing of inbound truck j starts before processing of inbound truck i is finished; 0, otherwise.

Table 3.3 Additional and altered notations for the continuous-time model formulations of the **TSFD-RC-F**.

Source: Own table.

TSFD-RC-F-CT1:

$$\text{Minimize} \quad W \tag{3.18}$$

$$\text{subject to} \quad \sum_{d \in \mathcal{D}} x_{id} = 1 \qquad \forall i \in \mathcal{I} \tag{3.19}$$

$$r_i \leq s_i \leq d_i \qquad \forall i \in \mathcal{I} \tag{3.20}$$

$$s_i + p_{id} x_{id} + \Lambda(x_{id} + x_{jd} + \phi_{ij} - 3) \leq s_j \qquad \forall i, j \in \mathcal{I} : i \neq j, d \in \mathcal{D} \tag{3.21}$$

$$\phi_{ij} + \phi_{ji} = 1 \qquad \forall i, j \in \mathcal{I} : i \neq j \tag{3.22}$$

$$\phi_{ij} + \phi_{jk} \leq 1 + \phi_{ik} \qquad \forall i, j, k \in \mathcal{I} \tag{3.23}$$

$$s_j - (s_i + \sum_{d \in \mathcal{D}} p_{id} x_{id}) \leq \Lambda(1 - \omega_{ji}) \qquad \forall i, j \in \mathcal{I} : i \neq j \tag{3.24}$$

$$(s_i + \sum_{d \in \mathcal{D}} p_{id} x_{id}) - s_j \leq \Lambda \omega_{ji} \qquad \forall i, j \in \mathcal{I} : i \neq j \tag{3.25}$$

$$\sum_{i \in \mathcal{I} : i \neq j} \kappa_i (\phi_{ij} + \omega_{ji} - 1) \leq W - \kappa_j \qquad \forall j \in \mathcal{I} \tag{3.26}$$

$$(s_i + \sum_{d \in \mathcal{D}} p_{id} x_{id}) - d_o \leq \Lambda y_{io} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.27}$$

$$\sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} y_{io} \leq (1 - \alpha) \sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} \tag{3.28}$$

$$W \in \mathbb{R}^+ \tag{3.29}$$

$$x_{id} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D} \tag{3.30}$$

$$s_i \in \mathbb{R}^+ \qquad \forall i \in \mathcal{I} \tag{3.31}$$

$$\phi_{ij}, \omega_{ji} \in \{0; 1\} \qquad \forall i, j \in \mathcal{I} \tag{3.32}$$

$$y_{io} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} \tag{3.33}$$

TSFD-RC-F-CT1 adopts the objective function from the **TSFD-RC-F-DT1**, that is to minimize the number of required operators W (3.18). Through constraints (3.19), every inbound truck is assigned to a dock-door, whereas constraints (3.20) guarantee that truck processing starts within a truck's time window. Inequalities (3.21) prevent multiple trucks from being processed simultaneously at the same dock-door. Constraints (3.22) are introduced to compel a well-defined precedence relation for truck pairs. The values of variables ω_{ji}, expressing whether processing truck j starts before processing of truck i is finished, are determined via the big-M formulations in (3.24) and (3.25). Furthermore, at any point in time, it must be assured that the number of operators required for processing trucks does not exceed the number of operators on duty. Luckily, this does not imply explicit checks for every point in time. Instead, it is sufficient to assure that whenever the processing of a truck j starts, at most $(W - \kappa_j)$ operators are busy with processing other trucks at that time (3.26). Specifically, this involves determining for every truck $i \in \mathcal{I} \setminus \{j\}$ whether it is **overlapping** the start time s_j of truck j. By definition, truck i is overlapping s_j if the following two conditions are met: (i) Processing truck i starts before processing of truck j (i.e., $\phi_{ij} = 1$), and (ii) processing of truck i ends after processing of truck j starts (i.e., $\omega_{ji} = 1$). Figure 3.1 shows an example to illustrate the overlapping conditions. In

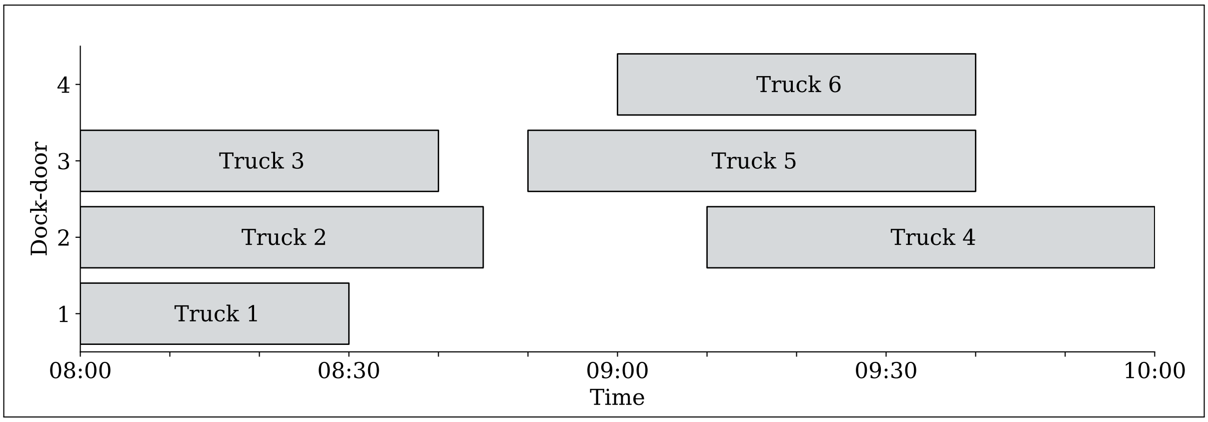


Figure 3.1 Exemplary truck schedule.
Source: Own figure.

this example, the processing of both trucks five and six overlap the starting time of truck four. Furthermore, neither truck four nor truck six overlap the starting time of truck five. The starting time of truck six is overlapped by truck five. The example also includes a situation in which three trucks, namely truck one, two, and three, start at the same time. While constraints (3.22) break ties between truck pairs, they do not necessarily break ties between more than two trucks in a logically correct way. For instance, consider the three inbound trucks $i \in \{1, 2, 3\}$ with identical start times in Figure 3.1. The model could break ties for truck pairs $(1, 2)$ and $(2, 3)$ by setting $\phi_{12} = \phi_{23} = 1$. In other words, the model defines that truck one starts before truck two and that truck two starts before truck three. However, note that constraints (3.22) then do not guarantee to set $\phi_{13} = 1$, i.e., truck one is not necessarily defined to start before truck three. In fact, constraints (3.26) incentivize the model to set $\phi_{13} = 0$ instead, since this would result in a lower (and hence wrong) number of required operators. To assure both a correct definition of ϕ_{ij} and calculation of the required operators when more than two trucks share an identical start time, therefore, so-called **transitivity constraints** (3.23) are introduced. In the example, these constraints set ϕ_{13} to one. Figure 3.2 illustrates this example. Moreover, constraints (3.27) and (3.28) determine the number of delayed product units which must not exceed the permitted maximum shortage quantity. Lastly, the decision variables are defined in (3.29) to (3.33).

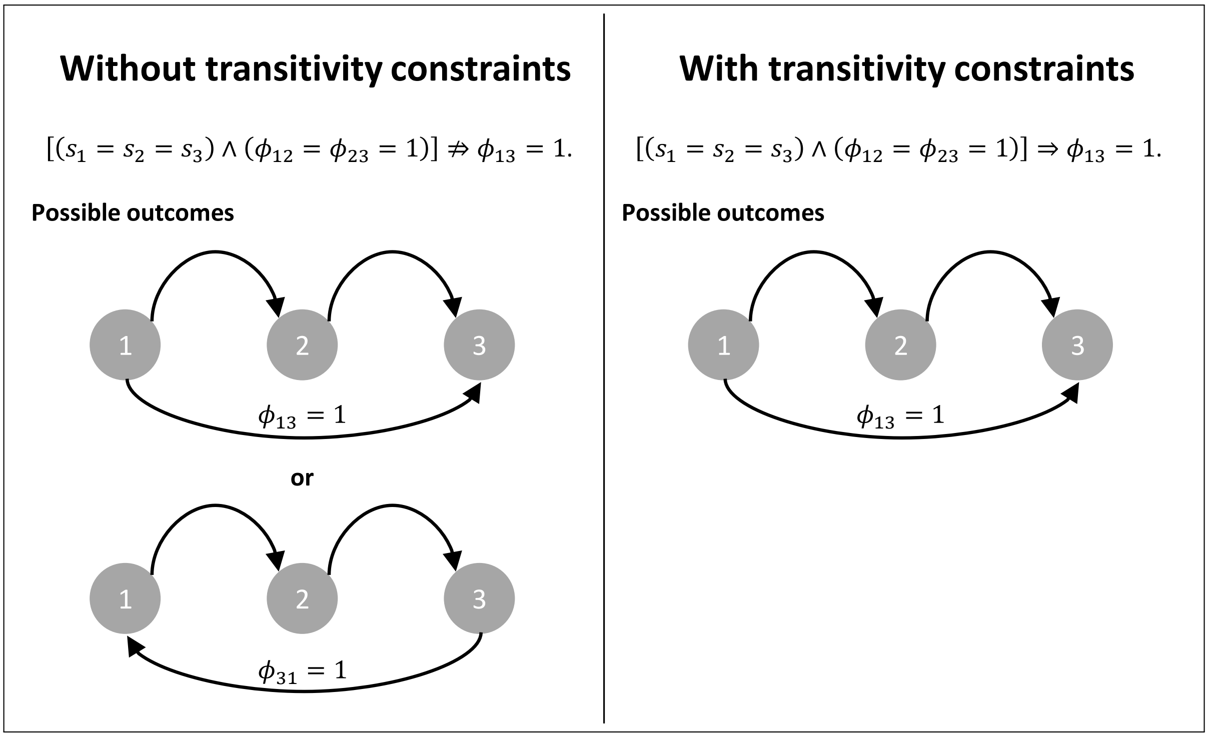


Figure 3.2 The effect of transitivity constraints.
Source: Own figure.

While the **TSFD-RC-F-CT1** involves fewer decision variables than the presented discrete-time model formulations, it comes along with a huge number of constraints. Especially the transitivity constraints, which are required for every triple (i, j, k) of trucks, constitute a major part of the constraints. Therefore, an alternative continuous-time formulation, denoted as **TSFD-RC-F-CT2**, that allows to reduce the number of constraints, is proposed. The **TSFD-RC-F-CT2** can be formulated with the objective function (3.34) and the constraints (3.35) to (3.49).

TSFD-RC-F-CT2:

$$\begin{aligned}
&\text{Minimize} && W && && (3.34)\\
&\text{subject to} && \sum_{d\in\mathcal{D}} x_{id} = 1 && \forall i \in \mathcal{I} && (3.35)\\
& && r_i \leq s_i \leq d_i && \forall i \in \mathcal{I} && (3.36)\\
& && s_i + p_{id}x_{id} + \Lambda(x_{id} + x_{jd} + \phi_{ij} - 3) \leq s_j && \forall i, j \in \mathcal{I} : i \neq j, d \in \mathcal{D} && (3.37)\\
& && \phi_{ij} + \phi_{ji} = 1 && \forall i, j \in \mathcal{I} : i \neq j && (3.38)\\
& && s_i + \lambda \leq s_j + \Lambda\phi_{ji} && \forall i, j \in \mathcal{I} && (3.39)\\
& && s_j - (s_i + \sum_{d\in\mathcal{D}} p_{id}x_{id}) \leq \Lambda(1 - \omega_{ji}) && \forall i, j \in \mathcal{I} : i \neq j && (3.40)\\
& && (s_i + \sum_{d\in\mathcal{D}} p_{id}x_{id}) - s_j \leq \Lambda\omega_{ji} && \forall i, j \in \mathcal{I} : i \neq j && (3.41)\\
& && \sum_{i\in\mathcal{I}:i\neq j} \kappa_i(\phi_{ij} + \omega_{ji} - 1) \leq W - \kappa_j && \forall j \in \mathcal{I} && (3.42)\\
& && (s_i + \sum_{d\in\mathcal{D}} p_{id}x_{id}) - d_o \leq \Lambda y_{io} && \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 && (3.43)\\
& && \sum_{i\in\mathcal{I}} \sum_{o\in\mathcal{O}} f_{io}y_{io} \leq (1 - \alpha) \sum_{i\in\mathcal{I}} \sum_{o\in\mathcal{O}} f_{io} && && (3.44)\\
& && W \in \mathbb{R}^+ && && (3.45)\\
& && x_{id} \in \{0; 1\} && \forall i \in \mathcal{I}, d \in \mathcal{D} && (3.46)\\
& && s_i \in \mathbb{R}^+ && \forall i \in \mathcal{I} && (3.47)\\
& && \phi_{ij}, \omega_{ji} \in \{0; 1\} && \forall i, j \in \mathcal{I} && (3.48)\\
& && y_{io} \in \{0; 1\} && \forall i \in \mathcal{I}, o \in \mathcal{O} && (3.49)
\end{aligned}$$

The **TSFD-RC-F-CT2** replaces **TSFD-RC-F-CT1's** transitivity constraints (3.23) with constraints (3.39). This constraint set, which is defined for every truck pair (i, j) and uses a very small constant λ (e.g., $\lambda = 0.01$), breaks ties between trucks with the same start time. Whenever two trucks i and j have the same start time, it sets $\phi_{ji} = 1$. This approach gets along with $(|\mathcal{I}|^3 - |\mathcal{I}|^2)$ fewer constraints than the **TSFD-RC-F-CT1**. Besides that, both model formulations are identical with respect to the objective function and remaining constraints.

Moreover, the search space can be further tightened by introducing the valid inequalities (3.50) and (3.51) on the decision variables ϕ_{ij}. According to preliminary tests, these inequalities help to reduce the solution time.

$$s_i - s_j \leq \Lambda(1 - \phi_{ij}) \qquad \forall i, j \in \mathcal{I} : i \neq j \tag{3.50}$$

$$s_j - s_i \leq \Lambda\phi_{ij} \qquad \forall i, j \in \mathcal{I} : i \neq j \tag{3.51}$$

3.1.3 Boundaries

Developing good lower and upper bounds is essential when solving an integer program, since they may help with fathoming and testing for optimality and hence reduce the solution time[1]. Therefore, both generic upper and lower bounds for the number of operators W that are intuitively appealing and computationally trivial are proposed in this section. These boundaries can be plugged into the mixed-integer programming formulations of the **TSFD-RC-F**. A **lower bound** for the objective function of the **TSFD-RC-F**, denoted as W^{LB}, is given in equation (3.52). It is defined by the minimum resource requirement of a truck ($\min_{i\in\mathcal{I}}\{\kappa_i\}$), the minimum total processing time of all trucks ($\sum_{i\in\mathcal{I}} \min_{d\in\mathcal{D}}\{p_{id}\}$), and the maximum time span that is available for processing trucks, i.e., the difference between the latest possible time an inbound truck could be finished and the earliest release time ($\max_{i\in\mathcal{I},d\in\mathcal{D}}\{d_i + p_{id} - 1\} - \min_{i\in\mathcal{I}}\{r_i\}$). Note that $\lceil * \rceil$ in 3.52 denotes the upper integer part of $*$.

$$W^{LB} = \left\lceil \min_{i\in\mathcal{I}}\{\kappa_i\} \cdot \frac{\sum\limits_{i\in\mathcal{I}} \min\limits_{d\in\mathcal{D}}\{p_{id}\}}{\max\limits_{i\in\mathcal{I},d\in\mathcal{D}}\{d_i + p_{id} - 1\} - \min\limits_{i\in\mathcal{I}}\{r_i\}} \right\rceil \tag{3.52}$$

For the calculation of a valid **upper bound** on the objective value, denoted as W^{UB}, first $\tilde{\kappa}_{it}$ is defined as the number of workers truck i would require if it was processed in time interval t. Note that a truck $i \in \mathcal{I}$ could potentially be processed in every time interval $t \in [r_i, d_i + \max_{d\in\mathcal{D}}\{\, p_{id}\,\} - 1] \cap \mathbb{Z}$, i.e., between its release time and the sum of its due time and maximum processing time. This leads to

$$\tilde{\kappa}_{it} = \begin{cases} \kappa_i & \text{if } r_i \leq t \leq d_i + \max_{d\in\mathcal{D}}\{p_{id}\} - 1 \\ 0 & \text{else.} \end{cases} \qquad \forall i \in \mathcal{I}, t \in \mathcal{T}. \tag{3.53}$$

[1] Cruz-Reyes et al. (2015, p. 42).

Based on this, for every $t \in \mathcal{T}$ a multiset $K_t = \{\, \tilde{\kappa}_{it} \mid i \in \mathcal{I} \,\}_b$ which contains all values $\tilde{\kappa}_{it}$ can be defined. Note that K_t is a multiset, i.e., it allows for multiple instances for each of its elements. Introducing W_t^{UB} for the maximum number of workers that could possibly be needed in parallel in time interval t, W_t^{UB}, hence, appears to be the sum of the $|D|$ biggest elements in the multiset K_t. This can be written as

$$W_t^{UB} = \underset{K'_t \subset K_t, |K'_t| = |D|}{\arg\max} \sum_{k \in K'_t} k \qquad \forall t \in \mathcal{T}. \tag{3.54}$$

Lastly, the upper bound W^{UB}, which can be plugged into the discrete-time formulations of the **TSFD-RC-F**, is the maximum of all W_t^{UB}:

$$W^{UB} = \max_{t \in \mathcal{T}} \left\{ W_t^{UB} \right\}. \tag{3.55}$$

3.1.4 Complexity

Large cross-docking platforms, which can easily have tens of thousands of square meters and dozens of dock-doors, often process several hundreds of trucks on a daily basis. Ladier and Alpan, for instance, report on a cross-docking platform of a less-than-truckload logistics service provider handling ca. 320 trucks every day[1)]. Similar numbers can be observed in the postal service industry[2)], whereas even bigger numbers of up to 480 trucks per day are reported in the automotive industry[3)]. For such large-sized problem instances, it can be very difficult to solve the **TSFD-RC-F**. Specifically, the number of inbound trucks $|\mathcal{I}|$, the number of dock-doors $|\mathcal{D}|$, the number of outbound trucks $|\mathcal{O}|$, and the number of time intervals $|\mathcal{T}|$ are critical factors that make a problem instance of the **TSFD-RC-F** either harder or easier to solve. They are the main complexity drivers of the **TSFD-RC-F** and have a strong impact on the model dimensions, that is the number of decision variables and number of constraints. Table 3.4 shows the mathematical expressions for calculating the number of decision variables and constraints for the different model formulations of the **TSFD-RC-F**. Moreover, Table 3.5 compares the model dimensions of the formulations for exemplary instances of various size.

1) Ladier and Alpan (2016, p. 156).

2) Boysen et al. (2017).

3) Battini et al. (2013, p. 210), Berghman and Leus (2015, p. 791).

Model formulation	$\#^{DV}$	$\#^{C}$
TSFD-RC-F-DT1	$\lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{D}\rvert \cdot \lvert\mathcal{T}\rvert + \lvert\mathcal{O}\rvert) + 1$	$\lvert\mathcal{T}\rvert \cdot (\lvert\mathcal{D}\rvert + 1) + \lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{O}\rvert + 1) + 1$
TSFD-RC-F-DT2	$\lvert\mathcal{I}\rvert \cdot \lvert\mathcal{D}\rvert \cdot \lvert\mathcal{T}\rvert + 1$	$\lvert\mathcal{T}\rvert \cdot (\lvert\mathcal{D}\rvert + 1) + \lvert\mathcal{I}\rvert + 1$
TSFD-RC-F-CT1	$\lvert\mathcal{I}\rvert \cdot (2 \cdot \lvert\mathcal{I}\rvert + \lvert\mathcal{D}\rvert + \lvert\mathcal{O}\rvert + 1) + 1$	$\lvert\mathcal{I}\rvert^3 + \lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{I}\rvert - 1) \cdot (\lvert\mathcal{D}\rvert + 5) + \lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{O}\rvert + 4) + 1$
TSFD-RC-F-CT2	$\lvert\mathcal{I}\rvert \cdot (2 \cdot \lvert\mathcal{I}\rvert + \lvert\mathcal{D}\rvert + \lvert\mathcal{O}\rvert + 1) + 1$	$\lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{I}\rvert - 1) \cdot (\lvert\mathcal{D}\rvert + 5) + \lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{I}\rvert + \lvert\mathcal{O}\rvert + 4) + 1$

$\#^{DV}$ and $\#^{C}$ denote the number of decision variables and the number of constraints, respectively. For the continuous-time models, $\#^{C}$ includes the valid inequalities (3.50) and (3.51).

Table 3.4 Number of decision variables and constraints for different model formulations of the **TSFD-RC-F**.
Source: Own table.

The tables indicate that the number of decision variables in the case of discrete-time models and the number of (big-M) constraints in the case of continuous-time models, can easily grow into the millions. The examples show that the continuous-time models deal with a considerably lower number of decision variables than the discrete-time models. If a fine time granularity is compulsory, continuous-time models get along with up to 94% fewer decision variables than the discrete-time models. However, this comes at the cost of an increasing number of constraints. The **TSFD-RC-F-CT1**, for instance, handles up to ca. 1,200 and 6,100 times more constraints than the **TSFD-RC-F-DT1** and **TSFD-RC-F-DT2**, respectively. The **TSFD-RC-F-CT2**, which spares out the transitivity constraints, on the other hand, requires ca. 80% less constraints than the **TSFD-RC-F-CT1**. It can be summarized that the huge complexity likely prevents real-world instances from being solved in a reasonable time by simply feeding the **TSFD-RC-F** into an off-the-shelf solver such as CPLEX or Gurobi. Even identifying a feasible solution to the **TSFD-RC-F** turns out to be very challenging. In fact, it is a strongly NP-complete task, as described in the following theorems.

Theorem 3.1.1. *Finding a feasible solution for the* ***TSFD-RC-F*** *is strongly NP-complete for* $|\mathcal{D}| \geq 1$.

Even without the service level constraint ($\alpha = 0$), i.e., for the special case that all product units could be delayed, finding a feasible solution remains NP-complete in the strong sense. This directly follows from the **TSFD-RC-F** without service level constraint being a generalization of the single machine scheduling problem with time windows, which is well known to be strongly NP-complete. Note that the feasibility of a given solution to the **TSFD-RC-F** can be tested in polynomial time.

Instances				TSFD-RC-F-DT1		TSFD-RC-F-DT2		TSFD-RC-F-CT1		TSFD-RC-F-CT2	
$\|\mathcal{I}\|$	$\|\mathcal{D}\|$	$\|\mathcal{O}\|$	$\|\mathcal{T}\|$	$\#^{DV}$	$\#^{C}$	$\#^{DV}$	$\#^{C}$	$\#^{DV}$	$\#^{C}$	$\#^{DV}$	$\#^{C}$
50	8	20	48	20,201	1,483	19,201	483	6,451	158,051	6,451	35,551
50	8	20	96	39,401	1,915	38,401	915	6,451	158,051	6,451	35,551
50	8	20	240	97,001	3,211	96,001	2,211	6,451	158,051	6,451	35,551
100	15	40	48	76,001	4,869	72,001	869	25,601	1,202,401	25,601	212,401
100	15	40	96	148,001	5,637	144,001	1,637	25,601	1,202,401	25,601	212,401
100	15	40	240	364,001	7,941	360,001	3,941	25,601	1,202,401	25,601	212,401
200	30	60	48	300,001	13,689	288,001	1,689	98,201	9,405,801	98,201	1,445,801
200	30	60	96	588,001	15,177	576,001	3,177	98,201	9,405,801	98,201	1,445,801
200	30	60	240	1,452,001	19,641	1,440,001	7,641	98,201	9,405,801	98,201	1,445,801
300	50	80	48	744,001	26,749	720,001	2,749	219,301	31,958,701	219,301	5,048,701
300	50	80	96	1,464,001	29,197	1,440,001	5,197	219,301	31,958,701	219,301	5,048,701
300	50	80	240	3,624,001	36,541	3,600,001	12,541	219,301	31,958,701	219,301	5,048,701

$\#^{DV}$ and $\#^{C}$ denote the number of decision variables and the number of constraints, respectively.

For the continuous-time models, $\#^{C}$ includes the valid inequalities (3.50) and (3.51).

Table 3.5 Exemplary model dimensions for different model formulations of the **TSFD-RC-F**.
Source: Own table.

Theorem 3.1.2. *Finding a feasible solution for the* ***TSFD-RC-F*** *is strongly NP-complete for* $|\mathcal{D}| \geq 1$, $0 < \alpha \leq 1$, *and* $(d_i - r_i) = |\mathcal{T}|$ *for all* $i \in \mathcal{I}$.

Proof. To prove theorem 3.1.2, a pseudopolynomial reduction from a single machine scheduling problem to minimize tardy task weights (**SMSP-TTW**) which is known to be NP-complete, is presented[1)].

SMS-TTW: Given a set $\mathcal{J}$ of independent jobs to be scheduled without preemption on a single machine. Each job $j \in \mathcal{J}$ has a processing time p_j, a weight f_j, a deadline d_j, and may start at any time in the planning horizon. Furthermore, a positive integer F is given. Is there a conflict-free schedule where all jobs $j \in \mathcal{J}$ are processed such that the sum of f_j, taken over all $j \in \mathcal{J}$ for which $s_j + p_j > d_j$, does not exceed F?

Any instance of the **SMS-TTW** can be transformed into an instance of the **TSFD-RC-F**. For each job j, a pair (i, o) with an inbound truck i and an outbound truck o is introduced. The release time and deadline of inbound truck i are 0 and $|\mathcal{T}|$, respectively. Moreover, outbound truck o has the deadline $d_o = d_j$. Inbound truck i holds f_j product units and solely supplies outbound truck o. That is, $f_{io} = f_j$. Hence, the question to be asked is whether there exists a feasible truck schedule, that is a schedule that has no overlaps and that satisfies $\alpha \geq 1 - \frac{F}{\sum_{j \in \mathcal{J}} f_j}$? The correspondence between the instances (and their solutions) is hence obvious. □

Corollary 3.1.3. *The* ***TSFD-RC-F*** *is strongly NP-hard.*

3.2 Multi-mode resource and truck scheduling problem

3.2.1 Introduction

The **TSFD-RC-F** assumed that the required number of operators for processing a truck is given and known in advance. For instance, a terminal manager could make use of rule of thumbs such as distinguishing between small trucks and big trucks and deploying one operator and two operators, respectively. In real-world settings, however, terminal managers have the further flexibility of adapting the resources for certain trucks. If a truck, for instance, is very time critical, a terminal manager could recede from his rule of thumb and deploy additional operators in

1) Garey and Johnson (1979, p. 236).

order to speed up truck processing. TADUMADZE ET AL. study this possibility in the context of cross-docking terminals and distribution centers[1]. In this section, a model for scheduling resources and trucks in a cross-docking terminal is presented. The model integrates the decision of how many resources should be deployed for truck processing.

3.2.1.1 Problem description

Unlike the **TSFD-RC-F**, the multi-mode resource and truck scheduling problem, referred to as **TSFD-RC-V**, does not assume that the number of deployed operators for processing a truck is given and known in advance. Instead, the **TSFD-RC-V** integrates the decision of **how many operators** should process each inbound truck. While deploying more operators accelerates truck processing, deploying fewer operators prolongs the processing time span. The remaining model assumptions of the **TSFD-RC-V**, which it has in common with the **TSFD-RC-F**, are briefly summarized below:

- **Fixed outbound departures:** Outbound trucks have already been scheduled, i.e., it is known at which dock-doors and at which times outbound trucks will be processed.
- **Exclusive service mode:** Inbound trucks can only be processed at inbound dock-doors and outbound trucks can only be processed at outbound dock-doors.
- **Truck time windows:** Processing of an inbound trucks must start within the truck's presupposed time window, which is defined through a release time and due time.
- **Standardized freight units:** Cargo is shipped in standardized freight units such as pallets.
- **Sort-at-receiving protocol:** Cargo is unloaded from inbound trucks and directly transferred to the associated outbound dock-doors. Upon arrival, the cargo is temporarily stored in front of the outbound dock-door until loading of the outbound truck starts.
- **Truck processing time:** The processing time of inbound trucks include the time for unloading all freight units and transshipping them to the associated outbound dock-doors. It is hence directly proportional to the number of freight units and depends on the distance between the unloading dock and the loading docks.
- **No preemption:** That is, (un-)loading a truck may not be interrupted. Moreover, an

[1] Tadumadze et al. (2019).

operator cannot process multiple trucks at the same time and must not start a new task prior to finishing her current task.

- **Delayed freight units:** Delaying cargo is allowed. Cargo that arrives at the outbound dock-door after loading of the outbound truck starts is regarded as delayed freight units. It is postponed until the next departure to the same destination.

- **Service level:** The required service level and hence the allowed number of delayed freight units is defined and known.

- **Operator characteristics and availability:** Trucks are processed by operators, i.e., workers equipped with material handling equipment. All operators have identical characteristics (i.e., identical skills, speed, etc.), are available throughout the whole planning horizon, and do not take any breaks.

Similar to the **TSFD-RC-F**, the **TSFD-RC-V** aims to find an efficient truck schedule, that is, a schedule that minimizes the number of operators active in the busiest period.

3.2.1.2 Related literature

In the following, the related literature for the **TSFD-RC-V** will be summarized. In order to avoid redundancies with the literature described in the context of the **TSFD-RC-F**, only studies which consider multi-mode processing will be included in this section.

Truck scheduling publications dealing with multi-mode scheduling are quite rare. To the best of the author's knowledge, Tadumadze et al. are the only ones who allow to adapt the workforce for processing trucks and consider a scarce workforce. Since they do not consider an objective function that is directly related to the internal resource requirements, their model is not immediately applicable to the **TSFD-RC-V**.

Speeding up jobs by assigning additional resources to them is also subject to task scheduling on processors. Błażewicz et al., for instance, introduced the problem of scheduling so-called malleable tasks, i.e., tasks which can be executed by multiple processors simultaneously in order to speed up processing[1)]. However, they do neither consider additional resources nor an objective function that is related to the resources. Publications dealing with resource related objective functions are in general comparatively rare. Gorczyca and Janiak are one of the

[1)] Błażewicz et al. (2004, p. 65).

few exceptions who consider a resource-leveling objective in the problem of scheduling non-preemptive tasks on identical parallel processors. In their problem, the processing speed of a task depends on the amount of a continuously divisible renewable resource assigned to this task[1]. Moreover, CHEN study the problem of scheduling jobs on identical parallel machines and allocating non-renewable resources to the jobs[2]. However, neither of the two studies are immediately applicable to the **TSFD-RC-V** which deals with unrelated parallel machines (i.e., dock-doors) and considers a discrete renewable resource (i.e., operators).

Moreover, multi-mode processing can also be found in the resource-constrained project scheduling problem literature. WĘGLARZ ET AL. survey the literature on multi-mode project scheduling problems[3]. Multi-mode processing and resource leveling objectives, two key characteristics of the **TSFD-RC-V**, can be found in the so-called multi-mode resource investment problem, which was firstly proposed by HSU AND KIM[4]. Moreover, GERHARDS creates a benchmark library and proposes new lower bounds for this problem. Even though the multi-mode resource investment problem applies a similar objective as the **TSFD-RC-V**, it generally fails to consider essential truck scheduling particularities, such as travel distances between inbound and outbound doors. Lastly, EMDE ET AL. propose a special type of multi-mode RCPSP with two renewable resources and a resource leveling objective for building up the unit load devices at an air-cargo terminal[5]. Their model, however, also cannot be applied to the **TSFD-RC-V** as it is not capable to model door-dependent truck processing times – a key characteristic of the **TSFD-RC-V**.

As existing research cannot directly be applied to the multi-mode resource and truck scheduling problem, a new solution concept needs to be developed.

3.2.2 Model formulations

In this section, both discrete-time and continuous-time model formulations for the multi-mode resource and truck scheduling problem will be presented.

1) Gorczyca and Janiak (2010, p. 32).
2) Chen (2004).
3) Węglarz et al. (2011).
4) Hsu and Kim (2005).
5) Emde et al. (2020).

3.2.2.1 Discrete-time model formulations

Similar to the **TSFD-RC-F**, the first discrete-time formulation for the multi-mode resource and truck scheduling problem, referred to as **TSFD-RC-V-DT1**, applies a set of binary decision variables x_{idkt}. These variables are indexed by truck, dock-door, operator mode, and time. They express at which dock-door and time the inbound trucks are unloaded and how many operators are deployed for processing trucks. The decision variable is defined so that $x_{idkt} = 1$ if truck i is processed in operator mode k at dock-door d and processing starts in time interval t. When applying the notation listed in Table 3.6, the **TSFD-RC-V-DT1** can be formulated with the objective function (3.56) and constraints (3.57) to (3.64).

TSFD-RC-V-DT1:

$$\text{Minimize} \quad W \tag{3.56}$$

$$\text{subject to} \quad \sum_{i\in\mathcal{I}}\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t'=\max\{0;t-p_{idk}+1\}}^{t} \kappa_{ik}x_{idkt'} \leq W \qquad \forall t \in \mathcal{T} \tag{3.57}$$

$$\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t=r_i}^{d_i} x_{idkt} = 1 \qquad \forall i \in \mathcal{I} \tag{3.58}$$

$$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\sum_{t'=\max\{0;t-p_{idk}+1\}}^{t} x_{idkt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{3.59}$$

$$\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}(t + p_{idk} - 1)x_{idkt} - d_o \leq \Lambda y_{io} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.60}$$

$$\sum_{i\in\mathcal{I}}\sum_{o\in\mathcal{O}} f_{io}y_{io} \leq (1-\alpha)\sum_{i\in\mathcal{I}}\sum_{o\in\mathcal{O}} f_{io} \tag{3.61}$$

$$W \in \mathbb{R}^+ \tag{3.62}$$

$$x_{idkt} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, k \in \mathcal{K}, t \in \mathcal{T} \tag{3.63}$$

$$y_{io} \in \{0;1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.64}$$

The **TSFD-RC-V-DT1** seeks to identify a feasible truck schedule which can be executed with a minimum number of operators W (3.56). In this context, a truck schedule is regarded as being feasible if it satisfies the resource constraints, assignment constraints, no-overlap constraints, and service level constraint. Specifically, the resource constraints (3.57), which set a lower

Sets:	
$\mathcal{I}$	Set of inbound trucks.
$\mathcal{O}$	Set of outbound trucks.
$\mathcal{D}$	Set of inbound doors.
$\mathcal{K}$	Set of operator modes.
$\mathcal{T}$	Set of time intervals.
Indices:	
i	Index used for inbound trucks, $i \in \mathcal{I}$.
o	Index used for outbound trucks, $o \in \mathcal{O}$.
d	Index used for inbound doors, $d \in \mathcal{D}$.
k	Index used for operator modes, $k \in \mathcal{K}$.
t, t'	Indices used for time intervals, $t, t' \in \mathcal{T}$.
Input parameters:	
r_i	Release time, i.e., earliest possible time to start processing inbound truck $i \in \mathcal{I}$; $r_i \in \mathcal{T}$.
d_i	Due time, i.e., latest possible time to start processing inbound truck $i \in \mathcal{I}$; $d_i \in \mathcal{T}$.
κ_{ik}	Number of operators required when processing inbound truck $i \in \mathcal{I}$ in operator mode $k \in \mathcal{K}$; $\kappa_{ik} \in \mathbb{Z}^+$.
d_o	Time when processing of outbound truck $o \in \mathcal{O}$ starts; $d_o \in \mathcal{T}$.
p_{idk}	Time for processing inbound truck $i \in \mathcal{I}$ at inbound dock-door $d \in \mathcal{D}$ in operator mode $k \in \mathcal{K}$; $p_{idk} \in \mathbb{Z}^+$.
f_{io}	Number of product units to be transferred from inbound truck $i \in \mathcal{I}$ to outbound truck $o \in \mathcal{O}$; $f_{io} \in \mathbb{Z}_0^+$.
α	Required minimum service level; $\alpha \in [0, 1]$.
Λ	Big number.
Decision variables:	
W	Continuous variable: Number of operators.
x_{idkt}	Binary variable: 1, if inbound truck $i \in \mathcal{I}$ is processed at inbound door $d \in \mathcal{D}$ in operator mode $k \in \mathcal{K}$ and starts processing in time interval $t \in \mathcal{T}$; 0, otherwise.
y_{io}	Binary variable: 1, if processing inbound truck $i \in \mathcal{I}$ is finished before processing outbound truck $o \in \mathcal{O}$ starts; 0, otherwise.

Table 3.6 Notations for the discrete-time model formulations of the **TSFD-RC-V**.
Source: Own table.

bound on the number of operators, assure that in each time interval a sufficient number of operators is deployed in order to handle the actual workload. Hence, the **TSFD-RC-V-DT1** deals with a **minmax** objective function. Due to the truck assignment constraints (3.58), every inbound truck i is assigned to exactly one dock-door and processed in exactly one operator mode. Moreover, the constraints compel that truck processing starts within a truck's time window. Note that this set of constraints may be further relaxed and expressed as $\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t=r_i}^{d_i} x_{idkt} \geq 1$ for all $i \in \mathcal{I}$. The no-overlap constraints (3.59) prevent multiple trucks from being processed in parallel at a dock-door. Once again, the binary variable y_{io} is used to express whether inbound truck i's cargo arrives in the outbound area before loading of outbound truck o starts. Specifically, y_{io} is set to 1, if inbound truck i's cargo arrives in the outbound area after outbound truck o's loading operations start, i.e., the cargo misses the deadline and cannot be loaded (3.60). The decision variables y_{io} are then used in the service level constraint (3.61), which compels that the number of delayed product units may not exceed the permitted number of delayed product units. In other words, constraint (3.61) assures that the predefined service level α is met. Lastly, the domain of the decision variables is defined by (3.62), (3.63), and (3.64). Note that W is defined as a continuous variable. Since κ_{ik} can only take positive integer values, it is guaranteed that $W \in \mathbb{Z}^+$.

Note that the **TSFD-RC-V-DT1's** big-M formulations in constraints (3.60) may produce a weak relaxation and lower bound. This, in turn, can make it more challenging to solve instances of the **TSFD-RC-V-DT1** with a default solver. Therefore, an alternative discrete-time formulation, denoted as **TSFD-RC-V-DT2**, is proposed below. Specifically, it contains preprocessing parameters a_{idkt} which express the delayed product units when truck i is processed in operator mode k at dock-door d, and processing starts in time interval t. Since the possible truck-to-door assignments, the number of operator modes, and the number of possible truck start times are finite, the computation of a_{idkt} is straightforward. More formally, the parameters a_{idkt} can be calculated as follows:

$$a_{idkt} = \sum_{o\in\mathcal{O}: t+p_{idk}-1>d_o} f_{io} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, k \in \mathcal{K}, t \in \mathcal{T} \qquad (3.65)$$

When applying the preprocessing parameters a_{idkt}, the **TSFD-RC-V-DT2** can be formulated with the objective function (3.66) and the constraints (3.67) to (3.72). The **TSFD-RC-V-DT2's** objective function (3.66), resource constraints (3.67), truck-to-door assignment constraints (3.68), and no-overlap constraints (3.69) remain unchanged compared with the **TSFD-RC-V-DT1**. Its service level constraint (3.70), on the other hand, differs from the **TSFD-RC-V-DT1's** service level constraint, since it applies the described preprocessing parameters a_{idkt} instead of the decision variables y_{io}.

TSFD-RC-V-DT2:

$$\text{Minimize} \quad W \tag{3.66}$$

$$\text{subject to} \quad \sum_{i\in\mathcal{I}}\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t'=\max\{0;t-p_{idk}+1\}}^{t} \kappa_i x_{idkt'} \leq W \qquad \forall t \in \mathcal{T} \tag{3.67}$$

$$\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t=r_i}^{d_i} x_{idkt} = 1 \qquad \forall i \in \mathcal{I} \tag{3.68}$$

$$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\sum_{t'=\max\{0;t-p_{id}+1\}}^{t} x_{idt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{3.69}$$

$$\sum_{i\in\mathcal{I}}\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} a_{idkt}x_{idkt} \leq (1-\alpha)\sum_{i\in\mathcal{I}}\sum_{o\in\mathcal{O}} f_{io} \tag{3.70}$$

$$W \in \mathbb{R}^+ \tag{3.71}$$

$$x_{idkt} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, k \in \mathcal{K}, t \in \mathcal{T} \tag{3.72}$$

3.2.2.2 Continuous-time model formulations

When applying the notations from Table 3.7, the **TSFD-RC-V-CT1** can be formulated with the objective function (3.73) and the constraints (3.74) to (3.90). In light of **TSFD-RC-V** being a generalization of the **TSFD-RC-F**, it is no surprise that the **TSFD-RC-V-CT1** is very similar to the **TSFD-RC-F-CT1**. Once again, the **TSFD-RC-V-CT1** aims to minimize the number of deployed operators (3.73). Constraints (3.74) assure that each truck is processed in one operator mode $k \in \mathcal{K}$ at one dock-door $d \in \mathcal{D}$. Moreover, violations of truck time windows are prevented by constraints (3.75). Constraints (3.76) assure that at most one truck is processed at a dock-door at a time. Constraints (3.77) and (3.78) are used to compel well-defined precedence relations between two trucks and more than two trucks with the same start time, respectively. Moreover, constraints (3.79) and (3.80) determine the values of decision variables ω_{ji}, that is, whether truck j starts before truck i is finished ($\omega_{ji} = 1$) or not ($\omega_{ji} = 0$). Both ϕ_{ij} and ω_{ji} are then used in constraints (3.81) in order to determine the value of the integer decision variable θ_{ji}. Recall that θ_{ji} expresses the number of deployed operators for processing truck i at truck j's start time s_j. Based on the values of variables θ_{ji}, constraints (3.82) assure that whenever a truck j starts, at most $(W - \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} \kappa_{jk}x_{jdk})$ operators are deployed for processing other trucks at that time. Moreover, constraints (3.83) and (3.84) allow to determine the number of delayed product units which must not exceed the permitted maximum shortage quantity. Lastly, the domains of the decision variables are defined by (3.85) to (3.90).

TSFD-RC-V-CT1:

$$\text{Minimize} \quad W \tag{3.73}$$

$$\text{subject to} \quad \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} x_{idk} = 1 \qquad \forall i \in \mathcal{I} \tag{3.74}$$

$$r_i \le s_i \le d_i \qquad \forall i \in \mathcal{I} \tag{3.75}$$

$$s_i + \sum_{k\in\mathcal{K}} p_{idk}x_{idk} + \Lambda(\sum_{k\in\mathcal{K}}[x_{idk} + x_{jdk}] + \phi_{ij} - 3) \le s_j \qquad \forall i,j \in \mathcal{I} : i \ne j, d \in \mathcal{D} \tag{3.76}$$

$$\phi_{ij} + \phi_{ji} = 1 \qquad \forall i,j \in \mathcal{I} : i \ne j \tag{3.77}$$

$$\phi_{ij} + \phi_{jl} \le 1 + \phi_{il} \qquad \forall i,j,l \in \mathcal{I} \tag{3.78}$$

$$s_j - (s_i + \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} p_{idk}x_{idk}) \le \Lambda(1 - \omega_{ji}) \qquad \forall i,j \in \mathcal{I} : i \ne j \tag{3.79}$$

$$(s_i + \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} p_{idk}x_{idk}) - s_j \le \Lambda\omega_{ji} \qquad \forall i,j \in \mathcal{I} : i \ne j \tag{3.80}$$

$$\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} \kappa_{ik}x_{idk} - \Lambda(2 - \phi_{ij} - \omega_{ji}) \le \theta_{ji} \qquad \forall i,j \in \mathcal{I} : i \ne j \tag{3.81}$$

$$\sum_{i\in\mathcal{I}:i\ne j} \theta_{ji} \le W - \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} \kappa_{jk}x_{jdk} \qquad \forall j \in \mathcal{I} \tag{3.82}$$

$$(s_i + \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} p_{idk}x_{idk}) - d_o \le \Lambda y_{io} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.83}$$

$$\sum_{i\in\mathcal{I}}\sum_{o\in\mathcal{O}} f_{io}y_{io} \le (1-\alpha)\sum_{i\in\mathcal{I}}\sum_{o\in\mathcal{O}} f_{io} \tag{3.84}$$

$$W \in \mathbb{R}^+ \tag{3.85}$$

$$x_{idk} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, k \in \mathcal{K} \tag{3.86}$$

$$s_i \in \mathbb{R}^+ \qquad \forall i \in \mathcal{I} \tag{3.87}$$

$$\phi_{ij}, \omega_{ji} \in \{0;1\} \qquad \forall i,j \in \mathcal{I} \tag{3.88}$$

$$\theta_{ji} \in \mathbb{Z}^+ \qquad \forall i,j \in \mathcal{I} \tag{3.89}$$

$$y_{io} \in \{0;1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} \tag{3.90}$$

Indices:	
j, l	Indices used for inbound trucks, $j, l \in \mathcal{I}$.
Decision variables:	
x_{idk}	Binary variable: 1, if inbound truck $i \in \mathcal{I}$ is processed at inbound door $d \in \mathcal{D}$ in operator mode $k \in \mathcal{K}$; 0, otherwise.
s_i	Continuous variable: Start time of inbound truck $i \in \mathcal{I}$.
ϕ_{ij}	Binary variable: 1, if processing of inbound truck i starts before processing of inbound truck j starts; 0, otherwise.
ω_{ji}	Binary variable: 1, if processing of inbound truck j starts before processing of inbound truck i is finished; 0, otherwise.
θ_{ji}	Integer variable: Number of deployed operators for processing truck i at truck j's start time s_j.

Table 3.7 Additional and altered notations for the continuous-time model formulations of the of the **TSFD-RC-V**.

Source: Own table.

Furthermore, using the identical approach as in the context of the continuous-time resource and truck scheduling problem allows to reduce the number of constraints. Once again, the **TSFD-RC-V-CT1's** transitivity constraints (3.78) can be replaced constraints (3.39), which were used in the **TSFD-RC-F-CT2**. The resulting continuous-time formulation, referred to as **TSFD-RC-V-CT2**, can then be formulated with objective function (3.91) and constraints (3.92) to (3.108). Besides using constraints (3.96) instead of the transitivity constraints, the **TSFD-RC-V-CT2** is identical with the **TSFD-RC-V-CT1**. Note that the valid inequalities (3.50) and (3.51) can also be applied in both continuous-time models in order to further tighten the search space.

TSFD-RC-V-CT2:

$$\text{Minimize} \quad W \tag{3.91}$$

$$\text{subject to} \quad \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} x_{idk} = 1 \qquad \forall i \in \mathcal{I} \tag{3.92}$$

$$r_i \leq s_i \leq d_i \qquad \forall i \in \mathcal{I} \tag{3.93}$$

$$s_i + \sum_{k\in\mathcal{K}} p_{idk}x_{idk} + \Lambda(\sum_{k\in\mathcal{K}}[x_{idk} + x_{jdk}] + \phi_{ij} - 3) \leq s_j \qquad \forall i,j \in \mathcal{I} : i \neq j, d \in \mathcal{D} \tag{3.94}$$

$$\phi_{ij} + \phi_{ji} = 1 \qquad \forall i,j \in \mathcal{I} : i \neq j \tag{3.95}$$

$$s_i + \lambda \leq s_j + \Lambda\phi_{ji} \qquad \forall i,j \in \mathcal{I} \tag{3.96}$$

$$s_j - (s_i + \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} p_{idk}x_{idk}) \leq \Lambda(1 - \omega_{ji}) \qquad \forall i,j \in \mathcal{I} : i \neq j \tag{3.97}$$

$$(s_i + \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} p_{idk}x_{idk}) - s_j \leq \Lambda\omega_{ji} \qquad \forall i,j \in \mathcal{I} : i \neq j \tag{3.98}$$

$$\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} \kappa_{ik}x_{idk} - \Lambda(2 - \phi_{ij} - \omega_{ji}) \leq \theta_{ji} \qquad \forall i,j \in \mathcal{I} : i \neq j \tag{3.99}$$

$$\sum_{i\in\mathcal{I}:i\neq j} \theta_{ji} \leq W - \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} \kappa_{jk}x_{jdk} \qquad \forall j \in \mathcal{I} \tag{3.100}$$

$$(s_i + \sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}} p_{idk}x_{idk}) - d_o \leq \Lambda y_{io} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.101}$$

$$\sum_{i\in\mathcal{I}}\sum_{o\in\mathcal{O}} f_{io}y_{io} \leq (1-\alpha)\sum_{i\in\mathcal{I}}\sum_{o\in\mathcal{O}} f_{io} \tag{3.102}$$

$$W \in \mathbb{R}^+ \tag{3.103}$$

$$x_{idk} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, k \in \mathcal{K} \tag{3.104}$$

$$s_i \in \mathbb{R}^+ \qquad \forall i \in \mathcal{I} \tag{3.105}$$

$$\phi_{ij}, \omega_{ji} \in \{0;1\} \qquad \forall i,j \in \mathcal{I} \tag{3.106}$$

$$\theta_{ji} \in \mathbb{Z}^+ \qquad \forall i,j \in \mathcal{I} \tag{3.107}$$

$$y_{io} \in \{0;1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} \tag{3.108}$$

3.2.3 Complexity

Compared to the **TSFD-RC-F**, the multi-mode resource and truck scheduling problem (**TSFD-RC-V**) comprises an additional complexity driver, namely the operator mode. Its impact on the model dimensions, as well as the complexity of the **TSFD-RC-V** in general, will be explained in more detail below.

While Table 3.8 provides the equations for calculating the number of decision variables and constraints in each model formulation, Table 3.9 compares the dimensions of the formulations for exemplary instances of different size.

Model formulation	$\#^{DV}$	$\#^{C}$
TSFD-RC-V-DT1	$\lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{D}\rvert \cdot \lvert\mathcal{K}\rvert \cdot \lvert\mathcal{T}\rvert + \lvert\mathcal{O}\rvert) + 1$	$\lvert\mathcal{T}\rvert \cdot (\lvert\mathcal{D}\rvert + 1) + \lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{O}\rvert + 1) + 1$
TSFD-RC-V-DT2	$\lvert\mathcal{I}\rvert \cdot \lvert\mathcal{D}\rvert \cdot \lvert\mathcal{K}\rvert \cdot \lvert\mathcal{T}\rvert + 1$	$\lvert\mathcal{T}\rvert \cdot (\lvert\mathcal{D}\rvert + 1) + \lvert\mathcal{I}\rvert + 1$
TSFD-RC-V-CT1	$\lvert\mathcal{I}\rvert \cdot (3 \cdot \lvert\mathcal{I}\rvert + \lvert\mathcal{D}\rvert \cdot \lvert\mathcal{K}\rvert + \lvert\mathcal{O}\rvert + 1) + 1$	$\lvert\mathcal{I}\rvert^3 + \lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{I}\rvert - 1) \cdot (\lvert\mathcal{D}\rvert + 6) + \lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{O}\rvert + 4) + 1$
TSFD-RC-V-CT2	$\lvert\mathcal{I}\rvert \cdot (3 \cdot \lvert\mathcal{I}\rvert + \lvert\mathcal{D}\rvert \cdot \lvert\mathcal{K}\rvert + \lvert\mathcal{O}\rvert + 1) + 1$	$\lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{I}\rvert - 1) \cdot (\lvert\mathcal{D}\rvert + 6) + \lvert\mathcal{I}\rvert \cdot (\lvert\mathcal{I}\rvert + \lvert\mathcal{O}\rvert + 4) + 1$

$\#^{DV}$ and $\#^{C}$ denote the number of decision variables and the number of constraints, respectively. For the continuous-time models, $\#^{C}$ includes the valid inequalities (3.50) and (3.51).

Table 3.8 Number of decision variables and constraints for different model formulations of the **TSFD-RC-V**.
Source: Own table.

The tables show that the **TSFD-RC-V's** discrete-time formulations deal with a significantly bigger number of decision variables than its continuous-time formulations. With respect to the number of constraints, on the other hand, it can be seen that the **TSFD-RC-V's** continuous-time formulations have a disadvantage over their discrete-time counterparts, which involve a considerably smaller number of constraints. Moreover, the tables illustrate the impact of the operator mode on the model dimensions. Since both the discrete-time and the continuous-time formulations use decision variables that are indexed by the operator mode, a positive relationship between the number of operator modes $|\mathcal{K}|$ and the number of decision variables can be observed for all formulations. The effect of $|\mathcal{K}|$ on the number of decision variables appears to be smaller in the continuous-time formulations compared to the discrete-time formulations. Moreover, it can be seen that higher values of $|\mathcal{K}|$ do not affect the number of constraints in the discrete-time formulations. In the continuous-time formulations, on the other hand, an increase in $|\mathcal{K}|$ leads to a slightly bigger number of constraints.

It can be summarized that the huge number of decision variables in the case of the discrete-time models, and the huge number of constraints in the case of the continuous-time models, likely prevents real-world instances from being solved in a reasonable time with the help of a default solver. The following theorems and corollary on the complexity status of the **TSFD-RC-V** verify

Instances					**TSFD-RC-V-DT1**		**TSFD-RC-V-DT2**		**TSFD-RC-V-CT1**		**TSFD-RC-V-CT2**	
$\|\mathcal{I}\|$	$\|\mathcal{D}\|$	$\|\mathcal{K}\|$	$\|\mathcal{O}\|$	$\|\mathcal{T}\|$	$\#^{DV}$	$\#^{C}$	$\#^{DV}$	$\#^{C}$	$\#^{DV}$	$\#^{C}$	$\#^{DV}$	$\#^{C}$
50	8	1	20	48	20,201	1,483	19,201	483	8,951	160,501	8,951	38,001
50	8	1	20	96	39,401	1,915	38,401	915	8,951	160,501	8,951	38,001
50	8	1	20	240	97,001	3,211	96,001	2,211	8,951	160,501	8,951	38,001
100	15	2	40	48	148,001	4,869	144,001	869	37,101	1,212,301	37,101	222,301
100	15	2	40	96	292,001	5,637	288,001	1,637	37,101	1,212,301	37,101	222,301
100	15	2	40	240	724,001	7,941	720,001	3,941	37,101	1,212,301	37,101	222,301
200	30	3	60	48	876,001	13,689	864,001	1,689	150,201	9,445,601	150,201	1,485,601
200	30	3	60	96	1,740,001	15,177	1,728,001	3,177	150,201	9,445,601	150,201	1,485,601
200	30	3	60	240	4,332,001	19,641	4,320,001	7,641	150,201	9,445,601	150,201	1,485,601
300	50	4	80	48	2,904,001	26,749	2,880,001	2,749	354,301	32,048,401	354,301	5,138,401
300	50	4	80	96	5,784,001	29,197	5,760,001	5,197	354,301	32,048,401	354,301	5,138,401
300	50	4	80	240	14,424,001	36,541	14,400,001	12,541	354,301	32,048,401	354,301	5,138,401

$\#^{DV}$ and $\#^{C}$ denote the number of decision variables and the number of constraints, respectively.

For the continuous-time models, $\#^{C}$ includes the valid inequalities (3.50) and (3.51).

Table 3.9 Exemplary model dimensions for different model formulations of the **TSFD-RC-V**.
Source: Own table.

this suspicion.

Theorem 3.2.1. *Finding a feasible solution for the **TSFD-RC-V** is strongly NP-complete for* $|\mathcal{D}| \geq 1$.

Theorem 3.2.1 directly follows from the **TSFD-RC-V** being a generalization of the resource and truck scheduling problem (**TSFD-RC-F**), which was proven to be strongly NP-complete in Theorem 3.1.1. Note that the feasibility of a given solution to the **TSFD-RC-V** can be tested in polynomial time.

Theorem 3.2.2. *Finding a feasible solution for the **TSFD-RC-V** is strongly NP-complete for* $|\mathcal{D}| \geq 1$, $0 < \alpha \leq 1$, *and* $(d_i - r_i) = |\mathcal{T}|$ *for all* $i \in \mathcal{I}$.

The **TSFD-RC-V** remains NP-complete even without truck time windows (i.e., $(d_i - r_i) = |\mathcal{T}|$ for all $i \in \mathcal{I}$). This directly follows from the **TSFD-RC-V** without time windows being a generalization of the **TSFD-RC-F** without time windows, which was proven to be strongly NP-complete in Theorem 3.1.2.

Corollary 3.2.3. *The **TSFD-RC-V** is strongly NP-hard.*

3.3 Shift and truck scheduling problem

3.3.1 Introduction

Both the **TSFD-RC-F** and the **TSFD-RC-V** make a couple of simplifying assumptions with respect to the availability of operators. Specifically, they assume one standard shift type for operators with a shift length equal to the planning horizon length and without breaks. Therefore, operators on duty are assumed to be available throughout the entire planning horizon. While these assumptions may hold when considering short planning horizons (e.g., a 4h planning horizon), they do not adequately represent the reality in many situations. When considering longer planning horizons (e.g., an 8h or 12h planning horizon), the labor laws must be taken into account. Under the German Working Time Act[1], for instance, an employee's working time is limited to a maximum of 8h per working day and 48h per week. Increasing the working time

[1] In German known as "Arbeitszeitgesetz".

up to 10h per working day is permitted as long as the average 8h working time per working day is not exceeded over a six months reference period.[1] In addition to the working time, the German Working Time Act also regulates an employee's right to rest. The law distinguishes between two types of rest periods, namely rest breaks and resting times. **Rest breaks** are breaks that must be granted during the working time. Employees who work between 6-9h are entitled to a minimum break of 30 minutes. **Resting times**, on the other hand, denote the time period between two working days. Employees are entitled to a minimum resting time of 11 hours between the end of a work period and the beginning of the next work period.[2] Furthermore, both the **TSFD-RC-F** and the **TSFD-RC-V** do not consider different workforce types or shift patterns. Real-world settings, however, can be far more complex. In the logistics industry, which is characterized by a highly variable demand, the workload can strongly fluctuate throughout the week and even over a day. Hence, it might not be sufficient to apply one standardized work pattern (e.g., a 8h standard shift) when creating the employees' timetables. Terminal managers can respond to demand fluctuations by using different shift patterns for workers.[3] Moreover, logistics companies often employ temporary staff in order to increase the flexibility[4].

In order to overcome the **TSFD-RC-F's** and **TSFD-RC-V's** shortcomings, this section sets out to develop mathematical models which allow to consider different types of workforce and work shifts.

3.3.1.1 Problem description

The shift and truck scheduling problem, denoted as **ISTSFD**, provides a framework for operations planning in cross-docking platforms, which integrates the employee timetabling task and the truck scheduling task. Especially in manual cross-docking terminals, which are often characterized by a high proportion of personnel expenses[5], efficient employee timetables are of great importance.

The **ISTSFD** allows to distinguish between different operator types. Specifically, it assumes that operators from different operator groups $g \in \mathcal{G}$ can be deployed. This enables decision-makers to distinguish between regular and temporary workers, full-time and part-time workers, etc.

[1] Goletz (2015).

[2] Breitschwerdt (2016).

[3] Ladier et al. (2014, p. 279).

[4] Weber et al. (2005, p. 22).

[5] Bartholdi and Gue (2004, p. 235-236), Pfohl (2005, p. 313).

For each operator group g, the maximum number of available operators $\overline{W}_g$ as well as a set of predefined shift patterns $\mathcal{S}_g$ is given. Operators belonging to operator group g can be deployed in any of the predefined shift patterns $s \in \mathcal{S}_g$. Each shift pattern is described via binary parameters γ_{gst}, which signal whether an operator who belongs to operator group g and works in shift pattern $s \in \mathcal{S}_g$ is on duty (i.e., $\gamma_{gst} = 1$) or not (i.e., $\gamma_{gst} = 0$). Table 3.10 shows exemplary shift patterns. The example considers an 8h planning horizon from 08:00-16:00 and shows

Time			Operator group 1				Operator group 2		
t	From	To	$s=1$	$s=2$	$s=3$	$s=4$	$s=1$	$s=2$	$s=3$
1	08:00	08:30	1	1	1	1	1	0	0
2	08:30	09:00	1	1	1	1	1	0	0
3	09:00	09:30	1	1	1	1	1	0	0
4	09:30	10:00	1	1	1	1	1	0	0
5	10:00	10:30	1	1	1	1	1	1	0
6	10:30	11:00	1	1	1	1	1	1	0
7	11:00	11:30	1	1	1	1	1	1	0
8	11:30	12:00	0	1	1	1	1	1	0
9	12:00	12:30	1	0	1	1	0	1	1
10	12:30	13:00	1	1	0	1	0	1	1
11	13:00	13:30	1	1	1	0	0	1	1
12	13:30	14:00	1	1	1	1	0	1	1
13	14:00	14:30	1	1	1	1	0	0	1
14	14:30	15:00	1	1	1	1	0	0	1
15	15:00	15:30	1	1	1	1	0	0	1
16	15:30	16:00	1	1	1	1	0	0	1

Table 3.10 Exemplary shift patterns.
Source: Own table.

predefined shift patterns for two operator groups (i.e., $g \in \{1, 2\}$). The four shift patterns for operator group $g = 1$ all begin at 08:00 and end at 16:00. Operators who are assigned to the first shift pattern (i.e., $(g, s) = (1, 1)$), for instance, are granted a 30 minute rest break from 11:30-12:00. Moreover, the example shows three predefined shift patterns for operator group $g = 2$. While all three have a shift length of 4h and no rest breaks, they differ with respect to their start times. Specifically, shift one, two, and three start at 08:00, 10:00, and 12:00, respectively.

For the sake of completeness, the remaining basic assumptions of the **ISTSFD** are briefly summarized below:

- **Fixed outbound departures:** Outbound trucks have already been scheduled, i.e., it is known at which dock-doors and at which times outbound trucks will be processed.

- **Exclusive service mode:** Inbound trucks can only be processed at inbound dock-doors and outbound trucks can only be processed at outbound dock-doors.
- **Truck time windows:** Processing of an inbound trucks must start within the truck's presupposed time window, which is defined through a release time and due time.
- **Standardized freight units:** Cargo is shipped in standardized freight units such as pallets.
- **Sort-at-receiving protocol:** Cargo is unloaded from inbound trucks and directly transferred to the associated outbound dock-doors. Upon arrival, the cargo is temporarily stored in front of the outbound dock-door until loading of the outbound truck starts.
- **Truck processing time:** The processing time of inbound trucks include the time for unloading all freight units and transshipping them to the associated outbound dock-doors. It is hence directly proportional to the number of freight units and depends on the distance between the unloading dock and the loading docks.
- **No preemption:** That is, (un-)loading a truck may not be interrupted. Moreover, an operator cannot process multiple trucks at the same time and must not start a new task prior to finishing her current task.
- **Delayed freight units:** Delaying cargo is allowed. Cargo that arrives at the outbound dock-door after loading of the outbound truck starts is regarded as delayed freight units. It is postponed until the next departure to the same destination.
- **Service level:** The required service level and hence the allowed number of delayed freight units is defined and known.
- **Operator characteristics:** Trucks are processed by operators, i.e., workers equipped with material handling equipment. All operators have identical characteristics (i.e., identical skills, speed, etc.).
- **Shift cost:** The costs for deploying an operator in a certain shift pattern are given and known in advance.

3.3.1.2 Related literature

The shift and truck scheduling problem bears some resemblance with the previously described (multi-mode) resource and truck scheduling problem. In order to avoid redundancies with the literature described in the context of the **TSFD-RC-F** and **TSFD-RC-V**, only studies which may be able to model the **ISTSFD's** employee timetabling characteristics will be summarized below.

Truck scheduling publications usually neglect employee timetabling aspects. Ladier and Alpan and Rosales et al. are among the few exceptions. Ladier and Alpan attempt to combine truck scheduling and employee rostering in a cross-docking terminal. They present two independent models, one for truck scheduling and one for employee rostering, and propose a solution approach where both models are solved iteratively until a stable point is reached. The study, however, does not include a fully integrated model.[1] Rosales et al. study the problem of scheduling inbound trailers in a cross-docking platform. They aim to minimize the operational costs consisting of the cost of forklift operations and labor cost. They presume that a worker is assigned to a certain dock-door and responsible for all trailers allocated to his dock-door. In addition, a set of constraints compel employee workload balancing[2]. However, the study assumes that the unloading start times for each trailer are given in advance. That is, the proposed model is rather an assignment problem than a scheduling problem.

Employee timetabling has also been integrated in the job-shop scheduling problem[3]. Some examples are Artigues et al. and Guyon et al.[4] However, these studies assume that each operation has a certain machine it must be processed on. This is different from the **ISTSFD** in which each inbound truck can theoretically be processed at every inbound dock-door. The flexible job-shop scheduling problem – a generalization of the job-shop scheduling problem – considers identical multi-purpose machines that can process different types of operations[5]. The inbound dock-doors in the **ISTSFD**, on the other hand, can be seen as unrelated parallel machines, which is not a feature of the classic flexible job-shop scheduling problem.

The **ISTSFD** also bears some resemblances with some extensions of the RCPSP, e.g., the RCPSP with time-dependent resource capacities[6]. The RCPSPS with time-dependent resource capacities usually assumes that the resource capacity in each time period is known and given in advance. In the **ISTSFD**, on the other hand, the resource capacity in each time period is an outcome instead of an input parameter.

---

[1] Ladier and Alpan (2015, p. 676).

[2] Rosales et al. (2009, pp. 325-326).

[3] The standard version of the job-shop scheduling problem considers a set of jobs. Each job consists of a set of operations that must be processed in a given order. Each operation must be processed on a specific machine (that is known in advance) and only one operation of a job can be performed at a time. Cf. Guyon et al. (2014, p. 148).

[4] Artigues et al. (2009), Guyon et al. (2014).

[5] Kress et al. (2019, p. 180).

[6] E.g., Hartmann (2013), Klein (2000), Sprecher and Drexl (1998).

As existing research cannot directly be applied to the shift and truck scheduling problem, a new solution concept needs to be developed.

3.3.2 Model formulations

Two variants of the **ISTSFD** are proposed in the following. Section 3.3.2.1 describes the single-mode shift and truck scheduling problem, which assumes fixed operator requirements for each inbound truck. A multi-mode variant of the shift and truck scheduling problem is presented in Section 3.3.2.2. The multi-mode problem does not consider fixed operator requirements but allows several operator modes in which each truck can be processed.

3.3.2.1 Single-mode model

The first model for shift and truck scheduling, referred to as **ISTSFD-F**, considers a single operator mode for each inbound truck. That is, similar to the **TSFD-RC-F**, the **ISTSFD-F** assumes that the number of required operators for processing a truck is given and known in advance. That is, the **ISTSFD-F** only considers one operator mode.

The first discrete-time formulation of the **ISTSFD-F**, referred to as **ISTSFD-F-DT1**, applies a set of binary variables x_{idt} in order to express at which dock-door $d \in \mathcal{D}$ and at which time $t \in \mathcal{T}$ processing of inbound truck i starts. Furthermore, it involves the decision variables W_{gs} and y_{io}. The integer variables W_{gs} express how many operators from operator group $g \in \mathcal{G}$ are rostered in shift pattern $s \in \mathcal{S}_g$. The binary decision variables y_{io}, on the other hand, are used to signal whether the cargo of inbound truck i arrives in the outbound area in-time so that the cargo can be loaded onto outbound truck o. When applying the notation listed in Table 3.11, the **ISTSFD-F-DT1** can be formulated with the objective function (3.109) and the constraints (3.110) to (3.118). Objective function (3.109) minimizes the total labor cost for deployed operators from various operator groups. Constraints (3.110) ensure that in each time interval a sufficient number of operators is present. Moreover, constraints (3.111) limit the number of operators rostered from each operator group $g \in \mathcal{G}$. They guarantee that the number of available operators per operator group is not exceeded. Constraints (3.112) enforce that each inbound truck is processed exactly once and render violating any time window impossible. Constraints (3.113) prevent multiple trucks from being processed in parallel at a dock-door. Constraints (3.114) in conjunction with (3.115) compels the service level by limiting the number of freight units which are delayed. Finally, (3.116) to (3.118) define the domain of the decision variables.

Sets:	
$\mathcal{I}$	Set of inbound trucks.
$\mathcal{O}$	Set of outbound trucks.
$\mathcal{D}$	Set of inbound doors.
$\mathcal{T}$	Set of time intervals.
$\mathcal{G}$	Set of operator groups.
$\mathcal{S}_g$	Set of shift patterns for operator group $g \in \mathcal{G}$.
Indices:	
i	Index used for inbound trucks, $i \in \mathcal{I}$.
o	Index used for outbound trucks, $o \in \mathcal{O}$.
d	Index used for inbound doors, $d \in \mathcal{D}$.
t, t'	Indices used for time intervals, $t, t' \in \mathcal{T}$.
g	Index used for operator groups, $g \in \mathcal{G}$.
s	Index used for shift patterns, $s \in \mathcal{S}_g$ with $g \in \mathcal{G}$.
Input parameters:	
r_i	Release time, i.e., earliest possible time to start processing inbound truck $i \in \mathcal{I}$; $r_i \in \mathcal{T}$.
d_i	Due time, i.e., latest possible time to start processing inbound truck $i \in \mathcal{I}$; $d_i \in \mathcal{T}$.
κ_i	Number of operators required to process inbound truck $i \in \mathcal{I}$; $\kappa_i \in \mathbb{Z}^+$.
d_o	Time when processing of outbound truck $o \in \mathcal{O}$ starts; $d_o \in \mathcal{T}$.
p_{id}	Time for processing inbound truck $i \in \mathcal{I}$ at inbound dock-door $d \in \mathcal{D}$; $p_{id} \in \mathbb{Z}^+$.
f_{io}	Number of product units to be transferred from inbound truck $i \in \mathcal{I}$ to outbound truck $o \in \mathcal{O}$; $f_{io} \in \mathbb{Z}_0^+$.
$\overline{W}_g$	Maximum number of operators in operator group $g \in \mathcal{G}$.
C_{gs}	Cost of shift pattern $s \in \mathcal{S}_g$ with $g \in \mathcal{G}$.
γ_{gst}	Binary parameter for expressing the shift work time: 1, if time interval $t \in \mathcal{T}$ is work time in shift $s \in \mathcal{S}_g$; 0, otherwise.
α	Required minimum service level; $\alpha \in [0, 1]$.
Λ	Big number.
Decision variables:	
W_{gs}	Integer variable: Number of operators belonging to operator group $g \in \mathcal{G}$ deployed in shift pattern $s \in \mathcal{S}_g$.
x_{idt}	Binary variable: 1, if inbound truck $i \in \mathcal{I}$ is processed at inbound door $d \in \mathcal{D}$ and starts processing in time interval $t \in \mathcal{T}$; 0, otherwise.
y_{io}	Binary variable: 1, if processing inbound truck $i \in \mathcal{I}$ is finished before processing outbound truck $o \in \mathcal{O}$ starts; 0, otherwise.

Table 3.11 Notations for the model formulations of the **ISTSFD-F**.
Source: Own table.

ISTSFD-F-DT1:

$$\text{Minimize} \quad \sum_{g \in \mathcal{G}} \sum_{s \in \mathcal{S}_g} C_{gs} W_{gs} \tag{3.109}$$

$$\text{subject to} \quad \sum_{i \in \mathcal{I}} \sum_{d \in \mathcal{D}} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} \kappa_i x_{idt'} \leq \sum_{g \in \mathcal{G}} \sum_{s \in \mathcal{S}_g} \gamma_{gst} W_{gs} \qquad \forall t \in \mathcal{T} \tag{3.110}$$

$$\sum_{s \in \mathcal{S}_g} W_{gs} \leq \overline{W}_g \qquad \forall g \in \mathcal{G} \tag{3.111}$$

$$\sum_{d \in \mathcal{D}} \sum_{t=r_i}^{d_i} x_{idt} = 1 \qquad \forall i \in \mathcal{I} \tag{3.112}$$

$$\sum_{i \in \mathcal{I}} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} x_{idt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{3.113}$$

$$\sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} (t + p_{id} - 1) x_{idt} - d_o \leq \Lambda y_{io} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.114}$$

$$\sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} y_{io} \leq (1 - \alpha) \sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} \tag{3.115}$$

$$W_{gs} \in \mathbb{Z}^+ \qquad \forall g \in \mathcal{G}, s \in \mathcal{S}_g \tag{3.116}$$

$$x_{idt} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, t \in \mathcal{T} \tag{3.117}$$

$$y_{io} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.118}$$

The **ISTSFD-F-DT1's** big-M formulations in the form of constraints (3.114) can be eliminated by applying the preprocessing parameters a_{idt}. Recall that a_{idt}, which was already applied in the **TSFD-RC-F**, expresses the number of delayed freight units associated with truck i if it is processed at dock-door d and processing starts in time interval t. Equation (3.10) can be used in order to compute the preprocessing parameters. Hence, an alternative discrete-time model formulation for the **ISTSFD-F**, denoted as **ISTSFD-F-DT2**, can be formulated with the objective function (3.119) and the constraints (3.120) to (3.126).

ISTSFD-F-DT2:

$$\text{Minimize} \quad \sum_{g\in\mathcal{G}} \sum_{s\in\mathcal{S}_g} C_{gs} W_{gs} \tag{3.119}$$

$$\text{subject to} \quad \sum_{i\in\mathcal{I}} \sum_{d\in\mathcal{D}} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} \kappa_i x_{idt'} \leq \sum_{g\in\mathcal{G}} \sum_{s\in\mathcal{S}_g} \gamma_{gst} W_{gs} \qquad \forall t \in \mathcal{T} \tag{3.120}$$

$$\sum_{s\in\mathcal{S}_g} W_{gs} \leq \overline{W}_g \qquad \forall g \in \mathcal{G} \tag{3.121}$$

$$\sum_{d\in\mathcal{D}} \sum_{t=r_i}^{d_i} x_{idt} = 1 \qquad \forall i \in \mathcal{I} \tag{3.122}$$

$$\sum_{i\in\mathcal{I}} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} x_{idt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{3.123}$$

$$\sum_{i\in\mathcal{I}} \sum_{d\in\mathcal{D}} \sum_{t\in\mathcal{T}} a_{idt} x_{idt} \leq (1-\alpha) \sum_{i\in\mathcal{I}} \sum_{o\in\mathcal{O}} f_{io} \tag{3.124}$$

$$W_{gs} \in \mathbb{Z}^+ \qquad \forall g \in \mathcal{G}, s \in \mathcal{S}_g \tag{3.125}$$

$$x_{idt} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, t \in \mathcal{T} \tag{3.126}$$

The **ISTSFD-F-DT2** requires only constraint (3.124) to enforce the service level. Compared with the **ISTSFD-F-DT1**, it hence involves up to $|\mathcal{I}| \cdot |\mathcal{O}|$ fewer decision variables and constraints. Its objective function (3.119), resource constraints (3.120), upper limit on the operator number (3.121), truck-to-door assignment constraints (3.122), and no-overlap constraint (3.123), on the other hand, remain unchanged compared with the **ISTSFD-F-DT1**.

3.3.2.2 Multi-mode model

The **ISTSFD-F** can be extended in order to allow for multiple operator modes. The multi-mode shift and truck scheduling problem (**ISTSFD-V**) integrates the decision of **how many operators** should process each inbound truck. Considering the set of operator modes $\mathcal{K}$ adds an additional degree of freedom. Specifically, this allows to accelerate or slow down truck processing by deploying more operators or fewer operators, respectively. Applying the additional (or altered) notation defined in Table 3.12, the resulting mixed-integer program, denoted as **ISTSFD-V-DT1**, consists of objective function (3.127) and constraints (3.128) to (3.136).

Sets:	
$\mathcal{K}$	Set of operator modes.
Indices:	
k	Index used for operator modes, $k \in \mathcal{K}$.
Input parameters:	
p_{idk}	Time for processing inbound truck $i \in \mathcal{I}$ at inbound dock-door $d \in \mathcal{D}$ in operator mode $k \in \mathcal{K}$; $p_{idk} \in \mathbb{Z}^+$.
Decision variables:	
x_{idkt}	Binary variable: 1, if inbound truck $i \in \mathcal{I}$ is processed at inbound door $d \in \mathcal{D}$ in operator mode $k \in \mathcal{K}$ and starts processing in time interval $t \in \mathcal{T}$; 0, otherwise.

Table 3.12 Additional and altered notations for the model formulations of the **ISTSFD-V**.
Source: Own table.

ISTSFD-V-DT1:

$$\text{Minimize} \quad \sum_{g \in \mathcal{G}} \sum_{s \in \mathcal{S}_g} C_{gs} W_{gs} \tag{3.127}$$

$$\text{subject to} \quad \sum_{i \in \mathcal{I}} \sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} \sum_{t'=\max\{0; t-p_{idk}+1\}}^{t} \kappa_{ik} x_{idkt'} \leq \sum_{g \in \mathcal{G}} \sum_{s \in \mathcal{S}_g} \gamma_{gst} W_{gs} \qquad \forall t \in \mathcal{T} \tag{3.128}$$

$$\sum_{s \in \mathcal{S}_g} W_{gs} \leq \overline{W}_g \qquad \forall g \in \mathcal{G} \tag{3.129}$$

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} \sum_{t=r_i}^{d_i} x_{idkt} = 1 \qquad \forall i \in \mathcal{I} \tag{3.130}$$

$$\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} \sum_{t'=\max\{0; t-p_{idk}+1\}}^{t} x_{idkt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{3.131}$$

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} (t + p_{idk} - 1) x_{idkt} - d_o \leq \Lambda y_{io} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.132}$$

$$\sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} y_{io} \leq (1 - \alpha) \sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} \tag{3.133}$$

$$W_{gs} \in \mathbb{Z}^+ \qquad \forall g \in \mathcal{G}, s \in \mathcal{S}_g \tag{3.134}$$

$$x_{idkt} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, k \in \mathcal{K}, t \in \mathcal{T} \tag{3.135}$$

$$y_{io} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{3.136}$$

The **ISTSFD-V-DT1** seeks to minimize the total labor cost for activated operators from all operator groups (3.127). While constraints (3.128) enforce that the number of active operators (i.e., operators on site) suffices, constraints (3.129) ensure that the maximum number workforce is not overextended. The assignment constraints (3.130) enforce that each inbound truck is processed and that truck processing starts within a truck's time window. Constraints (3.131) forbid parallel processing at a dock-door, while constraints (3.132) in combination with constraint (3.133) enforce the required service level by limiting the number of delayed freight units. Lastly, constraints (3.134) to (3.136) define the domain of the decision variables W_{gs}, x_{idkt}, and y_{io}.

Note that the **ISTSFD-V-DT1** involves big-M constraints for calculating the number of delayed freight units (3.132). As in the case of the **ISTSFD-F-DT1**, these constraints may result in a weak relaxation and, hence, make it more challenging to solve instances of the **ISTSFD-V-DT1** with an off-the-shelf solver. Therefore, an alternative discrete-time formulation, denoted as **ISTSFD-V-DT2**, which spares out big-M formulations, is proposed below. It applies preprocessing parameters in order to compute the delayed freight units and compel the service level. The preprocessing parameters a_{idkt} signal how many of inbound truck i's freight units cannot be loaded onto the designated outbound trucks since they arrive too late in the outbound area. The parameters can be computed with equation (3.65). The **ISTSFD-V-DT2** consists of objective function (3.137) and constraints (3.138) to (3.144). Note that **ISTSFD-V-DT2's** objective function (3.137), resource constraints (3.138), upper limit on the number of deployed operators (3.139), truck assignment constraints (3.140), and no-overlap constraints (3.141) remain unchanged compared with the **ISTSFD-V-DT1**. The service level constraint (3.142), however, includes the preprocessing parameters a_{idkt}. The **ISTSFD-V-DT2** consists of up to $|\mathcal{I}| \cdot |\mathcal{O}|$ fewer decision variables and constraints than the **ISTSFD-V-DT1**.

ISTSFD-V-DT2:

$$\text{Minimize} \quad \sum_{g\in\mathcal{G}}\sum_{s\in\mathcal{S}_g} C_{gs}W_{gs} \tag{3.137}$$

$$\text{subject to} \quad \sum_{i\in\mathcal{I}}\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t'=\max\{0;t-p_{idk}+1\}}^{t} \kappa_{ik}x_{idkt'} \leq \sum_{g\in\mathcal{G}}\sum_{s\in\mathcal{S}_g}\gamma_{gst}W_{gs} \qquad \forall t\in\mathcal{T} \tag{3.138}$$

$$\sum_{s\in\mathcal{S}_g} W_{gs} \leq \overline{W}_g \qquad \forall g\in\mathcal{G} \tag{3.139}$$

$$\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t=r_i}^{d_i} x_{idkt} = 1 \qquad \forall i\in\mathcal{I} \tag{3.140}$$

$$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}}\sum_{t'=\max\{0;t-p_{idk}+1\}}^{t} x_{idkt'} \leq 1 \qquad \forall t\in\mathcal{T}, d\in\mathcal{D} \tag{3.141}$$

$$\sum_{i\in\mathcal{I}}\sum_{d\in\mathcal{D}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} a_{idkt}x_{idkt} \leq (1-\alpha)\sum_{i\in\mathcal{I}}\sum_{o\in\mathcal{O}} f_{io} \tag{3.142}$$

$$W_{gs}\in\mathbb{Z}^+ \qquad \forall g\in\mathcal{G}, s\in\mathcal{S}_g \tag{3.143}$$

$$x_{idkt}\in\{0;1\} \qquad \forall i\in\mathcal{I}, d\in\mathcal{D}, k\in\mathcal{K}, t\in\mathcal{T} \tag{3.144}$$

3.3.3 Complexity

Table 3.13 provides the equations for calculating the number of decision variables and constraints of the single-mode and multi-mode shift and truck scheduling formulations.

Model formulation	$\#^{DV}$	$\#^{C}$
ISTSFD-F-DT1	$\|\mathcal{I}\|\cdot(\|\mathcal{D}\|\cdot\|\mathcal{T}\|+\|\mathcal{O}\|)+\sum_{g\in\mathcal{G}}\|\mathcal{S}_g\|$	$\|\mathcal{T}\|\cdot(\|\mathcal{D}\|+1)+\|\mathcal{I}\|\cdot(\|\mathcal{O}\|+1)+\|\mathcal{G}\|+1$
ISTSFD-F-DT2	$\|\mathcal{I}\|\cdot\|\mathcal{D}\|\cdot\|\mathcal{T}\|+\sum_{g\in\mathcal{G}}\|\mathcal{S}_g\|$	$\|\mathcal{T}\|\cdot(\|\mathcal{D}\|+1)+\|\mathcal{I}\|+\|\mathcal{G}\|+1$
ISTSFD-V-DT1	$\|\mathcal{I}\|\cdot(\|\mathcal{D}\|\cdot\|\mathcal{K}\|\cdot\|\mathcal{T}\|+\|\mathcal{O}\|)+\sum_{g\in\mathcal{G}}\|\mathcal{S}_g\|$	$\|\mathcal{T}\|\cdot(\|\mathcal{D}\|+1)+\|\mathcal{I}\|\cdot(\|\mathcal{O}\|+1)+\|\mathcal{G}\|+1$
ISTSFD-V-DT2	$\|\mathcal{I}\|\cdot\|\mathcal{D}\|\cdot\|\mathcal{K}\|\cdot\|\mathcal{T}\|+\sum_{g\in\mathcal{G}}\|\mathcal{S}_g\|$	$\|\mathcal{T}\|\cdot(\|\mathcal{D}\|+1)+\|\mathcal{I}\|+\|\mathcal{G}\|+1$

$\#^{DV}$ and $\#^{C}$ denote the number of decision variables and the number of constraints, respectively.

Table 3.13 Number of decision variables and constraints for different model formulations of the **ISTSFD**.
Source: Own table.

The **ISTSFD-F's** and **ISTSFD-V's** model dimensions are slightly bigger compared with the discrete-time **TSFD-RC-F** and **TSFD-RC-V**, respectively. Both the **ISTSFD-F** and **ISTSFD-V** contain $\sum_{g \in \mathcal{G}} |\mathcal{S}_g| - 1$ more decision variables and $|\mathcal{G}|$ more constraints than their **TSFD-RC-F** and **TSFD-RC-V** counterparts. Hence, in most real-world settings (where usually only a few operator groups and shift patterns are considered), the number of decision variables and constraints does not increase dramatically in comparison with the single-mode and multi-mode resource and truck scheduling problem. However, that also means that the **ISTSFD** may involve millions of decision variables for large-sized instances, which likely prevents large-sized instances from being solved with a default solver in a reasonable time. The following theorems and corollaries on the complexity status of the **ISTSFD** support this speculation.

Theorem 3.3.1. *Finding a feasible solution for the **ISTSFD-F** is strongly NP-complete for* $|\mathcal{D}| \geq 1$.

Theorem 3.3.2. *Finding a feasible solution for the **ISTSFD-V** is strongly NP-complete for* $|\mathcal{D}| \geq 1$.

Theorem 3.3.1 (Theorem 3.3.2) directly follows from the **ISTSFD-F** (**ISTSFD-V**) being a generalization of the **TSFD-RC-F** (**TSFD-RC-V**), which was proven to be strongly NP-complete in Theorem 3.1.1 (Theorem 3.2.1). Note that the feasibility of a given solution to the **ISTSFD-F** and **ISTSFD-V** can be tested in polynomial time.

Corollary 3.3.3. *The **ISTSFD-F** is strongly NP-hard.*

Corollary 3.3.4. *The **ISTSFD-V** is strongly NP-hard.*

3.4 Chapter summary

This chapter presented mathematical programs for scheduling resources and trucks in cross-docking facilities. First, the NP-hard resource and truck scheduling problem (**TSFD-RC-F**) was described in Section 3.1. Given a set of inbound trucks with known operator requirements to process over the planning horizon, the **TSFD-RC-F** seeks to determine at what time each truck should be processed in order to keep the demand for operators low over the planning horizon. A total of four mixed-integer programming formulations, two discrete-time and two continuous-time formulations, for the **TSFD-RC-F** were proposed. Next, the multi-mode resource and truck scheduling problem (**TSFD-RC-V**), also proven to be NP-hard, was described in Section 3.2.

Unlike the **TSFD-RC-F**, the **TSFD-RC-V** does not assume that the operator requirements for each truck are given. Instead, it aims to determine by how many operators and at what time each truck should be processed in order to minimize the demand for operators over the planning horizon. Again, discrete-time and continuous-time mixed integer programs were proposed. Lastly, the shift and truck scheduling problem (**ISTSFD**), which enables decision-makers to consider different operator groups and shift patterns, was described in Section 3.3. The **ISTSFD's** goal is to minimize the total labor cost for operators. Both a single-mode (**ISTSFD-F**) and a multi-mode (**ISTSFD-V**) variant for shift and truck scheduling were presented.

4 Solution procedures

4.1 Introduction

The complexity statuses of the discussed resource planning models make it difficult to solve large-sized instances with a default solver. Therefore, this chapter presents suited solution procedures that can be used to tackle the proposed planning problems. Specifically, this chapter sets out to develop heuristic solution procedures based on a column generation scheme. Column generation (CG) is often used when dealing with LP-relaxations of mixed-integer programs that contain too many columns (associated with variables) to handle explicitly – such as the proposed discrete-time formulations in this work. The basic idea is that instead of considering all potential columns right away, a restricted model that only contains a (small) subset of columns is defined. New columns that may improve the relaxation value are iteratively generated (by solving a pricing problem) and added to the restricted model. However, the majority of all possible columns are left out of the restricted LP-relaxation since most of these columns will very likely not be part of the optimal solution. The LP-relaxation of the restricted model has been solved to optimality once no columns with negative reduced costs, that is, columns that may improve the relaxation value, are excluded from the restricted model.[1)] Column generation has been successfully applied to a wide range of scheduling problems including truck scheduling[2)], parallel machine scheduling[3)], staff scheduling[4)], airline crew scheduling[5)], and resource-constrained project scheduling[6)]. A comprehensive overview on column generation

1) Chen et al. (2010, pp. 334-335).
2) E.g., Chmielewski et al. (2009).
3) E.g., Chen and Powell (1999), Chen (2004), Van Den Akker et al. (1999), Van Den Akker et al. (2012).
4) E.g., Fügener and Brunner (2019).
5) E.g., Zeighami and Soumis (2019).
6) E.g., Brucker and Knust (2003).

can be found in Desaulniers et al.[1]

The general structure of the column generation-based solution procedure, which will be explained in more detail below, is shown in Figure 4.1. The solution procedure defines a restricted problem, i.e., the LP-relaxation of the discrete-time model where only a subset $\overline{\mathcal{M}}$ of columns is included. Once the set $\overline{\mathcal{M}}$ has been initialized, absent columns that may improve the relaxation value are added iteratively using an approach based on column generation. In every iteration, the LP-relaxation of the restricted problem is solved to optimality with a default solver. The obtained optimal dual multipliers are passed to a pricing problem[2]. The pricing problem aims to identify at least one absent column with negative reduced cost or prove that no such column exists. If the optimal objective value of the pricing problem is negative, the corresponding column is added to the set $\overline{\mathcal{M}}$ and, hence, to the restricted problem. Once the appropriate columns have been added to the restricted problem, the LP-relaxation of the restricted problem is solved again, and the pricing process continues with the newly obtained optimal dual multipliers. The generation of new columns terminates when there are no more columns with negative reduced cost. Moreover, it is well known that the optimal value of the LP-relaxation of the restricted problem is equivalent to the optimal value of the LP-relaxation of the original problem once the column generation has run its course. In light of the restricted problem being an LP-relaxation, however, there is no guarantee that the solution obtained in the last iteration is integral. Therefore, integrality on the decision variables is reintroduced, and the restricted problem with the final column set $\overline{\mathcal{M}}$ is solved in order to obtain an integral solution.[3]

The remainder of this chapter is structured as follows. Section 4.2 presents a column generation-based solution procedure for the resource and truck scheduling problem (**TSFD-RC-F**). Specifically, the computation of initial columns and the generation of absent columns, which may improve the relaxation value, are described in more detail. Similarly, details regarding the heuristic solution procedures for the multi-mode resource and truck scheduling problem (**TSFD-RC-V**) and the shift and truck scheduling problem (**ISTSFD-V**) are presented in Section 4.3 and Section 4.4, respectively. The chapter ends with a short summary and concluding remarks provided in Section 4.5.

[1] Desaulniers et al. (2005).

[2] Note that there may also be multiple pricing problems. In the following, however, it is referred to as the pricing problem.

[3] Emde et al. (2020, p. 412).

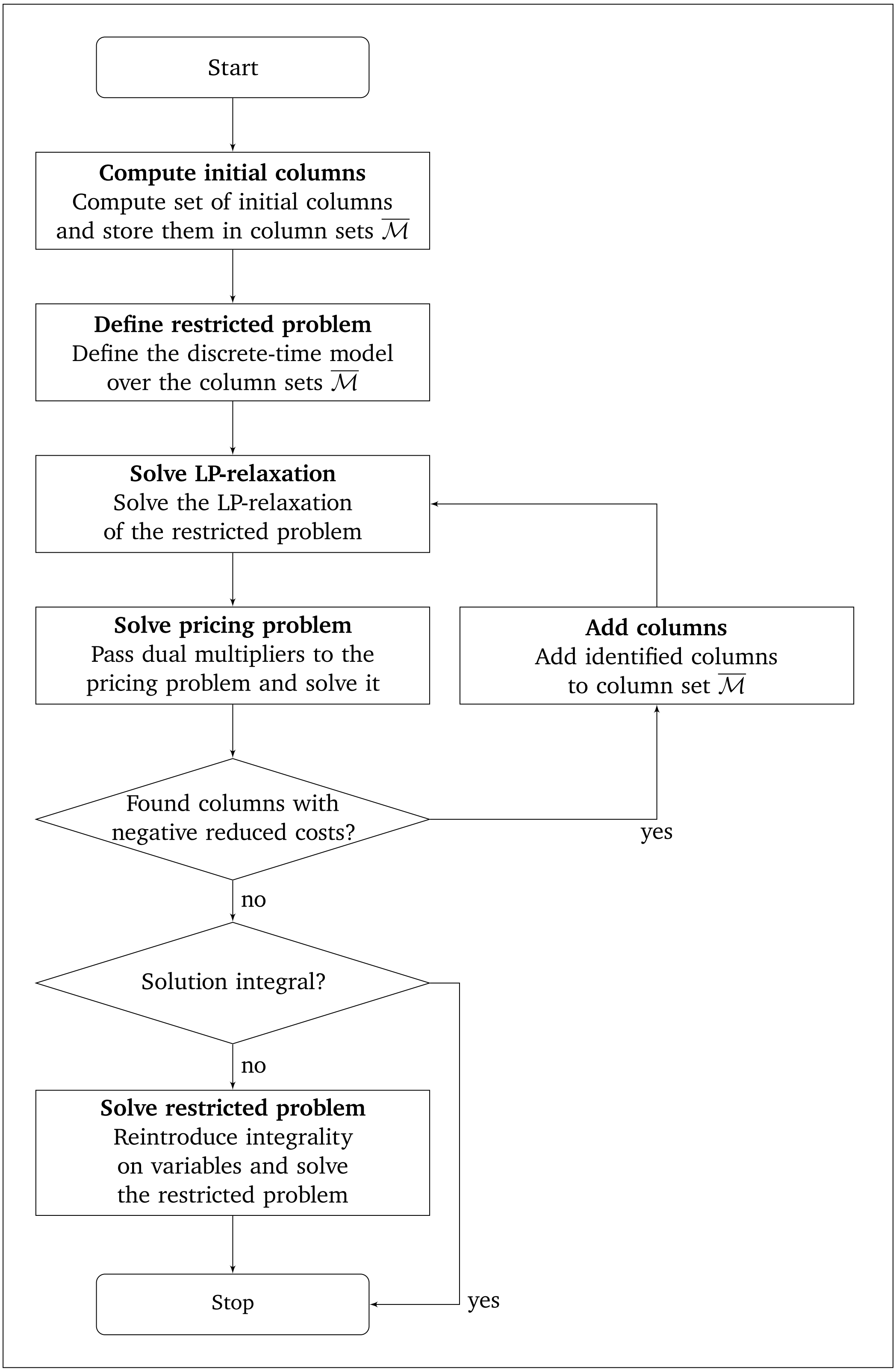


Figure 4.1 General structure of the column generation-based solution procedures.
Source: Own figure.

4.2 Resource and truck scheduling problem

In this section, a column generation-based solution procedure for the resource and truck scheduling problem (**TSFD-RC-F**) is proposed. Algorithm 1 summarizes the heuristic procedure.

Algorithm 1: Solution procedure for the **TSFD-RC-F**

Data: **TSFD-RC-F-DT2** and an empty column set $\overline{\mathcal{M}}$.

Result: Integer solution for the **TSFD-RC-F-DT2**.

1 Define the **R(TSFD-RC-F-DT2)** over the column set $\overline{\mathcal{M}}$;
2 Compute set of initial columns and add them to column set $\overline{\mathcal{M}}$;
3 Solve LP-relaxation of the **R(TSFD-RC-F-DT2)**;
4 Pass dual multipliers to the pricing problem and solve it;
5 **if** *columns with negative reduced costs were found* **then**
6   Add the columns to column set $\overline{\mathcal{M}}$;
7   Go back to step 3;
8 **if** *current solution fractional* **then**
9   Reintroduce integrality on variables and solve **R(TSFD-RC-F-DT2)**;

Specifically, the time-discrete model formulation **TSFD-RC-F-DT2** is used as a basis. First, a restricted version, denoted as **R(TSFD-RC-F-DT2)**, is defined over the column set $\overline{\mathcal{M}}$ (line 1). After adding a set of initial columns to the column set $\overline{\mathcal{M}}$ (line 2), the LP-relaxation of the **R(TSFD-RC-F-DT2)** is solved with a column generation procedure (line 3 to 7). If this results in a fractional solution, integrality on the decision variables is reintroduced in order to obtain an integer solution for the **TSFD-RC-F** (line 8 to 9). This section provides details concerning the computation of initial columns and the column generation procedure.

4.2.1 Initial columns

Prior to starting the column generation procedure, a non-empty set of initial columns is required. This section presents two procedures for computing initial columns for the **TSFD-RC-F-DT2**. First, a MIP formulation is presented which allows to generate a feasible truck schedule to the **TSFD-RC-F**. This schedule can be stored in the column set $\overline{\mathcal{M}}$. Afterwards, a heuristic approach for generating initial columns is presented. Both procedures can either be applied individually or in a sequential manner.

4.2.1.1 Generating initial columns via MIP

Even though it is not strictly necessary, having a feasible initial solution in the column set $\overline{\mathcal{M}}$ before starting the column generation procedure can be helpful. For instance, it can be guaranteed that the heuristic procedure terminates with a feasible integer solution. A feasible solution could be computed by solving a feasibility problem of the **TSFD-RC-F**, i.e., by simply setting the **TSFD-RC-F's** objective function to 1 and dropping the resource constraints. However, this approach performs poorly according to the pretests. The pretests revealed that when solving the integer program **TSFD-U-F**, with the objective function (4.1) and constraints (4.2) to (4.4), a feasible solution to the **TSFD-RC-F** can be identified in a shorter time.

TSFD-U-F:

$$\text{Minimize} \quad \sum_{i\in\mathcal{I}}\sum_{d\in\mathcal{D}}\sum_{t\in\mathcal{T}} a_{idt}x_{idt} \tag{4.1}$$

$$\text{subject to} \quad \sum_{d\in\mathcal{D}}\sum_{t=r_i}^{d_i} x_{idt} = 1 \qquad \forall i \in \mathcal{I} \tag{4.2}$$

$$\sum_{i\in\mathcal{I}}\sum_{t'=\max\{0;t-p_{id}+1\}}^{t} x_{idt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{4.3}$$

$$x_{idt} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, t \in \mathcal{T} \tag{4.4}$$

The **TSFD-U-F** drops the **TSFD-RC-F's** service level constraint and tries to find a truck schedule with a minimum number of delayed product units instead (4.1). That is, the objective function guarantees that the predefined service level α is met. Constraints (4.2) assure that each inbound truck is processed within its time window. Inequalities (4.3) guarantee that at most one inbound truck at a time is processed at a dock-door. Finally, constraints (4.4) represent the binary integrality requirement of 0-1 variables. Note that the **TSFD-RC-F's** resource constraints are not included. An optimal solution to the **TSFD-U-F** is a feasible solution to the **TSFD-RC-F** and hence can be added to column set $\overline{\mathcal{M}}$ and the **R(TSFD-RC-F-DT2)**. Note that this approach adds $|\mathcal{I}|$ columns to the **R(TSFD-RC-F-DT2)**, i.e., the least number of columns required.

4.2.1.2 Generating initial columns heuristically

Solving the **TSFD-U-F** is not a trivial task. Especially for large problem instances, it can require a long computational time. Therefore, a procedure which randomly adds "good" initial columns to the empty set $\overline{\mathcal{M}}$ is proposed in this section. Algorithm 2 summarizes the procedure.

Algorithm 2: Generating the set of initial columns for the **TSFD-RC-F**

Data: Empty column set $\overline{\mathcal{M}}$.
Result: Non-empty column set $\overline{\mathcal{M}}$.

1 **for** $i := 1$ **to** $|\mathcal{I}|$ **do**
2     $\overline{\mathcal{T}}_i \leftarrow \{r_i\}$;
3   **repeat**
4   Randomly select a time interval $t \in \{r_i + 1, \dots, d_i\} \setminus \overline{\mathcal{T}}_i$;
5         $\overline{\mathcal{T}}_i \leftarrow \overline{\mathcal{T}}_i \cup \{t\}$;
6   **until** $|\overline{\mathcal{T}}_i| \geq 1 + \left\lfloor \frac{|\{r_i+1,\dots,d_i\}|}{4} \right\rfloor$;
7     $p_i^B \leftarrow \min_{d \in \mathcal{D}}\{p_{id}\} + 0.1 \cdot (\max_{d \in \mathcal{D}}\{p_{id}\} - \min_{d \in \mathcal{D}}\{p_{id}\})$;
8   **for** $d := 1$ **to** $|\mathcal{D}|$ **do**
9   **if** $p_{id} \leq p_i^B$ **then**
10   Add columns $m = (i, d, t)$ with $t \in \overline{\mathcal{T}}_i$ to column set $\overline{\mathcal{M}}$;

First, a set of possible start times $\overline{\mathcal{T}}_i$ is determined for a given truck i (line 2 to 6). The set includes the release time r_i, which is expected to be of high relevance in many cases, as well as $\left\lfloor \frac{|\{r_i+1,\dots,d_i\}|}{4} \right\rfloor$ randomly chosen feasible start times from $\{r_i + 1, \dots, d_i\}$. That is, ca. 25% of all time intervals in which truck i could start, are included in set $\overline{\mathcal{T}}_i$. The quality of a column is mainly evaluated based on the processing time. Specifically, a column $m = (i, d, t)$ is considered as being "good", if it is characterized by a short truck processing time p_{id}. The upper limit p_i^B, up to which the processing time is considered as being short, is calculated for a given truck i (line 7). Based on that, all columns $m = (i, d, t)$ with $p_{id} \leq p_i^B$ and $t \in \overline{\mathcal{T}}_i$ are added to the column set $\overline{\mathcal{M}}$ (line 8 to 10).

4.2.2 Column generation

The LP-relaxation of the **TSFD-RC-F-DT2** can be solved with its restricted version **R(TSFD-RC-F-DT2)** (defined over column set $\overline{\mathcal{M}}$) and a column generation procedure. It is well known from optimization theory, that the optimal value of the LP-relaxation of the **R(TSFD-RC-F-DT2)** will be equivalent to the optimal value of the LP-relaxation of the **TSFD-RC-F-DT2**, if there are no columns with negative reduced costs excluded from the **R(TSFD-RC-F-DT2)**, i.e., such columns that may improve the relaxation value. A mathematical programming model for finding an improving column $m = (i, d, t)$, if one exists, defines the **pricing problem**. The pricing problem requires the dual multipliers of the relaxed **R(TSFD-RC-F-DT2)**. Using the dual variables λ_t for the resource constraints (3.12), ν_i for the truck assignment constraints (3.13), μ_{td} for the no-overlap constraints (3.14), , and π for the service level constraint (3.15), the dual of the relaxed **R(TSFD-RC-F-DT2)**, denoted as **d-R(TSFD-RC-F-DT2)**, is as follows:

d-R(TSFD-RC-F-DT2):

$$\text{Maximize} \quad \sum_{i\in\mathcal{I}} \nu_i - \sum_{t\in\mathcal{T}} \sum_{d\in\mathcal{D}} \mu_{td} - \pi(1-\alpha) \sum_{i\in\mathcal{I}} \sum_{o\in\mathcal{O}} f_{io} \tag{4.5}$$

$$\text{subject to} \quad \sum_{t\in\mathcal{T}} \lambda_t \leq 1 \tag{4.6}$$

$$\nu_i - \sum_{t'=t}^{t+p_{id}-1} \kappa_i \lambda_{t'} - \sum_{t'=t}^{t+p_{id}-1} \mu_{t'd} - a_{idt}\pi \leq 0 \qquad \forall (i,d,t) \in \overline{\mathcal{M}} \tag{4.7}$$

$$\lambda_t \in \mathbb{R}^+ \qquad \forall t \in \mathcal{T} \tag{4.8}$$

$$\mu_{td} \in \mathbb{R}^+ \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{4.9}$$

$$\nu_i \in \mathbb{R}^+ \qquad \forall i \in \mathcal{I} \tag{4.10}$$

$$\pi \in \mathbb{R}^+ \tag{4.11}$$

The reduced costs R_m of column $m = (i,d,t)$ can be computed by equation (4.12).

$$R_m = a_{idt}\pi + \sum_{t'=t}^{t+p_{id}-1} \kappa_i \lambda_{t'} + \sum_{t'=t}^{t+p_{id}-1} \mu_{t'd} - \nu_i \tag{4.12}$$

According to the primal simplex method, finding and adding an absent column with negative reduced cost could improve a current optimal solution to the LP-relaxation of the **R(TSFD-RC-F-DT2)**. For pricing out absent columns with negative reduced cost, if any exist, a pricing subproblem needs to be solved. The pricing problem can be decomposed by truck and dock-door into $|\mathcal{I}| \cdot |\mathcal{D}|$ pricing subproblems[1]. The pricing subproblem for the truck-door pair (i,d), referred to as **TSFD-RC-F-PP(i,d)**, can be formulated as follows:

[1] Other decompositions, e.g., by truck, are possible too.

TSFD-RC-F-PP(i,d):

$$\text{Minimize} \quad \pi \cdot \left(\sum_{t=r_i}^{d_i} a_{idt} x_t \right) + \sum_{t=r_i}^{d_i+p_{id}-1} \left[\lambda_t \kappa_i \cdot \left(\sum_{t'=\max\{r_i;t-p_{id}+1\}}^{t} x_{t'} \right) \right] + \sum_{t=r_i}^{d_i+p_{id}-1} \left[\mu_{td} \cdot \left(\sum_{t'=\max\{r_i;t-p_{id}+1\}}^{t} x_{t'} \right) \right] - \nu_i \tag{4.13}$$

$$\text{subject to} \quad \sum_{t=r_i}^{d_i} x_t = 1 \tag{4.14}$$

$$x_t \in \{0;1\} \qquad \forall t \in \{r_i, \ldots, d_i\} \tag{4.15}$$

Objective function (4.13) seeks to identify an absent column with minimum reduced cost. That is, the time interval t that minimizes the reduced cost must be found. Therefore, a set of binary decision variables x_t is defined so that $x_t = 1$ if processing truck i on dock-door d starts in time interval t. Constraint (4.14) compels that processing inbound truck i starts exactly once.

Note that it may be necessary to solve up to $|\mathcal{I}| \cdot |\mathcal{D}|$ mixed integer programs in every iteration to identify a column with negative reduced costs or to prove optimality for the LP-relaxation of the **R(TSFD-RC-F-DT2)**. Hence, this approach may be computationally expensive, especially for big problem instances. Next, I therefore propose an efficient algorithm that I have implemented to solve the pricing problem and add generated columns with negative reduced cost to the **R(TSFD-RC-F-DT2)**. The column generation procedure is shown in Algorithm 3.

The algorithm basically consists of two major phases: (i) Updating the dual multipliers λ_t, μ_{td}, ν_i, and π by solving the LP-relaxation of the **R(TSFD-RC-F-DT2)** (line 2), and (ii) finding absent columns with negative reduced cost, if any exist, and adding them to the **R(TSFD-RC-F-DT2)** (line 3 to 16). Both phases are executed alternately and the procedure stops once no new columns which can improve the optimal value of the LP-relaxation of the **R(TSFD-RC-F-DT2)** can be found (line 17). The generation of new columns is designed in a computationally efficient way. Specifically, I decompose the search space and determine separately for every truck-door-pair (i,d) the column $m^*_{id} \in \{\tilde{m} = (i, d, \tilde{t}) \mid r_i \leq \tilde{t} \leq d_i\}$ with the minimum reduced cost R^*_{id} (line 6 to 13). In other words, the "best" time interval to start processing truck i on dock-door d, i.e. the time interval that results in minimum reduced costs, is identified. All columns m^*_{id} with $R^*_{id} < 0$ are temporarily stored in the set $\mathcal{N}$ (line 14 to 15) and added to the **R(TSFD-RC-F-DT2)** at the end of each iteration (line 16). In every iteration, the algorithm

Algorithm 3: Column generation procedure for the **TSFD-RC-F**

Data: **R(TSFD-RC-F-DT2)** and the initial column set $\overline{\mathcal{M}}$.

Result: **R(TSFD-RC-F-DT2)** including all columns such that the optimal value of the relaxed **R(TSFD-RC-F-DT2)** is equivalent to the optimal value of the relaxed **TSFD-RC-F-DT2**.

repeat
  Solve the LP-relaxation of the **R(TSFD-RC-F-DT2)** over the set $\overline{\mathcal{M}}$ and store the values of the dual variables λ_t, μ_{td}, ν_i, and π;
  $\mathcal{N} \leftarrow \emptyset$;
  for $i := 1$ **to** $|\mathcal{I}|$ **do**
    for $d := 1$ **to** $|\mathcal{D}|$ **do**
      $\tilde{R} \leftarrow a_{i,d,r_i}\pi + \sum_{t=r_i}^{r_i+p_{id}-1} \kappa_i\lambda_t + \sum_{t=r_i}^{r_i+p_{id}-1} \mu_{td} - \nu_i$;
      $m^*_{id} \leftarrow (i, d, r_i)$;
      $R^*_{id} \leftarrow \tilde{R}$;
      for $t := r_i + 1$ **to** d_i **do**
        $\tilde{R} \leftarrow \tilde{R} + (a_{i,d,t+p_{id}-1} - a_{i,d,t-1})\pi + \kappa_i(\lambda_{t+p_{id}-1} - \lambda_{t-1}) + (\mu_{t+p_{id}-1,d} - \mu_{t-1,d})$;
        if $\tilde{R} < R^*_{id}$ **then**
          $m^*_{id} \leftarrow (i, d, t)$;
          $R^*_{id} \leftarrow \tilde{R}$;
      if $R^*_{id} < 0$ **then**
        $\mathcal{N} \leftarrow \mathcal{N} \cup \{m^*_{id}\}$;
  $\overline{\mathcal{M}} \leftarrow \overline{\mathcal{M}} \cup \mathcal{N}$;
until $\mathcal{N} = \emptyset$;

guarantees to identify the column with the minimum reduced cost in $O(|\mathcal{I}| \cdot |\mathcal{D}| \cdot |\mathcal{T}|)$ and up to $|\mathcal{I}| \cdot |\mathcal{D}|$ columns are added to the **R(TSFD-RC-F-DT2)**.

Once the column generation procedure terminates, the LP-relaxation of both the **R(TSFD-RC-F-DT2)** and **TSFD-RC-F-DT2** are solved to optimality. If the obtained solution only consists of integer variables, the optimal solution to the **TSFD-RC-F-DT2** is found. If the solution is fractional, on the other hand, integrality on the decision variables is reintroduced and the **R(TSFD-RC-F-DT2)** is solved to optimality as an integer program. By doing so, a feasible integer solution can be obtained. Note, however, that in this case the obtained solution to the **R(TSFD-RC-F-DT2)** is not necessarily the optimal solution to the **TSFD-RC-F-DT2**. Hence, the proposed procedure is a heuristic solution procedure.

4.3 Multi-mode resource and truck scheduling problem

In this section, a column generation-based solution procedure for the multi-mode resource and truck scheduling problem (**TSFD-RC-V**) is proposed. Algorithm 4 summarizes the heuristic procedure. Specifically, the time-discrete model formulation **TSFD-RC-V-DT2** is used as a basis. First, a restricted version, denoted as **R(TSFD-RC-V-DT2)**, is defined over the column set $\overline{\mathcal{M}}$ (line 1). After adding a set of initial columns to the column set $\overline{\mathcal{M}}$ (line 2), the LP-relaxation of the **R(TSFD-RC-V-DT2)** is solved with a column generation procedure (line 3 to 7). If this results in a fractional solution, integrality on the decision variables is reintroduced in order to obtain an integer solution for the **TSFD-RC-V** (line 8 to 9). This section provides details concerning the computation of initial columns and the column generation procedure.

Algorithm 4: Solution procedure for the **TSFD-RC-V**

Data: **TSFD-RC-V-DT2** and an empty column set $\overline{\mathcal{M}}$.
Result: Integer solution for the **TSFD-RC-V-DT2**.

1 Define the **R(TSFD-RC-V-DT2)** over the column set $\overline{\mathcal{M}}$;
2 Compute set of initial columns and add them to column set $\overline{\mathcal{M}}$;
3 Solve LP-relaxation of the **R(TSFD-RC-V-DT2)**;
4 Pass dual multipliers to the pricing problem and solve it;
5 **if** *columns with negative reduced costs were found* **then**
6   Add the columns to column set $\overline{\mathcal{M}}$;
7   Go back to step 3;
8 **if** *current solution fractional* **then**
9   Reintroduce integrality on variables and solve **R(TSFD-RC-V-DT2)**;

4.3.1 Initial columns

In this section, two approaches for computing initial columns for the **R(TSFD-RC-V-DT2)** are presented. The first approach computes an initial column set by solving a MIP with a default solver. The obtained solution is a feasible solution to the **TSFD-RC-V**. The second approach, on the other hand, heuristically generates initial columns that can be used in the **R(TSFD-RC-V-DT2)**. Both procedures can either be applied individually or in a sequential manner.

4.3.1.1 Generating initial columns via MIP

It can be beneficial to store a feasible solution in the restricted problem **R(TSFD-RC-V-DT2)** before starting the column generation procedure. Such a feasible solution can be obtained by solving the **TSFD-U-V** with the objective function (4.16) and constraints (4.17) to (4.19).

TSFD-U-V:

$$\text{Minimize} \quad \sum_{i\in\mathcal{I}} \sum_{d\in\mathcal{D}} \sum_{k=|\mathcal{K}|} \sum_{t\in\mathcal{T}} a_{idkt} x_{idkt} \tag{4.16}$$

$$\text{subject to} \quad \sum_{d\in\mathcal{D}} \sum_{k=|\mathcal{K}|} \sum_{t=r_i}^{d_i} x_{idkt} = 1 \qquad \forall i \in \mathcal{I} \tag{4.17}$$

$$\sum_{i\in\mathcal{I}} \sum_{k=|\mathcal{K}|} \sum_{t'=\max\{0;t-p_{idk}+1\}}^{t} x_{idkt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{4.18}$$

$$x_{idkt} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, k = |\mathcal{K}|, t \in \mathcal{T} \tag{4.19}$$

The **TSFD-U-V** is an extended version of the **TSFD-U-F**. Specifically, it only considers the fastest operator mode $k = |\mathcal{K}|$. This results in a smaller search space and may reduce the computation time. The **TSFD-U-V** aims to minimize the number of delayed product units (4.16). By solving the **TSFD-U-V** to optimality, hence, a solution with the highest possible service level is identified. This solution satisfies the service level constraint of the original multi-mode resource and truck scheduling problem **TSFD-RC-V**. Constraints (4.17) assure that each inbound truck is processed within its time window. Moreover, constraints (4.18) prevent overlaps, i.e., that multiple trucks are processed in parallel at a dock-door. The obtained solution includes $|\mathcal{I}|$ columns and can be added to the **R(TSFD-RC-V-DT2)**.

4.3.1.2 Generating initial columns heuristically

Solving the **TSFD-U-V** with a default solver can require a long computational time – especially for large problem instances. Therefore, an alternative procedure for determining an initial column set is presented. The procedure identifies "good" columns $m = (i, d, k, t)$ and adds them to set $\overline{\mathcal{M}}$. Algorithm 5 shows the pseudocode. For a given inbound truck i, its release time r_i and $\left\lfloor \frac{|\{r_i+1,\ldots,d_i\}|}{4} \right\rfloor$ randomly chosen start times from $\{r_i + 1, \ldots, d_i\}$ are stored as candidate start times in set $\overline{\mathcal{T}}_i$ (line 2 to 6). That is, ca. 25% of all time intervals in which truck i could

Algorithm 5: Generating the set of initial columns for the **TSFD-RC-V**

Data: Empty column sets $\overline{\mathcal{M}}$.
Result: Non-empty column set $\overline{\mathcal{M}}$.

1 **for** $i := 1$ **to** $|\mathcal{I}|$ **do**
2     $\overline{\mathcal{T}}_i \leftarrow \{r_i\}$;
3   **repeat**
4   Randomly select a time interval $t \in \{r_i + 1, \ldots, d_i\} \setminus \overline{\mathcal{T}}_i$;
5         $\overline{\mathcal{T}}_i \leftarrow \overline{\mathcal{T}}_i \cup \{t\}$;
6   **until** $|\overline{\mathcal{T}}_i| \geq 1 + \left\lfloor \frac{|\{r_i+1,\ldots,d_i\}|}{4} \right\rfloor$;
7     $p_i^B \leftarrow \min_{d \in \mathcal{D}}\{p_{id1}\} + 0.1 \cdot (\max_{d \in \mathcal{D}}\{p_{id1}\} - \min_{d \in \mathcal{D}}\{p_{id1}\})$;
8   **for** $d := 1$ **to** $|\mathcal{D}|$ **do**
9   **if** $p_{id1} \leq p_i^B$ **then**
10   Add columns $m = (i, d, k, t)$ with $k = 1$ and $t \in \overline{\mathcal{T}}_i$ to column set $\overline{\mathcal{M}}$;

start, are included in set $\mathcal{T}_i$. Next, the procedure adds columns that are associated with a convenient dock-door $d \in \mathcal{D}$ to the set $\overline{\mathcal{M}}$ (line 7 to 10). Line 7 computes an upper limit p_i^B for the processing time which is used to evaluate the convenience of a dock-door d. In this context, a dock-door with a short processing time is regarded as being convenient, since a short truck processing time may increase the chances that a truck's cargo reaches the outbound area in-time. Specifically, all columns $m = (i, d, k = 1, t)$ with $p_{id1} \leq p_i^B$ and $t \in \overline{\mathcal{T}}_i$ are added to the column set $\overline{\mathcal{M}}$ (line 8 to 10).

4.3.2 Column generation

Once the initial column set $\overline{\mathcal{M}}$ has been determined, the LP-relaxation of the **R(TSFD-RC-V-DT2)** can be solved to optimality with a column generation procedure. The column generation procedure aims to find absent columns (i.e., columns that are not in the column set $\overline{\mathcal{M}}$) with negative reduced costs that may improve the relaxation value. The equation for computing the reduced costs of an absent column $m = (i, d, k, t)$ can be derived from the dual of the relaxed **R(TSFD-RC-V-DT2)**. Using the dual variables λ_t for the resource constraints (3.67), ν_i for the truck assignment constraints (3.68), μ_{td} for the no-overlap constraints (3.69), and π for the service level constraint (3.70), the dual **d-R(TSFD-RC-V-DT2)** can be formulated with the objective function (4.20) and the constraints (4.21) to (4.26).

d-R(TSFD-RC-V-DT2):

$$\text{Maximize} \quad \sum_{i\in\mathcal{I}} \nu_i - \sum_{t\in\mathcal{T}}\sum_{d\in\mathcal{D}} \mu_{td} - \pi(1-\alpha)\sum_{i\in\mathcal{I}}\sum_{o\in\mathcal{O}} f_{io} \tag{4.20}$$

$$\text{subject to} \quad \sum_{t\in\mathcal{T}} \lambda_t \leq 1 \tag{4.21}$$

$$\nu_i - \sum_{t'=t}^{t+p_{idk}-1} \kappa_{ik}\lambda_{t'} - \sum_{t'=t}^{t+p_{idk}-1} \mu_{t'd} - a_{idkt}\pi \leq 0 \qquad \forall (i,d,k,t) \in \overline{\mathcal{M}} \tag{4.22}$$

$$\lambda_t \in \mathbb{R}^+ \qquad \forall t \in \mathcal{T} \tag{4.23}$$

$$\mu_{td} \in \mathbb{R}^+ \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{4.24}$$

$$\nu_i \in \mathbb{R}^+ \qquad \forall i \in \mathcal{I} \tag{4.25}$$

$$\pi \in \mathbb{R}^+ \tag{4.26}$$

In the LP-relaxation of the **R(TSFD-RC-V-DT2)**, the reduced cost R_m associated with the absent column $m = (i, d, k, t)$ can be calculated with equation (4.27).

$$R_m = a_{idkt}\pi + \sum_{t'=t}^{t+p_{idk}-1} \kappa_{ik}\lambda_{t'} + \sum_{t'=t}^{t+p_{idk}-1} \mu_{t'd} - \nu_i \tag{4.27}$$

R_m measures the extent by which constraint (4.22) is violated. It can be plugged in a pricing problem for identifying absent columns with negative reduced cost. The pricing problem can be decomposed by truck and dock-door into $|\mathcal{I}| \cdot |\mathcal{D}|$ pricing subproblems[1)]. The pricing subproblem for the truck-door pair (i, d), denoted as **TSFD-RC-V-PP(i,d)**, can be formulated with the objective function (4.28) and constraints (4.29) to (4.30).

[1)] Other decompositions, e.g., by truck or by truck, door, and operator mode, are possible too.

TSFD-RC-V-PP(i,d):

$$\text{Minimize} \quad \pi \cdot \left(\sum_{k \in \mathcal{K}} \sum_{t=r_i}^{d_i} a_{idkt} x_{kt} \right) + \sum_{k \in \mathcal{K}} \sum_{t=r_i}^{d_i+p_{idk}-1} \left[\lambda_t \kappa_{ik} \cdot \left(\sum_{t'=\max\{r_i;t-p_{idk}+1\}}^{t} x_{kt'} \right) \right] + \sum_{k \in \mathcal{K}} \sum_{t=r_i}^{d_i+p_{idk}-1} \left[\mu_{td} \cdot \left(\sum_{t'=\max\{r_i;t-p_{idk}+1\}}^{t} x_{kt'} \right) \right] - \nu_i \tag{4.28}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} \sum_{t=r_i}^{d_i} x_{kt} = 1 \tag{4.29}$$

$$x_{kt} \in \{0;1\} \qquad \forall k \in \mathcal{K}, \; t \in \{r_i, \ldots, d_i\} \tag{4.30}$$

The **TSFD-RC-V-PP(i,d)** aims to identify an absent column with minimum reduced cost. That is, for a given truck-door pair (i, d), it tries to find the best operator mode and start time that lead to minimum reduced costs (4.28). For this purpose, a set of binary decision variables x_{kt} is defined so that $x_{kt} = 1$ if truck i is processed in operator mode k on dock-door d and processing starts in time interval t. Constraint (4.29) assures that truck i is processed in exactly one operator mode and that processing starts once.

Using the **TSFD-RC-V-PP(i,d)** – a mixed-integer program – for identifying new columns and solving the LP-relaxation of the **R(TSFD-RC-V-DT2)** to optimality, may be computationally expensive. Therefore, an efficient algorithm that can be used to find and add columns with negative reduced cost to the **R(TSFD-RC-V-DT2)** is shown in Algorithm 6. The algorithm – an extension of the pricing algorithm previously described for the **TSFD-RC-F** – contains two phases. First, the LP-relaxation of the **R(TSFD-RC-V-DT2)** is solved in order to update the dual multipliers λ_t, μ_{td}, ν_i, and π (line 2). Once the dual multipliers have been updated, the procedure aims to identify absent columns $m = (i, d, k, t)$ with negative reduced costs and includes them in the column set $\overline{\mathcal{M}}$ and, hence, the restricted problem (line 3 to 19). For every truck-door pair (i, d), the procedure determines the column $m^*_{id} \in \{\tilde{m} = (i, d, \tilde{k}, \tilde{t}) \mid \tilde{k} \in \mathcal{K}, \tilde{t} \in \{r_i, \ldots, d_i\}\}$ with the minimum reduced costs R^*_{id} (line 6 to 16). That is, the operator mode and time interval that lead to the minimum reduced costs are determined. If column m^*_{id} has negative reduced costs (i.e., $R^*_{id} < 0$), it is temporarily stored in set $\mathcal{N}$ (line 17 to 18). Lastly, all columns in set $\mathcal{N}$ are added to the column set $\overline{\mathcal{M}}$ and, hence, to the restricted problem **R(TSFD-RC-V-DT2)**

Algorithm 6: Column generation procedure for the **TSFD-RC-V**

Data: **R(TSFD-RC-V-DT2)** and the initial column set $\overline{\mathcal{M}}$.

Result: **R(TSFD-RC-V-DT2)** including all columns such that the optimal value of the relaxed **R(TSFD-RC-V-DT2)** is equivalent to the optimal value of the relaxed **TSFD-RC-V-DT2**.

1 **repeat**
2   Solve the LP-relaxation of the **R(TSFD-RC-V-DT2)** over the set $\overline{\mathcal{M}}$ and store the values of the dual variables λ_t, μ_{td}, ν_i, and π;
3   $\mathcal{N} \leftarrow \emptyset$;
4   **for** $i := 1$ **to** $|\mathcal{I}|$ **do**
5   **for** $d := 1$ **to** $|\mathcal{D}|$ **do**
6       $R^*_{id} \leftarrow \infty$;
7   **for** $k := 1$ **to** $|\mathcal{K}|$ **do**
8         $\tilde{R} \leftarrow a_{i,d,k,r_i}\pi + \sum_{t=r_i}^{r_i+p_{idk}-1} \kappa_{ik}\lambda_t + \sum_{t=r_i}^{r_i+p_{idk}-1} \mu_{td} - \nu_i$;
9   **if** $\tilde{R} < R^*_{id}$ **then**
10           $m^*_{id} \leftarrow (i, d, k, r_i)$;
11           $R^*_{id} \leftarrow \tilde{R}$;
12   **for** $t := r_i + 1$ **to** d_i **do**
13           $\tilde{R} \leftarrow \tilde{R} + (a_{i,d,k,t+p_{idk}-1} - a_{i,d,k,t-1})\pi + \kappa_{ik}(\lambda_{t+p_{idk}-1} - \lambda_{t-1}) + (\mu_{t+p_{idk}-1,d} - \mu_{t-1,d})$;
14   **if** $\tilde{R} < R^*_{id}$ **then**
15             $m^*_{id} \leftarrow (i, d, k, t)$;
16             $R^*_{id} \leftarrow \tilde{R}$;
17   **if** $R^*_{id} < 0$ **then**
18         $\mathcal{N} \leftarrow \mathcal{N} \cup \{m^*_{id}\}$;
19   $\overline{\mathcal{M}} \leftarrow \overline{\mathcal{M}} \cup \mathcal{N}$;
20 **until** $\mathcal{N} = \emptyset$;

(line 19). Both steps, namely updating the dual multipliers and searching absent columns with negative reduced costs, are repeated until no new columns with negative reduced costs can be found (line 20).

Once the column generation procedure terminates, the LP-relaxation of both the **R(TSFD-RC-V-DT2)** and **TSFD-RC-V-DT2** are solved to optimality. If the obtained solution only consists of integer variables, the optimal solution to the **TSFD-RC-V-DT2** is found. If the solution is fractional, on the other hand, integrality on the decision variables is reintroduced and the **R(TSFD-RC-V-DT2)** is solved with a default solver. Note that in this case the obtained integer solution to the **R(TSFD-RC-V-DT2)** is not necessarily the optimal solution to the **TSFD-RC-V-**

DT2. Hence, the proposed procedure is a heuristic solution procedure.

4.4 Shift and truck scheduling problem

In this section, a column generation-based solution procedure for the shift and truck scheduling problem (**ISTSFD-V**) is proposed[1)]. Algorithm 7 summarizes the heuristic procedure. The time-discrete model formulation **ISTSFD-V-DT2** is used as a basis. First, a restricted version of the **ISTSFD-V-DT2**, denoted as **R(ISTSFD-V-DT2)**, is defined over the column sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$ (line 1). Both column sets are used to store truck columns and shift columns, respectively. After adding initial truck columns to $\overline{\mathcal{M}}_{truck}$ and initial shift columns to $\overline{\mathcal{M}}_{shift}$ (line 2), the LP-relaxation of the **R(ISTSFD-V-DT2)** is solved with a column generation procedure (line 3 to 14). If this results in a fractional solution, integrality on the decision variables is reintroduced in order to obtain an integer solution for the **ISTSFD-V** (line 15 to 16). This section provides details concerning the computation of initial columns and the column generation procedure.

4.4.1 Initial columns

4.4.1.1 Generating initial columns via MIP

There are multiple ways to generate initial columns via solving mixed-integer programs. In general, the computation of truck and shift columns could be either separated or integrated. This section presents one approach for computing the initial truck and shift column sets separately and simultaneously.

When separating the computation of initial truck and shift columns, the **TSFD-U-V**, which was proposed in the context of the multi-mode resource and truck scheduling problem, could be solved with a default solver for determining an initial set of truck columns for the **R(ISTSFD-V-DT2)**. Solving the **TSFD-U-V** to optimality provides $|\mathcal{I}|$ truck columns which can be added to the restricted problem **R(ISTSFD-V-DT2)**. Note, however, that the **TSFD-U-V** neither considers different operator shift types nor the maximum number of operators that are available in each operator group. Hence, it is not guaranteed that the obtained truck schedule can be executed with the available resources. This is problematic as it may hinder the default solver from

[1)] Note that the **ISTSFD-V** is a generalization of the **ISTSFD-F**. The presented solution procedure requires minor adjustments so that it can be applied to the **ISTSFD-F**.

Algorithm 7: Solution procedure for the **ISTSFD-V**

Data: **ISTSFD-V-DT2** and the empty column sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$.
Result: Integer solution for the **ISTSFD-V-DT2**.
Define the **R(ISTSFD-V-DT2)** over the column sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$;
Compute set of initial truck columns and shift columns and add them to column sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$, respectively;
repeat
    $b \leftarrow False$;
    Solve LP-relaxation of the **R(ISTSFD-V-DT2)** and save the dual multipliers;
    Pass dual multipliers to the truck pricing problem and solve it;
    if *truck columns with negative reduced costs were found* **then**
        Add the truck columns to truck column set $\overline{\mathcal{M}}_{truck}$;
        $b \leftarrow True$;
    Pass dual multipliers to the shift pricing problem and solve it;
    if *shift columns with negative reduced costs were found* **then**
        Add the shift columns to shift column set $\overline{\mathcal{M}}_{shift}$;
        $b \leftarrow True$;
until b = *False*;
if *current solution fractional* **then**
    Reintroduce integrality on variables and solve **R(ISTSFD-V-DT2)**;

obtaining a solution for the LP-relaxation of the restricted problem – a prerequisite for updating the dual multipliers which is required for pricing out new columns. Therefore, an infeasible dummy shift pattern $s = |\mathcal{S}_g| + 1$ can be defined for each operator group $g \in \mathcal{G}$ and loaded into the restricted problem in order to assure that the default solver can obtain a solution for the LP-relaxation of the restricted problem. Specifically, the dummy shift patterns are defined such that $\gamma_{g,|\mathcal{S}_g|+1,t} = 1$ for all $t \in \mathcal{T}$, i.e., operators deployed in this shift pattern work throughout the entire planning horizon and do not take any breaks. Furthermore, the dummy shifts' cost $C_{g,|\mathcal{S}_g|+1}$ are set prohibitively high in order to avoid that dummy shift patterns appear in the final solution.

The previously described approach is sufficient for column generation to work, however, it can be advantageous to compute a feasible solution to the **ISTSFD-V** and load it into the restricted problem **R(ISTSFD-V-DT2)** before starting the column generation procedure. Such a feasible solution can be obtained by solving the **ISTSFD-U-V** with the objective function (4.31) and constraints (4.32) to (4.37)[1].

ISTSFD-U-V:

$$\text{Minimize} \quad \sum_{i \in \mathcal{I}} \sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} a_{idkt} x_{idkt} \tag{4.31}$$

$$\text{subject to} \quad \sum_{i \in \mathcal{I}} \sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} \sum_{t'=\max\{0;t-p_{idk}+1\}}^{t} \kappa_{ik} x_{idkt'} \leq \sum_{g \in \mathcal{G}} \sum_{s \in \mathcal{S}_g} \gamma_{gst} W_{gs} \qquad \forall t \in \mathcal{T} \tag{4.32}$$

$$\sum_{s \in \mathcal{S}_g} W_{gs} \leq \overline{W}_g \qquad \forall g \in \mathcal{G} \tag{4.33}$$

$$\sum_{d \in \mathcal{D}} \sum_{k \in \mathcal{K}} \sum_{t=r_i}^{d_i} x_{idkt} = 1 \qquad \forall i \in \mathcal{I} \tag{4.34}$$

$$\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} \sum_{t'=\max\{0;t-p_{idk}+1\}}^{t} x_{idkt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{4.35}$$

$$x_{idkt} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, k \in \mathcal{K}, t \in \mathcal{T} \tag{4.36}$$

$$W_{gs} \in \mathbb{Z}^{+} \qquad \forall g \in \mathcal{G}, s \in \mathcal{S}_g \tag{4.37}$$

Similar to the **TSFD-U-V**, the **ISTSFD-U-V** aims to minimize the number of delayed product units (4.31). Constraints (4.32) guarantee that enough operators are on-site throughout the planning horizon while constraints (4.33) assure that the maximum number of available operators in each operator group is not exceeded. Moreover, constraints (4.34) and (4.35) are the truck assignment constraints and no-overlap constraints, respectively. Note that the obtained optimal solution to the **ISTSFD-U-V** satisfies the **ISTSFD-V's** service level constraint and, hence, is also a feasible solution to the original shift and truck scheduling problem **ISTSFD-V**. Solving the **ISTSFD-U-V** to optimality provides $|\mathcal{I}|$ truck columns and up to $|\mathcal{G}| \cdot |\mathcal{S}|$ shift columns which can be added to the restricted problem **R(ISTSFD-V-DT2)**.

[1] According to preliminary tests, the **ISTSFD-U-V** performed better than simply setting the **ISTSFD-V's** objective function to 1.

4.4.1.2 Generating initial columns heuristically

Generating initial columns by solving the previously proposed mixed-integer programs with a default solver can be computationally expensive. Hence, a heuristic approach for computing initial columns is presented below. The procedure heuristically identifies both truck columns $m = (i, d, k, t)$ and shift columns $n = (g, s)$ and adds them to the sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$, respectively. Algorithm 8 shows the pseudocode. The procedure consists of two phases: (i)

Algorithm 8: Generating the sets of initial columns for the **ISTSFD-V**

Data: Empty column sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$ for truck columns and shift columns, respectively.

Result: Non-empty column sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$.

1 **for** $i := 1$ **to** $|\mathcal{I}|$ **do**
2     $\overline{\mathcal{T}}_i \leftarrow \{r_i\}$;
3   **repeat**
4   Randomly select a time interval $t \in \{r_i + 1, \ldots, d_i\} \setminus \overline{\mathcal{T}}_i$;
5         $\overline{\mathcal{T}}_i \leftarrow \overline{\mathcal{T}}_i \cup \{t\}$;
6   **until** $|\overline{\mathcal{T}}_i| \geq 1 + \left\lfloor \frac{|\{r_i+1,\ldots,d_i\}|}{4} \right\rfloor$;
7     $p_i^B \leftarrow \min_{d \in \mathcal{D}}\{p_{id1}\} + 0.1 \cdot (\max_{d \in \mathcal{D}}\{p_{id1}\} - \min_{d \in \mathcal{D}}\{p_{id1}\})$;
8   **for** $d := 1$ **to** $|\mathcal{D}|$ **do**
9   **if** $p_{id1} \leq p_i^B$ **then**
10   Add columns $m = (i, d, k, t)$ with $k = 1$ and $t \in \overline{\mathcal{T}}_i$ to column set $\overline{\mathcal{M}}_{truck}$;

11 **for** $g := 1$ **to** $|\mathcal{G}|$ **do**
12     $\overline{\mathcal{S}}_g \leftarrow \emptyset$;
13   **repeat**
14   Randomly select a shift pattern $s \in \mathcal{S}_g \setminus \overline{\mathcal{S}}_g$;
15         $\overline{\mathcal{S}}_g \leftarrow \overline{\mathcal{S}}_g \cup \{s\}$;
16   **until** $|\overline{\mathcal{S}}_g| \geq \left\lfloor \frac{|\mathcal{S}_g|}{3} \right\rfloor$;
17   Define a dummy shift pattern $s = |\mathcal{S}_g| + 1$ with $C_{g,|\mathcal{S}_g|+1} = \infty$ and $\gamma_{g,|\mathcal{S}_g|+1,t} = 1$ with $t \in \mathcal{T}$;
18     $\overline{\mathcal{S}}_g \leftarrow \overline{\mathcal{S}}_g \cup \{s = |\mathcal{S}_g| + 1\}$;
19   Add columns $m = (g, s)$ with $s \in \overline{\mathcal{S}}_g$ to column set $\overline{\mathcal{M}}_{shift}$;

Adding a set of initial truck columns to the truck column set $\overline{\mathcal{M}}_{truck}$ (line 1 to 10), and (ii) adding a set of initial shift columns to the shift column set $\overline{\mathcal{M}}_{shift}$ (line 11 to 19). Note that the calculation of the truck columns is identical to the procedure in Algorithm 5, which was described in the context of the multi-mode resource and truck scheduling problem in Section 4.3. Therefore, only the generation of shift columns is explained in more detail below. For a given operator group g, $\left\lfloor \frac{|\mathcal{S}_g|}{3} \right\rfloor$ shift patterns are randomly chosen and temporarily stored in set $\overline{\mathcal{S}}_g$ (line 12 to 16). Furthermore, a dummy shift pattern $s = |\mathcal{S}_g| + 1$ – with infinite costs and an operator worktime throughout the whole planning horizon – is added to set $\overline{\mathcal{S}}_g$ (line 17 to 18). The dummy shift pattern guarantees that the default solver can obtain a solution for the LP-relaxation of the restricted problem. Lastly, all shift columns $n = (g, s)$ with $s \in \overline{\mathcal{S}}_g$ are added to the column set $\overline{\mathcal{M}}_{shift}$ (line 19).

4.4.2 Column generation

After the initial truck and shift column sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$ have been computed, a column generation procedure can be applied in order to solve the LP-relaxation of the **R(ISTSFD-V-DT2)** to optimality. A column generation procedure tries to detect absent truck and shift columns that may improve the relaxation value, and adds them to the restricted problem **R(ISTSFD-V-DT2)**. Therefore, the reduced cost of absent truck and shift columns need to be evaluated. The equations for computing the reduced cost of an absent truck column $m = (i, d, k, t)$ or an absent shift column $n = (g, s)$ can be derived from the dual of the relaxed **R(ISTSFD-V-DT2)**. Using the dual variables λ_t for the resource constraints (3.138), ψ_g for the upper limits of available operators (3.139), ν_i for the truck assignment constraints (3.140), μ_{td} for the no-overlap constraints (3.141), and π for the service level constraint (3.142), the dual **d-R(ISTSFD-V-DT2)** can be formulated with the objective function (4.38) and the constraints (4.39) to (4.44).

d-R(ISTSFD-V-DT2):

$$\begin{aligned}
\text{Maximize} \quad & \sum_{i\in\mathcal{I}} \nu_i - \sum_{g\in\mathcal{G}} \psi_g \overline{W}_g - \sum_{t\in\mathcal{T}} \sum_{d\in\mathcal{D}} \mu_{td} \\
& - \pi(1-\alpha) \sum_{i\in\mathcal{I}} \sum_{o\in\mathcal{O}} f_{io} && (4.38) \\
\text{subject to} \quad & \sum_{t\in\mathcal{T}} \lambda_t \gamma_{gst} + \psi_g \leq C_{gs} && \forall (g,s) \in \overline{\mathcal{M}}_{shift} \quad (4.39) \\
& \nu_i - \sum_{t'=t}^{t+p_{idk}-1} \kappa_{ik} \lambda_{t'} - \sum_{t'=t}^{t+p_{idk}-1} \mu_{t'd} && \forall (i,d,k,t) \in \overline{\mathcal{M}}_{truck} \quad (4.40) \\
& \qquad -a_{idkt}\pi \leq 0 \\
& \lambda_t \in \mathbb{R}^+ && \forall t \in \mathcal{T} \quad (4.41) \\
& \mu_{td} \in \mathbb{R}^+ && \forall t \in \mathcal{T}, d \in \mathcal{D} \quad (4.42) \\
& \nu_i \in \mathbb{R}^+ && \forall i \in \mathcal{I} \quad (4.43) \\
& \pi \in \mathbb{R}^+ && (4.44) \\
& \psi_g \in \mathbb{R}^+ && \forall g \in \mathcal{G} \quad (4.45)
\end{aligned}$$

In the LP-relaxation of the **R(ISTSFD-V-DT2)**, the reduced cost R_n and R_m associated with the absent shift column $n = (g, s)$ and truck column $m = (i, d, k, t)$, respectively, can be calculated as follows:

$$R_n = C_{gs} - \sum_{t\in T} \lambda_t \gamma_{gst} - \psi_g \tag{4.46}$$

$$R_m = a_{idkt}\pi + \sum_{t'=t}^{t+p_{idk}-1} \kappa_{ik} \lambda_{t'} + \sum_{t'=t}^{t+p_{idk}-1} \mu_{t'd} - \nu_i \tag{4.47}$$

R_n measures the extent by which constraint (4.39) is violated. R_m, on the other hand, measures the extent by which (4.40) is violated. Both expressions can be used in pricing problems for identifying absent truck and shift columns with negative reduced cost. Note that equation (4.47) for the reduced cost R_m of the absent truck column $m = (i, d, k, t)$ is identical to equation (4.27) associated with the **TSFD-RC-V**. Moreover, the pricing problem **TSFD-RC-V-PP(i,d)** can be used for identifying absent truck columns with negative reduced costs for the **R(ISTSFD-V-DT2)**. Hence, only the pricing problem for identifying absent shift columns with negative reduced cost will be explained in more detail below.

The shift pricing problem can be decomposed by the operator group into $|\mathcal{G}|$ pricing subproblems[1]. The shift pricing subproblem for operator group g, denoted as **ISTSFD-V-PP-S(g)**, can be formulated with the objective function (4.48) and constraints (4.49) to (4.50)

ISTSFD-V-PP-S(g):

$$\text{Minimize} \quad \sum_{s \in \mathcal{S}_g} C_{gs} z_s - \sum_{s \in \mathcal{S}_g} \sum_{t \in \mathcal{T}} \lambda_t \gamma_{gst} z_s - \psi_g \tag{4.48}$$

$$\text{subject to} \quad \sum_{s \in \mathcal{S}_g} z_s = 1 \tag{4.49}$$

$$z_s \in \{0; 1\} \qquad \forall s \in \mathcal{S}_g \tag{4.50}$$

The **ISTSFD-V-PP-S(g)** aims to identify an absent shift pattern with minimum reduced cost. That is, for a given operator group $g \in \mathcal{G}$, it tries to find the best shift pattern $s \in \mathcal{S}_g$ which leads to minimum reduced cost (4.48). For this purpose, a set of binary decision variables z_s is defined so that $z_s = 1$ if shift pattern s is chosen. Constraint (4.49) assures that exactly one shift pattern is chosen.

Using the **ISTSFD-V-PP-S(g)** – a mixed-integer program – for identifying new columns and solving the LP-relaxation of the **R(ISTSFD-V-DT2)** to optimality, may be computationally expensive. Therefore, the previously proposed efficient column generation algorithm for the multi-mode resource and truck scheduling problem is extended. Algorithm 9 shows the procedure in pseudocode that can be applied to find both truck and shift columns with negative reduced costs. It is an extension of the column generation procedure for the **TSFD-RC-V** and consists of three phases: (i) Solving the LP-relaxation of the **R(ISTSFD-V-DT2)** in order to update the dual multipliers (line 2), (ii) identifying and adding absent truck columns $m = (i, d, k, t)$ with negative reduced costs to the restricted problem (line 3 to 19), and (iii) identifying and adding absent shift columns $n = (g, s)$ with negative reduced costs to the restricted problem (line 20 to 30). Note that phases (i) and (ii) are identical to the phases described in the column generation algorithm for the **TSFD-RC-V**. Hence, only phase (iii), i.e., the pricing procedure for shift columns, will be described in more detail below. The column generation procedure for shifts aims to identify absent shift columns $n = (g, s)$ with negative reduced costs and includes them in the column set $\overline{\mathcal{M}}_{shift}$ and, hence, in the restricted problem. For every operator group $g \in \mathcal{G}$, the procedure determines the column $n_g^* \in \{\tilde{n} = (g, \tilde{s}) \mid \tilde{s} \in \mathcal{S}_g\}$ with the minimum reduced costs R_g^* (line 22 to 27). That is, the shift pattern which leads to the minimum reduced

[1] A decomposition is not mandatory. It would be also feasible to solve only one pricing problem.

Algorithm 9: Column generation procedure for the **ISTSFD-V**

Data: **R(ISTSFD-V-DT2)** and the initial column sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$.

Result: **R(ISTSFD-V-DT2)** including all columns such that the optimal value of the relaxed **R(ISTSFD-V-DT2)** is equivalent to the optimal value of the relaxed **ISTSFD-V-DT2**.

repeat
- Solve the LP-relaxation of the **R(ISTSFD-V-DT2)** over the sets $\overline{\mathcal{M}}_{truck}$ and $\overline{\mathcal{M}}_{shift}$ and store the values of the dual variables λ_t, μ_{td}, ν_i, π, and ψ_g;
- $\mathcal{N}_{truck} \leftarrow \emptyset$;
- **for** $i := 1$ **to** $|\mathcal{I}|$ **do**
  - **for** $d := 1$ **to** $|\mathcal{D}|$ **do**
    - $R^*_{id} \leftarrow \infty$;
    - **for** $k := 1$ **to** $|\mathcal{K}|$ **do**
      - $\tilde{R} \leftarrow a_{i,d,k,r_i}\pi + \sum_{t=r_i}^{r_i+p_{idk}-1} \kappa_{ik}\lambda_t + \sum_{t=r_i}^{r_i+p_{idk}-1} \mu_{td} - \nu_i$;
      - **if** $\tilde{R} < R^*_{id}$ **then**
        - $m^*_{id} \leftarrow (i, d, k, r_i)$;
        - $R^*_{id} \leftarrow \tilde{R}$;
      - **for** $t := r_i + 1$ **to** d_i **do**
        - $\tilde{R} \leftarrow \tilde{R} + (a_{i,d,k,t+p_{idk}-1} - a_{i,d,k,t-1})\pi + \kappa_{ik}(\lambda_{t+p_{idk}-1} - \lambda_{t-1})$ $+ (\mu_{t+p_{idk}-1,d} - \mu_{t-1,d})$;
        - **if** $\tilde{R} < R^*_{id}$ **then**
          - $m^*_{id} \leftarrow (i, d, k, t)$;
          - $R^*_{id} \leftarrow \tilde{R}$;
    - **if** $R^*_{id} < 0$ **then**
      - $\mathcal{N}_{truck} \leftarrow \mathcal{N}_{truck} \cup \{m^*_{id}\}$;
- $\overline{\mathcal{M}}_{truck} \leftarrow \overline{\mathcal{M}}_{truck} \cup \mathcal{N}_{truck}$;
- $\mathcal{N}_{shift} \leftarrow \emptyset$;
- **for** $g := 1$ **to** $|\mathcal{G}|$ **do**
  - $R^*_g \leftarrow \infty$;
  - **for** $s := 1$ **to** $|\mathcal{S}_g|$ **do**
    - $\tilde{R} \leftarrow C_{gs} - \sum_{t\in\mathcal{T}} \lambda_t\gamma_{gst} - \psi_g$;
    - **if** $\tilde{R} < R^*_g$ **then**
      - $n^*_g \leftarrow (g, s)$;
      - $R^*_g \leftarrow \tilde{R}$;
  - **if** $R^*_g < 0$ **then**
    - $\mathcal{N}_{shift} \leftarrow \mathcal{N}_{shift} \cup \{n^*_g\}$;
- $\overline{\mathcal{M}}_{shift} \leftarrow \overline{\mathcal{M}}_{shift} \cup \mathcal{N}_{shift}$;

until $\mathcal{N}_{truck} \cup \mathcal{N}_{shift} = \emptyset$;

costs is determined. If column n_g^* has negative reduced costs (i.e., $R_g^* < 0$), it is temporarily stored in set $\mathcal{N}_{shift}$ (line 28 to 29). Lastly, all columns in set $\mathcal{N}_{shift}$ are added to the column set $\overline{\mathcal{M}}_{shift}$ and, hence, to the restricted problem **R(ISTSFD-V-DT2)** (line 30). Phases (i)-(iii) are repeated until no new truck or shift columns with negative reduced cost can be found (line 31).

Once the algorithm terminates, the LP-relaxation of both the **R(ISTSFD-V-DT2)** and **ISTSFD-V-DT2** are solved to optimality. If the obtained solution only consists of integer variables, the optimal solution to the **ISTSFD-V-DT2** is found. If the solution is fractional, on the other hand, integrality on the decision variables is reintroduced and the **R(ISTSFD-V-DT2)** is solved with a default solver. Note that in this case the obtained integer solution to the **R(ISTSFD-V-DT2)** is not necessarily the optimal solution to the **ISTSFD-V-DT2**. Hence, the proposed procedure is a heuristic solution procedure.

4.5 Chapter summary

This chapter described solution procedures for the resource and truck scheduling problem, multi-mode resource and truck scheduling problem, and shift and truck scheduling problem. In Section 4.2, a heuristic solution procedure for tackling the resource and truck scheduling problem (**TSFD-RC-F**) was proposed. The solution procedure consisted of a column generation scheme which allowed to add decision variables iteratively to the discrete-time model formulation of the **TSFD-RC-F**. First, ways to generate an initial subset of columns that can be added to the restricted model were described. Specifically, a mixed-integer program and a heuristic procedure for computing a set of initial columns were developed. Next, a mixed-integer pricing program and an efficient pricing algorithm were proposed. They can be used in order to identify absent columns that may improve the relaxation value of the restricted problem. Moreover, the solution procedure served as a basis and was extended in Section 4.3 and Sec 4.4 in order to tackle the multi-mode resource and truck scheduling problem (**TSFD-RC-V**) and the shift and truck scheduling problem (**ISTSFD-V**), respectively. Both sections presented details on the computation of initial columns and the column generation procedure.

5 Computational experiments

This section contains numerical experiments for evaluating the performance of both the proposed mathematical programs and the proposed column generation-based solution procedures. Specifically, the research objectives in Table 5.1 are addressed.

Item	Research objective	Evaluation criteria	Section
RO1	Identify the best mixed-integer programming formulations for the proposed models.	Solution quality Computational time	5.2
RO2	Assess the performance of the proposed column generation-based solution procedures.	Solution quality Computational time	5.3
RO3	Derive managerial insights by benchmarking the proposed models against frequently used truck scheduling models.	Solution quality	5.4

Table 5.1 Research objectives.
Source: Own table.

The numerical study in Section 5.2 seeks for the best performing model formulations of the **TSFD-RC-F**, **TSFD-RC-V**, and **ISTSFDs** (research objective **RO1**), while the performance of the column generation-based solution procedures is analyzed in Section 5.3 (research objective **RO2**). Both studies employ the computational time and the solution quality for performance assessment. The numerical study in Section 5.4 sets out to address research objective **RO3** by investigating the solution quality of the proposed scheduling models. Specifically, the study benchmarks the **TSFD-RC-F**, **TSFD-RC-V**, and **ISTSFD-F** with respect to various key performance indicators (KPIs) against two frequently used truck scheduling models: (i) a model for minimizing the makespan, and (ii) a model for minimizing the total processing time of trucks. In addition, the impact of problem characteristics such as the length of time windows, the number of available inbound dock-doors, and the share of big trucks on the models' solution quality is analyzed.

All numerical experiments are conducted on a notebook with an Intel i7-8550 CPU and 16GB RAM. The MIP models are solved with IBM ILOG CPLEX Optimizer V12.10.0 and the column generation-based solution procedure is implemented in C++ (Visual Studio 2019) using the CPLEX API for solving the restricted master problem and pricing problem. If not mentioned otherwise, the solution time for all solution approaches is limited to 15 minutes per test instance.

Prior to the experiments, the next section outlines the test instance generator required to develop suitable testbeds for conducting the described computational experiments.

5.1 Instance generation

The presented research objectives call for a diverse set of test instances. Since there were no suitable testbeds available, it was necessary to implement a test instance generator. A total of six testbeds ranging from very small-sized instances (dubbed as **XS**) to very large-sized instances (dubbed as **XXL**) were generated. All generated problem instances share the following basic assumptions. They assume an 8h planning horizon (e.g., 08:00 – 16:00) with inbound truck arrival times and outbound truck deadlines uniformly distributed between 08:00 and 14:30, and 13:00 and 16:00, respectively. This overlap between inbound and outbound operations fosters that the deadlines of outbound trucks potentially affect the scheduling of inbound trucks. The number of outbound trucks supplied by each inbound truck is randomly chosen between five and seven[1)].

Moreover, it is distinguished between small inbound trucks and big inbound trucks. When dealing with single-mode processing, it is assumed that a small inbound truck must be processed by one operator (i.e., $\kappa_i = 1$) and that a big inbound truck must be processed by two operators (i.e., $\kappa_i = 2$). This seems to be a simple rule of thumb that may be used in practice. When dealing with multi-mode processing, on the other hand, two operator modes ($k \in \{1, 2\}$) are considered for inbound trucks $i \in \mathcal{I}$: A slow processing mode ($\kappa_{i,k=1} = 1$) and a fast processing mode ($\kappa_{i,k=2} = 2$). Truck processing times are generated as follows:

1. Randomly set the truck processing times for small trucks in the slow mode (using one operator) and the truck processing times for big trucks in the fast mode (using two operators) between 30 and 70 minutes, a similar ballpark as Tadumadze et al.[2)]. Note that the generated truck processing times can be used in the single-mode models.

2. For the multi-mode models, furthermore derive the truck processing times for small trucks

1) Rijal et al. (2019, p. 768).

2) Tadumadze et al. (2019, p. 351).

in the fast mode (using two operators) by using the equation $p_{i,k=2} = \frac{p_{i,k=1}}{\sqrt{2}}$. Moreover, derive the truck processing times for big trucks in the slow mode (using one operator) by using the equation $p_{i,k=1} = p_{i,k=2} \cdot \sqrt{2}$. This procedure is adapted from TADUMADZE ET AL. and considers a sub-additive performance increase[1)].

Based on these assumptions, the instance generator is initialized with the six input parameters from Table 5.2. These parameters are critical factors that make problem instances of the proposed models either harder or easier to solve. The number of inbound trucks $|\mathcal{I}|$ ranges from

	Generated testbeds					
Parameter	**XS**	**S**	**M**	**L**	**XL**	**XXL**
$\|\mathcal{I}\|$	30	50	80	150	250	350
$\|\mathcal{D}\|$	$\{4, 5\}$	$\{6, 7, 8\}$	$\{9, 11, 13\}$	$\{18, 21\}$	$\{30, 35\}$	$\{40, 45\}$
$\|\mathcal{O}\|$	20	30	30	50	50	70
$\|\mathcal{T}\|$	$\{48, 96, 240\}$					
$d_i - r_i$	$\{\sim U(30, 50), \sim U(60, 80)\}$					
β	$\{\sim U(0.0, 0.0), \sim U(0.2, 0.3), \sim U(0.4, 0.5)\}$					

Table 5.2 Parameters for instance generation.
Source: Own table.

30 in testbed **XS** to 350 in testbed **XXL**. Similarly, the number of outbound trucks varies between 20 in **XS** and 70 in **XXL**. For every testbed, different values for the number of available inbound dock-doors $|\mathcal{D}|$ are chosen such that the ratio $\frac{|\mathcal{I}|}{|\mathcal{D}|}$, i.e., the average number of inbound trucks per dock-door, varies between six and nine. The number of time intervals $|\mathcal{T}|$, which is a key complexity driver of the discrete-time models and expresses the granularity of time, is chosen from the set $\{48, 96, 240\}$. In other words, time interval lengths of ten minutes ($|\mathcal{T}| = 48$), five minutes ($|\mathcal{T}| = 96$), and two minutes ($|\mathcal{T}| = 240$) are considered. Lastly, the length of time windows and the share of big inbound trucks are varied in order to ensure a certain diversity within each testbed. The truck time windows $(d_i - r_i)$ are uniformly distributed and either of short length (30 to 50 minutes) or moderate length (60 to 80 minutes). The share of big trucks, denoted as β, is also uniformly distributed. Specifically, three cases are considered: (i) no big inbound trucks ($\sim U(0.0, 0.0)$), (ii) between 20-30% big inbound trucks ($\sim U(0.2, 0.3)$), and (iii) between 40-50% big inbound trucks ($\sim U(0.4, 0.5)$).

Moreover, the shift and truck scheduling problems (**ISTSFD-F** and **ISTSFD-V**) require operator

1) Tadumadze et al. (2019, p. 351).

group information and shift pattern information as input data. If not mentioned otherwise, one operator group (i.e., $|\mathcal{G}| = 1$) with the shift pattern information shown in Table 5.3 are used in the computational experiments.

Operator group	Shift pattern	Start	End	Work break	C_{gs}
1	1	08:00	16:00	11:30 - 12:00	1
1	2	08:00	16:00	12:00 - 12:30	1
1	3	08:00	16:00	12:30 - 13:00	1
1	4	08:00	16:00	13:00 - 13:30	1

Table 5.3 Standard shift pattern information.
Source: Own table.

The computational time and the solution quality need to be analyzed when seeking for the best performing MIP formulations of the proposed models. This demands test instances that can be solved to optimality by an off-the-shelf-solver in a reasonable time. Therefore, testbeds **XS** and **S** are chosen for investigating research objective **RO1**. In order to address research objective **RO2**, i.e., assessing the performance of the column generation-based solution procedures, the larger problem instances in testbeds **L**, **XL**, and **XXL** are chosen. Note that the size of these problem instances is representative for real-world facilities. Lastly, testbeds **S** and **M** are chosen for benchmarking the proposed models against existing truck scheduling models and for deriving managerial insights.

5.2 Performance of the mixed-integer programs

This section addresses research objective **RO1**, that is, it sets out to compare the computational performance of the **TSFD-RC-F's**, the **TSFD-RC-V's**, and the **ISTSFDs'** model formulations. For this purpose, the test instances from testbeds **XS** and **S** are used.

5.2.1 Resource and truck scheduling problem

According to preliminary tests, the number of inbound dock-doors $|\mathcal{D}|$ has only a marginal impact on the complexity of instances in **XS** and **S**. Therefore, $|\mathcal{D}|$ is set to the values of five in **XS** and seven in **S**. Since the granularity of time certainly impacts the computational performance of discrete-time models, all possible values for the number of time intervals are considered. Furthermore, different truck time window lengths and big truck shares are considered. Thus, the experiment includes a total of 360 instances, ten for each parameter combination. Table 5.4 presents the numerical results. The table shows the average CPU time for solving the MIP in column "CPU^s". It also reports the number of instances for which the solver identified a feasible integer solution (column "$\#^f$") and the number of instances that are solved to proven optimality (column "$\#^*$") within the time limit. Furthermore, the average optimality gaps of the best upper bound obtained within the time limit are reported in column "gap^*".

The results indicate that parameter $|\mathcal{T}|$ has a strong effect on the problem complexity of discrete-time models. In fact, the solution time grows disproportionally when the number of time intervals is increased. It can also be observed that the instances with larger truck time windows are harder to solve than instances with smaller time windows. Larger truck time windows increase the size of the MIP models (in the case of discrete-time models) and the size of the solution space which results in longer computational times. The parameter β also affects the solution time. Instances that only consider small inbound trucks ($\beta \sim U(0.0, 0.0)$) are solved in a shorter time than instances that incorporate small and big trucks. Both discrete-time models find the optimal solutions for all 360 instances. While **TSFD-RC-F-DT1** fails to prove optimality in one instance, **TSFD-RC-F-DT2** is not able to prove optimality for two instances within the time limit of 15 minutes. However, **TSFD-RC-F-DT2** performs slightly better than **TSFD-RC-F-DT1** as it has shorter solution times in most of the instances. Only in one instance, in which **TSFD-RC-F-DT2** fails while **TSFD-RC-F-DT1** succeeds to prove optimality within the time limit, **TSFD-RC-F-DT2** is clearly outperformed by **TSFD-RC-F-DT1**.

When comparing the average solution time of the continuous-time models, the results are more ambiguous. The solution times do not clearly show which model formulation is superior. For most instances in testbed **XS**, both continuous-time models are able to find the optimal solution. As the optimality gap indicates, both models struggle to find the optimal solutions for instances from testbed **S**. The results also reveal that the continuous-time models have difficulties with proving optimality. Specifically, **TSFD-RC-F-CT1** and **TSFD-RC-F-CT2** cannot prove optimality for 112 and 95 out of 360 instances, respectively. Surprisingly, both models are not able to identify a feasible integer solution in ca. 5% of the instances.

When comparing the discrete-time models with the continuous-time models, the table shows that the discrete-time models can be solved in a much shorter time. For many instances, their solution time is more than ten times shorter than the continuous-time models' solution time. Overall, it can be concluded that the discrete-time model formulations clearly outperform the continuous-time model formulations when seeking optimal solutions with a default solver.

Instances						TSFD-RC-F-DT1			TSFD-RC-F-DT2			TSFD-RC-F-CT1				TSFD-RC-F-CT2			
	Size	$\|\mathcal{D}\|$	$\|\mathcal{T}\|$	$d_i - r_i$	β	CPU^s	$\#^*$	gap^*	CPU^s	$\#^*$	gap^*	CPU^s	$\#^f$	$\#^*$	gap^*	CPU^s	$\#^f$	$\#^*$	gap^*
	XS	5	48	$\sim U(30,50)$	$\sim U(0.0,0.0)$	0.12	10	0.0	0.09	10	0.0	2.29	10	10	0.0	1.21	10	10	0.0
	XS	5	48	$\sim U(30,50)$	$\sim U(0.2,0.3)$	0.14	10	0.0	0.10	10	0.0	29.37	10	10	0.0	16.48	10	10	0.0
	XS	5	48	$\sim U(30,50)$	$\sim U(0.4,0.5)$	0.13	10	0.0	0.11	10	0.0	2.08	10	10	0.0	1.49	10	10	0.0
	XS	5	48	$\sim U(60,80)$	$\sim U(0.0,0.0)$	0.20	10	0.0	0.16	10	0.0	101.07	10	9	0.0	93.56	10	9	2.5
	XS	5	48	$\sim U(60,80)$	$\sim U(0.2,0.3)$	0.20	10	0.0	0.13	10	0.0	181.74	10	8	0.0	132.1	10	9	0.0
	XS	5	48	$\sim U(60,80)$	$\sim U(0.4,0.5)$	0.29	10	0.0	0.26	10	0.0	284.53	10	7	0.0	280.59	10	7	0.0
	XS	5	96	$\sim U(30,50)$	$\sim U(0.0,0.0)$	0.32	10	0.0	0.22	10	0.0	5.92	10	10	0.0	6.08	10	10	0.0
	XS	5	96	$\sim U(30,50)$	$\sim U(0.2,0.3)$	0.34	10	0.0	0.26	10	0.0	13.08	10	10	0.0	14.73	10	10	0.0
	XS	5	96	$\sim U(30,50)$	$\sim U(0.4,0.5)$	0.37	10	0.0	0.28	10	0.0	5.88	10	10	0.0	6.21	10	10	0.0
	XS	5	96	$\sim U(60,80)$	$\sim U(0.0,0.0)$	0.49	10	0.0	0.38	10	0.0	233.72	10	8	0.0	214.31	9	7	0.0
	XS	5	96	$\sim U(60,80)$	$\sim U(0.2,0.3)$	0.55	10	0.0	0.42	10	0.0	95.12	10	9	0.0	94.04	10	9	0.0
	XS	5	96	$\sim U(60,80)$	$\sim U(0.4,0.5)$	0.76	10	0.0	0.64	10	0.0	131.33	10	9	0.0	121.62	10	9	0.0
	XS	5	240	$\sim U(30,50)$	$\sim U(0.0,0.0)$	1.58	10	0.0	1.29	10	0.0	11.83	10	10	0.0	12.67	10	10	0.0
	XS	5	240	$\sim U(30,50)$	$\sim U(0.2,0.3)$	1.94	10	0.0	1.46	10	0.0	100.07	10	10	0.0	133.25	10	9	0.0
	XS	5	240	$\sim U(30,50)$	$\sim U(0.4,0.5)$	2.17	10	0.0	1.90	10	0.0	34.55	10	10	0.0	39.77	10	10	0.0
	XS	5	240	$\sim U(60,80)$	$\sim U(0.0,0.0)$	2.84	10	0.0	2.16	10	0.0	193.2	10	9	0.0	275.95	10	8	2.5
	XS	5	240	$\sim U(60,80)$	$\sim U(0.2,0.3)$	7.79	10	0.0	4.70	10	0.0	508.28	9	4	2.8	503.35	9	4	2.8
	XS	5	240	$\sim U(60,80)$	$\sim U(0.4,0.5)$	3.19	10	0.0	2.84	10	0.0	215.34	10	8	0.0	211.49	10	8	1.4
	S	7	48	$\sim U(30,50)$	$\sim U(0.0,0.0)$	0.36	10	0.0	0.24	10	0.0	601.69	9	4	2.2	513.17	9	5	2.2
	S	7	48	$\sim U(30,50)$	$\sim U(0.2,0.3)$	0.39	10	0.0	0.30	10	0.0	380.56	10	6	1.3	285.85	10	8	0.0
	S	7	48	$\sim U(30,50)$	$\sim U(0.4,0.5)$	0.39	10	0.0	0.27	10	0.0	278.72	10	8	1.1	214.51	10	8	1.1
	S	7	48	$\sim U(60,80)$	$\sim U(0.0,0.0)$	0.60	10	0.0	0.46	10	0.0	596.13	10	5	7.3	345.84	9	6	4.4
	S	7	48	$\sim U(60,80)$	$\sim U(0.2,0.3)$	0.62	10	0.0	0.52	10	0.0	660.02	9	3	9.4	559.6	9	5	4.8
	S	7	48	$\sim U(60,80)$	$\sim U(0.4,0.5)$	0.66	10	0.0	0.48	10	0.0	677.13	7	2	8.7	650.31	8	3	4.5
	S	7	96	$\sim U(30,50)$	$\sim U(0.0,0.0)$	1.03	10	0.0	0.72	10	0.0	385.44	10	6	0.0	341.55	9	6	0.0
	S	7	96	$\sim U(30,50)$	$\sim U(0.2,0.3)$	1.32	10	0.0	1.10	10	0.0	224.87	9	7	3.1	209.47	9	7	3.1
	S	7	96	$\sim U(30,50)$	$\sim U(0.4,0.5)$	4.75	10	0.0	0.96	10	0.0	248.33	9	7	1.4	220.14	9	7	2.8
	S	7	96	$\sim U(60,80)$	$\sim U(0.0,0.0)$	1.79	10	0.0	1.35	10	0.0	141.04	7	7	0.0	236.15	8	6	2.5
	S	7	96	$\sim U(60,80)$	$\sim U(0.2,0.3)$	2.30	10	0.0	1.99	10	0.0	664.64	9	3	17.3	561.64	10	4	9.8
	S	7	96	$\sim U(60,80)$	$\sim U(0.4,0.5)$	2.17	10	0.0	2.23	10	0.0	770.44	8	2	9.3	577.37	10	5	7.6
	S	7	240	$\sim U(30,50)$	$\sim U(0.0,0.0)$	4.67	10	0.0	4.12	10	0.0	136.59	8	7	0.0	132.31	8	7	0.0
	S	7	240	$\sim U(30,50)$	$\sim U(0.2,0.3)$	6.08	10	0.0	5.52	10	0.0	342.18	10	7	2.5	302.57	10	7	1.4
	S	7	240	$\sim U(30,50)$	$\sim U(0.4,0.5)$	13.41	10	0.0	15.14	10	0.0	520.77	9	4	2.8	582.12	10	4	2.5
	S	7	240	$\sim U(60,80)$	$\sim U(0.0,0.0)$	7.45	10	0.0	6.31	10	0.0	318.9	9	7	3.7	281.88	9	7	3.7
	S	7	240	$\sim U(60,80)$	$\sim U(0.2,0.3)$	33.89	10	0.0	101.64	9	0.0	802.2	9	1	9.4	653.68	9	3	10.1
	S	7	240	$\sim U(60,80)$	$\sim U(0.4,0.5)$	130.88	9	0.0	129.56	9	0.0	790.72	8	1	13.7	821.63	9	1	13.4
Avg.	**XS**					1.30		0.00	0.97		0.00	119.41			0.15	119.94			0.51
	S					11.82		0.00	15.16		0.00	474.47			5.17	416.10			4.11

$|\mathcal{I}| = 30$ and $|\mathcal{I}| = 50$ for instances in **XS** and **S**.
CPU^s: Avg. CPU time for solving the MIP; $\#^f$: Number of instances for which the solver identified a feasible integer solution; $\#^*$: Number of instances that are solved to proven optimality.
gap^*: Avg. optimality gap in %.

Table 5.4 Numerical results for different MIP formulations of the **TSFD-RC-F**.
Source: Own table.

5.2.2 Multi-mode resource and truck scheduling problem

It was shown in the last section that the **TSFD-RC-F's** continuous-time formulations were clearly outperformed by its discrete-time model formulations in terms of both solution quality and solution time. Due to the huge performance gap that was observed between the discrete-time and continuous-time model formulations, it is likely that the **TSFD-RC-V's** discrete-time formulations dominate its continuous-time formulations. To test this hypothesis, testbed **XS** is used for comparing the discrete-time and continuous-time model formulations of the **TSFD-RC-V**. The experiment includes 180 instances, ten for each parameter combination. Table 5.5 reports the average CPU times for solving the MIPs (columns "CPU^s"), the number of instances that are solved to proven optimality (columns "$\#^*$"), and the average optimality gaps (columns "gap^*") for all model formulations.

When comparing both continuous-time formulations, the results are rather ambiguous. The **TSFD-RC-V-CT2** has a slightly shorter average solution time but also a slightly bigger average optimality gap than the **TSFD-RC-V-CT1**. Both model formulations regularly struggle to find the optimal solution. With an average optimality gap of up to 22.83% for some parameter combinations, the continuous-time models often fail to find near-optimal solutions within a 15 minutes time limit. Since the discrete-time formulations solve all instances from **XS** to optimality and their average solution time is almost 200 times shorter than the continuous-time formulations' solution time, it can be concluded that the discrete-time formulations clearly outperform the continuous-time formulations.

Hence, only the discrete-time models are analyzed in more detail below. Both testbed **XS** and testbed **S** are used for this experiments and the results are reported in Table 5.6. Both discrete-time model formulations are able to solve all test instances in testbed **XS** to optimality within a few seconds. The differences between the **TSFD-RC-V-DT1** and **TSFD-RC-V-DT2** are marginal with the **TSFD-RC-V-DT2** having slightly shorter solution times for most of the parameter combinations in **XS**. For instances in **S**, on the other hand, the differences between both discrete-time formulations are clearly recognizable. When applying a default solver to the **TSFD-RC-V-DT1** and **TSFD-RC-V-DT2**, seven and one test instances cannot be solved to proven optimality within 15 minutes, respectively. Moreover, the **TSFD-RC-V-DT1** fails to find the optimal solution for two instances from testbed **S**. The **TSFD-RC-V-DT2**, on the other hand, always identified the optimal solution. With a ca. 70% shorter average solution time for instances from **S**, the **TSFD-RC-V-DT2** outperforms the **TSFD-RC-V-DT1**. Hence, the **TSFD-RC-V-DT2** seems to be the superior model formulation for the multi-mode resource and truck scheduling problem.

	Instances					**TSFD-RC-V-DT1**			**TSFD-RC-V-DT2**			**TSFD-RC-V-CT1**			**TSFD-RC-V-CT2**		
	Size	$\|\mathcal{D}\|$	$\|\mathcal{T}\|$	$d_i - r_i$	β	CPU^s	$\#^*$	gap^*	CPU^s	$\#^*$	gap^*	CPU^s	$\#^*$	gap^*	CPU^s	$\#^*$	gap^*
	XS	5	48	$\sim U(30,50)$	$\sim U(0.0,0.0)$	0.28	10	0.00	0.17	10	0.00	143.00	8	0.00	124.73	8	0.00
	XS	5	48	$\sim U(30,50)$	$\sim U(0.2,0.3)$	0.33	10	0.00	0.22	10	0.00	258.07	8	4.50	204.11	8	4.50
	XS	5	48	$\sim U(30,50)$	$\sim U(0.4,0.5)$	0.32	10	0.00	0.21	10	0.00	141.91	9	0.00	135.74	9	0.00
	XS	5	48	$\sim U(60,80)$	$\sim U(0.0,0.0)$	0.47	10	0.00	0.33	10	0.00	520.99	5	12.50	393.28	6	10.00
	XS	5	48	$\sim U(60,80)$	$\sim U(0.2,0.3)$	0.47	10	0.00	0.30	10	0.00	347.89	7	7.00	324.03	7	7.00
	XS	5	48	$\sim U(60,80)$	$\sim U(0.4,0.5)$	0.52	10	0.00	0.42	10	0.00	514.00	5	8.17	444.57	6	6.50
	XS	5	96	$\sim U(30,50)$	$\sim U(0.0,0.0)$	0.67	10	0.00	0.50	10	0.00	283.77	7	2.50	216.17	8	0.00
	XS	5	96	$\sim U(30,50)$	$\sim U(0.2,0.3)$	1.03	10	0.00	0.80	10	0.00	143.71	9	1.67	288.34	9	1.67
	XS	5	96	$\sim U(30,50)$	$\sim U(0.4,0.5)$	0.98	10	0.00	0.70	10	0.00	199.48	9	2.00	207.44	8	4.00
	XS	5	96	$\sim U(60,80)$	$\sim U(0.0,0.0)$	1.28	10	0.00	0.73	10	0.00	633.32	4	7.83	646.05	4	10.33
	XS	5	96	$\sim U(60,80)$	$\sim U(0.2,0.3)$	1.58	10	0.00	1.67	10	0.00	407.66	6	4.50	397.61	7	4.00
	XS	5	96	$\sim U(60,80)$	$\sim U(0.4,0.5)$	1.46	10	0.00	0.91	10	0.00	474.15	6	5.67	400.53	6	5.67
	XS	5	240	$\sim U(30,50)$	$\sim U(0.0,0.0)$	4.16	10	0.00	3.18	10	0.00	345.64	7	2.50	433.88	6	5.00
	XS	5	240	$\sim U(30,50)$	$\sim U(0.2,0.3)$	4.61	10	0.00	3.61	10	0.00	263.54	8	1.67	257.15	8	3.33
	XS	5	240	$\sim U(30,50)$	$\sim U(0.4,0.5)$	4.24	10	0.00	3.32	10	0.00	300.30	7	2.00	302.40	7	2.00
	XS	5	240	$\sim U(60,80)$	$\sim U(0.0,0.0)$	7.19	10	0.00	5.54	10	0.00	727.89	3	15.83	758.68	2	20.83
	XS	5	240	$\sim U(60,80)$	$\sim U(0.2,0.3)$	7.88	10	0.00	5.90	10	0.00	730.23	2	22.83	734.36	2	22.83
	XS	5	240	$\sim U(60,80)$	$\sim U(0.4,0.5)$	8.23	10	0.00	6.58	10	0.00	571.84	4	7.33	520.90	5	7.33
Avg.	**XS**					2.54		0.00	1.95		0.00	389.30		6.03	377.22		6.39

$|\mathcal{I}| = 30$ for instances in **XS**.
CPU^s: Avg. CPU time for solving the MIP.
$\#^*$: Number of instances that are solved to proven optimality.
gap^*: Avg. optimality gap in %.

Table 5.5 Numerical results for different MIP formulations of the **TSFD-RC-V**.
Source: Own table.

	Instances					TSFD-RC-V-DT1			TSFD-RC-V-DT2		
	Size	$\|\mathcal{D}\|$	$\|\mathcal{T}\|$	$d_i - r_i$	β	CPU^s	$\#^*$	gap^*	CPU^s	$\#^*$	gap^*
	XS	5	48	$\sim U(30, 50)$	$\sim U(0.0, 0.0)$	0.28	10	0.00	0.17	10	0.00
	XS	5	48	$\sim U(30, 50)$	$\sim U(0.2, 0.3)$	0.33	10	0.00	0.22	10	0.00
	XS	5	48	$\sim U(30, 50)$	$\sim U(0.4, 0.5)$	0.32	10	0.00	0.21	10	0.00
	XS	5	48	$\sim U(60, 80)$	$\sim U(0.0, 0.0)$	0.47	10	0.00	0.33	10	0.00
	XS	5	48	$\sim U(60, 80)$	$\sim U(0.2, 0.3)$	0.47	10	0.00	0.30	10	0.00
	XS	5	48	$\sim U(60, 80)$	$\sim U(0.4, 0.5)$	0.52	10	0.00	0.42	10	0.00
	XS	5	96	$\sim U(30, 50)$	$\sim U(0.0, 0.0)$	0.67	10	0.00	0.50	10	0.00
	XS	5	96	$\sim U(30, 50)$	$\sim U(0.2, 0.3)$	1.03	10	0.00	0.80	10	0.00
	XS	5	96	$\sim U(30, 50)$	$\sim U(0.4, 0.5)$	0.98	10	0.00	0.70	10	0.00
	XS	5	96	$\sim U(60, 80)$	$\sim U(0.0, 0.0)$	1.28	10	0.00	0.73	10	0.00
	XS	5	96	$\sim U(60, 80)$	$\sim U(0.2, 0.3)$	1.58	10	0.00	1.67	10	0.00
	XS	5	96	$\sim U(60, 80)$	$\sim U(0.4, 0.5)$	1.46	10	0.00	0.91	10	0.00
	XS	5	240	$\sim U(30, 50)$	$\sim U(0.0, 0.0)$	4.16	10	0.00	3.18	10	0.00
	XS	5	240	$\sim U(30, 50)$	$\sim U(0.2, 0.3)$	4.61	10	0.00	3.61	10	0.00
	XS	5	240	$\sim U(30, 50)$	$\sim U(0.4, 0.5)$	4.24	10	0.00	3.32	10	0.00
	XS	5	240	$\sim U(60, 80)$	$\sim U(0.0, 0.0)$	7.19	10	0.00	5.54	10	0.00
	XS	5	240	$\sim U(60, 80)$	$\sim U(0.2, 0.3)$	7.88	10	0.00	5.90	10	0.00
	XS	5	240	$\sim U(60, 80)$	$\sim U(0.4, 0.5)$	8.23	10	0.00	6.58	10	0.00
	S	7	48	$\sim U(30, 50)$	$\sim U(0.0, 0.0)$	0.74	10	0.00	0.53	10	0.00
	S	7	48	$\sim U(30, 50)$	$\sim U(0.2, 0.3)$	0.99	10	0.00	0.66	10	0.00
	S	7	48	$\sim U(30, 50)$	$\sim U(0.4, 0.5)$	1.03	10	0.00	0.95	10	0.00
	S	7	48	$\sim U(60, 80)$	$\sim U(0.0, 0.0)$	1.55	10	0.00	0.91	10	0.00
	S	7	48	$\sim U(60, 80)$	$\sim U(0.2, 0.3)$	2.03	10	0.00	1.68	10	0.00
	S	7	48	$\sim U(60, 80)$	$\sim U(0.4, 0.5)$	1.89	10	0.00	1.09	10	0.00
	S	7	96	$\sim U(30, 50)$	$\sim U(0.0, 0.0)$	2.71	10	0.00	2.01	10	0.00
	S	7	96	$\sim U(30, 50)$	$\sim U(0.2, 0.3)$	7.52	10	0.00	3.36	10	0.00
	S	7	96	$\sim U(30, 50)$	$\sim U(0.4, 0.5)$	3.08	10	0.00	2.20	10	0.00
	S	7	96	$\sim U(60, 80)$	$\sim U(0.0, 0.0)$	5.00	10	0.00	2.47	10	0.00
	S	7	96	$\sim U(60, 80)$	$\sim U(0.2, 0.3)$	4.61	10	0.00	3.52	10	0.00
	S	7	96	$\sim U(60, 80)$	$\sim U(0.4, 0.5)$	6.65	10	0.00	8.54	10	0.00
	S	7	240	$\sim U(30, 50)$	$\sim U(0.0, 0.0)$	14.87	10	0.00	9.37	10	0.00
	S	7	240	$\sim U(30, 50)$	$\sim U(0.2, 0.3)$	118.13	9	0.00	113.54	9	0.00
	S	7	240	$\sim U(30, 50)$	$\sim U(0.4, 0.5)$	21.98	10	0.00	19.10	10	0.00
	S	7	240	$\sim U(60, 80)$	$\sim U(0.0, 0.0)$	304.75	8	2.00	24.10	10	0.00
	S	7	240	$\sim U(60, 80)$	$\sim U(0.2, 0.3)$	326.60	8	1.67	40.99	10	0.00
	S	7	240	$\sim U(60, 80)$	$\sim U(0.4, 0.5)$	279.32	8	0.00	95.48	10	0.00
Avg.	**XS**					2.54		0.00	1.95		0.00
	S					61.30		0.20	18.36		0.00

$|\mathcal{I}| = 30$ and $|\mathcal{I}| = 50$ for instances in **XS** and **S**.
CPU^s: Avg. CPU time for solving the MIP.
$\#^*$: Number of instances that are solved to proven optimality.
gap^*: Avg. optimality gap in %.

Table 5.6 Numerical results for different discrete-time MIP formulations of the **TSFD-RC-V**.
Source: Own table.

5.2.3 Shift and truck scheduling problem

This section sets out to identify the best MIP formulations for the shift and truck scheduling problems. The experiment includes a total of four discrete-time model formulations, two for the single-mode problem and two for the multi-mode problem. A total of 360 instances from testbeds **XS** and **S** are included in the experiment. Table 5.7 shows the numerical results. It reports the average CPU times for solving the MIPs with a default solver in columns "CPU^s".

						Single-mode models						Multi-mode models					
	Instances					ISTSFD-F-DT1			ISTSFD-F-DT2			ISTSFD-V-DT1			ISTSFD-V-DT2		
	Size	$\|\mathcal{D}\|$	$\|\mathcal{T}\|$	$d_i - r_i$	β	CPU^s	$\#^*$	gap^*	CPU^s	$\#^*$	gap^*	CPU^s	$\#^*$	gap^*	CPU^s	$\#^*$	gap^*
	XS	5	48	$\sim U(30,50)$	$\sim U(0.0,0.0)$	0.19	10	0.00	0.12	10	0.00	0.40	10	0.00	0.26	10	0.00
	XS	5	48	$\sim U(30,50)$	$\sim U(0.2,0.3)$	0.26	10	0.00	0.17	10	0.00	0.44	10	0.00	0.33	10	0.00
	XS	5	48	$\sim U(30,50)$	$\sim U(0.4,0.5)$	0.23	10	0.00	0.18	10	0.00	0.43	10	0.00	0.28	10	0.00
	XS	5	48	$\sim U(60,80)$	$\sim U(0.0,0.0)$	0.30	10	0.00	0.19	10	0.00	0.65	10	0.00	0.36	10	0.00
	XS	5	48	$\sim U(60,80)$	$\sim U(0.2,0.3)$	0.34	10	0.00	0.23	10	0.00	0.63	10	0.00	0.39	10	0.00
	XS	5	48	$\sim U(60,80)$	$\sim U(0.4,0.5)$	0.35	10	0.00	0.25	10	0.00	0.72	10	0.00	0.59	10	0.00
	XS	5	96	$\sim U(30,50)$	$\sim U(0.0,0.0)$	0.52	10	0.00	0.34	10	0.00	1.04	10	0.00	0.65	10	0.00
	XS	5	96	$\sim U(30,50)$	$\sim U(0.2,0.3)$	0.47	10	0.00	0.39	10	0.00	1.21	10	0.00	0.77	10	0.00
	XS	5	96	$\sim U(30,50)$	$\sim U(0.4,0.5)$	0.48	10	0.00	0.38	10	0.00	1.28	10	0.00	0.92	10	0.00
	XS	5	96	$\sim U(60,80)$	$\sim U(0.0,0.0)$	0.74	10	0.00	0.53	10	0.00	1.80	10	0.00	1.14	10	0.00
	XS	5	96	$\sim U(60,80)$	$\sim U(0.2,0.3)$	0.79	10	0.00	0.71	10	0.00	2.01	10	0.00	1.24	10	0.00
	XS	5	96	$\sim U(60,80)$	$\sim U(0.4,0.5)$	1.00	10	0.00	0.81	10	0.00	2.15	10	0.00	1.69	10	0.00
	XS	5	240	$\sim U(30,50)$	$\sim U(0.0,0.0)$	2.22	10	0.00	1.89	10	0.00	4.72	10	0.00	3.69	10	0.00
	XS	5	240	$\sim U(30,50)$	$\sim U(0.2,0.3)$	3.00	10	0.00	2.59	10	0.00	5.74	10	0.00	4.44	10	0.00
	XS	5	240	$\sim U(30,50)$	$\sim U(0.4,0.5)$	2.68	10	0.00	2.40	10	0.00	5.93	10	0.00	4.99	10	0.00
	XS	5	240	$\sim U(60,80)$	$\sim U(0.0,0.0)$	3.72	10	0.00	3.80	10	0.00	9.71	10	0.00	7.12	10	0.00
	XS	5	240	$\sim U(60,80)$	$\sim U(0.2,0.3)$	5.61	10	0.00	5.22	10	0.00	10.82	10	0.00	7.95	10	0.00
	XS	5	240	$\sim U(60,80)$	$\sim U(0.4,0.5)$	8.66	10	0.00	8.46	10	0.00	10.38	10	0.00	8.35	10	0.00
	S	7	48	$\sim U(30,50)$	$\sim U(0.0,0.0)$	0.52	10	0.00	0.38	10	0.00	0.86	10	0.00	0.57	10	0.00
	S	7	48	$\sim U(30,50)$	$\sim U(0.2,0.3)$	0.49	10	0.00	0.34	10	0.00	1.09	10	0.00	0.62	10	0.00
	S	7	48	$\sim U(30,50)$	$\sim U(0.4,0.5)$	0.53	10	0.00	0.40	10	0.00	1.12	10	0.00	0.86	10	0.00
	S	7	48	$\sim U(60,80)$	$\sim U(0.0,0.0)$	0.73	10	0.00	0.52	10	0.00	1.47	10	0.00	0.99	10	0.00
	S	7	48	$\sim U(60,80)$	$\sim U(0.2,0.3)$	0.76	10	0.00	0.64	10	0.00	1.49	10	0.00	1.24	10	0.00
	S	7	48	$\sim U(60,80)$	$\sim U(0.4,0.5)$	0.74	10	0.00	0.63	10	0.00	2.10	10	0.00	1.60	10	0.00
	S	7	96	$\sim U(30,50)$	$\sim U(0.0,0.0)$	1.26	10	0.00	0.88	10	0.00	2.78	10	0.00	2.21	10	0.00
	S	7	96	$\sim U(30,50)$	$\sim U(0.2,0.3)$	1.44	10	0.00	1.19	10	0.00	3.46	10	0.00	2.58	10	0.00
	S	7	96	$\sim U(30,50)$	$\sim U(0.4,0.5)$	1.53	10	0.00	1.27	10	0.00	4.28	10	0.00	3.19	10	0.00
	S	7	96	$\sim U(60,80)$	$\sim U(0.0,0.0)$	2.21	10	0.00	1.85	10	0.00	4.76	10	0.00	2.92	10	0.00
	S	7	96	$\sim U(60,80)$	$\sim U(0.2,0.3)$	2.28	10	0.00	1.90	10	0.00	4.67	10	0.00	2.98	10	0.00
	S	7	96	$\sim U(60,80)$	$\sim U(0.4,0.5)$	2.86	10	0.00	3.14	10	0.00	6.47	10	0.00	4.99	10	0.00
	S	7	240	$\sim U(30,50)$	$\sim U(0.0,0.0)$	6.01	10	0.00	6.24	10	0.00	29.04	10	0.00	22.32	10	0.00
	S	7	240	$\sim U(30,50)$	$\sim U(0.2,0.3)$	6.79	10	0.00	7.45	10	0.00	18.14	10	0.00	19.26	10	0.00
	S	7	240	$\sim U(30,50)$	$\sim U(0.4,0.5)$	6.82	10	0.00	7.26	10	0.00	15.31	10	0.00	11.20	10	0.00
	S	7	240	$\sim U(60,80)$	$\sim U(0.0,0.0)$	8.88	10	0.00	9.32	10	0.00	85.40	10	0.00	22.83	10	0.00
	S	7	240	$\sim U(60,80)$	$\sim U(0.2,0.3)$	59.04	10	0.00	57.84	10	0.00	151.34	9	1.00	94.04	10	0.00
	S	7	240	$\sim U(60,80)$	$\sim U(0.4,0.5)$	17.24	10	0.00	15.95	10	0.00	47.90	10	0.00	36.38	10	0.00
Avg.	**XS**					1.75		0.00	1.59		0.00	3.34		0.00	2.51		0.00
	S					6.67		0.00	6.51		0.00	21.20		0.06	12.82		0.00

$|\mathcal{I}| = 30$ and $|\mathcal{I}| = 50$ for instances in **XS** and **S**.
CPU^s: Avg. CPU time for solving the MIP; $\#^*$: Number of instances that are solved to proven optimality; gap^*: Avg. optimality gap in %.

Table 5.7 Numerical results for different MIP formulations of the **ISTSFD-F** and **ISTSFD-V**.
Source: Own table.

Moreover, the table shows the number of instances for which the default solver identifies a feasible integer solution within the time limit of 15 minutes (columns "$\#^*$"). Finally, the average optimality gaps of the best upper bounds are reported in columns "gap^*".

First, some general observations for both the single-mode and multi-mode problem can be made. Dealing with more inbound trucks and time intervals, as well as wider truck time windows, usually results in longer solution times. Furthermore, instances which consider both small and big inbound trucks are oftentimes harder to solve than instances which only consider small inbound trucks. Moreover, the solution time significantly increases when considering multiple operator modes.

When comparing the model formulations for the single-mode shift and truck scheduling problem, only marginal differences can be observed. The default solver is able to solve all 360 test instances to optimality within a few seconds when using **ISTSFD-F-DT1** and **ISTSFD-F-DT2**. Due to its slightly shorter solution times for most of the parameter combinations, the **ISTSFD-F-DT2** should be used when tackling small-sized problem instances of the single-mode shift and truck scheduling problem.

When comparing the multi-mode formulations, on the other hand, more notable differences in the solution time can be observed. For most of the instances, the **ISTSFD-V-DT2** can be solved significantly faster than the **ISTSFD-V-DT1**. The average solution time for instances from testbed **S**, for example, can be almost cut by half when using the **ISTSFD-V-DT2** instead of the **ISTSFD-V-DT1**. Moreover, the default solver is able to obtain the optimal solution for all 360 test instances when using the **ISTSFD-V-DT2**. When using the **ISTSFD-V-DT1**, on the other hand, it fails to find the optimal solution for one test instance within the time limit of 15 minutes. Hence, the **ISTSFD-V-DT2** should be used when tackling small-sized problem instances for the multi-mode shift and truck scheduling problem.

5.3 Performance of the solution procedure

This section sets out to evaluate the performance of the proposed column generation-based solution procedures. To do so, the solution procedures are benchmarked against the best performing MIP formulations from the previous section. The experiment applies instances from testbeds **L**, **XL**, and **XXL**. The value of $|\mathcal{T}|$ is set to 96 in testbeds **XL** and **XXL**, resulting in time intervals with a rather fine granularity of five minutes. In order to assess the applicability of the solution procedure in scenarios that require a very fine time granularity, the experiment applies a time interval length of two minutes (i.e., $|\mathcal{T}| = 240$) for instances from testbeds **L**. Ten instances for each parameter combination are generated and each instance is solved twice – one time with a predefined service level of 100% (i.e., $\alpha = 1.0$) and one time with a predefined service level of 99% (i.e., $\alpha = 0.99$). In this context, the service level only applies to the product units that can theoretically reach the outbound area in time. Each experiment, hence, includes a total of 720 instances.

According to preliminary tests, it is faster to heuristically select the initial columns for the column generation-based solution procedures than by solving the proposed mixed-integer programs with a default solver. Hence, the initial columns are generated heuristically throughout the entire computational experiment. Furthermore, the proposed efficient algorithms for pricing out new columns are used. Table 5.8 summarizes the components that are used in the column generation-based solution procedures.

Model	Initial columns	Column generation
Resource and truck scheduling problem	Algorithm 2	Algorithm 3
Multi-mode resource and truck scheduling problem	Algorithm 5	Algorithm 6
Shift and truck scheduling problem	Algorithm 8	Algorithm 9

Table 5.8 Components used in the column generation-based solution procedure.
Source: Own table.

5.3.1 Resource and truck scheduling problem

Table 5.9 summarizes the numerical results for evaluating the **TSFD-RC-F-CG**, that is, the proposed column generation-based solution procedure for solving the **TSFD-RC-F**.

With respect to the **TSFD-RC-F-DT2**, the table shows the average solution time (column "CPU^s"), the number of instances that are solved to optimality (column "$\#^*$"), and the number of instances for which the best known solution is identified (column "$\#^{UB}$"). In addition, it includes the average optimality gap (column "gap^*") and the average gap to the best known solution (column "gap^{UB}") measured in percentage, as well as the number of columns in the MIP (column "$\#^{col}$"). The reporting structure for the **TSFD-RC-F-CG** is slightly different. The column "$\#^s$" contains the number of instances that are solved within the time limit, i.e., the default solver terminates within the time limit. Recall that solving an instance with the **TSFD-RC-F-CG** (a heuristic solution procedure) within the time limit does not necessarily lead to the global optimal solution of the problem instance. Therefore, the number of instances that are solved to

Instances					TSFD-RC-F-DT2						TSFD-RC-F-CG						
$\|\mathcal{T}\|$	Size	$\|\mathcal{D}\|$	$d_i - r_i$	β	CPU^s	$\#^*$	$\#^{UB}$	gap^*	gap^{UB}	$\#^{col}$	CPU^s	$\#^s$	$\#^p$	$\#^{UB}$	gap^*	gap^{UB}	$\#^{col}$
96	**XL**	30	$U(30,50)$	$U(0.0,0.0)$	22.21	20	20	0.00	0.00	752,251	4.84	20	20	20	0.00	0.00	5,508
96	**XL**	30	$U(30,50)$	$U(0.3,0.3)$	32.93	20	20	0.00	0.00	753,751	14.94	20	20	20	0.00	0.00	5,740
96	**XL**	30	$U(30,50)$	$U(0.4,0.5)$	53.63	20	20	0.00	0.00	753,751	16.68	20	20	20	0.00	0.00	5,755
96	**XL**	30	$U(60,80)$	$U(0.0,0.0)$	48.30	20	20	0.00	0.00	753,751	7.88	20	20	20	0.00	0.00	7,878
96	**XL**	30	$U(60,80)$	$U(0.3,0.3)$	106.32	20	20	0.00	0.00	753,001	23.40	20	19	20	0.00	0.00	7,634
96	**XL**	30	$U(60,80)$	$U(0.4,0.5)$	168.44	20	20	0.00	0.00	752,251	41.81	20	20	20	0.00	0.00	7,169
96	**XL**	35	$U(30,50)$	$U(0.0,0.0)$	22.06	20	20	0.00	0.00	877,626	3.49	20	20	20	0.00	0.00	6,201
96	**XL**	35	$U(30,50)$	$U(0.3,0.3)$	32.50	20	20	0.00	0.00	877,626	6.55	20	20	20	0.00	0.00	6,431
96	**XL**	35	$U(30,50)$	$U(0.4,0.5)$	50.72	20	20	0.00	0.00	878,501	11.27	20	20	20	0.00	0.00	6,658
96	**XL**	35	$U(60,80)$	$U(0.0,0.0)$	42.99	20	20	0.00	0.00	878,501	3.91	20	20	20	0.00	0.00	8,725
96	**XL**	35	$U(60,80)$	$U(0.3,0.3)$	68.73	20	20	0.00	0.00	876,751	7.91	20	20	20	0.00	0.00	8,751
96	**XL**	35	$U(60,80)$	$U(0.4,0.5)$	124.31	19	19	0.17	0.17	877,626	22.63	20	17	20	0.00	0.00	8,658
96	**XXL**	40	$U(30,50)$	$U(0.0,0.0)$	60.74	20	20	0.00	0.00	1,411,200	5.55	20	20	20	0.00	0.00	9,759
96	**XXL**	40	$U(30,50)$	$U(0.3,0.3)$	87.96	20	20	0.00	0.00	1,407,000	13.47	20	19	20	0.00	0.00	9,680
96	**XXL**	40	$U(30,50)$	$U(0.4,0.5)$	141.10	19	19	0.14	0.14	1,407,000	28.33	20	20	20	0.00	0.00	10,842
96	**XXL**	40	$U(60,80)$	$U(0.0,0.0)$	162.84	20	20	0.00	0.00	1,405,600	13.58	20	19	20	0.00	0.00	13,903
96	**XXL**	40	$U(60,80)$	$U(0.3,0.3)$	254.37	20	20	0.00	0.00	1,411,200	37.41	20	19	20	0.00	0.00	13,818
96	**XXL**	40	$U(60,80)$	$U(0.4,0.5)$	577.99	13	13	0.85	0.85	1,400,000	82.62	20	20	20	0.00	0.00	13,129
96	**XXL**	45	$U(30,50)$	$U(0.0,0.0)$	52.18	20	20	0.00	0.00	1,586,025	4.34	20	20	20	0.00	0.00	10,803
96	**XXL**	45	$U(30,50)$	$U(0.3,0.3)$	77.36	20	20	0.00	0.00	1,584,450	6.50	20	20	20	0.00	0.00	10,686
96	**XXL**	45	$U(30,50)$	$U(0.4,0.5)$	89.73	20	20	0.00	0.00	1,587,600	13.49	20	19	20	0.00	0.00	10,565
96	**XXL**	45	$U(60,80)$	$U(0.0,0.0)$	109.57	20	20	0.00	0.00	1,582,875	7.90	20	20	20	0.00	0.00	15,471
96	**XXL**	45	$U(60,80)$	$U(0.3,0.3)$	183.46	20	20	0.00	0.00	1,586,025	20.80	20	20	20	0.00	0.00	15,295
96	**XXL**	45	$U(60,80)$	$U(0.4,0.5)$	268.33	18	18	0.24	0.24	1,584,450	32.00	20	18	20	0.00	0.00	15,321
240	**L**	18	$U(30,50)$	$U(0.0,0.0)$	127.36	19	19	0.38	0.38	672,301	20.64	20	19	20	0.00	0.00	4,450
240	**L**	18	$U(30,50)$	$U(0.3,0.3)$	256.22	16	18	1.19	0.56	673,381	112.76	18	17	20	0.63	0.00	4,035
240	**L**	18	$U(30,50)$	$U(0.4,0.5)$	620.57	9	13	2.90	1.82	673,651	256.07	17	14	20	1.08	0.00	4,009
240	**L**	18	$U(60,80)$	$U(0.0,0.0)$	254.95	20	20	0.00	0.00	674,461	32.88	20	18	20	0.00	0.00	5,687
240	**L**	18	$U(60,80)$	$U(0.3,0.3)$	501.08	13	15	6.69	6.10	670,411	156.93	18	14	20	0.59	0.00	5,713
240	**L**	18	$U(60,80)$	$U(0.4,0.5)$	820.30	4	9	4.25	2.94	674,731	415.61	15	12	20	1.31	0.00	5,774
240	**L**	21	$U(30,50)$	$U(0.0,0.0)$	95.53	19	19	0.42	0.42	786,871	9.12	20	19	20	0.00	0.00	3,927
240	**L**	21	$U(30,50)$	$U(0.3,0.3)$	241.89	16	18	1.20	0.59	787,501	79.88	19	18	20	0.61	0.00	4,100
240	**L**	21	$U(30,50)$	$U(0.4,0.5)$	322.65	14	18	1.58	0.53	785,611	117.87	18	16	20	1.05	0.00	4,001
240	**L**	21	$U(60,80)$	$U(0.0,0.0)$	215.60	18	18	0.80	0.80	787,186	24.75	20	20	20	0.00	0.00	6,006
240	**L**	21	$U(60,80)$	$U(0.3,0.3)$	429.91	14	18	1.90	0.62	786,556	207.32	16	15	20	1.27	0.00	6,036
240	**L**	21	$U(60,80)$	$U(0.4,0.5)$	700.07	6	14	3.71	1.57	786,241	459.09	12	12	20	2.14	0.00	5,781
Avg.	**XL**				64.43			0.01	0.01	815,449	13.78				0.00	0.00	7,092
	XXL				172.14			0.10	0.10	1,496,119	22.17				0.00	0.00	12,439
	L				382.18			2.09	1.36	729,909	157.74				0.72	0.00	4,960

$|\mathcal{I}| = 150$, $|\mathcal{I}| = 250$ and $|\mathcal{I}| = 350$ for instances in **L**, **XL**, and **XXL**.
CPU^s: Avg. solution time; $\#^*$: Number of instances that are solved to optimality; $\#^{UB}$: Number of instances for which the best known solution is identified.
$\#^s$: Number of instances for which the procedure terminates within the time limit; $\#^p$: Number of instances that are solved to proven optimality.
gap^*: Avg. optimality gap; gap^{UB}: Avg. gap to the best known solution; $\#^{col}$: Number of columns in the MIP.

Table 5.9 Numerical results: Solving large problem instances with **TSFD-RC-F-DT2** and **TSFD-RC-F-CG**.
Source: Own table.

proven optimality are reported within column "$\#^p$". Specifically, the **TSFD-RC-F-CG** solves an instance to proven optimality if two conditions are met: (i) the instance is solved within the time limit, and (ii) the gap between the identified solution and the LP lower bound is less than one.

The results substantiate some of the observations made when analyzing the MIP formulations in one of the previous sections. It can be seen that the average solution times for instances with wider truck time windows and a higher share of big trucks are longer. Both wider truck time windows and varying operator requirements result in a bigger solution space and hence make an instance harder to solve. Furthermore, the number of available dock-doors appears to affect the solution time. In each testbed, the average solution time increases when less dock-doors are available for processing inbound trucks. Reducing the number of available dock-doors increases the average number of trucks per dock-door (i.e., a higher ratio $\frac{|\mathcal{I}|}{|\mathcal{D}|}$) and makes it more challenging to find feasible truck schedules.

Furthermore, some observations regarding the **TSFD-RC-F-DT2** can be made. The MIP formulation is able to solve the majority of the instances with $|\mathcal{T}| = 96$ from testbeds **XL** and **XXL** to optimality. Only 1 out of 240 instances from **XL** and 10 out of 240 instances from **XXL** cannot be solved to optimality within the time limit. For both testbeds, the average optimality gap is less than 1% for all parameter combinations. However, the **TSFD-RC-F-DT2** struggles to find the optimal solution when a very fine time granularity (i.e., $|\mathcal{T}| = 240$) is applied. Specifically, 72 out of 240 instances (30.0%) cannot be solved to proven optimality within the time limit. Moreover, in 41 out of 240 instances (ca. 17.1%) the solver fails to find the best known solution within the time limit.

The column generation-based solution procedure **TSFD-RC-F-CG**, on the other hand, solves all instances with $|\mathcal{T}| = 96$ within the time limit and identifies the best known solution in all instances. It also finds the best known solution for all instances with a time interval length of two minutes (i.e., $|\mathcal{T}| = 240$), but is unable to terminate within the time limit in 27 out of 240 instances (ca. 11.3%). The average optimality gap is less than 1% for 31 out of 36 parameter combinations and does not exceed 2.2%. Moreover, the **TSFD-RC-F-CG** can prove for ca. 98% of the instances from **XL** and **XXL** and ca. 91% of the solved instances from **L** that the identified solution is the global optimum. This result is very surprising since it is theoretically a heuristic solution procedure. The **TSFD-RC-F-CG** not only outperforms the **TSFD-RC-F-DT2** in terms of solution quality, but also with respect to the computational time. The solution procedure is able to solve the test instances from testbed **XL** and **XXL** on average in ca. 14 seconds and 22 seconds, respectively. The test instances with finer time intervals from testbed **L** have an average solution time of ca. 2.5 minutes. Note that its run time is on average ca. 69%, for some parameter combinations even up to ca. 93%, shorter than the MIP formulation's run time.

When solving problem instances with a very fine time granularity ($|\mathcal{T}| = 240$), both the **TSFD-RC-F-DT2** and **TSFD-RC-F-CG** sometimes struggle to terminate within the time limit. In light of **TSFD-RC-F** being an operational problem that has to be solved frequently, long computations are most likely not acceptable for practitioners. Therefore, the **TSFD-RC-F-DT2** and the **TSFD-RC-F-CG** are further compared with respect to the best found solution within a 60-s time limit. One minute of CPU time should be acceptable even for the most demanding applications. In this test, 40 out of the 72 problem instances from testbed **L** which could not be solved by the default solver within the 15 minutes time limit, are randomly selected. Table 5.10 summarizes the results. In 35 out of 40 instances, the default solver does not find a feasible solution within

	$\vert\mathcal{D}\vert = 18$			$\vert\mathcal{D}\vert = 21$		
ID	Best known solution	**DT2** Gap (%)	**CG** Gap (%)	Best known solution	**DT2** Gap (%)	**CG** Gap (%)
1	13	-	0.00	12	-	0.00
2	16	-	0.00	17	-	0.00
3	16	6.25	6.25	18	-	5.56
4	18	-	0.00	18	-	0.00
5	20	-	0.00	19	-	0.00
6	22	4.55	4.55	17	47.06	0.00
7	19	-	0.00	19	-	5.26
8	19	-	0.00	19	-	0.00
9	18	-	5.56	19	5.26	5.26
10	20	-	0.00	20	-	0.00
11	20	-	0.00	17	-	0.00
12	19	5.26	5.26	20	-	0.00
13	19	-	0.00	19	-	0.00
14	18	-	0.00	22	-	0.00
15	17	-	0.00	13	-	0.00
16	24	-	4.17	18	-	0.00
17	18	-	0.00	19	-	0.00
18	21	-	0.00	16	-	0.00
19	20	-	0.00	17	-	0.00
20	19	-	0.00	16	-	0.00
Avg.		5.35	1.29		26.16	0.80

A dash (-) denotes that no feasible solution was found within the time limit.
DT2: **TSFD-RC-F-DT2**; **CG**: **TSFD-RC-F-CG**.

Table 5.10 Comparison of the **TSFD-RC-F-DT2** and **TSFD-RC-F-CG** given a 60 seconds time limit (instances from testbed **L**).
Source: Own table.

the 60-s time limit when solving the **TSFD-RC-F-DT2**. In the few cases where it finds a feasible solution, the average gap to the best known solution is ca. 14%. The **TSFD-RC-F-CG**, on the other hand, is clearly superior at finding good solutions within the 60-s time limit. It lowers the average gap to the best known solution to ca. 1% and is able to identify the best known solution in 32 out of 40 test instances.

It can be summarized that the **TSFD-RC-F-CG** clearly outperforms the MIP formulation in terms of solution quality and computational time and hence should be used when tackling real-world instances.

5.3.2 Multi-mode resource and truck scheduling problem

Subsequently, the proposed column generation-based solution procedure for solving the multi-mode resource and truck scheduling problem, denoted as **TSFD-RC-V-CG**, will be evaluated. Recall that the number of decision variables in the **TSFD-RC-V** is considerably higher than in the **TSFD-RC-F**, since it also seeks to identify the best operator mode for each truck (i.e., how many operators should process each truck).

The solution procedure is benchmarked against the **TSFD-RC-V-DT2**, which was identified as the best performing MIP formulation before. The numerical results are reported in Table 5.11. With respect to the **TSFD-RC-V-DT2**, the table now also reports the number of instances for which the default solver is able to identify a feasible integer solution within the 15 minutes time limit (column "$\#^f$"). Besides that, the reporting structure is identical with the structure that was used when evaluating the **TSFD-RC-F-CG** in the previous section.

When feeding the **TSFD-RC-V-DT2** into a default solver, the solver is able to find the best known solution for ca. 89% of the instances from testbeds **XL** and **XXL**. The default solver is not able to identify a feasible solution for 6 out of 480 test instances with $|\mathcal{T}| = 96$. For instances from testbed **XL** and **XXL**, the average gap to the best known solution is ca. 0.1% and ca. 3.1%, respectively. Note that the gap considerably increased compared to the single-mode resource and truck scheduling problem. The results also show that the **TSFD-RC-V-DT2** struggles to solve instances with a very fine time granularity (i.e., $|\mathcal{T}| = 240$). Ca. 41.3% (99 out of 240 instances) cannot be solved to proven optimality within the time limit. Moreover, the solver is unable to find the best known solution in 25% of the instances and even fails to identify a feasible solution in 17 out of 240 instances (ca. 7%).

The heuristic solution procedure **TSFD-RC-V-CG**, on the other hand, finds the best known solution for all instances with $|\mathcal{T}| = 96$. Only in 3 out of 480 instances from testbed **XL** and **XXL**, the procedure does not terminate within the time limit. Moreover, the column generation-based

Instances					TSFD-RC-V-DT2							TSFD-RC-V-CG						
$\lvert\mathcal{T}\rvert$	Size	$\lvert\mathcal{D}\rvert$	$d_i - r_i$	β	CPU^s	$\#^f$	$\#^*$	$\#^{UB}$	gap^*	gap^{UB}	$\#^{col}$	CPU^s	$\#^s$	$\#^p$	$\#^{UB}$	gap^*	gap^{UB}	$\#^{col}$
96	**XL**	30	$\sim U(30,50)$	$\sim U(0.0,0.0)$	63.39	20	20	20	0.00	0.00	1,504,500	9.98	20	20	20	0.00	0.00	7,521
96	**XL**	30	$\sim U(30,50)$	$\sim U(0.3,0.3)$	149.33	20	19	19	0.22	0.22	1,507,500	7.64	20	20	20	0.00	0.00	6,570
96	**XL**	30	$\sim U(30,50)$	$\sim U(0.4,0.5)$	181.13	20	18	20	0.41	0.00	1,507,500	77.91	19	18	20	0.41	0.00	6,337
96	**XL**	30	$\sim U(60,80)$	$\sim U(0.0,0.0)$	143.93	20	20	20	0.00	0.00	1,507,500	9.66	20	20	20	0.00	0.00	9,704
96	**XL**	30	$\sim U(60,80)$	$\sim U(0.3,0.3)$	333.15	20	17	18	0.65	0.44	1,506,000	50.72	20	18	20	0.21	0.00	8,721
96	**XL**	30	$\sim U(60,80)$	$\sim U(0.4,0.5)$	352.12	20	17	18	0.63	0.42	1,504,500	80.22	19	18	20	0.21	0.00	8,071
96	**XL**	35	$\sim U(30,50)$	$\sim U(0.0,0.0)$	65.03	20	20	20	0.00	0.00	1,755,250	4.94	20	20	20	0.00	0.00	8,181
96	**XL**	35	$\sim U(30,50)$	$\sim U(0.3,0.3)$	68.31	20	20	20	0.00	0.00	1,755,250	4.76	20	20	20	0.00	0.00	7,266
96	**XL**	35	$\sim U(30,50)$	$\sim U(0.4,0.5)$	131.80	20	19	19	0.21	0.21	1,757,000	10.63	20	20	20	0.00	0.00	6,579
96	**XL**	35	$\sim U(60,80)$	$\sim U(0.0,0.0)$	163.33	20	19	19	0.25	0.25	1,757,000	7.17	20	20	20	0.00	0.00	10,846
96	**XL**	35	$\sim U(60,80)$	$\sim U(0.3,0.3)$	178.12	20	20	20	0.00	0.00	1,753,500	8.55	20	19	20	0.00	0.00	9,400
96	**XL**	35	$\sim U(60,80)$	$\sim U(0.4,0.5)$	211.97	20	20	20	0.00	0.00	1,755,250	12.36	20	20	20	0.00	0.00	8,583
96	**XXL**	40	$\sim U(30,50)$	$\sim U(0.0,0.0)$	217.83	20	20	20	0.00	0.00	2,822,400	16.21	20	19	20	0.00	0.00	12,958
96	**XXL**	40	$\sim U(30,50)$	$\sim U(0.3,0.3)$	228.48	20	19	20	0.17	0.00	2,814,000	26.27	20	19	20	0.17	0.00	11,601
96	**XXL**	40	$\sim U(30,50)$	$\sim U(0.4,0.5)$	410.62	20	19	19	0.15	0.15	2,814,000	28.61	20	19	20	0.00	0.00	10,370
96	**XXL**	40	$\sim U(60,80)$	$\sim U(0.0,0.0)$	542.70	20	19	19	0.18	0.18	2,811,200	21.49	20	20	20	0.00	0.00	17,165
96	**XXL**	40	$\sim U(60,80)$	$\sim U(0.3,0.3)$	711.37	20	10	10	1.64	1.64	2,822,400	44.01	20	19	20	0.00	0.00	14,953
96	**XXL**	40	$\sim U(60,80)$	$\sim U(0.4,0.5)$	862.03	17	2	2	18.03	17.88	2,800,000	52.36	20	19	20	0.14	0.00	13,427
96	**XXL**	45	$\sim U(30,50)$	$\sim U(0.0,0.0)$	207.62	20	20	20	0.00	0.00	3,172,050	22.36	20	17	20	0.00	0.00	14,203
96	**XXL**	45	$\sim U(30,50)$	$\sim U(0.3,0.3)$	324.55	20	19	19	0.17	0.17	3,168,900	16.59	20	19	20	0.00	0.00	12,229
96	**XXL**	45	$\sim U(30,50)$	$\sim U(0.4,0.5)$	332.29	20	19	19	0.15	0.15	3,175,200	9.53	20	20	20	0.00	0.00	10,776
96	**XXL**	45	$\sim U(60,80)$	$\sim U(0.0,0.0)$	524.72	20	17	17	0.54	0.54	3,165,750	15.02	20	20	20	0.00	0.00	19,061
96	**XXL**	45	$\sim U(60,80)$	$\sim U(0.3,0.3)$	575.78	19	15	15	5.65	5.65	3,172,050	19.94	20	19	20	0.00	0.00	16,071
96	**XXL**	45	$\sim U(60,80)$	$\sim U(0.4,0.5)$	697.75	18	13	14	10.76	10.61	3,168,900	70.60	19	17	20	0.16	0.00	14,118
240	**L**	18	$\sim U(30,50)$	$\sim U(0.0,0.0)$	349.69	20	16	18	1.54	0.77	1,344,600	103.21	18	18	20	0.77	0.00	4,773
240	**L**	18	$\sim U(30,50)$	$\sim U(0.3,0.3)$	561.85	20	12	17	2.72	1.00	1,359,180	262.88	15	14	20	1.72	0.00	4,707
240	**L**	18	$\sim U(30,50)$	$\sim U(0.4,0.5)$	745.14	20	7	15	3.97	1.51	1,384,560	378.36	13	12	20	2.46	0.00	5,066
240	**L**	18	$\sim U(60,80)$	$\sim U(0.0,0.0)$	514.76	19	15	17	6.57	5.80	1,348,920	67.20	19	18	20	0.77	0.00	6,249
240	**L**	18	$\sim U(60,80)$	$\sim U(0.3,0.3)$	839.36	17	3	6	19.45	18.48	1,366,200	225.39	18	16	20	1.34	0.00	6,619
240	**L**	18	$\sim U(60,80)$	$\sim U(0.4,0.5)$	887.45	15	1	3	30.97	28.25	1,387,260	497.61	10	7	20	3.48	0.00	7,082
240	**L**	21	$\sim U(30,50)$	$\sim U(0.0,0.0)$	249.67	20	18	20	0.77	0.00	1,573,740	57.96	19	18	20	0.77	0.00	4,860
240	**L**	21	$\sim U(30,50)$	$\sim U(0.3,0.3)$	341.04	20	17	20	1.08	0.00	1,606,500	117.94	18	17	20	1.08	0.00	4,916
240	**L**	21	$\sim U(30,50)$	$\sim U(0.4,0.5)$	402.90	20	16	20	1.27	0.00	1,614,060	161.16	17	14	20	1.27	0.00	4,860
240	**L**	21	$\sim U(60,80)$	$\sim U(0.0,0.0)$	519.15	19	16	16	6.19	6.19	1,574,370	103.48	19	19	20	0.38	0.00	6,720
240	**L**	21	$\sim U(60,80)$	$\sim U(0.3,0.3)$	647.21	18	14	16	11.40	10.69	1,609,020	145.67	18	17	20	1.05	0.00	6,965
240	**L**	21	$\sim U(60,80)$	$\sim U(0.4,0.5)$	815.51	15	6	12	27.77	25.94	1,612,800	308.60	14	11	20	2.47	0.00	6,673
Avg.	**XL**				170.13				0.20	0.13	1,630,896	23.71				0.07	0.00	8,148
	XXL				469.65				3.12	3.08	2,992,238	28.58				0.04	0.00	13,911
	L				572.81				9.48	8.22	1,481,768	202.46				1.46	0.00	5,791

$\lvert\mathcal{I}\rvert = 150$, $\lvert\mathcal{I}\rvert = 250$, and $\lvert\mathcal{I}\rvert = 350$ for instances in **L**, **XL**, and **XXL**.
CPU^s: Avg. solution time; $\#^f$: Number of instances for which a feasible solution is found; $\#^*$: Number of instances that are solved to optimality.
$\#^{UB}$: Number of instances for which the best known solution is identified; $\#^s$: Number of instances for which the procedure terminates within the time limit.
$\#^p$: Number of instances that are solved to proven optimality; gap^*: Avg. optimality gap; gap^{UB}: Avg. gap to the best known solution; $\#^{col}$: Number of columns in the MIP.

Table 5.11 Numerical results: Solving large problem instances with **TSFD-RC-V-DT2** and **TSFD-RC-V-CG**.
Source: Own table.

solution procedure also identifies the best known solution for all instances with a very fine time granularity ($|\mathcal{T}| = 240$). With an average optimality gap of less than 1.5%, it is capable to find near-optimal solutions. Surprisingly, the **TSFD-RC-V-CG** – a heuristic procedure – is able to prove for ca. 96% of the instances from **XL** and **XXL**, and ca. 91% of the solved instances from **L** that the identified solution is the global optimum. With an average solution time of less than 30 seconds for instances from **XL** and **XXL**, and less than 3 minutes for instances from **L**, the heuristic procedure achieves up to 97% shorter average run times than the default solver.

In 42 out of 240 instances from testbed **L** – almost double the number compared to the **TSFD-RC-F-CG** – the heuristic solution procedure does not terminate within the 15 minutes time limit. Therefore, the **TSFD-RC-V-DT2** and the **TSFD-RC-V-CG** are compared with respect to the best found solution within a 60-s time limit. For this purpose, 40 out of the 99 problem instances from testbed **L** which could not be solved by the default solver within the 15 minutes time limit, are randomly selected. Table 5.12 reports the results. In all instances, the default solver is unable to identify a feasible solution within the 60-s time limit when solving the **TSFD-RC-V-DT2**. This may be unacceptable for practitioners, since the multi-mode resource and truck scheduling problem is an operational problem which must be solved frequently. When applying the heuristic solution procedure **TSFD-RC-V-CG**, on the other hand, near-optimal solution can be found within the time limit. Specifically, the procedure is capable to find the best known solution within a 60-s time limit in 31 out of 40 test instances.

Again, it can be concluded that the proposed heuristic solution procedure clearly outperforms the MIP formulation in terms of solution quality and computational time.

5.3.3 Shift and truck scheduling problem

This section sets out to evaluate the performance of the heuristic solution procedure proposed for the shift and truck scheduling problem. Therefore, the multi-mode shift and truck scheduling problem **ISTSFD-V** is used for the comparison. It deals with a considerably larger number of decision variables than the **ISTSFD-F** as it also determines the best operator mode for each inbound truck.

The heuristic solution procedure, in the following denoted as **ISTSFD-V-CG**, is benchmarked against the **ISTSFD-V-DT2**, which was identified as the best performing MIP formulation before. Table 5.13 reports the numerical results. The reporting structure is identical with the structure that was used when evaluating the **TSFD-RC-F-CG** in one of the previous sections.

ID	$\|\mathcal{D}\| = 18$ Best known solution	**DT2** Gap (%)	**CG** Gap (%)	$\|\mathcal{D}\| = 21$ Best known solution	**DT2** Gap (%)	**CG** Gap (%)
1	13	-	0.00	14	-	0.00
2	14	-	0.00	14	-	0.00
3	13	-	0.00	14	-	0.00
4	13	-	0.00	17	-	0.00
5	16	-	0.00	14	-	0.00
6	15	-	0.00	16	-	0.00
7	16	-	0.00	18	-	0.00
8	16	-	0.00	17	-	0.00
9	15	-	0.00	16	-	0.00
10	20	-	5.00	13	-	0.00
11	17	-	5.88	14	-	7.14
12	17	-	0.00	16	-	6.25
13	14	-	0.00	16	-	0.00
14	13	-	7.69	16	-	0.00
15	16	-	0.00	18	-	0.00
16	17	-	0.00	18	-	0.00
17	15	-	0.00	17	-	0.00
18	15	-	0.00	18	-	0.00
19	18	-	5.56	16	-	6.25
20	16	-	6.25	16	-	6.25
Avg.		-	1.52		-	1.29

A dash (-) denotes that no feasible solution was found within the time limit.
DT2: **TSFD-RC-V-DT2**; **CG**: **TSFD-RC-V-CG**.

Table 5.12 Comparison of the **TSFD-RC-V-DT2** and **TSFD-RC-V-CG** given a 60 seconds time limit (instances from testbed **L**).
Source: Own table.

Compared to the **TSFD-RC-V**, the average solution time for solving the **ISTSFD-V** with a default solver has decreased. While the default solver is able to solve almost all instances from testbed **XL**, it is unable to find the optimal solution for 15 and 51 instances from **XXL** and **L**, respectively. Moreover, the default solver sometimes struggles to identify the best known solution within the time limit. With an average gap of ca. 0.25%, however, the default solver reliably finds solutions that are close to the best known solutions.

Regarding the heuristic solution procedure, it can be seen that the **ISTSFD-V-CG** contains less than 0.5% of the MIP's number of columns. This, in turn, results in considerably shorter solution times compared to the **ISTSFD-V-DT2**. Specifically, instances in **XL**, **XXL**, and **L** are on average solved ca. eleven times, seven times, and three times faster, respectively. Only one instance from testbed **XXL** and 21 instances from testbed **L** cannot be solved within the 15 minutes time limit. However, the **ISTSFD-V-CG** is able to identify the best known solution in all 720 instances.

Instances					ISTSFD-V-DT2						ISTSFD-V-CG						
$\|\mathcal{T}\|$	Size	$\|\mathcal{D}\|$	$d_i - r_i$	β	CPU^s	$\#^*$	$\#^{UB}$	gap^*	gap^{UB}	$\#^{col}$	CPU^s	$\#^s$	$\#^p$	$\#^{UB}$	gap^*	gap^{UB}	$\#^{col}$
96	**XL**	30	$U(30,50)$	$U(0.0,0.0)$	51.37	20	20	0.00	0.00	1,504,500	5.39	20	20	20	0.00	0.00	7,391
96	**XL**	30	$U(30,50)$	$U(0.3,0.3)$	63.22	20	20	0.00	0.00	1,543,500	7.10	20	20	20	0.00	0.00	6,781
96	**XL**	30	$U(30,50)$	$U(0.4,0.5)$	114.86	20	20	0.00	0.00	1,549,500	11.28	20	20	20	0.00	0.00	6,536
96	**XL**	30	$U(60,80)$	$U(0.0,0.0)$	90.68	20	20	0.00	0.00	1,507,500	9.41	20	19	20	0.00	0.00	9,898
96	**XL**	30	$U(60,80)$	$U(0.3,0.3)$	139.36	20	20	0.00	0.00	1,549,500	15.48	20	18	20	0.00	0.00	8,962
96	**XL**	30	$U(60,80)$	$U(0.4,0.5)$	271.47	18	19	0.36	0.18	1,552,500	23.65	20	18	20	0.18	0.00	8,343
96	**XL**	35	$U(30,50)$	$U(0.0,0.0)$	54.47	20	20	0.00	0.00	1,755,250	7.33	20	20	20	0.00	0.00	8,378
96	**XL**	35	$U(30,50)$	$U(0.3,0.3)$	56.99	20	20	0.00	0.00	1,793,750	5.17	20	20	20	0.00	0.00	7,431
96	**XL**	35	$U(30,50)$	$U(0.4,0.5)$	66.96	20	20	0.00	0.00	1,811,250	6.38	20	18	20	0.00	0.00	6,739
96	**XL**	35	$U(60,80)$	$U(0.0,0.0)$	129.24	19	20	0.22	0.00	1,757,000	6.85	20	19	20	0.22	0.00	10,895
96	**XL**	35	$U(60,80)$	$U(0.3,0.3)$	111.80	20	20	0.00	0.00	1,804,250	7.36	20	19	20	0.00	0.00	9,575
96	**XL**	35	$U(60,80)$	$U(0.4,0.5)$	154.09	20	20	0.00	0.00	1,811,250	9.19	20	20	20	0.00	0.00	8,785
96	**XXL**	40	$U(30,50)$	$U(0.0,0.0)$	131.22	20	20	0.00	0.00	2,822,400	9.78	20	20	20	0.00	0.00	13,057
96	**XXL**	40	$U(30,50)$	$U(0.3,0.3)$	211.51	19	19	0.14	0.14	2,898,000	32.27	20	20	20	0.00	0.00	11,797
96	**XXL**	40	$U(30,50)$	$U(0.4,0.5)$	298.67	17	17	0.41	0.41	2,909,200	74.65	20	20	20	0.00	0.00	10,666
96	**XXL**	40	$U(60,80)$	$U(0.0,0.0)$	195.74	20	20	0.00	0.00	2,811,200	14.85	20	19	20	0.00	0.00	17,391
96	**XXL**	40	$U(60,80)$	$U(0.3,0.3)$	373.40	18	18	0.29	0.29	2,895,200	29.96	20	20	20	0.00	0.00	15,317
96	**XXL**	40	$U(60,80)$	$U(0.4,0.5)$	503.61	14	15	0.78	0.64	2,909,200	192.64	19	19	20	0.14	0.00	13,843
96	**XXL**	45	$U(30,50)$	$U(0.0,0.0)$	108.40	20	20	0.00	0.00	3,172,050	8.30	20	18	20	0.00	0.00	14,400
96	**XXL**	45	$U(30,50)$	$U(0.3,0.3)$	136.39	20	20	0.00	0.00	3,244,500	9.25	20	20	20	0.00	0.00	12,468
96	**XXL**	45	$U(30,50)$	$U(0.4,0.5)$	193.02	19	20	0.14	0.00	3,266,550	12.11	20	19	20	0.14	0.00	10,912
96	**XXL**	45	$U(60,80)$	$U(0.0,0.0)$	217.40	20	20	0.00	0.00	3,165,750	13.38	20	20	20	0.00	0.00	19,202
96	**XXL**	45	$U(60,80)$	$U(0.3,0.3)$	267.77	19	20	0.14	0.00	3,263,400	20.44	20	17	20	0.14	0.00	16,296
96	**XXL**	45	$U(60,80)$	$U(0.4,0.5)$	342.48	19	19	0.13	0.13	3,272,850	17.38	20	19	20	0.00	0.00	14,518
240	**L**	18	$U(30,50)$	$U(0.0,0.0)$	131.88	20	20	0.00	0.00	1,344,600	11.41	20	18	20	0.00	0.00	4,616
240	**L**	18	$U(30,50)$	$U(0.3,0.3)$	335.42	16	18	1.24	0.61	1,374,300	90.70	19	15	20	0.62	0.00	4,858
240	**L**	18	$U(30,50)$	$U(0.4,0.5)$	639.11	9	17	2.97	0.77	1,385,640	388.83	13	11	20	2.19	0.00	5,165
240	**L**	18	$U(60,80)$	$U(0.0,0.0)$	370.67	17	17	1.08	1.08	1,348,920	62.77	20	19	20	0.00	0.00	6,422
240	**L**	18	$U(60,80)$	$U(0.3,0.3)$	423.82	16	18	1.10	0.54	1,366,200	117.72	19	17	20	0.56	0.00	6,835
240	**L**	18	$U(60,80)$	$U(0.4,0.5)$	613.90	10	14	2.40	1.35	1,387,260	251.94	17	16	20	1.04	0.00	6,986
240	**L**	21	$U(30,50)$	$U(0.0,0.0)$	139.13	20	20	0.00	0.00	1,573,740	24.34	20	19	20	0.00	0.00	4,848
240	**L**	21	$U(30,50)$	$U(0.3,0.3)$	231.93	18	19	0.61	0.33	1,606,500	26.78	20	17	20	0.28	0.00	4,912
240	**L**	21	$U(30,50)$	$U(0.4,0.5)$	289.13	17	19	0.87	0.28	1,614,060	114.07	18	14	20	0.59	0.00	4,939
240	**L**	21	$U(60,80)$	$U(0.0,0.0)$	306.55	17	18	1.00	0.65	1,574,370	114.00	19	15	20	0.36	0.00	6,938
240	**L**	21	$U(60,80)$	$U(0.3,0.3)$	477.10	13	17	2.18	0.88	1,609,020	214.73	16	15	20	1.29	0.00	7,110
240	**L**	21	$U(60,80)$	$U(0.4,0.5)$	405.93	16	18	1.12	0.53	1,612,800	126.47	18	17	20	0.59	0.00	6,976
Avg.	**XL**				108.71			0.05	0.02	1,661,646	9.55				0.03	0.00	8,309
	XXL				248.30			0.17	0.13	3,052,525	36.25				0.04	0.00	14,156
	L				363.71			1.21	0.59	1,483,118	128.65				0.63	0.00	5,884

$|\mathcal{I}| = 150$, $|\mathcal{I}| = 250$ and $|\mathcal{I}| = 350$ for instances in **L**, **XL**, and **XXL**.
CPU^s: Avg. solution time; $\#^*$: Number of instances that are solved to optimality; $\#^{UB}$: Number of instances for which the best known solution is identified.
$\#^s$: Number of instances for which the procedure terminates within the time limit; $\#^p$: Number of instances that are solved to proven optimality.
gap^*: Avg. optimality gap; gap^{UB}: Avg. gap to the best known solution; $\#^{col}$: Number of columns in the MIP.

Table 5.13 Numerical results: Solving large problem instances with **ISTSFD-V-DT2** and **ISTSFD-V-CG**.
Source: Own table.

Furthermore, the **ISTSFD-V-CG** can prove for almost 94% of the solved instances that that the identified solution is the global optimum.

Since both the default solver and the heuristic solution procedure do not terminate within the time limit for some of the problem instances from testbed **L**, they are further compared in terms of the best found solution within a 60-s time limit. For this analysis, 40 out of the 51 instances from testbed **L** which could not be solved by the default solver, are randomly selected. The numerical results are reported in Table 5.14. The default solver fails to solve the instances to

	$\|\mathcal{D}\| = 18$			$\|\mathcal{D}\| = 21$		
ID	Best known solution	**DT2** Gap (%)	**CG** Gap (%)	Best known solution	**DT2** Gap (%)	**CG** Gap (%)
1	21	-	0.00	15	-	6.67
2	19	-	0.00	19	-	0.00
3	17	-	5.88	18	-	5.56
4	20	-	0.00	17	-	0.00
5	19	-	0.00	19	-	0.00
6	20	-	0.00	15	-	0.00
7	19	-	0.00	15	-	6.67
8	20	-	0.00	16	-	6.25
9	18	-	0.00	16	-	0.00
10	21	-	4.76	18	-	5.56
11	19	-	5.26	17	-	0.00
12	17	-	0.00	16	-	0.00
13	13	-	7.69	17	-	5.88
14	15	-	6.67	16	-	6.25
15	18	-	5.56	17	-	0.00
16	19	-	0.00	19	-	0.00
17	18	-	0.00	17	-	0.00
18	25	-	4.00	20	-	5.00
19	20	-	0.00	18	-	5.56
20	21	-	0.00	15	-	0.00
Avg.		-	1.99		-	2.67

A dash (-) denotes that no feasible solution was found within the time limit.
DT2: **ISTSFD-V-DT2**; **CG**: **ISTSFD-V-CG**.

Table 5.14 Comparison of the **ISTSFD-V-DT2** and **ISTSFD-V-CG** given a 60 seconds time limit (instances from testbed **L**).
Source: Own table.

feasibility within the 60-s time limit. With the **ISTSFD-V** being an operational problem that must be solved frequently, this may be unacceptable for practical applications. When tackling large instances with a fine time granularity with the **ISTSFD-V-CG**, on the other hand, good solutions close to the best known solutions can be identified in 60s. The procedure even finds the best known solution in 24 out of 40 test instances.

Again, it can be concluded that the proposed heuristic solution procedure outperforms the MIP formulation in terms of solution quality and computational time.

5.4 Managerial insights

This section evaluates the solution quality of the **TSFD-RC-F**, **TSFD-RC-V**, and **ISTSFDs** and, hence, the potential benefit of utilizing resource-related performance measures in truck scheduling models.

The numerical studies, which call for test instances that can be solved to proven optimality within a short time, use the small- and moderate-sized instances from testbeds **S** and **M**. Throughout the entire experiment, $|\mathcal{T}|$ is set to 48. The number of dock-doors $|\mathcal{D}|$ is chosen from the sets $\{6, 7, 8\}$ and $\{9, 11, 13\}$ for instances from **S** and **M**, respectively. Hence, the average number of inbound trucks per dock-door $\frac{|\mathcal{I}|}{|\mathcal{D}|}$ varies between six and nine. Similar to the experiments in the previous sections, test instances with short and moderate truck time windows (i.e., $(d_i - r_i) \in \{\sim U(30, 50), \sim U(60, 80)\}$) as well as test instances with different shares of big trucks (i.e., $\beta \in \{\sim U(0.0, 0.0), \sim U(0.2, 0.3), \sim U(0.4, 0.5)\}$), are included. Hence, each numerical experiment includes a total of 720 test instances with 20 instances for each parameter combination.

5.4.1 Resource and truck scheduling problem

In order to evaluate the solution quality of the **TSFD-RC-F**, it is benchmarked against a truck scheduling model for minimizing the makespan, denoted as **TSFD-MS-F**, and a truck scheduling model for minimizing the total truck processing time, denoted as **TSFD-PT-F**. Both the **TSFD-MS-F** and **TSFD-PT-F** can be found in Appendix. Moreover, this section investigates the operator utilization and the role of other factors (e.g., width of truck time windows, service level, etc.) in the **TSFD-RC-F**.

Benefits of utilizing operator requirements as the key performance indicator

This section sets out to analyze whether utilizing the operator requirements as the dominant key performance indicator results in more efficient cross-docking operations. For this purpose, the **TSFD-RC-F** is compared with a truck scheduling model for makespan minimization, denoted as

TSFD-MS-F, and a truck scheduling model for total truck processing time minimization, denoted as **TSFD-PT-F**. The makespan and the total truck processing time are frequently used objective functions in truck scheduling models that do not explicitly consider the required operators. The three models are compared in terms of the following performance indicators:

1. Makespan C: The time span that elapses from the start of processing the first inbound truck until the end of processing the last inbound truck.
2. Total truck throughput time (TPT): The cumulated time that all trucks spend at the cross-docking terminal. Specifically, it includes:
   a) Total truck waiting time H: The cumulated time that inbound trucks spend on the yard prior to being unloaded at one of the dock-doors.
   b) Total truck processing time P: The cumulated time that is required to unload all inbound trucks and transfer the product units from the inbound area to the outbound area.
3. Required operators W: The number of operators required to accomplish the workload.

The truck throughput time is included as a performance indicator to acknowledge for the truck driver detention at warehousing facilities, a significant issue for the trucking industry. It was estimated that motor carriers in the United States could save over $3 billion annually from eliminating time inefficiencies related to loading and unloading operations[1]. Truck drivers in the United States wait on average ca. 2.5h (often without getting paid during that time) and experience more frequent pickup and delivery delays at warehousing facilities. These detentions regularly create cascading effects on subsequent pick-ups and deliveries, and reduce the earnings of both trucking companies and drivers.[2] Costello and Karickhoff even identify the reduction of truck throughput times at facilities as a measure to reduce the truck driver shortage in the United States. By reducing a truck driver's throughput time at warehouse facilities, he can drive more miles within the hours-of-service limits which increases his and, ultimately, the whole trucking industry's effective capacity.[3]

Table 5.15 reports the descriptive statistics for the numerical experiment. Note that, naturally, the **TSFD-MS-F**, **TSFD-PT-F**, and **TSFD-RC-F** compute truck schedules with the minimum makespan,

[1] Belella et al. (2009, p. 20).

[2] Speltz and Murray (2019, p. 9).

[3] Costello and Karickhoff (2019, p. 14).

		Makespan			Total truck waiting time			Total truck processing time			Required operators		
Size		**MS**	**PT**	**RC**	**MS**	**PT**	**RC**	**MS**	**PT**	**RC**	**MS**	**PT**	**RC**
S	N	360	360	360	360	360	360	360	360	360	360	360	360
	Mean	404.3	423.1	429.5	570.1	676.8	635.8	1,953.2	1,759.4	1,856.4	9.5	9.1	7.7
	Median	400.0	430.0	430.0	530.0	630.0	590.0	1,940.0	1,750.0	1,850.0	10.0	9.0	8.0
	SD	20.5	19.6	22.0	225.2	241.2	258.4	107.0	68.3	96.7	2.2	2.0	1.7
M	N	360	360	360	360	360	360	360	360	360	360	360	360
	Mean	402.2	426.2	434.7	861.6	1,023.9	927.1	3,031.5	2,627.9	2,791.7	14.2	13.1	10.5
	Median	400.0	430.0	440.0	830.0	970.0	880.0	3,010.0	2,610.0	2,780.0	14.0	13.0	10.0
	SD	17.4	15.8	20.5	278.3	329.0	328.6	145.6	82.4	130.7	3.2	2.7	2.0

$|\mathcal{I}| = 50$ and $|\mathcal{I}| = 80$ for instances in **S** and **M**.

N: Number of observations; SD: Standard deviation.

Mean, median, and standard deviation in minutes.

MS: **TSFD-MS-F**; **PT**: **TSFD-PT-F**; **RC**: **TSFD-RC-F**.

Table 5.15 Descriptive statistics for different single-mode truck scheduling models and key performance indicators.
Source: Own table.

minimum total truck processing time, and minimum operator requirements, respectively. The statistics reveal that the **TSFD-PT-F** and **TSFD-RC-F** compute truck schedules with a ca. 6-8% longer makespan than the truck schedules computed with the **TSFD-MS-F**. Since the required service level is met in all cases, this marginal increase has no negative effect on the internal operations. It can also be seen that the total truck waiting time is smaller when using the **TSFD-MS-F**. In order to minimize the makespan, the model tends to start processing trucks soon after their arrival at the cross-docking terminal. This yields to ca. 18% and 10% shorter average truck waiting times compared to the **TSFD-PT-F** and **TSFD-RC-F**, respectively. With an average waiting time of ca. 11 minutes per truck, however, this is a rather small improvement. In addition, the table provides the models' average performance regarding the truck processing time. As in the case of truck waiting time, a longer truck processing time can be considered negative since it could turn into a lengthened truck throughput time. The average truck processing time, however, only slightly increases by ca. 2 minutes and ca. 4-5 minutes when applying the **TSFD-RC-F** and **TSFD-MS-F**, respectively. Achieving a minimum makespan (**TSFD-MS-F**) or a minimum total truck processing time (**TSFD-PT-F**) comes at a price, as it requires a significantly larger number of operators according to the statistics. Compared with the **TSFD-RC-F**, both models generate truck schedules that use ca. 20-35% more operators. This strong surge is unfavorable since it turns into higher labor cost and, consequently, into higher total operational cost of the cross-docking terminal. Hence, it can be concluded that the **TSFD-RC-F** outperforms both the makespan and processing time models.

Impact of exogenous factors on operator requirements

In order to assess whether and how exogenous factors affect the dominance of the **TSFD-RC-F**, the models' operator requirements for different parameter combinations are explored in the following. Specifically, the influence of factors such as the number of available dock-doors $|\mathcal{D}|$, the width of time windows $(d_i - r_i)$, and the share of big trucks β on the number of operators W is analyzed. Table 5.16 reports the operator requirements of the single-mode truck scheduling models for different parameter combinations. It can be seen from the table that the makespan model performs worst. For each parameter combination, it generates truck schedules with the highest average operator requirements. By applying the **TSFD-PT-F**, the number of operators can be reduced in most cases. As expected, the **TSFD-RC-F** outperforms the other two single-mode truck scheduling models as it guarantees to compute truck schedules that can be executed with a minimum number of operators. Interestingly, the potential saving strongly varies among the different parameter combinations. In very few cases, the **TSFD-RC-F** is able to reduce the operator requirements by less than 10% compared to the makespan and processing time models.

Instances				Required operators (avg.)			Gap w/ **TSFD-RC-F**	
Size	$\|\mathcal{D}\|$	$d_i - r_i$	β	W_{MS}^{avg}	W_{PT}^{avg}	W_{RC}^{avg}	Δ_{MS}	Δ_{PT}
S	6	$\sim U(30,50)$	$\sim U(0.0,0.0)$	6.00	6.00	5.75	4.3%	4.3%
S	6	$\sim U(30,50)$	$\sim U(0.2,0.3)$	9.30	9.05	8.00	16.3%	13.1%
S	6	$\sim U(30,50)$	$\sim U(0.4,0.5)$	10.60	10.40	9.60	10.4%	8.3%
S	6	$\sim U(60,80)$	$\sim U(0.0,0.0)$	6.00	6.00	5.85	2.6%	2.6%
S	6	$\sim U(60,80)$	$\sim U(0.2,0.3)$	9.25	9.15	7.80	18.6%	17.3%
S	6	$\sim U(60,80)$	$\sim U(0.4,0.5)$	10.70	10.40	9.50	12.6%	9.5%
S	7	$\sim U(30,50)$	$\sim U(0.0,0.0)$	7.00	6.80	6.05	15.7%	12.4%
S	7	$\sim U(30,50)$	$\sim U(0.2,0.3)$	9.80	9.60	8.00	22.5%	20.0%
S	7	$\sim U(30,50)$	$\sim U(0.4,0.5)$	11.75	11.20	9.50	23.7%	17.9%
S	7	$\sim U(60,80)$	$\sim U(0.0,0.0)$	7.00	6.95	6.05	15.7%	14.9%
S	7	$\sim U(60,80)$	$\sim U(0.2,0.3)$	9.85	9.80	7.40	33.1%	32.4%
S	7	$\sim U(60,80)$	$\sim U(0.4,0.5)$	11.55	11.45	8.90	29.8%	28.7%
S	8	$\sim U(30,50)$	$\sim U(0.0,0.0)$	7.90	7.50	6.20	27.4%	21.0%
S	8	$\sim U(30,50)$	$\sim U(0.2,0.3)$	10.90	9.70	8.15	33.7%	19.0%
S	8	$\sim U(30,50)$	$\sim U(0.4,0.5)$	12.10	10.95	8.45	43.2%	29.6%
S	8	$\sim U(60,80)$	$\sim U(0.0,0.0)$	7.85	7.35	5.75	36.5%	27.8%
S	8	$\sim U(60,80)$	$\sim U(0.2,0.3)$	11.05	10.25	7.90	39.9%	29.7%
S	8	$\sim U(60,80)$	$\sim U(0.4,0.5)$	12.60	11.95	8.90	41.6%	34.3%
M	9	$\sim U(30,50)$	$\sim U(0.0,0.0)$	9.00	8.95	8.55	5.3%	4.7%
M	9	$\sim U(30,50)$	$\sim U(0.2,0.3)$	12.95	12.90	11.00	17.7%	17.3%
M	9	$\sim U(30,50)$	$\sim U(0.4,0.5)$	14.85	14.15	12.75	16.5%	11.0%
M	9	$\sim U(60,80)$	$\sim U(0.0,0.0)$	9.00	9.00	8.55	5.3%	5.3%
M	9	$\sim U(60,80)$	$\sim U(0.2,0.3)$	12.95	12.75	10.95	18.3%	16.4%
M	9	$\sim U(60,80)$	$\sim U(0.4,0.5)$	15.50	15.30	12.85	20.6%	19.1%
M	11	$\sim U(30,50)$	$\sim U(0.0,0.0)$	10.95	10.53	8.74	25.3%	20.5%
M	11	$\sim U(30,50)$	$\sim U(0.2,0.3)$	14.89	13.21	10.68	39.4%	23.7%
M	11	$\sim U(30,50)$	$\sim U(0.4,0.5)$	17.30	15.60	12.40	39.5%	25.8%
M	11	$\sim U(60,80)$	$\sim U(0.0,0.0)$	11.00	10.55	8.45	30.2%	24.9%
M	11	$\sim U(60,80)$	$\sim U(0.2,0.3)$	15.15	13.75	10.15	49.3%	35.5%
M	11	$\sim U(60,80)$	$\sim U(0.4,0.5)$	17.35	16.00	12.35	40.5%	29.6%
M	13	$\sim U(30,50)$	$\sim U(0.0,0.0)$	12.10	10.60	8.15	48.5%	30.1%
M	13	$\sim U(30,50)$	$\sim U(0.2,0.3)$	16.25	14.15	10.35	57.0%	36.7%
M	13	$\sim U(30,50)$	$\sim U(0.4,0.5)$	18.55	16.40	12.05	53.9%	36.1%
M	13	$\sim U(60,80)$	$\sim U(0.0,0.0)$	12.45	11.45	8.65	43.9%	32.4%
M	13	$\sim U(60,80)$	$\sim U(0.2,0.3)$	16.55	14.30	10.05	64.7%	42.3%
M	13	$\sim U(60,80)$	$\sim U(0.4,0.5)$	18.90	16.75	12.15	55.6%	37.9%

$|\mathcal{I}| = 50$ and $|\mathcal{I}| = 80$ for instances in **S** and **M**.
Gap w/ **TSFD-RC-F** calculated as follows: $\Delta_* = ((W_*^{avg} - W_{RC}^{avg})/W_{RC}^{avg}) \cdot 100\%$.

Table 5.16 Comparison of resource requirements for different single-mode truck scheduling models.
Source: Own table.

For most of the cases, however, the relative advantage of the **TSFD-RC-F** in terms of operator requirements varies between 15-35% and can reach values of up to ca. 65%. The varying impact of the exogenous factors on the operator requirements can be further explored with the help of Figure 5.1. The figure illustrates how changes in the number of available dock-doors,

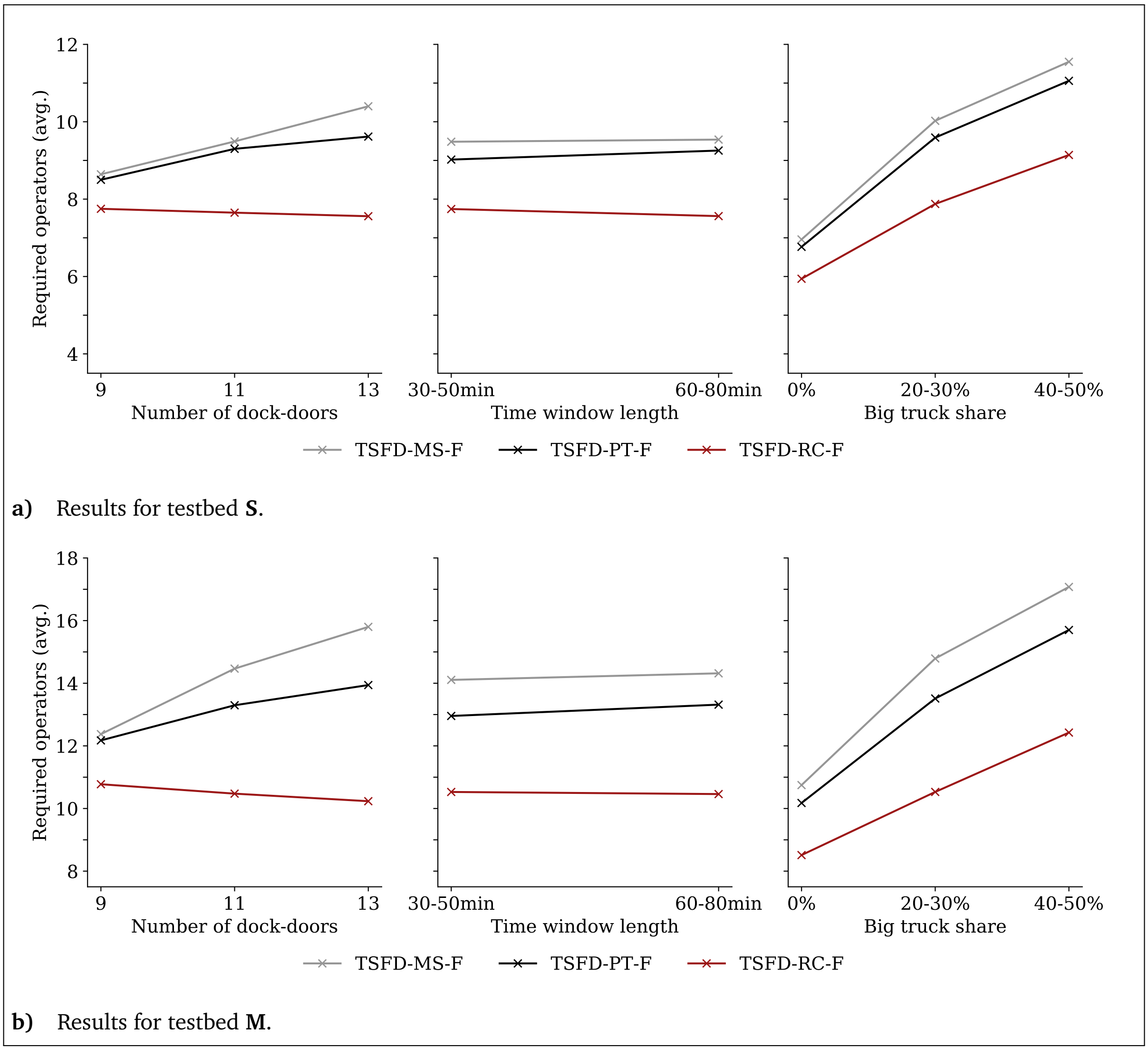


Figure 5.1 Effect of exogenous factors on the number of operators in different single-mode truck scheduling models.
Source: Own figure.

width of time windows, and share of big trucks affect the operator requirements in the different single-mode truck scheduling models. The figure suggests a positive relationship between the number of dock-doors $|\mathcal{D}|$ and the operator requirements W when the makespan or processing time model, both models that pursue to process trucks in parallel, is applied. Increasing the number of available dock-doors facilitates parallel processing and hence causes a surge in demand for operators. The opposite can be observed for the **TSFD-RC-F**. As $|\mathcal{D}|$ increases, the demand for operators slightly decreases. More available dock-doors, i.e., additional flexibility, help the **TSFD-RC-F** to reduce (undesired) parallel processing. These countervailing effects also amplify the differences in operator requirements between the models. The more dock-doors are available for processing inbound trucks, the bigger the discrepancy between the models' operator requirements. In situations with plentiful available dock-doors, the relative operator gap with the **TSFD-RC-F** can easily exceed values of 50% and 35% for the makespan model and processing time model, respectively. Furthermore, the width of time windows appears to neither have a strong effect on the number of operators nor amplifying the differences in operator requirements between the models. When applying the makespan or processing time model, the impact of the big truck share β is somehow comparable with the impact of the available dock-doors $|\mathcal{D}|$. The figure suggests a degressive relationship between β and W, i.e., the effect is decreasing in higher values. When applying the **TSFD-RC-F**, higher values of β also eventuate in higher operator requirements. However, the figure suggests a linear relationship between β and W when the **TSFD-RC-F** is applied. Moreover, higher values of β magnify the differences in operator requirements between the models, since the effect of β appears to be smaller in the **TSFD-RC-F** compared to the **TSFD-MS-F** and **TSFD-PT-F**.

Operator utilization

The previous two sections revealed that the **TSFD-RC-F** usually generates truck schedules that can be executed with a considerably smaller number of operators compared to the **TSFD-MS-F** and **TSFD-PT-F**. In order to get a better idea of how efficiently the different models utilize the available resources, the operator utilization over the planning horizon is studied in this section. Table 5.17 provides the summary statistics on the average operator utilization[1] and shows the time span with a full operator utilization and an operator utilization of less than 50% relative to the length of the planning horizon. For the makespan and processing time models, average utilization rates of ca. 50-55% are realized. That is, operators on-site experience ca. 4h of idle time in schedules generated by the **TSFD-MS-F** and **TSFD-PT-F**. With utilization rates of ca. 65%, the **TSFD-RC-F** allows to considerably reduce the idle times of operators. The statistics regarding the full and low operator utilization indicate that both the makespan and processing time model unevenly distribute the workload over the planning horizon. Note that the facility operates at full capacity for less than 1h (<11%) when applying these models. The **TSFD-RC-F**, on the other hand, appears to achieve a more evenly distributed workload which can also be seen in the representative example displayed in Figure 5.2. The figure depicts the

[1] In this context, the operator utilization is defined as the amount of the available operator time (i.e., $W \cdot |\mathcal{T}|$) that is used for unloading inbound trucks, expressed as a percentage.

	Average operator utilization			Time w/ full operator utilization			Time w/ operator utilization below 50%		
Size	**MS**	**PT**	**RC**	**MS**	**PT**	**RC**	**MS**	**PT**	**RC**
S	54.0%	50.5%	63.3%	10.9%	7.7%	24.2%	41.2%	48.3%	33.2%
M	55.9%	52.2%	68.8%	8.3%	6.0%	29.0%	35.8%	43.5%	26.9%

$|\mathcal{I}| = 50$ and $|\mathcal{I}| = 80$ for instances in **S** and **M**.
MS: **TSFD-MS-F**; **PT**: **TSFD-PT-F**; **RC**: **TSFD-RC-F**.

Table 5.17 Operator utilization statistics for different single-mode truck scheduling models.
Source: Own table.

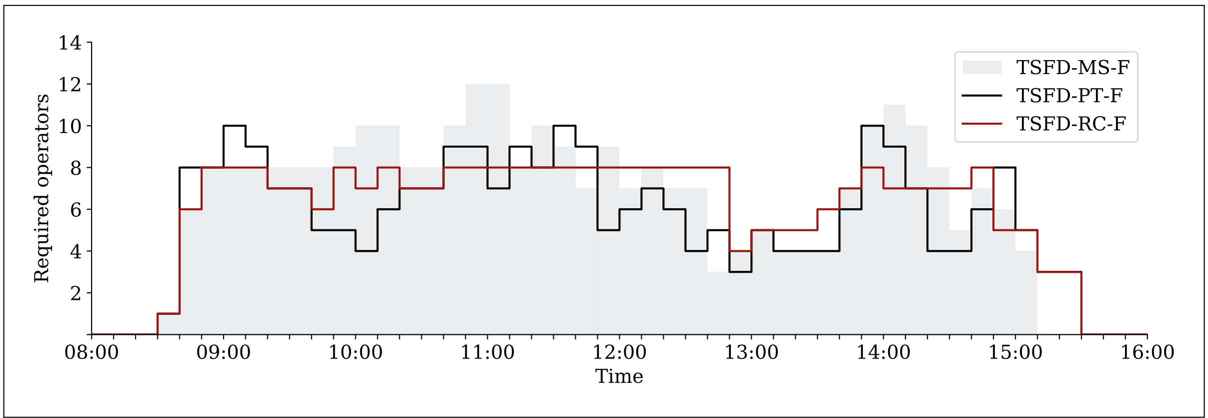
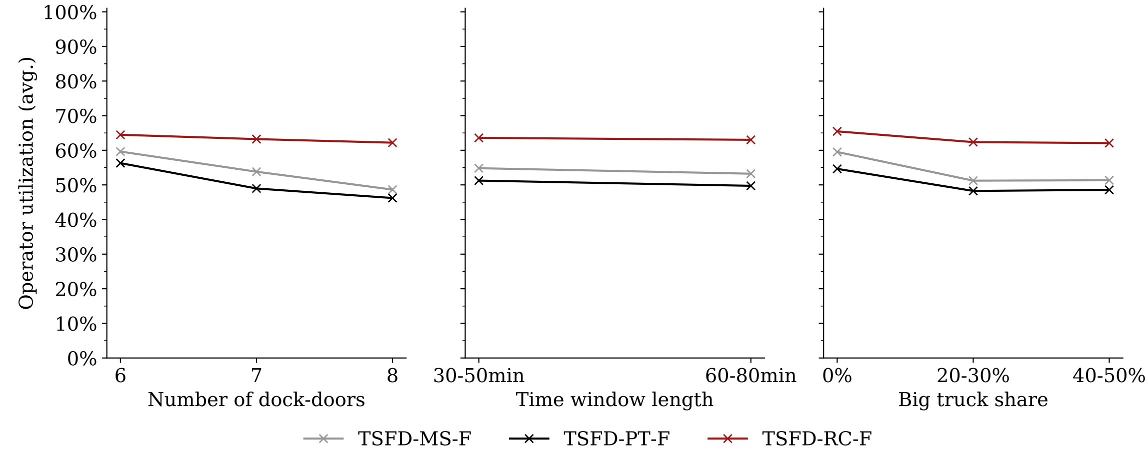


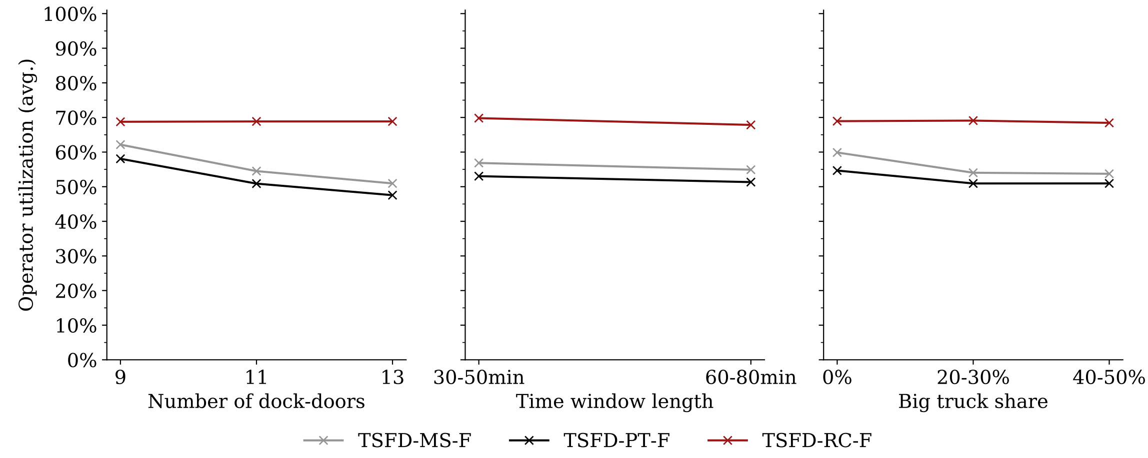
Figure 5.2 Exemplary operator utilization over the planning horizon for different single-mode truck scheduling models.
Source: Own figure.

total number of occupied operators for each time interval of the 8h planning horizon. Recall that the **TSFD-RC-F** is the benchmark as it generates a truck schedule that can be executed with the minimum number of operators. In both examples, it requires a considerably larger number of operators to execute a truck schedule with a minimum makespan or minimum total processing time. When applying the **TSFD-MS-F** to the instance from the figure, up to 12 operators must be on-duty at a time for processing the inbound trucks. That amounts to a surplus of 4 operators as compared with the **TSFD-RC-F's** resource efficient plan. Looking at the figure, it is apparent that the additional operators are mainly utilized during peak hours - in the example between 10:00-11:30 and 14:00-14:30. Outside of the peak hours, and hence most of the time, the additional operators are barely utilized. Similar observations can be made for the **TSFD-PT-F**.

Moreover, Figure 5.3 displays the effect of exogenous factors on the operator utilization. The figure shows that the **TSFD-RC-F** yields a higher operator utilization than the makespan and processing time models, regardless of the factor values. Furthermore, the **TSFD-RC-F** is robust against changes to the number of available dock-doors, the length of time windows, and the



a) Results for testbed **S**.



b) Results for testbed **M**.

Figure 5.3 Effect of exogenous factors on the operator utilization in different single-mode truck scheduling models.
Source: Own figure.

share of big trucks. In the case of the makespan and processing time models, on the other hand, the operator utilization decreases as the number of available dock-doors and the share of big trucks increases.

Effect of service level adjustments in the TSFD-RC-F

The previous sections examined how factors such as the number of available dock-doors, the width of time windows, and the big truck share, affect the operator requirements in the **TSFD-RC-F**. It was shown, for example, that the operator requirements decrease as the number of available dock-doors increases and the share of big trucks decreases. These factors, however, are either hard to change or even cannot be changed at all. The number of dock-doors, for instance, is set during the construction phase of the cross-docking facility and is impossible to adjust at the time of computing the truck schedules. It is also difficult to reduce the share of big trucks as it is determined by the cargo volume. The defined service level α, on the other hand, is a design parameter that can be influenced by the decision-maker. This section, therefore, sets out to investigate how sensitive the operator requirement is to changes in the service level and if the operational efficiency can be improved by marginally lowering the service level. All instances from testbed **M** are solved again assuming different values for the service level α. Throughout the experiment, α is chosen from the set $\{1.0, 0.99, 0.98, 0.97\}$. In this context, the service level only applies to the product units that can theoretically reach the outbound area in time. Figure 5.4 shows the average operator requirements and the average number of additionally delayed product units for each service level. The figure suggests a regressive relationship between α and W, that is, the effect is decreasing in lower values. It can be seen from the figure that by allowing to delay 1% of the overall cargo (i.e., $\alpha = 0.99$), the operator requirements can be reduced by almost 10% (ca. 1 operator) on average. In 44% of the test instances, the operator requirements can be reduced by one, while in ca. 20% of the instances reductions of two or more operators can be realized. Even bigger operator reductions can be achieved more frequently when further reducing the service level requirements. However, the benefit comes at a price, as a lower service level implies a higher number of delayed product units. It can be seen from the figure that a service level reduction of 1% corresponds on average to ca. 13 additionally delayed product units in instances from testbed **M**. Given the fact that it requires one dedicated operator (paid for the whole 8h shift) to avoid delaying 13 pallets, adjusting the service level could be a reasonable means to further improve the operational efficiency in a cross-docking facility.

Effect of time window adjustments in the TSFD-RC-F

This section analyzes the effect of time window adjustments on the operator requirements in the **TSFD-RC-F**. It is analyzed whether the number of required operators can be further reduced by slightly extending the truck time windows. It is suspected that prolonging truck time windows adds additional flexibility and may help to further smoothing out the workload

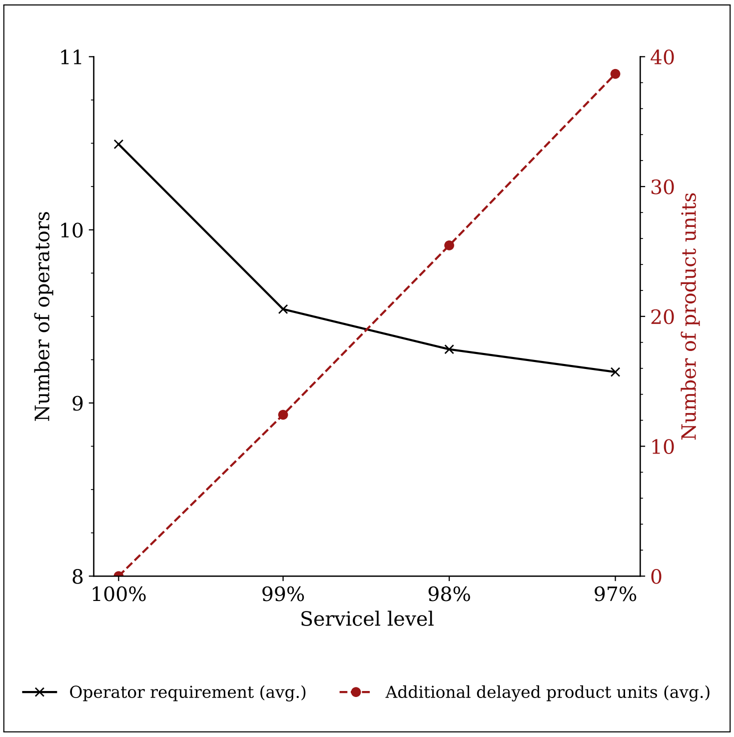


Figure 5.4 Impact of service level adjustments in the **TSFD-RC-F** (testbed **M**).
Source: Own figure.

over the planning horizon. Note that it might be difficult to adjust the time windows as they usually must be reconciled with the transportation provider due to their possible impact on the truck routing. However, in some cases, for instance when both the transportation service and the cross-docking service are offered by the same provider, time window adjustments could be an option.

For this experiment, time window extensions of 10 and 20 minutes are considered. The original time windows are extended in such a way that all trucks are released 10 and 20 minutes earlier, respectively. The resulting operator requirements are then compared against the base case with the original time windows. In addition, the influence on the truck throughput time is analyzed. Figure 5.5 presents the numerical results for testbed **M**. The figure shows that with 10 minutes longer time windows, ca. 1.3 operators, equating to a work time of ca. 620 minutes, can be saved on average. In ca. 1/3 of the instances, savings of two or more operators can be realized. Once again, these operator reductions come at a price, as extended time windows result in longer truck throughput times. Specifically, the figure shows that the average truck throughput time increases by ca. 6 minutes as the time windows are extended by 10 minutes. That is, the cumulated truck throughput time increases by roughly 480 minutes. The aggregated

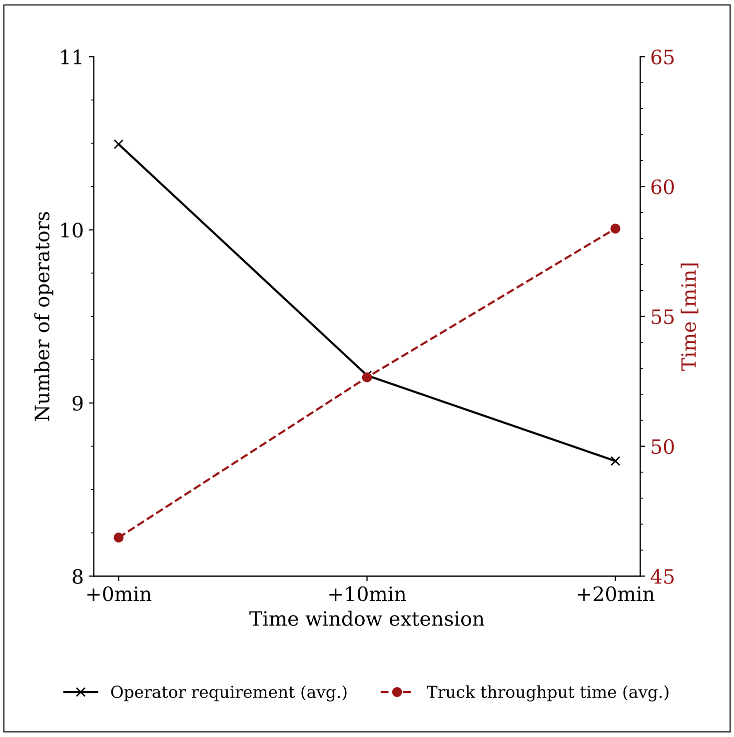


Figure 5.5 Impact of time window adjustments in the **TSFD-RC-F** (testbed **M**).
Source: Own figure.

results suggest that time window extensions are unfavorable. However, extending time windows for smaller truck subsets instead, could be a reasonable option to further reduce the operator requirements. This could be analyzed in a future study.

5.4.2 Multi-mode resource and truck scheduling problem

The **TSFD-RC-F**, which was analyzed in the previous section, assumes that the resource requirement for each truck is known and given in advance. Throughout the experiment, it was assumed that one and two operators are deployed for processing small and big trucks, respectively. This rule of thumb, however, might not result in the most efficient truck schedule. The **TSFD-RC-V**, on the other hand, has the additional flexibility of adapting the workforce for processing inbound trucks. This section sets out to explore the potential benefits of considering multiple operator modes for each inbound truck by benchmarking the **TSFD-RC-V** against the **TSFD-RC-F**. In the following, both models are compared with respect to various key performance indicators. Moreover, the role of influencing factors (e.g., width of truck time windows, service level, etc.) in the **TSFD-RC-V** will be investigated.

Benefits of the multiple operator modes

In the following, the **TSFD-RC-V** is benchmarked against the **TSFD-RC-F** in terms of the makespan, truck waiting and processing time, and operator requirements[1]. Table 5.18 reports the descriptive statistics for the numerical experiment.

Size		Makespan		Total truck wating time		Total truck processing time		Required operators	
		RC-F	**RC-V**	**RC-F**	**RC-V**	**RC-F**	**RC-V**	**RC-F**	**RC-V**
S	N	360	360	360	360	360	360	360	360
	Mean	429.5	433.5	635.8	704.1	1,856.4	1,807.0	7.7	6.5
	Median	430.0	430.0	590.0	670.0	1,850.0	1,800.0	8.0	6.0
	SD	22.0	23.1	258.4	249.5	96.7	137.6	1.7	1.1
M	N	360	360	360	360	360	360	360	360
	Mean	434.7	438.7	927.1	1,009.3	2,791.7	2,706.5	10.5	9.0
	Median	440.0	440.0	880.0	950.0	2,780.0	2,720.0	10.0	9.0
	SD	20.5	19.0	328.6	333.1	130.7	169.0	2.0	1.3

$|\mathcal{I}| = 50$ and $|\mathcal{I}| = 80$ for instances in **S** and **M**.
N: Number of observations; SD: Standard deviation.
Mean, median, and standard deviation in minutes.
RC-F: **TSFD-RC-F**; **RC-V**: **TSFD-RC-V**.

Table 5.18 Descriptive statistics for single-mode and multi-mode resource and truck scheduling problem and different key performance indicators.
Source: Own table.

Having the additional flexibility of adapting the workforce for processing inbound trucks (**TSFD-RC-V**) allows to reduce the average number of required operators by ca. 15%. This significant reduction comes at a price, as it leads to truck schedules with a slightly longer makespan. With an average increase of less than five minutes compared with the **TSFD-RC-F** and in light of **TSFD-RC-V** meeting the service level, however, the difference has no negative effect on the internal operations. Moreover, it can be seen that a truck's waiting time increases by ca. 1.5 minutes when using the **TSFD-RC-V**. The average processing time per truck, on the other hand, can be reduced by almost the same extent when applying the **TSFD-RC-V** instead of the **TSFD-RC-F**. That is, the average truck throughput times of the single-mode and multi-mode models are almost identical. Hence, it can be concluded that the **TSFD-RC-V** is superior to the **TSFD-RC-F** and allows to further improve the operational efficiency in a cross-docking facility.

[1] The **TSFD-RC-V** is not compared against the multi-mode versions of the makespan or cumulated truck processing time model. These models choose the fastest operator mode for every truck according to preliminary test. As this, in turn, results in extremely high operator requirements, a comparison would make little or no sense.

Impact of exogenous factors on operator requirements

The previous section revealed that truck schedules computed with the **TSFD-RC-V** can be executed with ca. 15% fewer operators than schedules computed with the **TSFD-RC-F**. In order to better understand under which circumstances the **TSFD-RC-V's** flexibility of adapting the workforce may or may not pay off, the impact of various exogenous factors (namely the number of dock-doors, width of time windows, and share of big trucks) on the operator requirements is investigated in more detail below. Table 5.19 reports the operator requirements in the **TSFD-RC-F** and **TSFD-RC-V** for different parameter combinations.

For all 36 parameter combinations, applying the single-mode model results in a higher average number of required operators than the multi-mode model. The relative advantage of the **TSFD-RC-V** in terms of operator requirements varies between 1-20% among the reported parameter combinations. It can be seen that the **TSFD-RC-V's** relative advantage over the **TSFD-RC-F** is strongly impacted by the share of big trucks β. While its relative advantage varies between 1-9% for instances without big trucks (i.e., $\beta \sim U(0.0, 0.0)$), it reaches values between 13-20% for instances which include both small and big inbound trucks (i.e., $\beta \sim U(0.2, 0.3)$ and $\beta \sim U(0.4, 0.5)$). Variations in the number of dock-doors and the time window length, on the other hand, seem to have a weaker impact on the relative gap between the operator requirements in the **TSFD-RC-V** and the **TSFD-RC-F**. The influence of the three exogenous factors can be further explored with the help of Figure 5.6. The figure illustrates how the number of dock-doors $|\mathcal{D}|$, the width of time windows $(d_i - r_i)$, and the share of big trucks β affect the operator requirements W in the single-mode and multi-mode resource and truck scheduling problem.

As the number of inbound dock-doors increases, the demand for operators slightly decreases when applying the **TSFD-RC-F** or **TSFD-RC-V**. That is, more available dock-doors offer additional flexibility which helps both models to reduce (undesired) parallel processing. With the two models' curves being almost parallel to each other, the number of inbound dock-doors does not seem to influence the absolute difference in operator requirements between the two models. Furthermore, the width of time windows appears to neither have a strong effect on the number of operators nor amplifying the differences in operator requirements between the two models. The figure also illustrates the significant impact of the share of big trucks β on the number of required operators W. The figure suggests an almost linear relationship between β and W when the **TSFD-RC-F** or **TSFD-RC-V** is applied. It can be seen that the effect of β is smaller in the **TSFD-RC-V** compared to the **TSFD-RC-F**. Therefore, higher values of β magnify the differences in operator requirements between the two models.

Instances				Required operators (avg.)		Gap w/ **TSFD-RC-F**
Size	$\|\mathcal{D}\|$	$d_i - r_i$	β	W^{avg}_{RC-F}	W^{avg}_{RC-V}	Δ_{RC-V}
S	6	$\sim U(30,50)$	$\sim U(0.0,0.0)$	5.75	5.65	-1.7%
S	6	$\sim U(30,50)$	$\sim U(0.2,0.3)$	8.00	6.50	-18.8%
S	6	$\sim U(30,50)$	$\sim U(0.4,0.5)$	9.60	7.75	-19.3%
S	6	$\sim U(60,80)$	$\sim U(0.0,0.0)$	5.85	5.65	-3.4%
S	6	$\sim U(60,80)$	$\sim U(0.2,0.3)$	7.80	6.50	-16.7%
S	6	$\sim U(60,80)$	$\sim U(0.4,0.5)$	9.50	7.75	-18.4%
S	7	$\sim U(30,50)$	$\sim U(0.0,0.0)$	6.05	5.80	-4.1%
S	7	$\sim U(30,50)$	$\sim U(0.2,0.3)$	8.00	6.55	-18.1%
S	7	$\sim U(30,50)$	$\sim U(0.4,0.5)$	9.50	7.75	-18.4%
S	7	$\sim U(60,80)$	$\sim U(0.0,0.0)$	6.05	5.55	-8.3%
S	7	$\sim U(60,80)$	$\sim U(0.2,0.3)$	7.40	6.25	-15.5%
S	7	$\sim U(60,80)$	$\sim U(0.4,0.5)$	8.90	7.40	-16.9%
S	8	$\sim U(30,50)$	$\sim U(0.0,0.0)$	6.20	5.85	-5.6%
S	8	$\sim U(30,50)$	$\sim U(0.2,0.3)$	8.15	6.55	-19.6%
S	8	$\sim U(30,50)$	$\sim U(0.4,0.5)$	8.45	6.90	-18.3%
S	8	$\sim U(60,80)$	$\sim U(0.0,0.0)$	5.75	5.40	-6.1%
S	8	$\sim U(60,80)$	$\sim U(0.2,0.3)$	7.90	6.45	-18.4%
S	8	$\sim U(60,80)$	$\sim U(0.4,0.5)$	8.90	7.15	-19.7%
M	9	$\sim U(30,50)$	$\sim U(0.0,0.0)$	8.55	8.10	-5.3%
M	9	$\sim U(30,50)$	$\sim U(0.2,0.3)$	11.00	9.15	-16.8%
M	9	$\sim U(30,50)$	$\sim U(0.4,0.5)$	12.75	10.30	-19.2%
M	9	$\sim U(60,80)$	$\sim U(0.0,0.0)$	8.55	8.20	-4.1%
M	9	$\sim U(60,80)$	$\sim U(0.2,0.3)$	10.95	9.20	-16.0%
M	9	$\sim U(60,80)$	$\sim U(0.4,0.5)$	12.85	10.65	-17.1%
M	11	$\sim U(30,50)$	$\sim U(0.0,0.0)$	8.74	8.05	-7.9%
M	11	$\sim U(30,50)$	$\sim U(0.2,0.3)$	10.68	9.16	-14.2%
M	11	$\sim U(30,50)$	$\sim U(0.4,0.5)$	12.40	10.00	-19.4%
M	11	$\sim U(60,80)$	$\sim U(0.0,0.0)$	8.45	7.95	-5.9%
M	11	$\sim U(60,80)$	$\sim U(0.2,0.3)$	10.15	8.75	-13.8%
M	11	$\sim U(60,80)$	$\sim U(0.4,0.5)$	12.35	9.95	-19.4%
M	13	$\sim U(30,50)$	$\sim U(0.0,0.0)$	8.15	7.60	-6.7%
M	13	$\sim U(30,50)$	$\sim U(0.2,0.3)$	10.35	8.35	-19.3%
M	13	$\sim U(30,50)$	$\sim U(0.4,0.5)$	12.05	9.75	-19.1%
M	13	$\sim U(60,80)$	$\sim U(0.0,0.0)$	8.65	7.95	-8.1%
M	13	$\sim U(60,80)$	$\sim U(0.2,0.3)$	10.05	8.50	-15.4%
M	13	$\sim U(60,80)$	$\sim U(0.4,0.5)$	12.15	9.90	-18.5%

$|\mathcal{I}| = 50$ and $|\mathcal{I}| = 80$ for instances in **S** and **M**.
Gap w/ **TSFD-RC-F** calculated as follows: $\Delta_* = \left(\left(W^{avg}_* - W^{avg}_{RC-F}\right)/W^{avg}_{RC-F}\right) \cdot 100\%$.

Table 5.19 Comparison of resource requirements for the **TSFD-RC-V** and **TSFD-RC-F**.
Source: Own table.

Operator utilization

This section benchmarks the **TSFD-RC-V** against the **TSFD-RC-F** with respect to the operator utilization. The comparison attempts to shed a light on how efficiently resources are utilized when applying both models. Table 5.20 provides the summary statistics for both models.

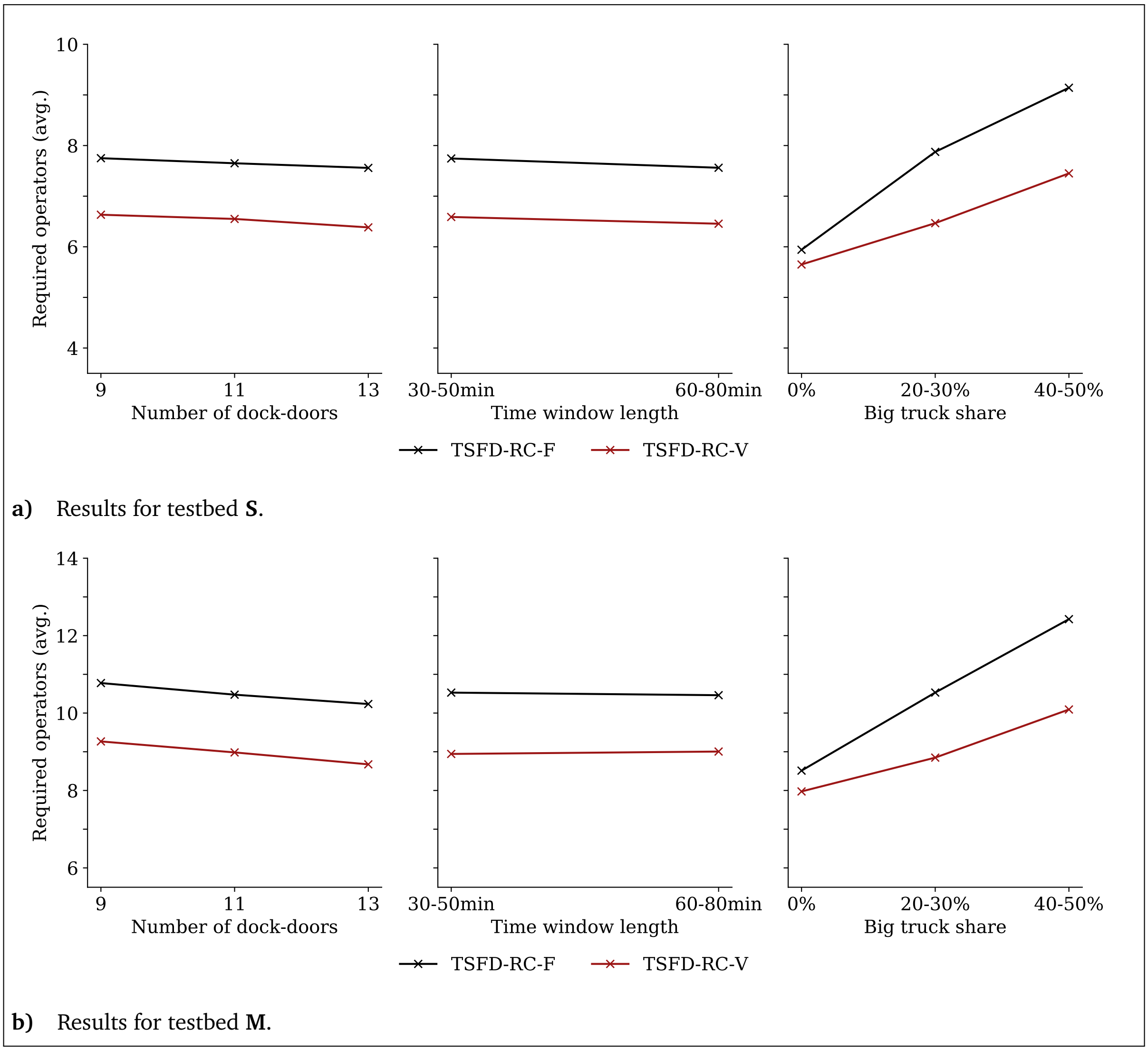


Figure 5.6 Effect of exogenous factors on the number of operators in the **TSFD-RC-F** and **TSFD-RC-V**.
Source: Own figure.

It can be seen from the table that a significantly higher operator utilization[1)] can be achieved by using the multi-mode resource and truck scheduling model. Specifically, average increases of ca. 9% and ca. 7% for instances from testbed **S** and testbed **M** can be realized, respectively. Moreover, the statistics on the time span with full and low operator utilization indicate that the **TSFD-RC-V** yields a more evenly distributed workload over the planning horizon. That is, the

[1)] Once again, the operator utilization is defined as the amount of the available operator time (i.e., $W \cdot |\mathcal{T}|$) that is used for unloading inbound trucks, expressed as a percentage.

	Average operator utilization		Time w/ full operator utilization		Time w/ operator utilization below 50%	
Size	**RC-F**	**RC-V**	**RC-F**	**RC-V**	**RC-F**	**RC-V**
S	63.3%	72.2%	24.2%	39.1%	33.2%	23.3%
M	68.8%	75.7%	29.0%	40.4%	26.9%	20.6%

$|\mathcal{I}| = 50$ and $|\mathcal{I}| = 80$ for instances in **S** and **M**.
RC-F: **TSFD-RC-F**; **RC-V**: **TSFD-RC-V**.

Table 5.20 Operator utilization statistics for **TSFD-RC-F** and **TSFD-RC-V**.
Source: Own table.

TSFD-RC-V is doing better at avoiding peaks of operator demand during the planning horizon which, in turn, allows to reduce the number of required operators. This aspect is also illustrated in Figure 5.7, which shows for an exemplary test instance how operators are utilized over the planning horizon in both models.

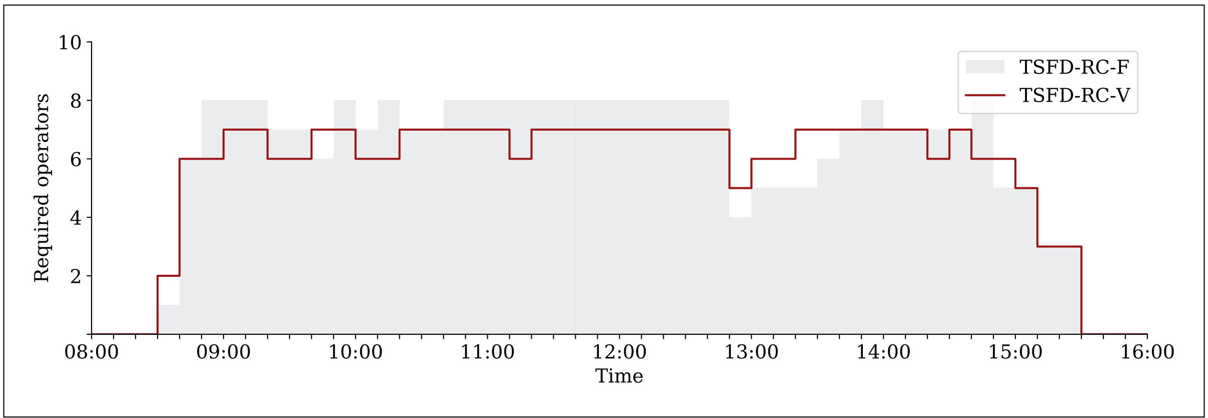
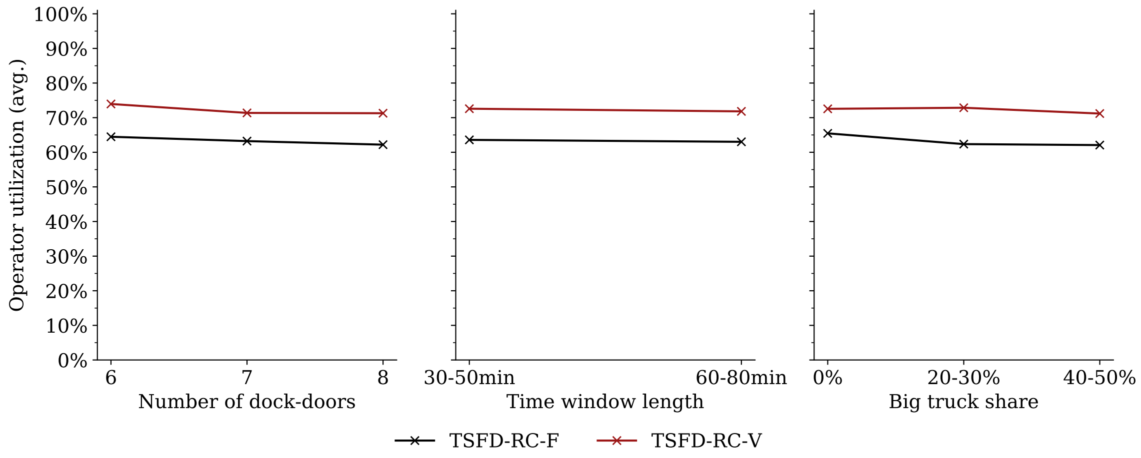


Figure 5.7 Exemplary operator utilization over the planning horizon for **TSFD-RC-F** and **TSFD-RC-V**.
Source: Own figure.

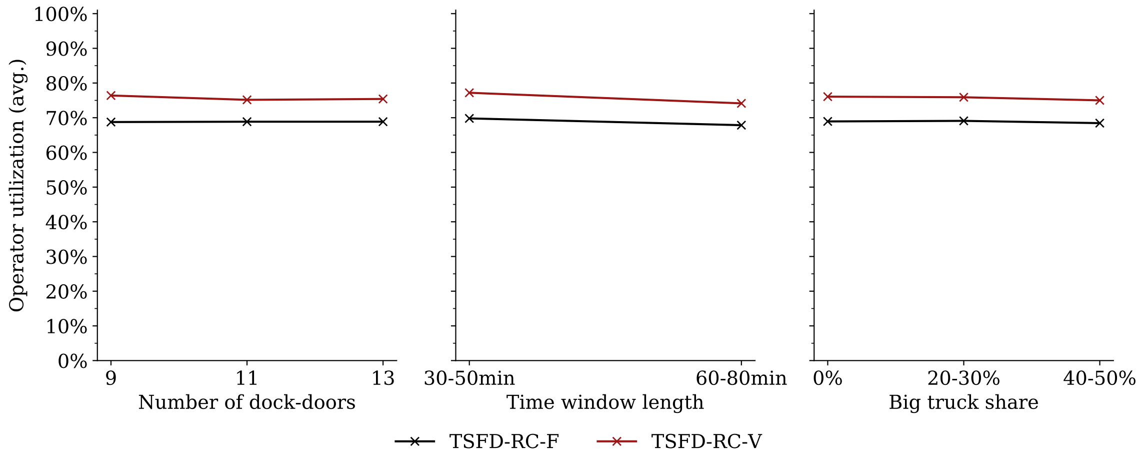
During the busiest period from 09:00 to 15:00, the operator demand fluctuates between four and eight when applying the **TSFD-RC-F**. When applying the **TSFD-RC-V**, on the other hand, a more level schedule with smaller operator demand fluctuations can be realized. More than that, by smoothing out the workload over the planning horizon, the **TSFD-RC-V** computes a schedule which allows a cut down on the required operators of 12.5% compared to the **TSFD-RC-F**.

Lastly, Figure 5.8 allows to explore how the number of dock-doors $|\mathcal{D}|$, the width of truck time windows $(d_i - r_i)$, and the share of big trucks β affect the average operator utilization in the **TSFD-RC-V**. The **TSFD-RC-F's** utilization rates are included as a reference.

The **TSFD-RC-V** always seems to achieve a considerably higher average operator utilization



a) Results for testbed **S**.



b) Results for testbed **M**.

Figure 5.8 Effect of exogenous factors on the operator utilization in **TSFD-RC-F** and **TSFD-RC-V**.
Source: Own figure.

than the **TSFD-RC-F**, no matter which factor level is chosen. Moreover, the nearly horizontal course of the utilization rate curve shows the **TSFD-RC-V's** robustness against changes of the factor values.

It can be summarized that the **TSFD-RC-V's** inherent additional flexibility of adapting the workforce for processing inbound trucks makes it possible to cut down the required number of operators by smoothing out the workload over the planning horizon.

Effect of service level adjustments in the TSFD-RC-V

Previously, it was shown in the context of the **TSFD-RC-F** that adjustments to the service level requirement can have a huge impact on the operator demand. This section sets out to explore whether service level adjustments have a similar effect when considering multiple operator modes like in the **TSFD-RC-V**. For this purpose, all problem instances in testbed **M** are solved multiple times using different service levels $\alpha \in \{1.0, 0.99, 0.98, 0.97\}$[1]. The results are illustrated in Figure 5.9. The figure shows the average operator requirements and the additional number of delayed freight units for different service levels.

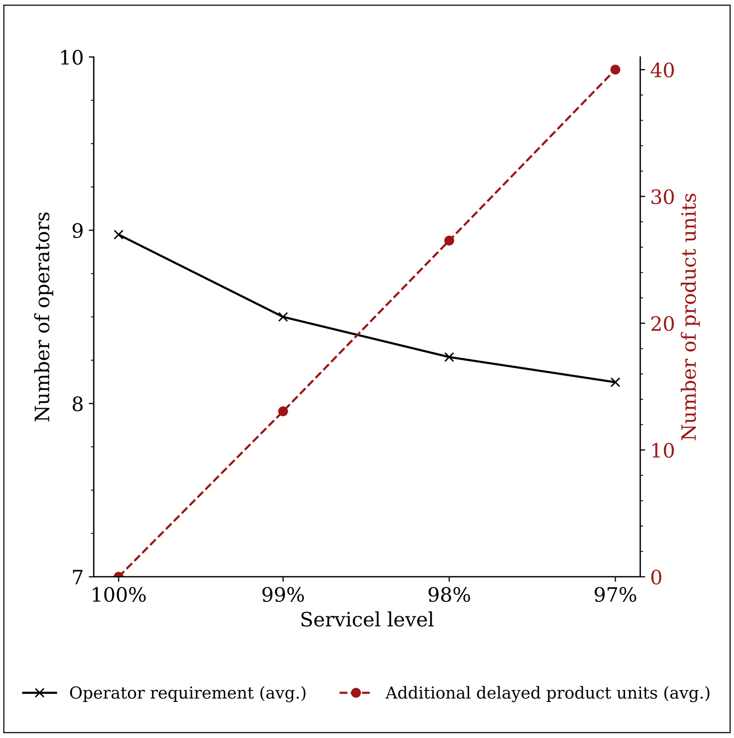


Figure 5.9 Impact of service level adjustments in the **TSFD-RC-V** (testbed **M**).
Source: Own figure.

The figure suggest a regressive relationship between the service level and the operator requirements. That is, the effect of the service level on the operator requirements is decreasing in lower values – an observation that was already made for the **TSFD-RC-F** in one of the previous sections. From the figure, it can be seen that reducing the service level requirement from 100%

[1] In this context, the service level only applies to the product units that can theoretically reach the outbound area in time.

to 99% (an 1% service level reduction is equivalent to ca. 13 additionally delayed freight units) makes it possible to cut down the required operators by ca. 5.5% (ca. 0.5 operators). Compared with the **TSFD-RC-F**, in which operator demand reductions of ca. 10% (ca. 1 operator) could be obtained when reducing the required service level from 100% to 99%, service level adjustments seem to have a smaller impact in the **TSFD-RC-V**. Even though the operator demand reacts less sensitive to changes in the service level, adjusting the service level could still be a reasonable means to further improve the operational efficiency when considering multiple operator modes.

Effect of time window adjustments in the TSFD-RC-V

It was revealed in the context of the single-mode resource and truck scheduling problem that truck time window extensions may help to reduce the operator demand. This section sets out to investigate if this also holds for the **TSFD-RC-V**. Subsequently, it is analyzed if prolonging the truck time windows can help smoothing out the workload over the planning horizon which, in turn, may help to further reduce the peak workforce. Once again, time window extensions of 10 and 20 minutes are considered. Specifically, the original time windows are prolonged in such a way that all inbound trucks are released 10 and 20 minutes earlier, respectively. The instances with the original truck time windows are used as a reference in order to assess whether time window extensions pay off or not. The three scenarios are compared with respect to the operator demand and the average truck throughput time. Figure 5.10 depicts the numerical results for testbed **M**.

The figure suggests a regressive relationship between the width of time windows and the operator requirements – an observation that was already made for the **TSFD-RC-F** in one of the previous sections. It can be seen that by prolonging the truck time windows by ten minutes, the average operator demand can be reduced by ca. 9% (ca. 0.8 operators). This reduction is equivalent to a ca. 380 minutes reduction in work time. Compared with the **TSFD-RC-F**, in which reductions of more than 12% could be obtained, the effect of time window extensions seem to be smaller. Moreover, the reduced operator demand comes at a price, as the average truck throughput time increases by ca. 13% (ca. 6 minutes). Since the benefits in the form of a reduced operator demand are offset by the longer truck throughput times, time window extensions seem to be an unfavorable choice in the **TSFD-RC-V**.

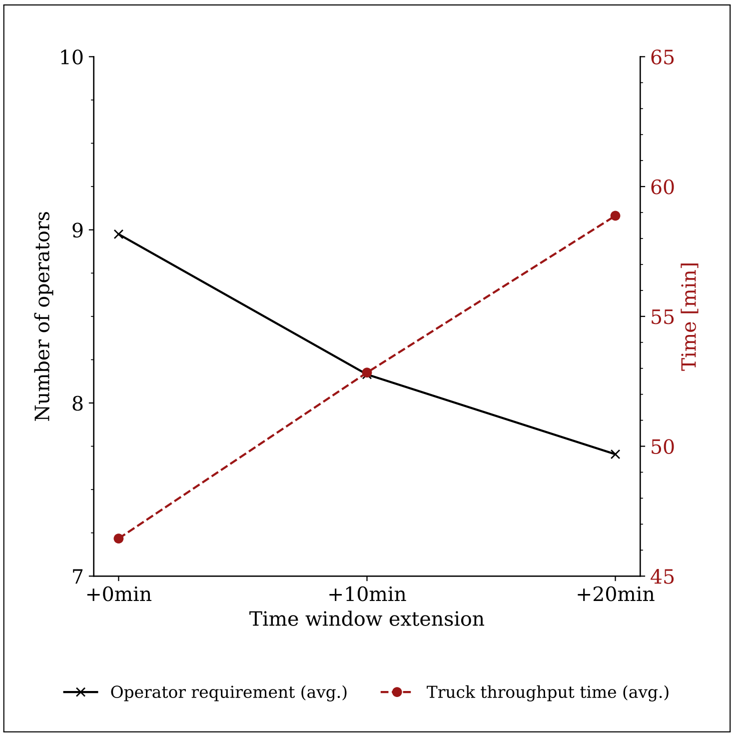


Figure 5.10 Impact of time window adjustments in the **TSFD-RC-V** (testbed **M**).
Source: Own figure.

5.4.3 Shift and truck scheduling problem

This sections sets out to evaluate the solution quality of the **ISTSFD**.

In the following, the **ISTSFD-F** will be benchmarked against sequential scheduling approaches which first create the truck schedules and then create the employee timetables. According to Ladier and Alpan, sequential approaches are commonly used in practice[1]. Specifically, the **ISTSFD-F** is benchmarked against the following two sequential approaches:

- **TSFD-MS-F/SP:** First, a truck schedule is created by solving the truck scheduling model for minimizing the makespan (**TSFD-MS-F**). After that, employee timetables are created by solving the shift planning model **SP**.
- **TSFD-PT-F/SP:** First, a truck schedule is created by solving the truck scheduling model for minimizing the total processing time (**TSFD-PT-F**). After that, employee timetables are created by solving the shift planning model **SP**.

1) Ladier and Alpan (2015, p. 679)

The shift planning model **SP** uses the obtained truck schedule as an input. It aims to minimize the total operator costs and must assure that the operator demand never exceeds the operator supply. The **SP** can be found in Appendix.

Then, the benefits of having the additional flexibility of adapting the workforce for processing inbound trucks (i.e., multiple operator modes) is investigated for the shift and truck scheduling problem. Moreover, the impact of work breaks is analyzed.

If not mentioned otherwise, the shift patterns $s \in \{1, 2, 3, 4\}$, which were described in Table 5.3 are applied.

Benefits of utilizing operator requirements as the key performance indicator

It was shown in the context of the resource and truck scheduling problem that utilizing the operator requirements as the main key performance indicator can significantly improve the cross-dock efficiency. Recall, however, that the **TSFD-RC-F** only considers one standard shift type for operators and does not consider work breaks. Hence, it cannot be expected that the finding also holds when multiple shift patterns and work breaks are considered. In the following, it will be analyzed whether using the operator demand as the main performance indicator also pays off in planning situations with multiple shift patterns and work breaks. The **ISTSFD-F** will be compared with the **TSFD-MS-F/SP** and **TSFD-PT-F/SP** in terms of the makespan, truck waiting and processing times, and operator requirements. Table 5.21 reports the descriptive statistics for the numerical experiment. First, the table shows that the **TSFD-PT-F/SP** and **ISTSFD-F** compute truck schedules with a ca. 5-7% longer makespan than the truck schedules computed with the **TSFD-MS-F/SP**. This increase of less than 30 minutes, however, has no negative effect on the internal operations. With respect to the operator requirements, it can be seen from the table that both sequential approaches generate plans which require can 18-35% more operators than plans generated with the **ISTSFD-F**. However, the significantly lower operator demand in the integrated shift and truck scheduling model comes at a price. Compared with the **TSFD-PT-F/SP**, which realizes the shortest average truck throughput times, the **ISTSFD-F's** average truck throughput time slightly increases by ca. 4% (equivalent to ca. two minutes). Note, however, that it requires ca. 20% more operators to achieve this marginal truck throughput time reduction. This strong surge in operator demand and hence operational costs is unfavorable. It can be concluded that utilizing the operator demand as the main performance indicator is also beneficial in planning situations with multiple shift patterns and work breaks.

		Makespan			Total truck waiting time			Total truck processing time			Required operators		
Size		**MS/SP**	**PT/SP**	**IST**	**MS/SP**	**PT/SP**	**IST**	**MS/SP**	**PT/SP**	**IST**	**MS/SP**	**PT/SP**	**IST**
S	N	360	360	360	360	360	360	360	360	360	360	360	360
	Mean	404.3	423.1	426.4	570.1	676.8	645.1	1,953.2	1,759.4	1,900.4	11.0	10.3	8.6
	Median	400.0	430.0	430.0	530.0	630.0	620.0	1,940.0	1,750.0	1,890.0	11.0	10.0	8.0
	SD	20.5	19.6	21.9	225.2	241.2	199.3	107.0	68.3	92.6	2.2	2.1	1.8
M	N	360	360	360	360	360	360	360	360	360	360	360	360
	Mean	402.2	426.2	430.1	861.6	1,023.9	931.1	3,031.5	2,627.9	2,855.2	16.6	14.8	11.9
	Median	400.0	430.0	430.0	830.0	970.0	890.0	3,010.0	2,610.0	2,850.0	16.0	15.0	12.0
	SD	17.4	15.8	19.9	278.2	328.9	249.6	145.6	82.4	131.7	3.2	2.7	2.3

$|\mathcal{I}| = 50$ and $|\mathcal{I}| = 80$ for instances in **S** and **M**.

N: Number of observations; SD: Standard deviation.

Mean, median, and standard deviation in minutes.

MS/SP: **TSFD-MS-F/SP**; **PT/SP**: **TSFD-PT-F/SP**; **IST**: **ISTSFD-F**.

Shift patterns $s \in \{1, 2, 3, 4\}$ used in all models.

Table 5.21 Descriptive statistics for different single-mode shift and truck scheduling models and key performance indicators.
Source: Own table.

Benefits of the multiple operator modes

Moving on now to investigate whether having multiple operator modes in the shift and truck scheduling problem may or may not be beneficial, the **ISTSFD-V** is benchmarked against the **ISTSFD-F**[1]. Again, both models are compared with respect to the makespan, truck processing and waiting times, and operator requirements. Table 5.22 reports the descriptive statistics for the numerical experiment.

		Makespan		Total truck waiting time		Total truck processing time		Required operators	
Size		**F**	**V**	**F**	**V**	**F**	**V**	**F**	**V**
S	N	360	360	360	360	360	360	360	360
	Mean	426.4	430.9	645.1	681.6	1900.4	1836.1	8.6	7.4
	Median	430.0	430.0	620.0	659.0	1890.0	1845.0	8.0	7.0
	SD	21.9	20.7	199.3	209.1	92.6	109.8	1.8	1.2
M	N	360	360	360	360	360	360	360	360
	Mean	430.1	436.2	931.1	984.7	2855.2	2707.7	11.9	10.2
	Median	430.0	438.0	890.0	954.0	2850.0	2717.0	12.0	10.0
	SD	19.9	18.5	249.6	264.0	131.7	157.1	2.3	1.5

$|\mathcal{I}| = 50$ and $|\mathcal{I}| = 80$ for instances in **S** and **M**.
N: Number of observations; SD: Standard deviation.
Mean, median, and standard deviation in minutes.
F: **ISTSFD-F**; **V**: **ISTSFD-V**.
Shift patterns $s \in \{1, 2, 3, 4\}$ used in all models.

Table 5.22 Descriptive statistics for **ISTSFD-V** and **ISTSFD-F** and different key performance indicators.
Source: Own table.

The table shows that the **ISTSFD-V** computes truck schedules that have a ca. 5 minute longer makespan than the **ISTSFD-F** does. However, this marginal increase does not have a negative impact on the internal operations since the service level requirements are still met. The average truck throughput time, i.e., the sum of truck waiting time and truck processing time, can be slightly reduced when having the additional flexibility of multiple operator modes. Specifically, the slightly longer truck waiting times are offset by shorter average truck processing times. While the differences in terms of the makespan and truck throughput times are rather small, significant differences with respect to the operator requirements can be observed. The table

[1] The **ISTSFD-V** is not compared against the multi-mode versions of the presented sequential models. The models, which seek to minimize the makespan or cumulated truck processing time at the first stage, choose the fastest operator mode for every truck according to preliminary test. As this, in turn, results in extremely high operator requirements, a comparison would make little or no sense.

shows that the operator demand can be reduced by ca. 14% by considering multiple operator modes instead of applying the proposed "rule of thumb", which deploys one operator and two operators for processing small and big trucks, respectively. That is, ca. 1.2 and 1.7 fewer operators are needed in instances from testbed **S** and testbed **M**, respectively. As the operator demand considerably decreases and the average truck throughput time slightly decreases, it can be summarized that having multiple operator modes clearly pays off in the **ISTSFD**.

Impact of work breaks

This section is devoted to exploring the effect of work breaks on the operator requirements. Specifically, it attempts to analyze whether the operator requirements can be reduced by considering more potential start times for work breaks. From a theoretical perspective, considering more work break patterns offers additional flexibility and hence allows to better match the operator supply with the operator demand. However, it also increases the complexity of both the planning task and the facility operations. It hence becomes important to identify a reasonable number of work break patterns which provides sufficient flexibility to match operator supply and demand.

A total of ten different shift patterns, which are shown in Table 5.23, are considered for the analysis. Each work shift starts at 08:00, ends at 16:00, includes a 30 minute work break

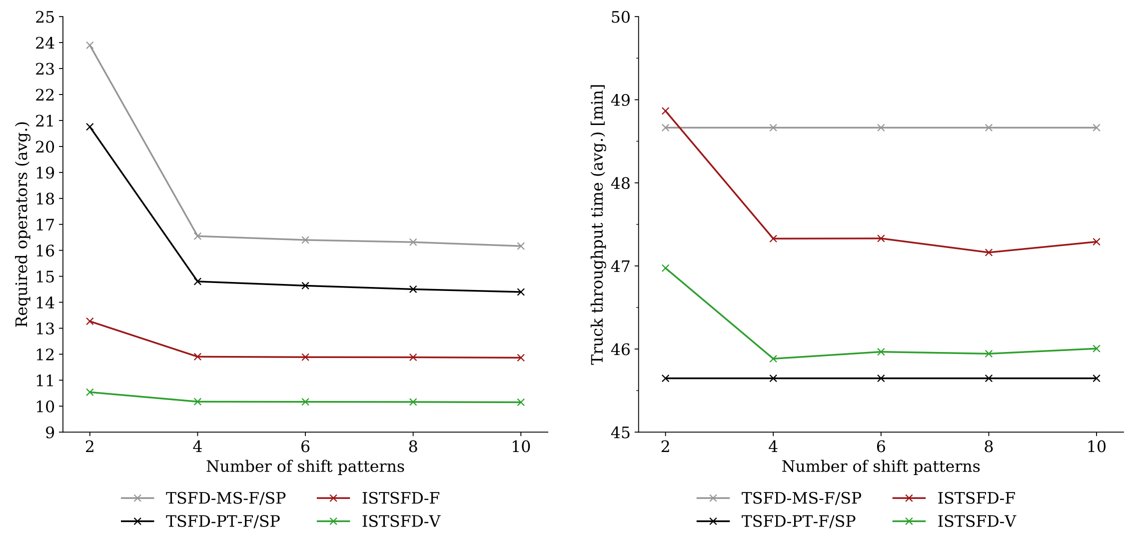
Operator group	Shift pattern	Start	End	Work break	C_{gs}
1	1	08:00	16:00	11:30 - 12:00	1
1	2	08:00	16:00	12:00 - 12:30	1
1	3	08:00	16:00	12:30 - 13:00	1
1	4	08:00	16:00	13:00 - 13:30	1
1	5	08:00	16:00	11:40 - 12:10	1
1	6	08:00	16:00	12:10 - 12:40	1
1	7	08:00	16:00	11:50 - 12:20	1
1	8	08:00	16:00	12:20 - 12:50	1
1	9	08:00	16:00	12:40 - 13:10	1
1	10	08:00	16:00	12:50 - 13:20	1

Table 5.23 Shift pattern information for analyzing the impact of work breaks on the operator requirements.
Source: Own table.

between 11:30 and 13:30 (e.g., a lunch break), and has identical shift costs. However, each shift pattern is characterized by a unique start time for the work break. Based on that, the test instances from testbed **M** are solved for the following five scenarios:

- **Two shift patterns:** $s \in \{1, 2\}$
- **Four shift patterns:** $s \in \{1, 2, 3, 4\}$
- **Six shift patterns:** $s \in \{1, 2, 3, 4, 5, 6\}$
- **Eight shift patterns:** $s \in \{1, 2, 3, 4, 5, 6, 7, 8\}$
- **Ten shift patterns:** $s \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

Figure 5.11 depicts the average operator requirements and the average truck throughput time for the five scenarios and different sequential and integrated models for shift and truck scheduling.



a) Operator requirements. **b)** Truck throughput time.

Figure 5.11 Impact of the number of shift patterns (and work breaks) on the operator requirements and truck throughput times in different models for shift and truck scheduling (testbed **M**).
Source: Own figure.

Figure 5.11a suggests a regressive relationship between the number of work/break patterns and the average operator requirements. The L-shaped curves indicate that the effect decreases in higher values. When increasing the number of work break patterns from two to four, large operator reductions of ca. 30% in both the **TSFD-MS-F/SP** and the **TSFD-PT-F/SP**, and moderate operator reductions of ca. 10% in the **ISTSFD-F** can be achieved. The effect is smaller in the **ISTSFD-V**, where the operator demand can be reduced by ca. 4%. Further increasing the number of work break patterns does not lead to significant reductions in the operator demand. Figure 5.11b shows how the number of work break patterns impacts the truck throughput times in the different models. The number of work break patterns does not effect the average truck throughput time in the sequential models. This result is as expected, since the truck throughput times are determined in stage one which neglects any shift planning aspects. For the integrated models, on the other hand, it can be seen that the average truck throughput times drop significantly when increasing the number of work break patterns from two to four. All in all, considering the four shift patterns $s \in \{1, 2, 3, 4\}$ seems to be a good compromise as it provides enough flexibility and does not add too much complexity to the problem of scheduling workforce and trucks.

5.5 Chapter summary

This chapter set out to evaluate the performance of both the proposed mathematical programs and the proposed column generation-based solution procedures.

In Section 5.2, the different MIP formulations for the (multi-mode) resource and truck scheduling problems and the shift and truck scheduling problem were evaluated with respects to the solution quality and computational time. It was shown that the discrete-time models are clearly outperforming the continuous-time model formulations in terms of both the solution quality and computational time. Moreover, the results indicated that applying preprocessing parameters for calculating the number of delayed freight units and compelling the service level is beneficial, as it helps to reduce the solution time for most of the instances. The performance of the proposed column generation-based solution procedures was analyzed in Section 5.3. It could be seen that the heuristic solution procedures are able to find high-quality solutions. The developed solution procedures clearly outperformed the MIP formulations in terms of solution quality and computational time and hence should be used when tackling real-world instances. Finally, Section 5.4 aimed to derive managerial insights for the **TSFD-RC-F**, **TSFD-RC-V**, and the **ISTSFDs**. Amongst others, the following take-home messages could be derived:

- Scheduling models which utilize a metric directly related to the internal resource requirements as the objective function obtain plans with a more evenly distributed workload that can be executed with a significantly smaller number of operators than the most frequently used truck scheduling models.

- Considering multi-mode processing, i.e., providing the additional flexibility of adapting the workforce for truck processing, further helps to achieve considerable efficiency gains.
- The service level requirement has a strong impact on the operator demand. Lowering the required service level can be a reasonable means to further improve the operational efficiency in a cross-docking facility.
- Too low a number of work break patterns may result in a strong surge in operator demand.

6 Conclusion

This chapter marks the end of this work. First, the study and its main research results will be summarized. Moreover, opportunities for future research will be described.

6.1 Summary

While operational cross-docking decision problems such as the assignment of trucks to dock-doors and the scheduling of trucks have been addressed, the literature largely overlooked the importance of considering resource planning aspects in operational decision problems. This is very surprising considering that labor cost is “very often the first cost center of a logistics platform”[1] and that material handling equipment such as forklifts requires a significant investment. Therefore, this study’s overall research objective was to “develop novel planning tools that allow cross-docking managers to allocate and schedule internal resources more efficiently”. This was done by developing novel mathematical models that combine two interdependent operational problems faced by cross-docking managers, namely the scheduling of internal resources and the scheduling of trucks.

First, the resource and truck scheduling problem (**TSFD-RC-F)** was introduced. The **TSFD-RC-F** deals with the problem of scheduling inbound trucks in a cross-docking platform, where the departure times of outbound trucks follow a given schedule. In this setting, trucks are processed by operators (that is, a worker equipped with suitable material handling equipment), and the number of required operators for processing a truck is given and known in advance. The goal is to identify a feasible truck schedule that can be executed with a minimum number of operators. Various discrete-time and continuous-time mixed-integer programming formulations were proposed for this novel problem. In light of **TSFD-RC-F** being an NP-hard problem, a

[1] Ladier and Alpan (2016, p. 147).

column generation-based solution procedure was developed to solve large problem instances.

Next, the multi-mode resource and truck scheduling problem (**TSFD-RC-V**) was proposed. The **TSFD-RC-V** is a generalization of the **TSFD-RC-F** since it discards the assumption that the number of required operators for processing a truck is given and known in advance. It integrates the decision of how many resources should be deployed for truck processing. Hence, it better represents real-world settings in which terminal managers usually have the flexibility of adapting the resources for certain trucks, e.g., very time-critical trucks. The **TSFD-RC-V's** objective, however, is identical to the objective of the **TSFD-RC-F**. It seeks to find a truck schedule that can be executed with a minimum number of operators. Again, different mixed-integer programming formulations and a heuristic solution procedure were presented.

Moreover, a further extension of the resource and truck scheduling problem was proposed. The so-called shift and truck scheduling problem (**ISTSFD**) discards some of the **TSFD-RC's** simplifying assumptions regarding the availability of operators. While both the **TSFD-RC-F** and the **TSFD-RC-V** assume one standard shift type without work breaks for operators and a shift length equal to the planning horizon, the **ISTSFD** attempts to provide a framework that integrates the employee timetabling task and the truck scheduling task. Specifically, the **ISTSFD** distinguishes between different operator types (e.g., temporary and regular staff or part-time and full-time staff) and different shift patterns (including work breaks). It seeks to find a truck schedule that can be executed at minimum labor costs. Thus, the **ISTSFD** can be a suitable decision-making tool for labor-intense cross-docking platforms that are often characterized by a high proportion of personnel expenses[1]. The **ISTSFD** was modeled for single-mode operations (**ISTSFD-F**) and multi-mode operations (**ISTSFD-V**). In addition, a column generation-based solution procedure was proposed.

To assess the computational performance of the MIP formulations and the heuristic solution procedures, as well as to derive managerial insights, large-sized computational experiments were conducted. It was shown that the discrete-time MIP formulations clearly outperform the continuous-time MIP formulations in terms of both solution quality and computational time. Moreover, the solution time could be reduced by using the proposed preprocessing parameters for calculating the number of delayed freight units and compelling the service level. While a default solver can solve the discrete-time MIPs for small and medium-sized instances in a reasonable time, it often fails to provide good solutions for very large problem instances with a fine time granularity. The proposed heuristics solution procedures, on the other hand, can

[1] Pfohl (2005, p. 313).

provide high-quality for very large problem instances in a short time and clearly outperform commercial solvers. In addition, extensive computational experiments were conducted in order to derive managerial insights for the novel models. Some of the take-home messages that could be derived were:

- By using the internal resource requirements instead of the frequently used makespan or processing time as the primary performance metrics, the operational efficiency of the cross-docking platform can be significantly increased.
- By integrating the decision of how many resources should be deployed for truck processing (i.e., considering multi-mode processing), further gains in the operational efficiency can be realized.
- The defined service level has a significant impact on the operator demand. Lowering the required service level can be a reasonable means to further improve the operational efficiency in a cross-docking facility.
- The work break patterns have a significant impact on the operator requirements. Too low a number of work break patterns may result in a strong surge in operator demand.

6.2 Future research

Moving forward, there are additional opportunities for future research.

The numerical results indicated that minor adjustments to the predefined service level could further reduce the number of deployed resources and hence improve the operational efficiency. Hence, future research could focus on incorporating the decision of defining an adequate service level. The aggregated results suggested that time window extensions (for all trucks) are unfavorable. However, extending time windows for smaller truck subsets could be a reasonable option to further reduce the operator requirements without significantly increasing the truck throughput times. Fixed outbound departures, as assumed in this work, might not be applicable in some industries. Therefore, the integration of outbound operations could also be an interesting task for future research. To account for uncertainties such as uncertain truck arrival times and truck processing times, it may be worthwhile to develop a robust optimization approach in the future.

From an algorithmic perspective, the proposed column generation procedures may also form the basis of branch-and-price schemes. Even though the heuristics deliver near-optimal solutions for large-sized instances and hence are sufficient for practical purposes, embedding the column generation procedures into exact branch-and-price algorithms can be interesting from a theoretical perspective. It may also be worthwhile to develop Benders decomposition algorithms to tackle the proposed problems. The proposed continuous-time formulations could be used as a starting point.

Appendix

Additional models

The truck scheduling model for minimizing the makespan (**TSFD-MS-F**) in the single-mode context is as follows:

TSFD-MS-F:

$$\text{Minimize} \quad C \tag{A.1}$$

$$\text{subject to} \quad \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} (t + p_{id} - 1) x_{idt} \leq C \qquad \forall i \in \mathcal{I} \tag{A.2}$$

$$\sum_{d \in \mathcal{D}} \sum_{t = r_i}^{d_i} x_{idt} = 1 \qquad \forall i \in \mathcal{I} \tag{A.3}$$

$$\sum_{i \in \mathcal{I}} \sum_{t' = \max\{0; t - p_{id} + 1\}}^{t} x_{idt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{A.4}$$

$$\sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} (t + p_{id} - 1) x_{idt} - d_o \leq \Lambda y_{io} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{A.5}$$

$$\sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} y_{io} \leq (1 - \alpha) \sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}} f_{io} \tag{A.6}$$

$$C \in \mathbb{R}^+ \tag{A.7}$$

$$x_{idt} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, t \in \mathcal{T} \tag{A.8}$$

$$y_{io} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{A.9}$$

The truck scheduling model for minimizing the total truck processing time (**TSFD-PT-F**) in the single-mode context is as follows:

TSFD-PT-F:

$$\text{Minimize} \quad \sum_{i\in\mathcal{I}} \sum_{d\in\mathcal{D}} \sum_{t=r_i}^{d_i} p_{id} x_{idt} \tag{A.10}$$

$$\text{subject to} \quad \sum_{d\in\mathcal{D}} \sum_{t=r_i}^{d_i} x_{idt} = 1 \qquad \forall i \in \mathcal{I} \tag{A.11}$$

$$\sum_{i\in\mathcal{I}} \sum_{t'=\max\{0;t-p_{id}+1\}}^{t} x_{idt'} \leq 1 \qquad \forall t \in \mathcal{T}, d \in \mathcal{D} \tag{A.12}$$

$$\sum_{d\in\mathcal{D}} \sum_{t\in\mathcal{T}} (t + p_{id} - 1) x_{idt} - d_o \leq \Lambda y_{io} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{A.13}$$

$$\sum_{i\in\mathcal{I}} \sum_{o\in\mathcal{O}} f_{io} y_{io} \leq (1-\alpha) \sum_{i\in\mathcal{I}} \sum_{o\in\mathcal{O}} f_{io} \tag{A.14}$$

$$x_{idt} \in \{0;1\} \qquad \forall i \in \mathcal{I}, d \in \mathcal{D}, t \in \mathcal{T} \tag{A.15}$$

$$y_{io} \in \{0;1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O} : f_{io} > 0 \tag{A.16}$$

The shift planning model (**SP**) requires the input parameters w_t^D expressing the operator demand in time interval $t \in \mathcal{T}$ as an input. It computes an employee timetable with minimum total operator costs (if a feasible employee timetable for the given truck schedule exists) and can be formulated as follows:

SP

$$\text{Minimize} \quad \sum_{g\in\}} \sum_{s\in\mathcal{S}_g} C_{gs} W_{gs} \tag{A.17}$$

$$\text{subject to} \quad \sum_{g\in\mathcal{G}} \sum_{s\in\mathcal{S}_g} \gamma_{gst} W_{gs} \geq w_t^D \qquad \forall t \in \mathcal{T} \tag{A.18}$$

$$\sum_{s\in\mathcal{S}_g} W_{gs} \leq \overline{W}_g \qquad \forall g \in \mathcal{G} \tag{A.19}$$

$$W_{gs} \in \mathbb{Z}^+ \qquad \forall g \in \mathcal{G}, s \in \mathcal{S}_g \tag{A.20}$$

Bibliography

Acar, K., Yalcin, A., and Yankov, D. (2012). Robust door assignment in less-than-truckload terminals. *Computers and Industrial Engineering*, 63(4):729–738.

Alpan, G., Ladier, A.-L., Larbi, R., and Penz, B. (2011a). Heuristic solutions for transshipment problems in a multiple door cross docking warehouse. *Computers and Industrial Engineering*, 61(2):402–408.

Alpan, G., Larbi, R., and Penz, B. (2011b). A bounded dynamic programming approach to schedule operations in a cross docking platform. *Computers and Industrial Engineering*, 60(3):385–396.

Álvarez-Pérez, G. A., González-Velarde, J. L., and Fowler, J. W. (2009). Crossdocking just in time scheduling: An alternative solution approach. *Journal of the Operational Research Society*, 60(4):554–564.

Amini, A., Tavakkoli-Moghaddam, R., and Omidvar, A. (2014). Cross-docking truck scheduling with the arrival times for inbound trucks and the learning effect for unloading/loading processes. *Production and Manufacturing Research*, 2(1):784–804.

Apte, U. M. and Viswanathan, S. (2000). Effective cross docking for improving distribution efficiencies. *International Journal of Logistics*, 3(3):291–302.

Artigues, C., Gendreau, M., Rousseau, L.-M., and Vergnaud, A. (2009). Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound. *Computers and Operations Research*, 36(8):2330–2340.

Bartholdi, J. J. and Gue, K. R. (2000). Reducing labor costs in an LTL crossdocking terminal. *Operations Research*, 48(6):823–832.

Bartholdi, J. J. and Gue, K. R. (2004). The best shape for a crossdock. *Transportation Science*, 38(2):235–244.

Bartholdi, J. J., Gue, K. R., and Kang, K. (2008). Staging protocols for unit-load crossdocking. In Lahmar, M., editor, *Facility Logistics. Approaches and Solutions to Next Generation Challenges*, pages 153–171. Auerbach Publications, Boca Raton, FL.

Bartz-Beielstein, T., Chmielewski, A., Janas, M., Naujoks, B., and Scheffermann, R. (2006). Optimizing door assignment in LTL-terminals by evolutionary multiobjective algorithms. In *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pages 41–47.

Battini, D., Boysen, N., and Emde, S. (2013). Just-in-time supermarkets for part supply in the automobile industry. *Journal of Management Control*, 24(2):209–217.

Belella, P., Maggiore, M., Rychlik, N., Beshers, E., Manzo, T., Keppler, S., Fayez, S., and Ang-Olson, J. (2009). Motor carrier efficiency study phase I. Technical report, Federal Motor Carrier Safety Administration, U.S. Department of Transportation, Washington, DC.

Bellanger, A., Hanafi, S., and Wilbaut, C. (2013). Three-stage hybrid-flowshop model for cross-docking. *Computers and Operations Research*, 40(4):1109–1121.

Berghman, L. and Leus, R. (2015). Practical solutions for a dock assignment problem with trailer transportation. *European Journal of Operational Research*, 246(3):787–799.

Bermúdez, R. and Cole, M. H. (2001). A genetic algorithm approach to door assignments in breakbulk terminals. Technical report, Mack-Blackwell Rural Transportation Center, University of Arkansas, Arkansas.

Bhaskaran, S. (1992). Identification of transshipment center locations. *European Journal of Operational Research*, 63(2):141–150.

Błażewicz, J., Machowiak, M., and Węglarz, J. (2004). Scheduling malleable tasks on parallel processors to minimize the makespan. *Annals of Operations Research*, 129:65–80.

Bodnar, P., De Koster, R., and Azadeh, K. (2017). Scheduling trucks in a cross-dock with mixed service mode dock doors. *Transportation Science*, 51(1):112–131.

Boloori Arabani, A. R., Fatemi Ghomi, S. M. T., and Zandieh, M. (2010). A multi-criteria cross-docking scheduling with just-in-time approach. *International Journal of Advanced Manufacturing Technology*, 49:741–756.

Boloori Arabani, A. R., Fatemi Ghomi, S. M. T., and Zandieh, M. (2011a). Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert Systems with Applications*, 38(3):1964–1979.

Boloori Arabani, A. R., Zandieh, M., and Fatemi Ghomi, S. M. T. (2011b). Multi-objective genetic-based algorithms for a cross-docking scheduling problem. *Applied Soft Computing*, 11(8):4954–4970.

Boloori Arabani, A. R., Zandieh, M., and Fatemi Ghomi, S. M. T. (2012). A cross-docking scheduling problem with sub-population multi-objective algorithms. *International Journal of Advanced Manufacturing Technology*, 58:741–761.

Boysen, N., Briskorn, D., and Tschöke, M. (2013). Truck scheduling in cross-docking terminals with fixed outbound departures. *OR Spectrum*, 35(2):479–504.

Boysen, N., Fedtke, S., and Weidinger, F. (2017). Truck scheduling in the postal service industry. *Transportation Science*, 51(2):723–736.

Boysen, N. and Fliedner, M. (2010). Cross dock scheduling: Classification, literature review and research agenda. *Omega*, 38(6):413–422.

Boysen, N., Fliedner, M., and Scholl, A. (2010). Scheduling inbound and outbound trucks at cross docking terminals. *OR Spectrum*, 32(1):135–161.

Bozer, Y. A. and Carlo, H. J. (2008). Optimizing inbound and outbound door assignments in less-than-truckload crossdocks. *IIE Transactions (Institute of Industrial Engineers)*, 40(11):1007–1018.

Breitschwerdt, J. (2016). What rights do workers have to rest breaks in Germany? https://www.globalworkplaceinsider.com/2016/12/what-rights-do-workers-have-to-rest-breaks-in-germany/, accessed at July 23, 2020.

Briskorn, D., Choi, B. C., Lee, K., Leung, J., and Pinedo, M. (2010). Complexity of single machine scheduling subject to nonnegative inventory constraints. *European Journal of Operational Research*, 207(2):605–619.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., and Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1):3–41.

Brucker, P. and Knust, S. (2003). Lower bounds for resource-constrained project scheduling problems. *European Journal of Operational Research*, 149(2):302–313.

Buijs, P., Vis, I. F., and Carlo, H. J. (2014). Synchronization in cross-docking networks: A research classification and framework. *European Journal of Operational Research*, 239(3):593–608.

Carlo, H. J. and Bozer, Y. A. (2011). Analysis of optimum shape and door assignment problems in rectangular unit-load crossdocks. *International Journal of Logistics Research and Applications*, 14(3):149–163.

Chen, D.-S., Batson, R. G., and Dang, Y. (2010). *Applied Integer Programming*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Chen, F. and Lee, C. Y. (2009). Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*, 193(1):59–72.

Chen, F. and Song, K. (2009). Minimizing makespan in two-stage hybrid cross docking scheduling problem. *Computers and Operations Research*, 36(6):2066–2073.

Chen, Z.-L. (2004). Simultaneous job scheduling and resource allocation on parallel machines. *Annals of Operations Research*, 129(1-4):135–153.

Chen, Z. L. and Powell, W. B. (1999). Solving parallel machine scheduling problems by column generation. *INFORMS Journal on Computing*, 11(1):78–92.

Chiarello, A., Gaudioso, M., and Sammarra, M. (2018). Truck synchronization at single door cross-docking terminals. *OR Spectrum*, 40(2):395–447.

Chmielewski, A., Naujoks, B., Janas, M., and Clausen, U. (2009). Optimizing the door assignment in LTL-terminals. *Transportation Science*, 43(2):198–210.

Cohen, Y. and Keren, B. (2009). Trailer to door assignment in a synchronous cross-dock operation. *International Journal of Logistics Systems and Management*, 5(5):574–590.

Cook, R. L., Gibson, B., and MacCurdy, D. (2005). A lean approach to cross docking. *Supply Chain Management Review*, 9:54–59.

Costello, B. and Karickhoff, A. (2019). Truck driver shortage analysis 2019. Technical report, American Trucking Association.

Cruz-Reyes, L., Hernández, P., Melin, P., Mar-Ortiz, J., Huacuja, H. J. F., Soberanes, H. J. P., and Barbosa, J. J. G. (2015). Lower and upper bounds for the master bay planning problem. *International Journal of Combinatorial Optimization Problems and Informatics*, 6(1):42–52.

Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors (2005). *Column generation*. Springer Science+Business Media, New York.

Drexl, A. and Kimms, A. (2001). Optimization guided lower and upper bounds for the resource investment problem. *Journal of the Operational Research Society*, 52(3):340–351.

Edis, E. B., Oguz, C., and Ozkarahan, I. (2013). Parallel machine scheduling with additional resources: Notation, classification, models and solution methods. *European Journal of Operational Research*, 230(3):449–463.

Emde, S., Abedinnia, H., Lange, A., and Glock, C. H. (2020). Scheduling personnel for the build-up of unit load devices at an air cargo terminal with limited space. *OR Spectrum*, 42(2):397–426.

Fanjul-Peyro, L., Perea, F., and Ruiz, R. (2017). Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources. *European Journal of Operational Research*, 260(2):482–493.

Fanti, M. P., Stecco, G., and Ukovich, W. (2014). Scheduling internal operations in post-distribution cross docking systems. *IEEE Transactions on Automatic Control*, pages 1–17.

Fazel Zarandi, M. H., Khorshidian, H., and Akbarpour Shirazi, M. (2016). A constraint programming model for the scheduling of JIT cross-docking systems with preemption. *Journal of Intelligent Manufacturing*, 27(2):297–313.

Fleszar, K. and Hindi, K. S. (2018). Algorithms for the unrelated parallel machine scheduling problem with a resource constraint. *European Journal of Operational Research*, 271(3):839–848.

Forger, G. (1995). UPS starts world's premiere cross-docking operation. *Modern Material Handling*, pages 36–38.

Forouharfard, S. and Zandieh, M. (2010). An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems. *International Journal of Advanced Manufacturing Technology*, 51(9):1179–1193.

Fügener, A. and Brunner, J. O. (2019). Planning for overtime: The value of shift extensions in physician scheduling. *INFORMS Journal on Computing*, 31(4):732–744.

Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability. A guide to the theory of NP-completeness*. W. H. Freeman, New York.

Gelareh, S., Glover, F., Guemri, O., Hanafi, S., Nduwayo, P., and Todosijević, R. (2020). A comparative study of formulations for a cross-dock door assignment problem. *Omega (United Kingdom)*, 91.

Gendreau, M. and Potvin, J. Y. (2019). Tabu search. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics*, pages 37–55. Springer International Publishing, Cham, 3 edition.

Gerhards, P. (2020). The multi-mode resource investment problem: A benchmark library and a computational study of lower and upper bounds. *OR Spectrum*, 42(4):901–933.

Ghobadian, E., Tavakkoli-Moghaddam, R., Javanshir, H., and Naderi-Beni, M. (2012). Scheduling trucks in cross docking systems with temporary storage and dock repeat truck holding pattern using GRASP method. *International Journal of Industrial Engineering Computations*, 3(5):777–786.

Goletz, B. (2015). Working time – are there any restrictions on an employee's working week in Germany? https://www.globalworkplaceinsider.com/2015/05/working-time-are-there-any-restrictions-on-an-employees-working-week-in-germany/, accessed at July 23, 2020.

Golias, M. M., Saharidis, G. K. D., Boile, M., and Theofanis, S. (2012). Scheduling of inbound trucks at a cross-docking facility: Bi-objective vs bi-level modeling approaches. *International Journal of Information Systems and Supply Chain Management*, 5(1):20–37.

Golias, M. M., Saharidis, G. K. D., Ivey, S., and Haralambides, H. E. (2013). Advances in truck scheduling at a cross dock facility. *International Journal of Information Systems and Supply Chain Management*, 6(3):40–62.

Gorczyca, M. and Janiak, A. (2010). Resource level minimization in the discrete-continuous scheduling. *European Journal of Operational Research*, 203(1):32–41.

Gue, K. R. (1999). The effects of trailer scheduling on the layout of freight terminals. *Transportation Science*, 33(4):419–428.

Gümüş, M. and Bookbinder, J. H. (2004). Cross-docking and its implications in location-distribution systems. *Journal of Business Logistics*, 25(2):199–228.

Guyon, O., Lemaire, P., Pinson, É., and Rivreau, D. (2014). Solving an integrated job-shop problem with human resource constraints. *Annals of Operations Research*, 213(1):147–171.

Hartmann, S. (2013). Project scheduling with resource capacities and requests varying with time: A case study. *Flexible Services and Manufacturing Journal*, 25(1-2):74–93.

Hartmann, S. and Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1–14.

Hermel, D., Hasheminia, H., Adler, N., and Fry, M. J. (2016). A solution framework for the multi-mode resource-constrained cross-dock scheduling problem. *Omega*, 59:157–170.

Hsu, C.-C. and Kim, D. S. (2005). A new heuristic for the multi-mode resource investment problem. *Journal of the Operational Research Society*, 56(4):406–413.

Jarrah, A. I., Qi, X., and Bard, J. F. (2016). The destination-loader-door assignment problem for automated package sorting centers. *Transportation Science*, 50(4):1314–1336.

Joo, C. M. and Kim, B. S. (2013). Scheduling compound trucks in multi-door cross-docking terminals. *International Journal of Advanced Manufacturing Technology*, 64:977–988.

Kinnear, E. (1997). Is there any magic in cross-docking? *Supply Chain Management. An International Journal*, 2(2):49–52.

Klein, R. (2000). Project scheduling with time-varying resource constraints. *International Journal of Production Research*, 38(16):3937–3952.

Kress, D., Müller, D., and Nossack, J. (2019). A worker constrained flexible job shop scheduling problem with sequence-dependent setup times. *OR Spectrum*, 41(1):179–217.

Ladier, A.-L. (2014). *Scheduling cross-docking operations: Integration of operational uncertainties and resource capacities*. Ph.d. thesis, Université de Grenoble.

Ladier, A. L. and Alpan, G. (2015). Integrating truck scheduling and employee rostering in a cross-docking platform - An iterative approach. In *Proceedings of the international conference on industrial engineering and systems management 2015*, number October, pages 676–685, Seville, Spain.

Ladier, A. L. and Alpan, G. (2016). Cross-docking operations: Current research versus industry practice. *Omega (United Kingdom)*, 62:145–162.

Ladier, A.-L. and Alpan, G. (2018). Crossdock truck scheduling with time windows: earliness, tardiness and storage policies. *Journal of Intelligent Manufacturing*, 29(3):569–583.

Ladier, A. L., Alpan, G., and Penz, B. (2014). Joint employee weekly timetabling and daily rostering: A decision-support tool for a logistics platform. *European Journal of Operational Research*, 234(1):278–291.

Lamorgese, L. and Mannino, C. (2019). A noncompact formulation for job-shop scheduling problems in traffic management. *Operations Research*, 67(6):1586–1609.

Larbi, R., Alpan, G., Baptiste, P., and Penz, B. (2011). Scheduling cross docking operations under full, partial and no information on inbound arrivals. *Computers and Operations Research*, 38:889–900.

Li, Y., Lim, A., and Rodrigues, B. (2004). Crossdocking - JIT scheduling with time windows. *Journal of the Operational Research Society*, 55(12):1342–1351.

Liao, T. W., Egbelu, P. J., and Chang, P. C. (2012). Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations. *Applied Soft Computing*, 12(11):3683–3697.

Liao, T. W., Egbelu, P. J., and Chang, P. C. (2013). Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations. *International Journal of Production Economics*, 141(1):212–229.

Madani-Isfahani, M., Tavakkoli-Moghaddam, R., and Naderi, B. (2014). Multiple cross-docks scheduling using two meta-heuristic algorithms. *Computers and Industrial Engineering*, 74:129–138.

Maknoon, M. Y. and Baptiste, P. (2009). Cross-docking: Increasing platform efficiency by sequencing incoming and outgoing semi-trailers. *International Journal of Logistics Research and Applications*, 12(4):249–261.

Maknoon, M. Y., Kone, O., and Baptiste, P. (2014). A sequential priority-based heuristic for scheduling material handling in a satellite cross-dock. *Computers and Industrial Engineering*, 72:43–49.

McWilliams, D. L. (2009). A dynamic load-balancing scheme for the parcel hub-scheduling problem. *Computers and Industrial Engineering*, 57(3):958–962.

McWilliams, D. L., Stanfield, P. M., and Geiger, C. D. (2005). The parcel hub scheduling problem: A simulation-based solution approach. *Computers and Industrial Engineering*, 49(3):393–412.

Molavi, D., Shahmardan, A., and Sajadieh, M. S. (2018). Truck scheduling in a cross docking systems with fixed due dates and shipment sorting. *Computers and Industrial Engineering*, 117(January):29–40.

Mousavi, S. M. and Tavakkoli-Moghaddam, R. (2013). A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain. *Journal of Manufacturing Systems*, 32(2):335–347.

Musa, R., Arnaout, J.-P., and Jung, H. (2010). Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Computers and Industrial Engineering*, 59(1):85–92.

Nassief, W., Contreras, I., and Asad, R. (2016). A mixed-integer programming formulation and Lagrangean relaxation for the cross-dock door assignment problem. *International Journal of Production Research*, 54(2):494–508.

Neumann, K. and Zimmermann, J. (2000). Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. *European Journal of Operational Research*, 127(2):425–443.

Oh, Y., Hwang, H., Cha, C. N., and Lee, S. (2006). A dock-door assignment problem for the Korean mail distribution center. *Computers & Industrial Engineering*, 51:288–296.

Peck, K. E. (1983). *Operational analysis of freight terminals handling less than container load shipments*. Ph.d. thesis, University of Illinois at Urbana-Champaign,.

Pfohl, H.-C. (2005). Erfolgreicher Einsatz von Personal in der Logistik. In Eßig, M., editor, *Perspektiven des Supply Management - Konzepte und Anwendungen*, pages 309–328. Springer, Berlin, Heidelberg.

Pinedo, M. L. (2016). *Scheduling. Theory, Algorithms, and Systems*. Springer International Publishing, Cham.

Rahmanzadeh Tootkaleh, S., Fatemi Ghomi, S. M., and Sheikh Sajadieh, M. (2016). Cross dock scheduling with fixed outbound trucks departure times under substitution condition. *Computers and Industrial Engineering*, 92:50–56.

Rijal, A., Bijvank, M., and de Koster, R. (2019). Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors. *European Journal of Operational Research*, 278(3):752–771.

Romanova, A. A. (2015). Algorithms for minimizing the makespan in a two-machine cross-docking flow shop problem. *Journal of Applied and Industrial Mathematics*, 9(4):570–579.

Rosales, C. R., Fry, M. J., and Radhakrishnan, R. (2009). Transfreight reduces costs and balances workload at Georgetown crossdock. *Interfaces*, 39(4):316–328.

Ross, A. and Jayaraman, V. (2008). An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design. *Computers and Industrial Engineering*, 55(1):64–79.

Saddle Creek Corporation (2011). 2011 cross-docking trends report. Technical report.

Sadykov, R. (2012). Scheduling incoming and outgoing trucks at cross docking terminals to minimize the storage cost. *Annals of Operations Research*, 201:423–440.

Serrano, C., Delorme, X., and Dolgui, A. (2017). Scheduling of truck arrivals, truck departures and shop-floor operation in a cross-dock platform, based on trucks loading plans. *International Journal of Production Economics*, 194:102–112.

Shahram fard, S. and Vahdani, B. (2019). Assignment and scheduling trucks in cross-docking system with energy consumption consideration and trucks queuing. *Journal of Cleaner Production*, 213:21–41.

Shakeri, M., Low, M. Y. H., Turner, S. J., and Lee, E. W. (2012). A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. *Computers and Operations Research*, 39(11):2564–2577.

Shiguemoto, A. L., Cavalcante Netto, U. S., and Bauab, G. H. S. (2014). An efficient hybrid meta-heuristic for a cross-docking system with temporary storage. *International Journal of Production Research*, 52(4):1231–1239.

Soltani, R. and Sadjadi, S. J. (2010). Scheduling trucks in cross-docking systems: A robust meta-heuristics approach. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):650–666.

Speltz, E. and Murray, D. (2019). Driver detention impacts on safety and productivity. Technical Report September, American Transportation Research Institute, Minneapolis, MN.

Sprecher, A. and Drexl, A. (1998). Multi-mode resource-constrained project scheduling by a simple , general and powerful sequencing algorithm. *European Journal of Operational Research*, 107:431–450.

Stalk, G., Evans, P., and Shulman, L. E. (1992). Competing on capabilities: The new rules of corporate strategy. *Harvard Business Review*, 70(2):57–68.

Stephan, K. and Boysen, N. (2011). Cross-docking. *Journal of Management Control*, 22(1):129–137.

Sung, C. S. and Song, S. H. (2003). Integrated service network design for a cross-docking supply chain network. *Journal of the Operational Research Society*, 54(12):1283–1295.

Sung, C. S. and Yang, W. (2008). An exact algorithm for a cross-docking supply chain network design problem. *Journal of the Operational Research Society*, 59(1):119–136.

Tadumadze, G., Boysen, N., Emde, S., and Weidinger, F. (2019). Integrated truck and workforce scheduling to accelerate the unloading of trucks. *European Journal of Operational Research*, 278(1):343–362.

Tsui, L. Y. and Chang, C.-H. (1990). A microcomputer based decision support tool for assigning dock doors in freight yards. *Computers and Industrial Engineering*, 19(1-4):309–312.

Tsui, L. Y. and Chang, C.-H. (1992). An optimal solution to a dock door assignment problem. *Computers and Industrial Engineering*, 23(1-4):283–286.

Vahdani, B., Soltani, R., and Zandieh, M. (2010). Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics. *International Journal of Advanced Manufacturing Technology*, 46(5):769–783.

Vahdani, B. and Zandieh, M. (2010). Scheduling trucks in cross-docking systems: Robust meta-heuristics. *Computers & Industrial Engineering*, 58:12–24.

Van Belle, J., Valckenaers, P., and Cattrysse, D. (2012). Cross-docking: State of the art. *Omega*, 40(6):827–846.

Van Belle, J., Valckenaers, P., Vanden Berghe, G., and Cattrysse, D. (2013). A tabu search approach to the truck scheduling problem with multiple docks and time windows. *Computers and Industrial Engineering*, 66(4):818–826.

Van Den Akker, J. M., Hoogeveen, J. A., and Van Kempen, J. W. (2012). Using column generation to solve parallel machine scheduling problems with minmax objective functions. *Journal of Scheduling*, 15(6):801–810.

Van Den Akker, M., Hoogeveen, J. A., and Van Velde, S. L. (1999). Parallel machine scheduling by column generation. *Operations Research*, 47(6):862–872.

Vis, I. F. A. and Roodbergen, K. J. (2008). Positioning of goods in a cross-docking environment. *Computers and Industrial Engineering*, 54(3):677–689.

Weber, J., Lukassen, P., Mahlendorf, M., Rommel, K., and Wallenburg, C. M. (2005). Flexibilität in der Logistik. Potentiale und Grenzen von Personaldienstleistungen. Technical report, WHU - Otto Beisheim School of Management, Vallendar.

Węglarz, J., Józefowska, J., Mika, M., and Waligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes - A survey. *European Journal of Operational Research*, 208(3):177–205.

Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F., and Laporte, G. (2009). Vehicle routing with cross-docking. *Journal of the Operational Research Society*, 60(12):1708–1718.

Witt, C. E. (1998). Crossdocking: concepts demand choice. *Material Handling Engineering*, 53(7):44–49.

Yazdani, M., Naderi, B., and Mousakhani, M. (2015). A model and metaheuristic for truck scheduling in multi-door cross-dock problems. *Intelligent Automation and Soft Computing*, 21(4):633–644.

Yu, W. and Egbelu, P. J. (2008). Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research*, 184(1):377–396.

Zeighami, V. and Soumis, F. (2019). Combining Benders' decomposition and column generation for integrated crew pairing and personalized crew assignment problems. *Transportation Science*, 53(5):1479–1499.

Zhang, T., Saharidis, G. K. D., Theofanis, S., and Boile, M. (2010). Scheduling of inbound and outbound trucks at cross-docks: Modeling and analysis. *Transportation Research Record*, (2162):9–16.