

The Impact of Load Carrier Types and Staging-Level Designs on Cross-Docking Performance under Uncertainty

A Discrete Event Simulation Study

A.W.B. Hofstee

Delft University of Technology



The Impact of Load Carrier Types and Staging-Level Designs on Cross-Docking Performance under Uncertainty

A Discrete Event Simulation Study

by

A.W.B. Hofstee

to obtain the degree of Master of Science
at Delft University of Technology,
to be defended publicly
on Tuesday September 5, 2023 at 13:30 PM.

Student number:	4438388
Project duration:	September 5, 2022 – September 5, 2023
Thesis committee:	Prof. dr. R.R. Negenborn, TU Delft, chair Ir. M.B. Duinkerken, TU Delft, supervisor Dr. S. Fazi, TU Delft, supervisor S. Baljet, Budbee, supervisor

Glossary

- 4PL** Fourth Party Logistics Provider. 29, 77
- ATA** Actual Time of Arrival. 26–28
- ATD** Actual Time of Departure. 26–28
- CDF** cross-docking facility. iii, 2, 15, 22, 81
- DES** Discrete Event Simulation. v, 13
- EPT** Electric Pallet Truck. 19, 30, 81
- ETA** Expected Time of Arrival. 26, 28
- ETD** Expected Time of Departure. 26, 28
- FIFO** First In First Out. 47, 48
- FTL** Full Truckload. xi, 5, 40
- KPI** Key Performance Indicator. xiii, 42
- LIFO** Last In First Out. 47
- LTL** Less Than Truckload. 5
- MPT** Manual Pallet Truck. 19, 30, 81
- UL** Unit Load. 36, 37, 39, 42–45, 47, 48, 50
- WMS** Warehouse Management System. 7, 24

Preface

This report represents the culmination of my Master's thesis for the Master's program in Transport, Infrastructure, and Logistics (TIL) at the Delft University of Technology, which was conducted in collaboration with Budbee. I want to express my sincere gratitude to all those who provided support and guidance during this journey.

First and foremost, I am indebted to my supervisors, Mark Duinkerken and Stefano Fazi, for their invaluable guidance and insightful perspectives. Additionally, I am grateful to Sam Baljet, who provided essential feedback and input data and ensured that my time at Budbee was delightful. I would also like to acknowledge the tremendous assistance I received from David T. Sturrock on the Simio Insiders forum, as his explanations on the simulation software helped me develop my simulation model.

Throughout my academic journey, the Delft University of Technology has taught me to gather, integrate, and interpret diverse source materials to translate them into valuable insights and knowledge. Significantly, this thesis has deepened my understanding of the intricate complexities within the TIL domain by considering problems from different perspectives. This enables me to reflect critically on technical, scientific, and policy dimensions while considering the associated implications and limitations.

The academic activities undertaken during this thesis have allowed me to form informed judgments, draw meaningful conclusions, and provide well-founded recommendations based on my research findings. Furthermore, I have embraced the practice of academic reflection, drawing from relevant literature and insights shared by experts in the field.

Lastly, I would like to extend my heartfelt appreciation to my family and friends for their unwavering support throughout my thesis and academic journey.

*A.W.B. Hofstee
Delft, April 2023*

Summary

There has been a significant increase in e-commerce sales in recent years, with a reported growth of 290% between 2014 and 2020. This growth is expected to continue, with forecasts predicting a further 56% increase by 2026, resulting in approximately 8.1 trillion dollars in worldwide retail e-commerce sales. However, with this growth comes an increase in product returns, with a return rate of 30%, particularly during the Covid-19 pandemic. This has added complexity to already complex supply chains.

To accommodate the increase in e-commerce sales, logistics companies have expanded their assets and increased efficiency in their operations. One of the strategies to improve logistic efficiency is cross-docking. Cross-docking is a logistics strategy that reduces distribution centers' handling and storage time and associated costs. In this process, goods delivered by inbound trucks are reloaded onto outbound trucks to expedite the process. Cross-docking is also employed in the retail industry, where inbound goods are sorted, consolidated, and stored until they are ready to be shipped out. Essentially, a cross-docking facility (CDF) operates like a warehouse but with the added benefit of minimal handling operations and a storage period of fewer than 24 hours for unloaded loads. Particularly in cross-docking operations, synchronization is crucial for using available resources optimally and optimizing efficiencies in logistics. However, cross-docking operations are prone to uncertainties, such as truck arrival times, which undermine the synchronization of the complex operations. This study aims to improve the performance of cross-docking operations under uncertainties using a logistical service provider case study. Therefore, the following question arises.

“How to improve cross-docking operations under uncertainty?”

To address this question, first, a greater understanding of CDFs and theories on CDFs are searched for in the available literature. This literature research is performed to answer the following sub-question.

“What is a cross-docking terminal, and how can uncertainty be mitigated in literature?”

A cross-docking terminal (CDF) is a facility used in supply chains to consolidate and redistribute goods without long-term storage. Goods typically spend less than 24 hours at a CDF. The main activities at a CDF include unloading incoming trucks, sorting goods, and loading outbound trucks for delivery [31]. Uncertainty in arrival times, processing times, and freight volumes can impact CDF performance [6].

Several ways to mitigate uncertainty at cross-docking terminals are suggested in the literature:

- Increasing dock door availability reduces congestion from variable truck arrivals. Dock doors can be sized based on trucks, loading times, hours, and a safety factor [40].
- Using mixed-mode dock doors (both inbound and outbound) improves performance over exclusive doors. At least 60% mixed-mode is recommended [11, 50, 10].
- Adjusting internal transportation methods via conveyors, vehicles, and load carriers can smooth material flows under fluctuating volumes [8, 40].
- Truck-to-dock assignment policies help reduce delays from uncertain arrival times. Optimization is limited, so policies based on known data are preferred [30].
- Simulation modeling incorporates interdependencies and variability that analytical optimization methods cannot handle well [8, 30].
- Adding staging storage areas and optimizing layouts can mitigate downstream bottlenecks when processing volumes are variable [8, 23].

- Monitoring freight levels and truck status enables dynamic scheduling and assignments to handle real-time changes [30].

While optimization excels at finding ideal solutions under fixed conditions, simulation better captures operational uncertainties though it cannot identify optimal parameters [8, 30]. Research gaps remain around assessing load carriers and staging designs under uncertainty [59, 23].

To model cross-docking operations under uncertainty, analytical optimization methods are preferred for problems requiring (near) optimal results, such as truck-to-door assignment and scheduling optimization. However, simulation can capture interdependencies and uncertainty in demand and arrival times. The performance of cross-docking operations is measured using time-related, inventory-related, or resource utilization-related performance indicators such as makespan, inventory levels, door utilization, and waiting times. Design alternatives to mitigate uncertainty include increasing the availability of dock doors and using mixed-mode dock doors. Internal transportation methods can also impact performance, but limited research has been conducted. Simulation is often used to model complex interdependencies, but optimization methods are more suitable for obtaining optimal solutions. Overall, the effects of internal transportation methods have not been fully addressed for different load carriers.

Now that a detailed understanding of CDFs is obtained and means to improve the performance of cross-docking terminals from the literature, the current state of the CDF in the case study must be examined. This helps identify possible means for improving the performance of the cross-docking operations under uncertainty in practice. To this end, the following research question is answered:

“What is the current state of cross-docking at the CDF of Budbee?”

The main research topic requires assessing the current state of cross-docking operations in the case study. It is crucial to note that the terminal's layout is characterized by staging areas that differ in capacities, and trucks are pre-assigned to dock doors based on historical volumes. However, the lack of data monitoring prohibits truck scheduling optimization. Additionally, the CDF functions as an RT with direct injection of demand volume, preventing the terminal from being viewed as a pure CDF. However, the literature mentioned that most CDFs operate as hybrid cross-docking and warehousing terminals.

The activities conducted in the CDF are receiving inbound trucks, unloading inbound trucks, simultaneously moving pallets to the outbound staging area, reloading outbound trucks, and dispatching them. With each dock door having its own assigned EPT, which performs all unloading, moving, and reloading of the related door, it is essential to note that the main resources used in cross-docking are EPTs, MPTs, and workers. Dock doors are also considered resources, and the number of trucks served exceeds the total number of dock doors.

It is imperative to recognize that the current state of the cross-docking terminal is uncertain due to varying demand volumes and truck arrival times. The load carrier type mainly determines the mode of transport, with pallets being transported using EPTs or MPTs, and roll containers being manually moved by terminal personnel. However, the conflicting interests between the truck drivers and the terminal operator, with drivers wanting to minimize waiting times and the terminal operator aiming to minimize the total makespan, pose a significant challenge.

Two potential design alternatives have been identified to complement the current state and improve operations. The two design alternatives are changing the internal transportation method or redesigning the staging level strategy of the CDF. However, it is crucial to note that the lack of a well-functioning WMS restrains the CDF from instigating truck scheduling optimization, which could benefit the cross-docking terminal. Additionally, due to the lack of monitoring incoming freight, the CDF must unload and store all pallets at their assigned staging area before reloading can commence. This currently limits the system's maximum throughput to the staging area capacity. The design alternatives identified are changing the internal transportation method or redesigning the staging level strategy of the CDF.

- A0: Single-Stage Cross-docking - Current State (do nothing)
- A1: Two-Stage Cross-Docking with Pallets
- A2: Two-Stage Cross-Docking with Roll Containers

The alternatives mentioned above are identified as the most useful for the case study, and their performance will be tested using modeling. However, it is crucial to note that the lack of a well-functioning WMS restrains the CDF from instigating truck scheduling optimization and imposes the need to unload and store all pallets in the corresponding staging area before reloading can commence. This limits the system's maximum throughput to the outbound staging area capacity.

To analyze the system in the case study, it is important to create a model that can test different alternatives. To achieve this, the following sub-question needs to be addressed.

“How to develop a model for testing the design alternatives for a cross-docking facility under uncertainty?”

To test the impact of different design alternatives, a simulation model is developed to capture the interdependencies in the system. This includes factors such as arriving trucks, resource occupancy, and queuing. Regarding the model inputs, the model must include the number of ULs generated(1), truck arrival time window (2), and cross-docking purity (3). Considering the design alternatives, the model must simulate two-stage cross-docking with pallets and two-stage cross-docking with roll containers. To this end, the load carrier must be dynamically altered, and single-stage and two-stage process steps must be included. The processes and activities to be modeled must include truck and load carrier generation, receiving incoming trucks, unloading, moving to stage areas, reloading, and dispatching. Additionally, the model must measure the performance of the design alternatives by recording the KPIs identified as the *Total Makespan*, *Individual Truck Waiting Time*, and *Throughput per hour*.

It's important to note that models are never 100% accurate due to the need to make assumptions while modeling. In this research, one of the assumptions is that the truck-to-dock assignment is fixed and doesn't change. Additionally, it is assumed that each dock has the same properties and that the staging area has unlimited capacity. Furthermore, it is assumed that the CDF starts and finishes empty as backlog ULs are transported by dedicated trucks. Other assumptions include EPT and Worker traveling speeds, pick-up and drop-off speeds, truck load capacities for different consolidation types, and assuming sufficient employees to operate the EPTs.

The simulation model is implemented in SIMIO, an object-oriented Discrete Event Simulation (DES) modeling software, which suits the problem as it can handle underlying data regarding the origin, destination, number of entities, travel time, and operational time underlying the model objects. The conceptual and computer models undergo verification and validation, followed by calibration of model parameters and assessment of input data validity. The conceptual model is then validated through visual checks, tracing, and black-box and white-box validation. Black-box validation shows that the difference in means for '*Throughput*' and '*Total Makespan*' between simulation and empirical data is statistically insignificant, with a 95% confidence level. Statistical analysis also reveals no significant differences in data distribution between the two. These verification and validation steps have greatly increased confidence in the simulation model, although pure CDF data validation remains incomplete. Further experiments will be conducted on the case study to achieve generalization. Additionally, pure CDF configurations are tested as well for scientific interests.

However, the literature mentioned that most CDFs operate in non-pure CDF configurations, showing hybrid cross-docking and warehousing characteristics. Since the model parameters can be altered to examine pure CDF configurations and there is interest in testing the purity of cross-docking, the model will also be tested for pure CDF configurations. However, since pure CDF data is lacking, the model's validation for pure CDFs cannot be ensured.

Now that the model is formulated for testing the design alternatives, the model performance is tested for different scenarios by answering the following sub-question.

“What are the effects of different design alternatives under the case study?”

Various scenarios are created with varying input parameters to examine the different design options in the case study. The study's results indicate that uncertainty greatly impacts the effectiveness of cross-docking operations, and the chosen staging level plays a significant role. For configurations that are not purely cross-docking, utilizing a two-stage process with pallets can reduce the total time needed

by 22% while improving the hourly throughput rate by 28%. Decisions regarding load carrier type can also improve cross-docking performance by reducing intra-terminal congestion.

For pure cross-docking configurations, Two-stage cross-docking with pallets is more effective than other methods, reducing makespan by 15%, truck waiting time by 11%, and total throughput by 23%. However, the study did not address congestion issues, resulting in weaker and biased results.

The study didn't focus on cost modeling, so estimated costs of alternatives are used. More roll containers are needed to transport the same parcels as Euro pallets, making operating costs 4.2 times higher. However, reducing worker hours could offset this cost. Two-stage cross-docking requires twice the space and increases CAPEX by 2.

Based on this case study conducted at Budbee's cross-docking terminal, the results of the simulation testing showed that cross-docking operations are significantly influenced by the designed staging level, particularly in uncertain conditions. Two-stage cross-docking with pallets was highly effective by reducing the total makespan by 22% and boosting the average throughput per hour by 28% in non-pure cross-docking configurations. Additionally, two-stage cross-docking with pallets in pure cross-docking configurations showed to reduce the total makespan by 15% and increase the total throughput by 23% while reducing the average truck waiting time by 11%.

A case study was conducted at a cross-docking terminal to evaluate staging design and consolidation methods. Both non-pure and pure cross-docking configurations were tested.

For non-pure scenarios, two-stage cross-docking with pallets significantly reduced makespan and increased throughput compared to alternatives. This demonstrates staging design can improve performance under uncertainty.

In pure cross-docking tests, two-stage with pallets also outperformed other options in makespan, waiting times, and throughput. However, congestion issues reduced successful simulations, so these results are considered weak. Specifically, the conceptual model lacked defined processes or rules to resolve deadlock situations caused by congestion. Without the capabilities to cope with deadlocks, the simulation failed to run properly under pure cross-docking conditions.

Operating costs were estimated based on parcel and vehicle capacities. Roll containers have higher acquisition costs but may reduce labor needs. Actual cost trade-offs require further analysis.

While specific improvements were quantified, trends likely apply directionally to other facilities but will vary based on parameters. Consolidation methods significantly impact congestion risks. Careful selection is needed based on cross-docking purity level.

Limitations include unresolved congestion in pure scenarios and single case study constraints. Areas for future research include congestion modeling, multi-case analyses, scheduling strategies, and comparisons under additional real-world complexities like mixed flows and due times.

By evaluating staging and consolidation designs under uncertainty, this study provides valuable insights into improving cross-docking operations across contexts. Further work can build on the simulation methods and findings.

This research has some limitations due to assumptions and simplifications made during the simulation testing. Future studies could focus on different areas, such as incorporating due times, developing advanced scheduling strategies, and analyzing individual makespans. Additionally, comparing roll containers with pallets in pure cross-docking scenarios and studying deadlock situations under different consolidation methods would provide valuable insights into improving cross-docking operations. By addressing these areas, we can enhance our understanding of cross-docking operations and improve logistics systems.

This research drew significant findings on cross-docking operations under uncertainty. The study was conducted through a case study at Budbee's cross-docking terminal to answer the main research question. The results show that the performance of cross-docking operations under uncertainty is significantly impacted by the designed staging level and consolidation method. Specifically, two-stage cross-docking with pallets can reduce the total makespan by 22% and enhance the average throughput per hour by 28%. Additionally, the consolidation methods considerably impact cross-docking performance due to their effects on intra-terminal congestion. However, the purity of cross-docking operations affects the number of intra-terminal transshipments, subsequently impacting the probabilities of intra-terminal congestion. Therefore, the consolidation method of choice depends on the purity of cross-docking.

The study on cross-docking operations is scientifically relevant as it fills part of the scientific gap of comparison of internal transportation methods. This research does so by conducting a pair-wise analysis of two load carriers (pallets & roll containers) in cross-docking operations for a logistics service provider. This new analysis stresses the importance of selecting the appropriate load carrier for logistical systems. Additionally, this research expands the available literature on the effects of staging-level design and consolidation methods. This research suggests that decisions in staging-level design depend on queuing theory factors that can differ for other CDF configurations.

Furthermore, the study identified several areas for future research that could enhance the performance of cross-docking operations even further. These include exploring the potential benefits of roll containers in reducing the total makespan in highly congested configurations, analyzing individual makespans within cross-docking operations to identify areas for improvement, investigating the behavior and performance of alternative consolidation methods under congestion scenarios, incorporating due times into cross-docking studies, and developing a framework or scale to measure cross-docking purity levels.

Based on these findings, the study provides valuable managerial insights for other logistics service providers. This research stressed the importance of considerations for staging level design, consolidation methods, and hybrid warehouse characteristics. Additionally, the model in this research showed the capability of simulation modeling to improve cross-docking operations' performance under uncertainty. By implementing these recommendations and insights, logistics service providers can ensure that their cross-docking operations are efficient, effective, and able to adapt to changing conditions.

Contents

1	Introduction	1
1.1	Background	1
1.2	Research questions	3
1.3	Research methodology.	3
1.3.1	Literature research	3
1.3.2	Case study	3
1.3.3	Modeling	4
1.3.4	Experimenting & data analysis.	4
1.4	Report structure	4
2	Literature research	5
2.1	Background of decisions in cross-docking	5
2.2	Pure cross-docking versus warehousing	8
2.3	Mitigating uncertainty.	8
2.3.1	Cross-dock terminal design	8
2.3.2	Cross-dock terminal planning	10
2.3.3	Cross-dock terminal scheduling	10
2.3.4	Uncertainty in cross-docking.	11
2.4	Modeling methods under uncertainty	12
2.5	Measuring performance of cross-docking operations.	13
2.6	Potential design alternatives	15
2.7	Conclusion literature review	15
3	System analysis: Case Study	17
3.1	Layout and facility components cross-docking terminal	17
3.2	Resources and load carrier types	19
3.3	Activities and processes in cross-docking.	20
3.3.1	IDEF0 A1-level representation.	20
3.3.2	Receiving	21
3.3.3	Unloading trucks and moving pallets	21
3.3.4	Loading	22
3.3.5	Dispatching	22
3.3.6	IDEF0 A0-level representation.	22
3.4	System uncertainties and occurring problems	23
3.4.1	Transported volumes.	23
3.4.2	Arrival and departure times	25
3.4.3	Dock door availability.	29
3.4.4	Internal handling time	29
3.5	Current performance	30
3.6	Design alternatives identification	32
3.7	Conclusion problem description	33
4	Model formulation	35
4.1	Model goal and scope	35
4.1.1	Network versus local scope	35
4.1.2	Layout and processes	36
4.1.3	Modeling method: simulation	37

4.2	Theories and alternatives design	38
4.2.1	Alternative design 0: Keep as is	38
4.2.2	Alternative design 1: Two-stage cross-docking	38
4.2.3	Alternative design 2: Two-stage cross-docking with roll containers	39
4.3	Requirements and assumptions	41
4.3.1	Requirements	41
4.3.2	Assumptions	42
4.4	Key performance indicators	42
4.5	Implementation and specifications	42
4.5.1	Model implementation	42
4.5.2	Process step 1: Generating demand in trucks at regional terminals	43
4.5.3	Process step 2: Truck arrival at unloading dock	46
4.5.4	Process step 3: Unloading and staging procedure	47
4.5.5	Process step 4: Reloading truck and dispatching	50
4.6	Verification & validation	52
4.6.1	Verification	52
4.6.2	Calibration	54
4.6.3	Data validation	54
4.6.4	Conceptual model validation	55
4.6.5	White-box validation	55
4.6.6	Black-box validation	55
4.6.7	Conclusion verification & validation	56
4.7	Conclusion model formulation	57
5	Experimentation and results	58
5.1	Scenario definition	58
5.2	Design alternative specification	59
5.3	Experimental design	60
5.4	Results	61
5.4.1	Non-pure CD configurations	64
5.4.2	Pure CD configurations	66
5.5	Discussion	66
6	Conclusion	69
6.1	Key findings and research question	69
6.2	Scientific relevance	71
6.3	Recommendations for future research	71
6.4	Managerial insights	72
	Bibliography	73
A	Case description	77
A.1	Case introduction	77
A.2	Network	77
A.3	Numerical analysis	78
A.3.1	Number of total and cross-docked unit loads	78
A.4	Conclusion and scope system	80
A.5	Goals and needs	81
A.6	Conclusion	81
A.7	Logic flow	81
B	Simulation model	83
B.1	Entity generation	85
B.2	Arrival and unloading	90
B.3	Staging and internal transshipment	91
B.4	Reloading and Dispatching	93
B.5	Model trace	95

C	Data analysis	121
C.1	Input parameter fitting	121
C.1.1	Fitting distributions of the total pallets cross-docked	125
C.1.2	Fitting distributions of the arrival times	128
C.2	Analysis empirical and simulation data: Student's T-Test with Bootstrapping and Chi-Squared Test	131
D	Results	138
D.1	Box plots	138
D.2	Pair-wise Student's T-Testing	141
D.3	Central Limit Theorem	144
E	Research paper	145

List of Figures

1.1	Supply chain network example	2
2.1	Schematic representation of a cross-docking terminal [12]	6
2.2	Cross-docking problem classes [14]	6
2.3	Popular cross-dock shapes [9]	9
2.4	Staging-levels in cross-docking	10
3.1	Floor plan terminal UTR	18
3.2	Available resources and freight carrier types	19
3.3	IDEF0 A1-level representation of the case study	21
3.4	IDEF0 A0-level representation of the case study	23
3.5	Distribution of actual arrival and departure times	27
3.6	Distribution of difference in expected and actual arrival and departure times	28
3.7	Box plot of the difference in ETAs and ATAs in the data set	28
3.8	Correlation and regression of total unit loads and total makespan	30
3.9	Total cross-docking times	31
3.10	Makespan cross-docking operations	32
4.1	Local scope within network scope	36
4.2	System scope and boundaries CDF	37
4.3	Alternative 1: Two-stage cross-docking	39
4.4	Average cross-docking time per Full Truckload (FTL) per mode over distance	40
4.5	Alternative 2: Combination of reallocating resources and roll container usage	41
4.6	Object definition in SIMIO [56]	43
4.7	Destination assignment logic	45
4.8	Step 1: Generation process of unit loads and trucks	46
4.9	Step 2: Truck-to-dock assignment	47
4.10	Step 3: Decision process Single-stage and Two-stage unloading procedure	49
4.11	Full overview simulation model	51
4.12	Verification and validation in the modelling process [51]	52
4.13	Unit loads flow conservation verification (end of verification run)	53
4.14	Time conservation (end of verification run)	53
4.15	Zero demand input flow conservation (end of verification run)	54
4.16	Black-box validation [51]	55
4.17	Histograms of empirical data and simulation generated base case data (copy from section C.2)	56
4.18	Scatter plot empirical data and simulation model data	56
5.1	Scenario definition from IDEF0 A-0 level	58
5.2	Scatter plots of results for non-pure CD	64
5.3	Histograms of total makespan for A0, A1, and A2 for non-pure CD	64
5.4	Histograms of total makespan for three experiments under varying width arrival time window	65
5.5	Histograms of total makespan for three experiments under varying cross-docking level	66
A.1	Schematic representation of the hub-spoke network in the middle-mile	78
A.2	Histogram of pallets handled in cross-docking	79
A.3	Box plot of pallets handled in cross-docking	79
A.4	Box plot of the generated pallet demand levels per origin in the data set	80

A.5	Logic flow diagram	82
B.1	Simulation model overview in SIMIO	84
B.2	Entity and transporter visualization	85
B.3	Entity generation	85
B.4	Process step: AssignOriginDestination	87
B.5	Process step: AssignOrigin	88
B.6	Process step: AssignDestination	89
B.7	Unloading dock object and inbound staging area object	90
B.8	Process Step: RoutingToStagingArea	91
B.9	Terminal Pathway Roundabout	92
B.10	Loading Dock	93
C.1	Total number of pallets cross-docked per day (before outlier removal)	126
C.2	Total number of pallets cross-docked per day (after outlier removal)	126
C.3	Best fitting distributions to the total number of pallets cross-docked per day	127
C.4	Histogram ATA before outlier removal	128
C.5	Histogram ATA after outlier removal	129
C.6	Best fitting distributions on Actual Times of Arrival	129
C.7	Histograms of empirical data and simulation generated base case data	136
C.8	Box plots of empirical data and simulation generated base case data	136
C.9	Bootstrap plots of empirical data and simulation generated base case data	136
D.1	Box plots Total Makespan	139
D.2	Box plots Average Waiting Time Per Truck	140
D.3	Margin of error for sample sizes	144

List of Tables

2.1	Characteristics of other studies versus this study	14
3.1	Available staging area	18
3.2	Assigned dock doors	18
3.3	Intra-terminal distances [meters]	19
3.4	Number of line hauls per origin	24
3.5	Origin-destination matrix of scanned parcels for three months	24
3.6	Three-month average distribution per origin and destination	25
3.7	Departure and arrival times trucks at terminal	26
3.8	Example of data on arrival times trucks	26
3.9	Descriptive statistics empirical data	27
3.10	Box plot values of the difference in ETAs and ATAs in the data set	28
4.1	Hypothetical performance per consolidation method and mode	39
4.2	Key Performance Indicators (KPIs) in this research	42
4.3	Unit load properties	44
4.4	Truck properties	44
4.5	Unit Load Source properties	44
4.6	Truck Source properties	44
4.7	Demand volume distribution input parameters	45
4.8	Truck-to-dock assignment	47
4.9	Unloading dock specification	47
4.10	EPT specification	48
4.11	Worker specification	48
4.12	EPT/Worker to door assignment	49
4.13	Inbound staging area assignment	49
4.14	Inbound staging area specification	50
4.15	Reloading dock specification	50
4.16	Parameter calibration	54
5.1	Input parameter specification	59
5.2	Experimental Design	60
5.3	Computer characteristics	61
5.4	Number of successful replications per scenario for the alternatives	61
5.5	Results	62
5.6	Change in KPIs the for non-pure cross-docking scenarios relative to A0 (the Base Case)	63
5.7	T-test values of Total Makespan for A0 (Current State) versus A1 (Two-Stage Pallets)	65
A.1	Box plot values of the pallet demand levels in the data set	80
B.1	Properties Forward unit load Source	86
B.2	Properties Return unit load Source	86
B.3	Properties Truck Source	86
B.4	DataTable: VolumeGeneration	87
B.5	Probability tables	89
B.6	Properties Regional Terminal Combiner	89
B.7	Properties Travel Time Path	90
B.8	Properties Terminal Arrival Node	90
B.9	Properties Unloading Dock	91

B.10 OD matrix door-to-door distance [meters]	92
B.11 OD matrix Inbound Staging Area (ISA) to Door [meters]	92
B.12 Properties Reloading Dock	94
B.13 Model trace verification	96
C.1 Weibull Minimum Extreme Value Distribution Parameters	127
C.2 JohnsonSU Distribution Parameters	127
C.3 Fisk Distribution Parameters	128
C.4 Fisk Distribution Parameters	130
C.5 Lognormal Distribution Parameters	130
C.6 Normal Distribution Parameters	130
C.7 Student's T-Test Results	137
C.8 Chi-squared Goodness of Fit Test Results	137
D.1 Pair-wise Student's T-testing of the Total Makespan for the experiments per scenario	143

Introduction

This chapter introduces the subject under study by providing background information in section 1.1. After introducing the subject of interest, the research questions are presented in section 1.2. Next, the methodology for answering the main research questions is elaborated in section 1.3. Finally, the report structure is addressed in section 1.4.

1.1. Background

E-commerce sales have increased substantially in recent years. Statista reported a 290% growth in worldwide retail e-commerce sales between 2014 and 2020 [16]. Chevalier [16] forecasts this figure to grow by 56% over the following years, accumulating to about 8.1 trillion dollars by 2026. Accompanying this growth in e-commerce sales is a product return rate of 30% [19]. Many e-commerce businesses saw a steep increase in product returns regarding the online shopping boom during the Covid-19 pandemic [62]. These reverse logistics add complexity to already complex supply chains.

To ensure product delivery by e-commerce merchants, logistical companies have built extensive supply chain networks consisting of warehouses, sorting terminals, transportation modes, and cross-docking terminals. The increase in e-commerce sales and the transported volume subsequently requires logistics companies to expand their assets and increase efficiency in their operations. Zhang, Atasoy, and Negenborn [67] mention that synchronization is key for using available resources optimally and optimizing efficiencies in logistics. Especially in cross-docking operations, operations must be carefully synchronized due to an absent or limited storage buffer inside a cross-dock terminal [14]. Figure 1.1 shows an example of a supply chain network, including forward and return flow. This example emphasizes the crucial role of coordination at the cross-dock for supply chain performance.

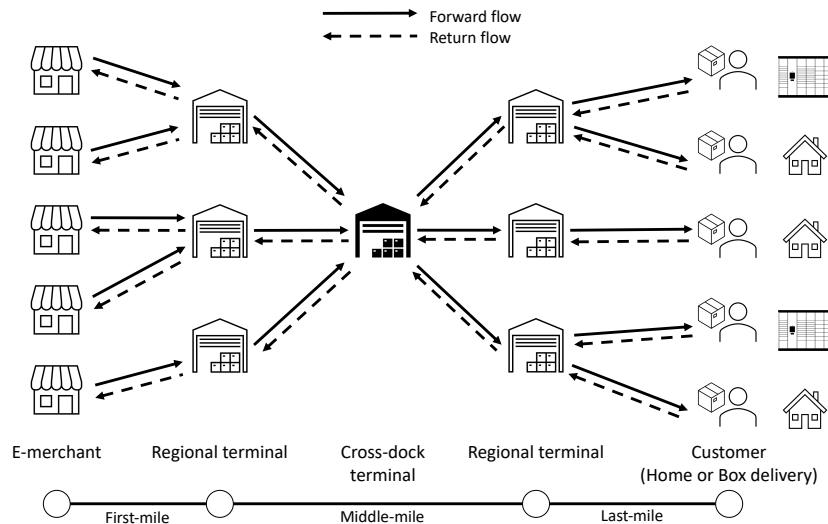


Figure 1.1: Supply chain network example

A cross-docking terminal is designed to transfer goods directly from inbound trucks to outbound trucks with as little intermediate storage as possible to achieve “just-in-time” (JIT) logistics [32]. Ideally, the inbound and outbound trucks are all available and docked simultaneously to start transferring loads immediately and minimize the trucks’ total time at the cross-dock door without temporarily storing goods. However, this is rarely the case since trucks can be delayed or early, and the number of dock doors is often smaller than the number of trucks, resulting in waiting times, delays, and loss of efficiency [30].

Gallo et al. [22] underlines that synchronization between inbound and outbound operations in cross-docking leads to cost reduction and increases in on-time deliveries. Still, cross-docking operations are sensitive to uncertainties such as truck arrival times, number of products, and other external or internal factors.

As e-commerce sales and transported volume increase, cross-docking terminals are put under pressure, and uncertain arrival times of trucks undermine the synchronization of complex cross-docking operations. Therefore, this study aims to reduce the total operation time and increase the throughput of cross-docking facilities (CDFs) by mitigating uncertainties in cross-docking operations. To this end, research will be conducted using a case study of the cross-docking terminal operated by Budbee, a parcel delivery company based in Sweden. A detailed system analysis will be provided in chapter 3. The case company has a network of regional terminals and cross-docking facilities for parcel distribution.

The cross-docking terminal under study is located in Utrecht, Netherlands, and serves as a consolidation hub for 13 regional terminals. It has a layout of 10 dock doors on one side in an I-shape. The terminal has a surface area of around 700 m².

Some key characteristics of the cross-docking operations are:

- Parcel freight is consolidated onto pallets
- Mixed-mode dock doors are used for both unloading and loading
- Volumes fluctuate and cause uncertainty in incoming freight
- Most volume originates from the Utrecht terminal itself (around 70%)
- Arrival times are uncertain, with trucks arriving outside the scheduled window
- Staging areas of varying capacities are assigned to each dock door
- Main resources are electric pallet trucks, manual trucks, and workers
- Lack of monitoring and data on arriving freight and truck status

The cross-docking terminal functions more as a hybrid facility rather than a pure cross-dock. The primary uncertainties are in freight volumes and truck arrival times. The key metrics the company uses are total makespan and throughput. This case study provides insights into improving operations under uncertainty for a real-world parcel delivery cross-dock. Both external and internal uncertainty factors characterize the CDF of Budbee. External factors are mainly the truck arrival times and incoming freight, whereas the main internal uncertainty factor is the internal transportation time.

Now that the background is addressed, the research questions can be formulated in the next section.

1.2. Research questions

This research aims to improve the performance of cross-docking operations under uncertainty. However, costs are also an important consideration. One main research question and several sub-questions are composed to structure this research objective. Özkaraca [42] states that quantitative research usually follows the following steps: system definition, parameter determination, objective formulation, constraint identification, model creation, and solving. The next main research question and sub-questions are constructed following this structure line.

“How to improve cross-docking operations under uncertainty?”

1. *What is a cross-docking terminal, and how can uncertainty be mitigated in literature?*
2. *What is the current state of cross-docking at the CDF of Budbee?*
3. *How to develop a model for testing the design alternatives for a cross-docking facility under uncertainty?*
4. *What are the effects of different design alternatives under the case study?*

As the research questions are identified, the next section elaborates on the methodology for answering the research questions.

1.3. Research methodology

Different research methods are used to answer the main research question and sub-questions. The methods used are described in the following subsections.

1.3.1. Literature research

Literature research is performed to comprehend the current state-of-the-art cross-docking operations under uncertainty fully. Knowledge obtained from this literature research forms the basis for expanding this research. This knowledge is used for theories, hypotheses, and designs to help answer the main research question and apply it to the problem. The literature research will be provided in chapter 2.

First, the research is on the design of cross-docking operations and to what extent uncertainty is accounted for in cross-docking. Next, measuring the performance of cross-docking operations is assessed. Third, reverse logistics' influence on cross-docking operations' performance is examined. Fourth, literature on the modeling methods for researching cross-docking under uncertainty is searched for. Fifth, potential design alternatives for improving cross-docking performance under uncertainty are identified. Finally, suitable modeling methods are identified.

1.3.2. Case study

By examining a real-world case of a cross-docking facility, the current state of cross-docking activities, processes, and factors causing uncertainty can be assessed. The case study acts as a real-world example of a complex problem. This case study will be held at the CDF of Budbee, which can provide useful insights for other similar logistical service providers. A case study has limitations since a model is tested to one case only [27]. However, due to limited time, only the case of one logistic service provider will be examined in this research.

The case study will be analysed in chapter 3. Questions related to the case study are on the actual layout of a cross-docking terminal and the activities and processes during cross-docking. The resources used, the uncertainties in the problem under study, and the associated problems are also identified. Finally, the current performance is analyzed and compared to the desired performance.

1.3.3. Modeling

After researching the literature and identifying a problem from the case under study, the extracted theories, hypotheses, and designs can be applied to the problem. By modeling the problem and implementing the designs, the results of potential design alternatives can be assessed and compared to real-world testing. The advantages of model testing are less time, cost, and risk.

Modeling the problem can provide us by identifying relevant interdependencies and relevant uncertainties. This will lead to scoping of the cross-docking system under study and a working model for the base case. Having identified the model's scope, the applicable design alternatives are discussed. Based on the design alternatives, the requirement of the model and the assumptions made can be stated. Next, the base case model is verified and validated to ensure the functionality of the model.

1.3.4. Experimenting & data analysis

After modeling, successfully verifying, and validating the model, experimentation is conducted with the design alternatives and scenarios for which the results are presented. Additionally, numerical tests and data analysis will help answer the research question and provide useful managerial insights for recommendations.

1.4. Report structure

This thesis will continue by providing the literature research in chapter 2 to identify the state-of-the-art of improving cross-docking operations under uncertainty and identify potential solution methods, theories, and hypotheses to construct design alternatives.

Next, the cross-docking system under study is introduced in chapter 3 and the problem definition.

After reviewing the literature on uncertainty in cross-docking operations, the model of the system under study is formulated and presented in chapter 4. This chapter also includes verification and validation to analyze the model's behavior.

Next, the obtained results are presented and discussed in chapter 5.

Finally, the thesis is concluded, and recommendations for future research and the following managerial insights are provided in chapter 6.

2

Literature research

This chapter presents an extensive literature review on improving cross-docking performance under uncertainty. This knowledge will validate the design choices made in this research. The topics will be elaborated on in terms of definition, previous research conducted, and future research recommendations to pinpoint the design alternatives.

This chapter starts with background information on decision-making in cross-docking in section 2.1. Next, designs and methods to mitigate uncertainty in cross-docking operations are discussed in section 2.3. Additionally, methods for modeling uncertainty in cross-docking operations are studied in section 2.4. Moreover, measures for monitoring performance in cross-docking are identified in section 2.5. Then, potential design alternatives are identified in section 2.6 using the information gathered in this chapter. Finally, the literature review is concluded in section 2.7.

2.1. Background of decisions in cross-docking

Cross-docking became a logistics strategy to minimize handling or storage time and related costs in a distribution center, where goods delivered by inbound trucks are immediately reloaded onto outbound trucks [68]. In retail, cross-docking entails sorting inbound goods, consolidating them, and storing them until the outbound shipments are ready to leave [66]. A cross-dock is like a warehouse. However, the unloaded loads are held for less than 24 hours before leaving the cross-dock with minimal handling operations [39, 60]. This additional transshipment caused by cross-docking slows the distribution process down compared to point-to-point deliveries [12]. However, economies of scale in transportation costs can be realized with the use of cross-docking by enabling consolidation of Less Than Truckload (LTL) with the same destination to FTL shipments [5]. An example of a cross-docking terminal is shown in Figure 2.1.

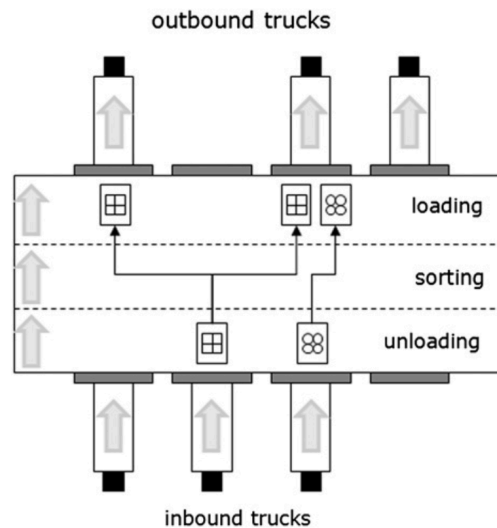


Figure 2.1: Schematic representation of a cross-docking terminal [12]

The example of a cross-docking terminal in Figure 2.1 depicts an inbound side and an outbound side with dedicated inbound or outbound doors. This results in a one-way flow through the cross-docking terminal. However, mixed-mode service doors are also possible, which are used for both inbound and outbound. Similarly, Figure 2.1 implicitly regards two sets of trucks: inbound and outbound. In reality, however, individual trucks can be used for both inbound and outbound operations. These are called mixed-mode trucks. Additionally, temporary storage at the cross-dock's staging area is possible but not depicted in Figure 2.1.

Buijs, Vis, and Carlo [14] state that problems in cross-docking occur on strategic, tactical, and operational levels and the problems can be considered locally or within a network. Strategic-level decisions involve long-term planning and focus on the overall goals and objectives of the cross-docking operation. Tactical-level decisions involve medium-term planning and focus on coordinating different processes and activities within the cross-docking operation. Operational-level decisions involve short-term planning and focus on the execution of day-to-day activities within the cross-docking operation. Buijs, Vis, and Carlo [14] therefore propose the following six cross-docking problem classes as shown in Figure 2.2, which are elaborated on further.

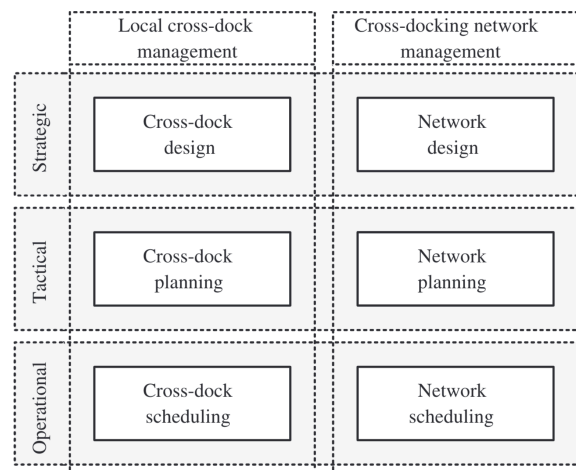


Figure 2.2: Cross-docking problem classes [14]

The first problem class is *cross-dock design*. Cross-dock design decisions specify the internal structure and the contour of the cross-dock. The aim is to provide sufficient capacity for the throughput

requirements and enable fast transshipment. Cross-dock design decisions include, for example, the number of dock doors, internal configurations of sorting areas, and temporary storage space [14].

The second problem class is *cross-dock planning*. Cross-dock planning decisions aim to minimize the time or handling effort to transship goods from inbound to outbound doors, which are determined by dock door assignment [14]. Buijs, Vis, and Carlo [14] mentions that cross-docks that serve a fixed set of origins and destinations with relatively constant volumes tend to assign dock doors for 3 to 6 months. Cross-dock planning also determines the workforce and equipment needed to efficiently handle the freight volume within the limited time available during cross-docking [14].

The third problem class is *cross-dock scheduling*. Cross-dock scheduling decisions allocate resources within the cross-dock over time. In contrast to assigning dock doors, truck scheduling decisions often only consider situations where the number of trucks far exceeds the number of available dock doors [14]. Trucks are scheduled before the start of operations but can also alter dynamically during ongoing operations if uncertainties develop. Internal workforce and utilization of storage locations have to be scheduled as well.

The fourth problem class is *network design*. Network design decisions determine the network structure of possible facility locations and routes to transport the freight volume [14]. The required number of facilities in the cross-docking network is determined based on the expected demand and the network structure. Also, choices on outsourcing are made at this strategic level [14].

The fifth problem class is *network planning*. Network planning determines freight flow throughout the network by assigning volumes to routes. This also determines whether opportunities for consolidation at a cross-dock occur [14].

The sixth problem class is *network scheduling*. Network scheduling determines decisions on planning concerning capacity and time window constraints, such as vehicle routing to collect and deliver shipments from and to a cross-dock [14].

Boysen and Flidner [12] listed the decision problems relevant to a cross-docking terminal in order of long-term strategic problems to short-term operational problems as follows:

- Location of cross-docking terminal(s)
- Layout of the cross-dock
- Assignment of destinations to dock doors
- Vehicle routing
- Truck scheduling
- Resource scheduling inside the terminal
- (Un-)Packing loads into (from) trucks

Mavi et al. [37] mention that several models in cross-docking research studies aim to solve different problems and improve cross-docking operations such as scheduling, dock door assignment, transshipment, vehicle routing, product allocation, and layout design and network. Both optimization and simulation are used extensively in cross-docking. Mavi et al. [37] also mention that in cross-docking operations, often Warehouse Management Systems (WMSs) are present designed to address inventory and data visibility, traceability, supply chain agility, and operational adaptability.

Gallo et al. [22] developed a stochastic genetic optimization algorithm based on a scenarios tree for scheduling cross-docking operations under uncertainty. Their research underlines that synchronization between inbound and outbound operations leads to cost reduction and increases in on-time deliveries, but the operations are sensitive to uncertain truck arrivals. Boysen and Flidner [12] mentions that synchronization of inbound and outbound trucks is required to ensure on-time deliveries. This so-called truck scheduling problem decides on the sequencing of inbound and outbound trucks for a set of dock doors at the cross-docking terminal [12]. Kuo [30] mentions that, usually, the number of inbound trucks and outbound trucks is greater than the number of inbound doors and outbound doors. Moreover, a fraction of the inbound and outbound trucks can be unloaded and loaded simultaneously, and the remaining inbound and outbound trucks need to wait. If a product is on an inbound truck waiting in the queue, but its corresponding outbound truck is ready for loading,

delays will occur. The outbound truck has to wait and cannot be replaced by other outbound trucks, and the efficiency of cross-docking is reduced. As a result, the sequence of all inbound and outbound trucks in the queue can influence the efficiency of cross-docking. Bodnar, De Koster, and Azadeh [11] mention that optimization in cross-docking mainly includes decisions on truck scheduling, assigning trucks to docks, and assigning products to locations.

2.2. Pure cross-docking versus warehousing

Apte and Viswanathan [5] mention that there is agreement on a continuum of cross-docking from "pure" to "short warehouse time". In a pure cross-docking facility, products do not enter the WMS, and all labeling and packing activities are already completed before the item enters the warehouse. Warehousing, on the other hand, mainly focuses on the storage process of goods [18].

Although not mentioned in the literature, the position on the continuum of cross-docking impacts the design of cross-docking facilities by the share of unit loads pre-injected and subsequently determines the number of unit loads handled in cross-docking. Due to this vague continuum in cross-docking, designs of cross-docking facilities differ greatly. The position on the continuum of cross-docking can be called the *cross-docking purity level*.

2.3. Mitigating uncertainty

To fully comprehend the current state-of-the-art in mitigating uncertainties in cross-docking, the decision levels in local cross-dock management as shown in Figure 2.2 are studied. First, the design decisions of cross-docking terminals are studied in subsection 2.3.1. Next, the planning decisions in cross-docking are discussed in subsection 2.3.2. Third, the scheduling problems arising in cross-docking are discussed in subsection 2.3.2. Finally, factors of uncertainty in cross-docking and their effects on the performance are discussed in subsection 2.3.4.

2.3.1. Cross-dock terminal design

As seen in Figure 2.2, cross-dock terminal design decisions are of a strategic level. Designing a cross-dock facility entails many aspects. Following Boysen and Fliedner [12], after determining the locations of the cross-docking terminal, the layout of the terminal is designed. The internal and external dimensions, shapes, and arrangements are part of the cross-dock terminal layout [59]. Van Belle, Valckenaers, and Cattrysse [59] distinguish three types of characteristics: physical characteristics, operational characteristics, and flow characteristics, which will be assessed in this paragraph respectively.

Physical characteristics The physical characteristics of cross-docking terminals are the shape, the number of dock doors, and the internal transportation method [59]. Additionally, sizes of staging areas can be considered a physical characteristic as well.

Internal transportation methods are often determined by the type of load carrier used in transportation. For example, pallets can only be moved using terminal transportation equipment (manual, electric, or automatic).

The shape of a cross-docking terminal is described by the letter corresponding to the shape (I, L, U, T, H, E, X, ...). Van Belle, Valckenaers, and Cattrysse [59] mention that I-shaped cross-docks are more efficient for smaller cross-docks (less than 150 doors). For intermediate-sized cross-docks, a T-shape is best, and for more than 200 doors, an X-shape is considered best due to the proximity values of the dock doors [59]. Examples of cross-dock shapes are shown in Figure 2.3.

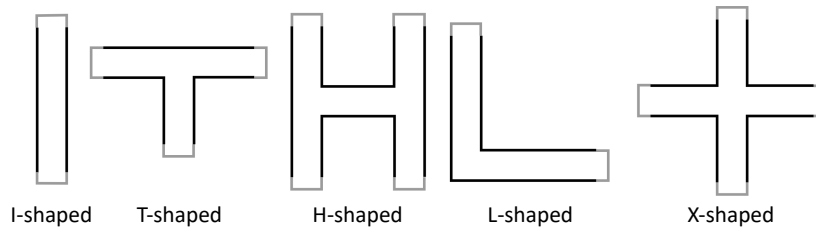


Figure 2.3: Popular cross-dock shapes [9]

Regarding the fact that Bartholdi and Gue [9] varied the dock size in their study from 40 to 350 doors and a terminal with 100 to 150 doors is considered a medium-sized facility, the cross-docking terminal under study is considered fairly small. Due to this relatively small terminal size, one could argue that the intra-facility distance between the dock doors and the dock doors and staging spaces is small enough for intra-terminal transshipment time to be only proportional to the number of pallets moved instead of also being proportional to distances covered. Design choices in the layout of the terminal, such as the number and positioning of dock doors, the number and locations of staging areas, and sorting machines, influence internal transshipment times. Department [20] reports that the rent of a warehouse is 60-73 euros per square meter.

Mulcahy [40] states that the number of cross-docks required can be calculated using Equation 2.1.

$$D = N \times \frac{T_{\text{load}}}{T_{\text{work}}} \times SF \quad (2.1)$$

D	No. dock positions needed
N	No. trucks per day
T_{load}	Hours needed for loading/unloading
T_{work}	Work hours per day
SF	Safety factor of 25%

Additionally, internal cross-dock transportation can be executed either manually (workers with forklifts), automated (conveyor belts), or a combination depending on the type of freight [59].

Operational characteristics Operational characteristics also influence the cross-dock design and thus the performance. Operational characteristics include the number of dock doors and their service mode, the number of trucks to be served, and the allowance of preemption.

The service mode of the dock doors determines the degrees of freedom in assigning trucks to dock doors [12]. In exclusive mode, the doors are dedicated to inbound or outbound trucks only. Whereas in mixed mode, both inbound and outbound trucks can be serviced at the same door [59]. A combination of mixed and exclusive dock doors is also possible.

Additionally, interruption of (un)loading trucks can be allowed. This so-called preemption allows another truck to replace an already docked unfinished truck. This unfinished truck then has to be docked later to finish (un)loading [59].

Flow characteristics Flow characteristics of the processed goods can differ too. Flow characteristics include arrival patterns, departure patterns, and the type of freight which can determine the storage requirements or other processes.

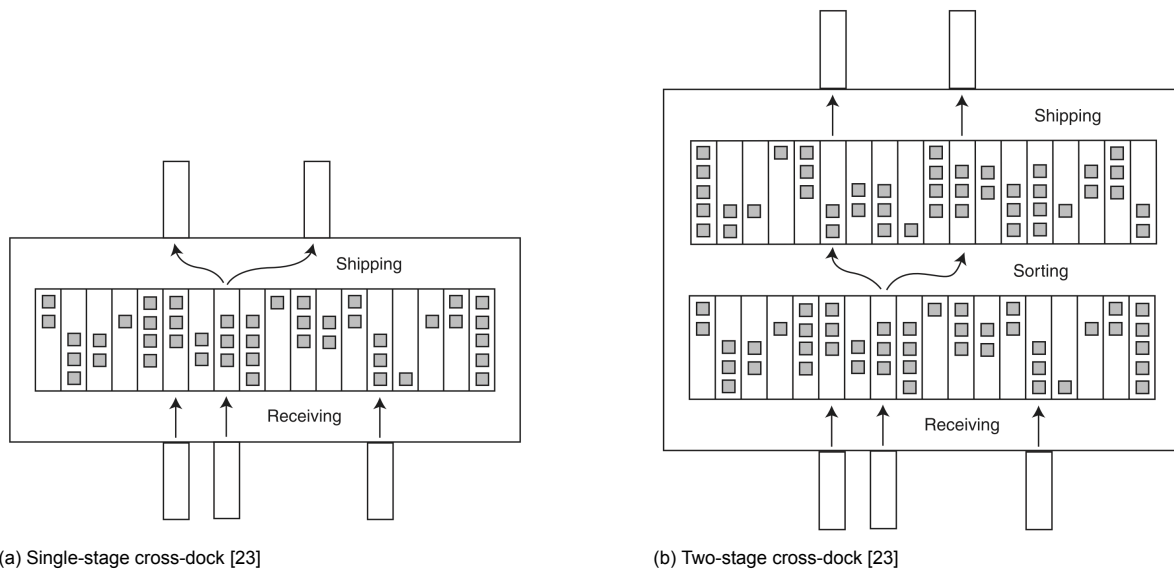
The arrival times of the inbound trucks determine the arrival pattern of the goods. Van Belle, Valckenaers, and Cattrysse [59] mention that usually, the arrival patterns are concentrated at one or more periods. These arrival patterns influence cross-dock congestion as well as worker scheduling.

Departure times of the outbound trucks can be restricted in case the truck has to arrive at a certain location before a certain time. Sometimes, the departure times of trucks are not restricted, and the trucks leave after all freight is cross-docked.

Temporary storage is unnecessary when all arriving freight can immediately be transferred to the outbound trucks. However, Van Belle, Valckenaers, and Cattrysse [59] mention that this is rarely the case, and goods are generally stored on the terminal floor. The designated floor storage area is called the staging area. Sometimes goods cannot be stored temporarily due to their freight characteristics. For example, refrigerated products need cooled storage which often is not possible in the staging area.

Distribution level design & staging level design A cross-dock can be designed in two additional ways [8]. First, a cross-dock can be either a pre-distribution or a post-distribution cross-dock. In pre-distribution, the pallets are sorted and labeled upon arrival at the cross-dock. In post-distribution, the cross-docking facility must sort the freight of the pallets and assign destinations to the shipments. Yang, Balakrishnan, and Cheng [65] mention that post-distribution requires more floor space than pre-distribution.

Secondly, cross-docking operations can differ in staging methods. In a single-stage cross-dock, the pallets are directly staged into areas of their destination. A two-stage CDF allows the re-sorting of pallets, but more floor space is needed as well as additional travel time. A free-staging cross-dock is also possible, where not any queues are used to place pallets at one end and pull from the other. Gue and Kang [23] show that a similar single-stage cross-dock facility produces higher throughput and productivity than a two-stage cross-dock. Additionally, Gue and Kang [23] mention that using flow racks over floor storage produces too little benefit. According to Yang, Balakrishnan, and Cheng [65] additional handling by staging freight at the doors reduces congestion due to the difference in time needed for (un)loading and time needed for internal transshipment. Therefore, less handling is not always considered to be better. Figure 2.4 depicts single-stage and two-stage cross-docking.



(a) Single-stage cross-dock [23]

(b) Two-stage cross-dock [23]

Figure 2.4: Staging-levels in cross-docking

2.3.2. Cross-dock terminal planning

As seen in Figure 2.2, terminal planning is of a tactical level. Truck-to-dock assignment is a planning decision and entails the problem of assigning inbound and/or outbound trucks to docks [11]. Trucks are often assigned to dock doors for long periods and alter over a medium time horizon. Zhou, Lei, and Zong [68] mention that considering both the truck-to-dock assignment and truck scheduling problem at the same door is established to minimize the cost during the inner transportation procedure and the truck waiting process.

Since both inbound and outbound trucks can be assigned to the docks, it is possible for inbound trucks to follow a different assignment than the outbound trucks.

2.3.3. Cross-dock terminal scheduling

Cross-dock terminal scheduling problems are short-term operational decisions. These problems include truck scheduling and product-to-location assignment, which are assessed in this subsection.

The truck scheduling problem entails the assignment of inbound and outbound trucks to the different dock doors at a cross-dock [60]. Gunawan et al. [24] describe a cross-dock as a transshipment facility in a Vehicle Routing Problem that enables the consolidation process of products from suppliers to customers or retail chains with marginal handling or storage time. Cross-dock is a possible source of

cost savings for companies as it eliminates long origin-to-destination paths and reduces the fleet size compared to direct shipment [24].

Boysen and Fliedner [12] studied the scheduling process of inbound trucks, in which the outbound trucks depart at a fixed, predetermined time. The internal transshipment time was also considered. In this research, the focus was only on scheduling inbound trucks instead of both inbound and outbound trucks. Serrano, Delorme, and Dolgui [54] considered a MILP model for scheduling inbound trucks at a cross-dock with capacitated temporary storage zones minimizing penalty costs due to an unbalanced workload. Shakeri et al. [55] based their truck scheduling problem on a multi-door cross-dock situation constrained by a limited number of forklifts and developed two metaheuristics to minimize the total schedule length. The scheduled length is also referred to as the makespan [55]. Bodnar, De Koster, and Azadeh [11] considered the problem of scheduling inbound and outbound trucks subject to time windows at a multi-door cross-dock with temporary storage. In this research, the dock doors handle both inbound and outbound trucks. Bodnar, De Koster, and Azadeh [11] also mention that solving a truck scheduling problem is notoriously hard since the problem is $\mathcal{NP}$ -hard and that future research should be more focused on either the stochastic nature of the problem or simultaneous objectives such as the scheduling and assignment of the trucks.

Zhou, Lei, and Zong [68] proposed a multi-door cross-dock truck scheduling problem in which the inner transportation cost in the cross-dock and the total truck waiting cost are minimized. This research introduced product-related time window constraints and a temporary storage area. Zhou, Lei, and Zong [68] mention that this research can be extended by considering uncertain arrival times.

Wang and Regan [63] mentions that time-based policies outperform First Come, First Serve (FCFS) and look-ahead policies in reliability and productivity when cross-docking. Some research considered truck-to-dock assignment solely by heuristics and policies such as 'First Come, First Serve' [65, 13].

Product-to-location assignment emerges when the arriving goods cannot immediately be transferred to the outbound truck [11]. Bodnar, De Koster, and Azadeh [11] mention that some models assume an infinite storage capacity, whereas capacitated temporary storage location is more realistic. For temporary storage at the cross-dock door, the product-to-location assignment problem arises too.

Vis and Roodbergen [61] consider handling cost in cross-docking for determining the short-term storage locations to minimize the total travel distance within the facility, assuming prior truck scheduling and truck-to-dock assignment.

2.3.4. Uncertainty in cross-docking

Uncertainty factors in cross-docking operations can be categorized into external and internal factors. According to Ardakani and Fei [6], demand is one of the main external uncertainty factors in supply chains. Yang, Balakrishnan, and Cheng [65] show that pre-distribution is preferred when demand is stable. However, when demand is uncertain, post-distribution is preferred. Since suppliers have performed all necessary preparation, pre-distribution has less impact on cross-docking operations, while in post-distribution, higher operational costs are induced since preparation happens inside the cross-docking facility. Arnaout et al. [7] show that truck utilization can be decreased by using larger trucks, but also assumed that cross-docking centers have infinite space, which is not true in practice.

Supply is also considered an external uncertain factor in supply chains [6]. However, supply is out of scope for a 4PL as the 4PL does not rely on producing and manufacturing processes but only transports the demand handed to them.

Arrival time is often considered one of the most critical external factors impacting the planning and scheduling of cross-dock facilities [12, 32]. Konur and Golias [28] point out that monitoring the arrival times of trucks can improve scheduling and update schedules when arrival times change.

Internal factors are considered to be processing time, available resources, and departure times [63, 6]. Yang, Balakrishnan, and Cheng [65] also assessed uncertainties in travel time due to pathway blocking of pallet trucks in a cross-docking terminal. Additionally, Yang, Balakrishnan, and Cheng [65] assumed that pallet trucks were assigned to individual unloading doors and found that the number of pallet trucks available limits unloading. In contrast, reloading is more uncertain since other pallet trucks may visit the same shipping door simultaneously.

Ardakani and Fei [6] state that the departure time of trucks mainly relies on arrival time, truck processing time, and availability of resources inside the cross-docking facilities. However, existing

work mainly considers the impact of demand uncertainty on the network but leaves the effect of internal cross-docking operations unaddressed.

2.4. Modeling methods under uncertainty

Nasiri et al. [41] state that truck arrival times are the main source of uncertainty in a cross-docking system. However, also demand volume and internal handling times within the facility (e.g. transportation times) can be uncertain. Peidro et al. [43] describes that uncertainty can be modeled using one of four distinct approaches: analytical models, Artificial Intelligence (AI) Models, simulation models, and hybrid models.

Analytical models are great for managing random uncertainties but are not powerful enough to model complex scenarios due to the many assumptions needed to simplify the real world. Analytical models are mostly used for optimization problems [43].

AI Models use (meta-)heuristics to obtain approximate solutions, but the solutions can be limited due to the lack of a global view of the system and preliminary assumptions [43].

Simulation models are more capable of capturing the behavior of complex systems and analyzing scenarios without the need for many assumptions. This enables simulation to include uncertainties and interdependencies more easily. However, simulation models require a large development and running time [43].

Finally, hybrid models integrate the best capabilities of analytical and simulation models, but the coordination of information provided by the individual models is complex [43].

Moreover, uncertainty can be dealt with by one of three main strategies: proactive, reactive, and predictive-reactive strategy [41].

Nasiri et al. [41] state that in a proactive strategy, robust optimization provides a robust master schedule but does not consider rescheduling if uncertainties further develop. Such proactive strategies have been proposed in other work [28, 29, 47, 4, 32, 46, 64]. Dealing with uncertainty in proactively designing cross-dock operations is seen in calculations by Mulcahy [40] by including a safety factor.

Reactive strategies have been proposed by Wang and Regan [63], Larbi, Alpan, and Penz [33], and Acar, Yalcin, and Yankov [1]. Nasiri et al. [41] mention that a reactive strategy suits systems under high uncertainties. However, the cross-dock manager has some information about truck arrivals, loads, and processes in the cross-dock facility. Nasiri et al. [41] argue that a reactive strategy may provide sub-optimal schedules due to its myopic approach.

A predictive-reactive system, as proposed by Nasiri et al. [41], is designed to reschedule and manage truck loading and unloading operations at a cross-dock facility considering delays in the inbound trucks' arrival times by minimizing the deviation between the original schedule and the new schedule. Nasiri et al. [41] argue that this predictive-reactive approach is a medium between the other two uncertainty-handling strategies. However, the authors considered all outbound trucks available from the start and assumed infinite temporary storage.

Table 2.1 shows that similar relevant studies are performed using analytical and simulation models or a combination of the two methods. As the level of uncertainty in the system rises, studies tend to favor simulation instead of optimization. Truck-to-dock assignment is usually performed analytically. However, studies were found where the truck-to-dock assignment was performed using heuristics as 'First Come, First Serve' are implemented [65, 13]. Simulation methods generally do not optimize to an optimum, therefore solving is considered not to be the main function of simulation.

Relevant studies shown in Table 2.1 lack considering both mixed-mode trucks [12] as analytical research often assumes all inbound trucks or outbound trucks to be available from $t = 0$. This implies that the inbound trucks are not the same individual trucks as the outbound trucks. The combination of mixed-mode trucks and mixed-mode doors has only been studied by Xi et al. [64]. However, Xi et al. [64] omitted limited storage capacity and assumed the demand to be known. Additionally, no research was conducted on the influence of reverse flow on cross-docking operations.

Based on findings in this chapter and Table 2.1, simulation is more suited for modeling cross-docking operations under many uncertainties and interdependencies. Although stochasticity of input parameters can be modeled more accurately using simulation, problems such as truck-to-dock assignment, product-to-location assignment, and truck scheduling are usually better to model using optimization. However, these optimization problems can be simplified and simulated using certain

scheduling or assignment policies (heuristics). The main shortcomings of Discrete Event Simulation (DES) resides in its inability to solve problems to optimality and the limited generalizability of research findings [21].

2.5. Measuring performance of cross-docking operations

To measure the performance of cross-docking operations, Ladier and Alpan [31] listed various possible measures. These indicators are often included in objective functions for optimization problems [31]. Their most relevant performance measures are itemized below.

- **Inventory level**

One of the goals of cross-docking is to minimize inventory. Therefore, the inventory level is an important indicator to track. In this case, the staging area stores the inventory. Hence, the storage levels of the staging areas are measured.

- **Makespan**

The makespan or schedule of the cross-docking operations is the time difference between the first received truck and the last dispatched truck. The shorter the makespan, the earlier operations are finished and the better the performance.

- **Total loading or unloading time**

The makespan mainly consists of loading and unloading of trucks. By minimizing the total time needed for unloading and loading the trucks, the doors are available sooner, and performance increases.

- **Waiting times**

Additionally to the *makespan* and *total (un)loading time*, trucks might need to wait before unloading and loading. By measuring waiting times, inefficiencies or congestion at doors can be identified.

- **Door utilization**

Closely related to the *total (un)loading time*, the door utilization rate measures the use of the individual doors and can identify certain doors as bottlenecks.

- **Travel distance**

The longer distance a pallet needs to be moved inside of the terminal, the longer the total cross-docking operation. The traveled distance is proportional to the number of pallets moved and therefore averaged to the total number of parcels.

- **Total pallets stay time**

To minimize the inventory, freight needs to leave the cross-dock as soon as possible. The total pallets stay time measures the average time a product spends in the facility.

- **Pallets not loaded**

When pallets cannot be loaded onto the outbound truck, they are moved to the backlog and either need to wait until the next day or have a dedicated truck pick them up. The number of pallets not loaded measures the missed orders, corresponding to lost profit. Not loaded pallets are called 'backlog'.

As seen from the itemization above, cross-docking performance measures mainly include time-based and inventory indicators.

Table 2.1: Characteristics of other studies versus this study

Reference	Scheduling problems		Physical characteristics		Process control		Product flow		Uncertainty factors			Method	Solving	
	Truck-to-dock assignment	Product-to-location assignment	Truck scheduling	Limited capacity staging area	Number of dock doors	Mixed-mode trucks	Mixed-mode dock doors	Reverse logistics	Forward logistics	Arrival times	Demand			Internal handling time
Boysen and Flidner [12]	•												MILP	Exact
Bodnar, De Koster, and Azadeh [11]	•	•	•		•	•	•	•	•	•			ILP	Exact & ALNS
Konur and Golias [28]	•		•							•			Bi-level MILP	GA
Konur and Golias [29]										•			Bi-level MILP	GA
Vis and Roodbergen [61]			•	•	•				•				MILP	DP
Serrano, Delorme, and Dolgui [54]		•	•	•	•		•						MILP	Exact
Shakeri et al. [55]	•	•	•						•				DP	VNS, TS
Rahmanzadeh Tootkaleh, Fatemi Ghomi, and Sheikh Sajadieh [47]	•		•		•				•				MILP	Heuristic
Amini and Tavakkoli-Moghaddam [4]	•		•		•	•			•				Multi-objective MILP	NSGA-II, MOSA, MODE
Rahbari et al. [46]	•		•	•	•					•			Bi-level MILP	Exact
Xi et al. [64]	•		•			•	•		•	•		•	MILP	C&CG
Larbi, Alpan, and Penz [33]			•	•		•			•				DP	Heuristic
Acar, Yalcin, and Yankov [1]	•		•		•					•			MIQP	Heuristic
Zhou, Lei, and Zong [68]	•		•	•		•			•			•	MILP	LR, GrA
Nasiri et al. [41]	•		•	•		•			•				MILP	Exact
Buijs, Danhof, and Wortmann [13]	•		•				•		•		•	•	DES	n/a
McWilliams, Stanfield, and Geiger [38]	•		•		•				•				Simulation-optimization	GA
Yang, Balakrishnan, and Cheng [65]	•			•	•	•			•	•	•	•	DES	n/a
Buijs, Danhof, and Wortmann [13]	•		•		•	•			•	•	•	•	DES	n/a
Arnaout et al. [7]					•	•			•	•	•	•	DES	n/a
Soto Zuluaga, Thiell, and Colomé Perales [57]								•					MILP	Exact
This study					•	•	•	•	•	•	•	•	DES	n/a

ALNS = Adaptive Local Neighborhood Search, GA = Genetic Algorithm, DP = Dynamic Programming, VNS = Variable Neighborhood Search, TS = Tabu Search, NSGA-II = Non-dominated sorting genetic algorithm II, MOSA = Multi-Objective Simulated Annealing, MODE = Multi-Objective Differential Evolutionary, C&CG = Column and Constraint Generation, LR = Lagrangian Relaxation, GrA = Greedy Algorithm, MILP = Mixed Integer Linear Programming, ILP = Integer Linear Programming, MIQP = Mixed Integer Quadratic Programming, DES = Discrete Event Simulation, n/a = not applicable.

2.6. Potential design alternatives

Following the research on decision-making theories in cross-docking operations section 2.3, design alternatives are constructed. Many means are found to improve cross-docking operations under uncertainty in literature. The most important means are identified to be the following.

- Storage area sizing [45]
- Layout designs [45]
- Facility sizing [40]
- Number of dock doors [40]
- Facility shape [40]
- Service mode dock doors [12]
- Truck-to-dock assignment [12]
- Truck scheduling [12]
- Internal transportation methods [40, 8]
- Staging-level design [8]

However, not all means stated above can be applied when addressing a specific problem under study. Therefore, chapter 3 will elaborate on a case study to identify key characteristics of a real-life cross-docking terminal and its suitable potential design alternatives.

2.7. Conclusion literature review

By concluding this literature review, the first sub-question of the main research question can be answered.

A cross-docking facility (CDF) is a physical facility used in supply chains to perform cross-docking, which aims to minimize point-to-point deliveries of goods. Unlike a warehouse, a CDF typically holds the goods for less than 24 hours. The goods are often consolidated on load carriers which determine the internal transportation method. The main activities taking place in cross-docking are unloading trucks, sorting goods, and (re)loading trucks.

Synchronization between the inbound and outbound operations is key in cross-docking. Local cross-docking operations can be managed on three levels. On the strategic level, the design of a cross-docking terminal impacts a facility's performance and robustness for uncertainty. On the tactical level, cross-docking operations entail assigning inbound or outbound trucks to dock doors. Often optimization is used to determine the best assignment tactic. On the operational level, terminal scheduling consists of truck scheduling and product-to-location assignment.

Cross-docking can be positioned on a continuum ranging from 'pure' to 'short warehouse time'. In a pure CDF, products are labeled and packed before arriving. Different facilities have different practices, affecting the number of unit loads processed and pre-injected unit loads. This can be called the cross-docking purity level.

Uncertainty is seen in cross-docking by either external or internal factors. The most influential external factors are demand and truck arrival time. Internal factors are processing time, available resources, and departure times. Internal processing can be uncertain due to the pathway blocking of pallet trucks and the skill level of the pallet truck driver. Due to the external factor of uncertain demand, the total internal transportation is also uncertain. This is considered an inter-dependency.

In problems requiring (near) optimal results, such as truck-to-door assignment and truck scheduling, analytical optimization methods are preferred. However, these optimization models require known data and can only handle limited uncertainty and interdependencies. As a result, uncertainty in demand and arrival times are hard to capture in optimization methods. Simulation can capture these interdependencies but cannot be used to find optimal solutions.

Performance of cross-docking operations is mainly measured using time-related, inventory-related, or resource utilization-related performance indicators such as tracking the makespan of the operation, inventory (or backlog) levels, door utilization, and waiting times.

When considering an existing cross-docking facility, the following design alternatives are applicable. Since trucks need available dock doors to start unloading and reloading as early as possible, uncertainty in truck arrival times can be mitigated by increasing the availability of the dock doors. The number of dock doors can be determined using the number of visiting trucks, the time needed for (un)loading, hours of work per day, and a safety factor by Mulcahy [40].

Service modes of the dock doors impact performance as well. Although little research was found on the performance of different service modes under uncertainty, most research indicates that mixed-mode dock doors outperform exclusive-mode dock door configurations [11]. However, research shows the additional benefit of more than 60% mixed-mode service doors is negligible [11, 50, 10].

Internal transportation methods can influence the performance of cross-docking operations. However, to the best of the author's knowledge, limited research can be found on the performance of different internal transportation alternatives.

Truck-to-dock assignment can be used to mitigate uncertainty in arrival times. However, truck-to-dock assignment alternatives can only be designed using policies rather than optimization methods in case of limited truck arrival and demand monitoring abilities.

A problem characterized by many interdependencies and uncertainty is often studied using simulation instead of optimization. However, the results of simulation research are less easily generalized, and simulation models are unsuitable for obtaining optimal solutions.

Although often mentioned by researchers, the effects of internal transportation methods have not been addressed fully for different load carriers as identified by Van Belle, Valckenaers, and Cattrysse [59]. Additionally, little research is performed on the effects of staging-level design [23].

To the author's knowledge, no research has been conducted on improving cross-docking performance by addressing the effects of different load carriers and staging-level designs under external and internal uncertainty factors. Therefore, this research aims to study the performance of staging-level design and other load carriers in a cross-docking environment under uncertainty. Although costs are important and cost trade-offs are associated with different cross-dock designs and configurations, this research mainly focuses on improving performance metrics.

The following chapter elaborates on the modeling method and construction of the design alternatives.

System analysis: Case Study

This chapter introduces the case under study and analyses the current state of the case study CDF. This functions as the case description and is used for identifying potential alternatives. First, the cross-dock terminal layout of the case under investigation is presented in section 3.1. Next, the activities and processes during cross-docking at the case study are elaborated on in section 3.3. The resources used in the cross-dock operations at the case CDF are discussed in section 3.2. Finally, section 3.4 touches upon the uncertainties occurring in the case under study and its associated problems.

3.1. Layout and facility components cross-docking terminal

The floor plan of the cross-docking terminal is shown in Figure 3.1. The terminal can be split into the lower half of the terminal and the upper half of the terminal. In the lower half of the terminal, the dock doors and staging areas are located. Dock door D10 is situated to the left and is furthest away from dock door D1. The staging areas are placed throughout the lower half of the terminal floor plan and differ in distance from the door docks. All dock doors are used for both unloading and loading arriving trucks. These are called mixed-mode service doors. Other areas are the sorting area for injected parcels, an entrance, offices, and a staging area for regionally injected parcels and backlog. Terminal pathways and staging areas are not marked and not strictly enforced.

As seen in Figure 3.1, the terminal under study is somewhat I-shaped since it only has doors over one side. The terminal also has a surface of roughly 700 m^2 and consists of 10 dock doors. Bartholdi and Gue [9] argues that an I-shape is best for small to mid-sized cross-docks because this design lets the freight move directly across the dock from the inbound door to the outbound door. However, Bartholdi and Gue [9] regarded opposite inbound and outbound doors. In contrast, the terminal under study only has one side of mixed-mode dock doors. The intra-terminal distances are presented in Table 3.3.

Each staging area is exclusively assigned to a single dock door, and the staging areas differ in storage capacity. The largest storage capacities are at S8, S9, and S10. The smallest storage capacities are found in S4 and S6. Since volumes per destination on average follow the distribution shown in Table 3.6, staging areas are pre-assigned to a destination and only hold pallets corresponding to that assigned destination. The cross-docking terminal is not exclusively used for cross-docking but also functions as a regional terminal. At night, parcels are injected and stored at the destined staging areas.

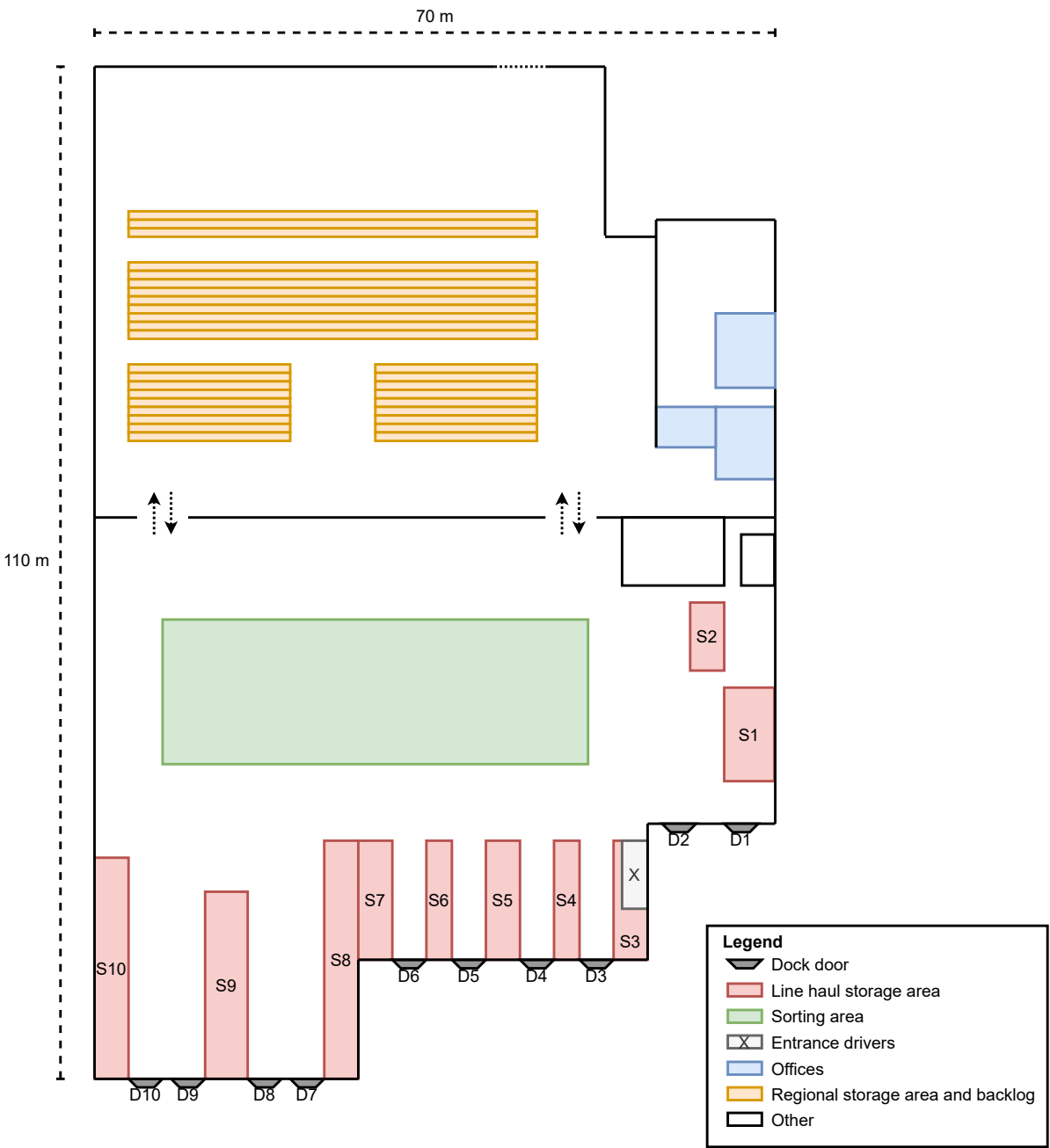


Figure 3.1: Floor plan terminal UTR

Table 3.1: Available staging area

Area	Pallets [#]
S1	15
S2	12
S3	18
S4	7
S5	21
S6	7
S7	14
S8	42
S9	44
S10	33

Table 3.2: Assigned dock doors

Dock door	Assigned line hauls
D1	GRO
D2	ANT
D3	HGL
D4	LWD, ZWL
D5	EIN, GEL
D6	AHM, BRE
D7	AMS
D8	HAG
D9	HTB
D10	RTM

Table 3.3: Intra-terminal distances [meters]

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	Storage CDF
D1	-	12	24	28	37	41	53	57	66	70	50
D2	12	-	20	24	33	37	49	53	62	66	62
D3	24	20	-	18	27	31	43	47	60	60	74
D4	28	24	18	-	23	27	39	43	52	56	78
D5	37	33	27	23	-	18	30	34	43	47	87
D6	41	37	31	27	18	-	26	30	39	43	93
D7	53	49	43	39	30	26	-	24	33	37	87
D8	57	53	47	43	34	30	24	-	29	33	83
D9	66	62	60	52	43	39	33	29	-	24	74
D10	70	66	60	56	47	43	37	33	24	-	60
Storage CDF	50	62	74	78	87	93	87	83	74	60	-

As seen in Table 3.3, the distances from door to door increase when the doors are located further apart. The distances to the storage area of the CDF (URT) are largest for doors located in the middle of the CDF. This is because one must travel around the sorting machine to get to the back of the terminal, where the storage area of the cross-docking facility itself is located.

3.2. Resources and load carrier types

Several resources depicted in Figure 3.2 are used during cross-docking operations. To transship the goods within the terminal, the terminal under study has 10 Electric Pallet Truck (EPT) and 2 Manual Pallet Truck (MPT) at its disposal. However, MPTs are rarely used. The EPT and MPT are used for transporting pallets. (Figure 3.2d). Other load carrier types, such as roll containers, can be used too to transport the freight (Figure 3.2c).

Additionally, the employees need to scan the inbound and outbound goods using a hand scanner to scan labels with bar codes and a picker phone to receive information on the scanned label. Picker phones and hand scanners are available for all terminal workers.

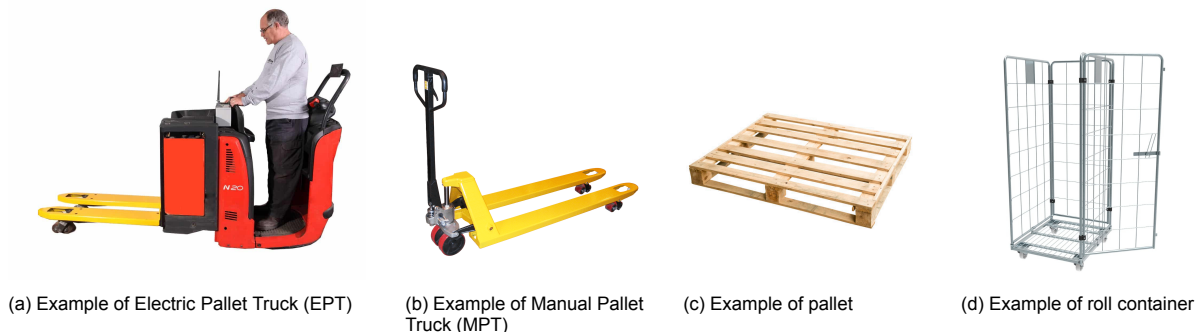


Figure 3.2: Available resources and freight carrier types

Subsequently, the employees operate these pallet trucks during operations, and since employees cost money, driving the pallet trucks induces costs. With a top speed limited to 6 km/h, EPTs travel faster than MPTs (4 km/h). However, EPTs can only be operated well by trained workers, whereas MPTs can also be operated by untrained workers. According to the terminal manager, unloading one pallet using an EPT takes 1 minute on average, whereas an MPT needs 3 minutes on average to unload a pallet.

Other resources are the dock doors, which only one truck can dock, and the limited staging area capacity shown in Figure 3.1. The staging area, however, can be viewed as a throughput bottleneck since the policy is that all pallets must be staged before reloading. Hence, the maximum throughput of the terminal is determined by the staging area capacity under the current policy.

Although the cross-docking terminal under study uses Euro pallets as load carrier type shown in Figure 3.2d other load carrier types exist. Among these load carrier types are roll containers Figure 3.2c

and postal bags. These load carrier types are currently not used during cross-docking in the case study. However, it can be interesting to study how the performance would change under other load carrier types. Euro pallets can hold 1,500 KG, whereas roll containers can hold 500 KG. Roll containers can carry a third of the weight that Euro pallets can, limiting the number of parcels fitting into a roll container.

Regarding the costs per load carrier type, one Euro pallet costs 15-21 euros, whereas a roll container costs over 70 euros. Since pallets generally can hold more volume than roll containers, pallets seem to be the more economical load carrier type. Additionally, with prices ranging between 1,200-4,000 euros, EPTs are more expensive than MPT, which cost little over 300 euros on average. Finally, the current average price per square meter of warehousing surface in the Netherlands is 60-73 euros.

3.3. Activities and processes in cross-docking

Identifying the activities and processes involved in cross-docking is essential to provide a clear overview of the current operations. Cross-docking operations can start when trucks arrive at the terminal. There, incoming trucks are unloaded based on their moment of arrival. Hence, this terminal uses a First In, First Out (FIFO) policy for serving trucks at the doors. However, trucks are assigned to a specific door and are generally not allowed to dock at other doors. The cross-docking operations consist of four consecutive steps; the arrival or receiving of inbound trucks (1), unloading of the inbound trucks and directly moving pallets to their staging area (2), loading the outbound truck (3), and dispatching the outbound trucks (4). For the cross-docking operations of the case under study, a logic flow diagram is constructed together with the CDF terminal manager and cross-docking supervisors, as seen in Figure A.5. Departing from the logic flow diagram, the cross-docking system is conceptualized by functions and activity. To this end, the IDEF0 diagram of the cross-docking process is constructed.

3.3.1. IDEF0 A1-level representation

The system is depicted using an IDEF0 diagram. An IDEF0 diagram is a common technique for structured analysis and design of systems [35]. An IDEF0 functional model consists of six components as described by Li and Chen [35]. The arrow from the left depicts inputs transformed into outputs to the right by a function. Controls are represented by the arrow from above the function, which are used to ensure accurate output conditions. Using the inserting arrow below the function, mechanisms that support function execution are depicted, while call arrows link models and provide additional details between boxes.

Using the description of Li and Chen [35], the system is represented in Figure 3.4 and Figure 3.3. First, activities or processes are depicted by boxes. Second, inputs are represented by arrows flowing into the left side of the box, while outputs are shown by arrows flowing out of the right side. Third, constraints or controls on the activities are represented by arrows flowing into the top of the box. Fourth, the mechanisms that carry out the activity are shown by arrows flowing into the bottom of the box. Finally, the model provides a comprehensive view of the system by representing all of these components clearly and concisely.

When zooming in on the activities in cross-docking at the case study, the A1 level of the system shown in Figure 3.3 emerges.

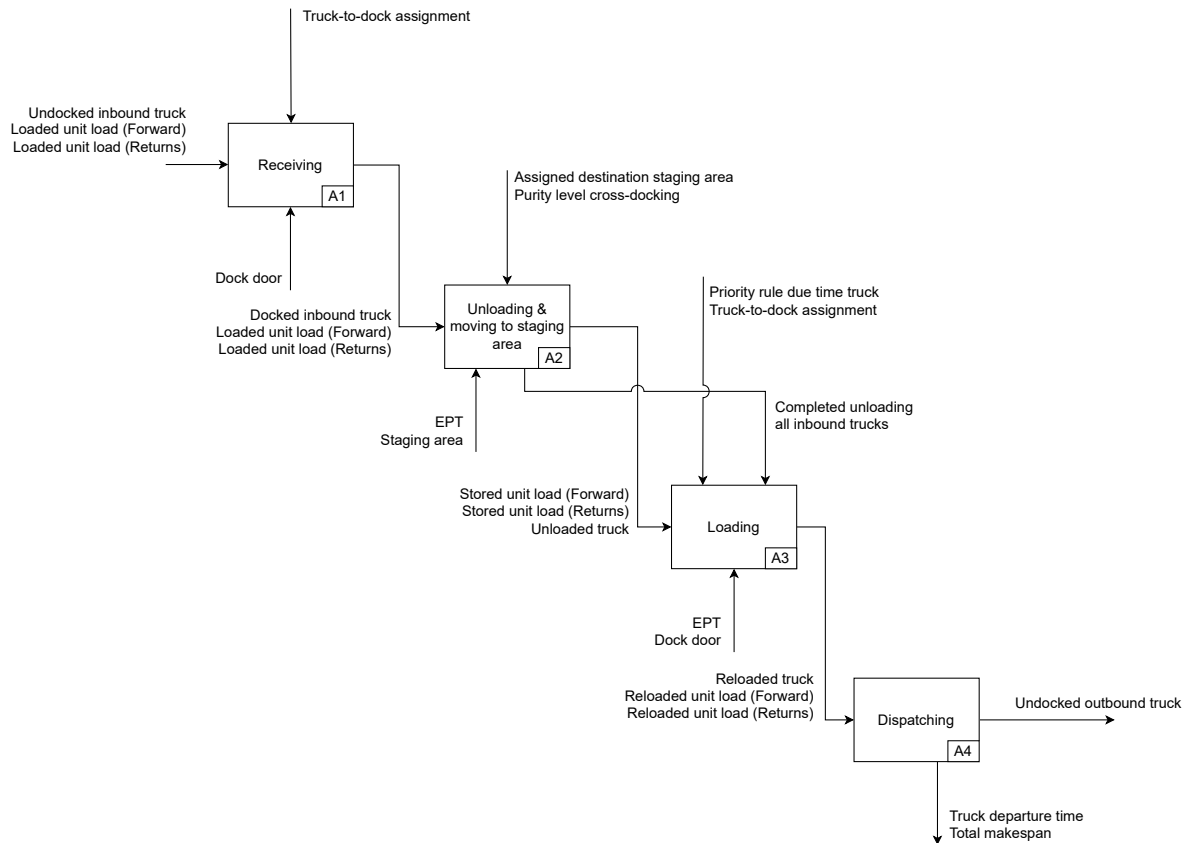


Figure 3.3: IDEF0 A1-level representation of the case study

The following subsections elaborate on the individual activities during cross-docking.

3.3.2. Receiving

In practice, unloading and moving pallets to the staging area are not consecutive steps but are performed iteratively and simultaneously. Each day, cross-docking is prepared at 5:45 AM and the drivers are expected to arrive at 6:15 AM. The planned departure time of the drivers is 7:00 AM to arrive on time for sorting at their regional terminal. Processes such as storing injected goods the day before can require more storage capacity than the available designated capacity, resulting in the usage of staging area capacity reserved from cross-docking. Other activities and processes at the terminal are irrelevant for cross-docking.

Arriving trucks can dock immediately for unloading or wait for docking if the assigned door is unavailable. The arrival of a truck is determined by the arrival time, and receiving is performed at a pre-assigned unloading dock door. The arriving truck contains loaded pallets of either Home, Box, or Return volume type. As soon as a dock door is available, the truck docks at the door, and unloading can begin.

All pallets arriving have already been sorted at their originating regional terminals. Therefore, the incoming pallets do not need extensive sorting and can be directly moved to the

3.3.3. Unloading trucks and moving pallets

Each dock door has its own EPT assigned, which performs all movements regarding its assigned dock door. Before unloading a pallet, an inbound scan is given to check in the pallet at the cross-docking terminal. This scan provides the pallet truck driver with information on which outbound truck the pallet must be loaded. If the outbound truck is already docked, the pallet is directly loaded into the outbound truck. If unavailable, the pallet is first stored at the staging area of the dock door. The inbound pallet is either a forward flow with a destination or a return flow exclusively destined for the regional terminal

handling returns. Additionally, the forward flow pallets are differentiated by the type of delivery method (home delivery or box delivery). The delivery method determines the priority to which the freight is reloaded in the trucks later.

An unloaded pallet is directly moved to the staging of its respective destination. The time needed for unloading a truck depends on the number of pallets stored in the inbound truck, the number of pallet trucks available, and whether the inbound pallets can be loaded onto the outbound trucks directly or are temporarily stored. The distance covered by the transshipment of the inbound pallets to the staging area affects the unloading time. The terminal pathways are narrow, and EPTs can not pass each other. Hence, EPTs often must wait at intersections to let other EPTs pass. Subsequently, EPTs cannot overtake one another in the terminal pathways. Additionally, some employees are more experienced with pallet truck driving and, therefore, faster, influencing the unloading time. Preemption of unloading is not allowed, meaning that once the unloading of a truck has begun unloading cannot be paused until completed.

The assigned dock door of the truck determines the distance to be covered. However, due to differences in staging area capacity, the truck-to-dock assignment is not performed regularly to minimize internal travel time.

Inbound pallets can be temporarily stored at the staging area near the dock doors. It is impossible to keep pallets other than at the dock door due to the risk of losing track of pallets. The storage capacity of the staging area areas differs per dock door. Therefore, as the average distribution of historical volume per destination is known, trucks are currently assigned to dock doors, of which the capacity of the staging area is sufficient to store pallets before loading.

Following Yang, Balakrishnan, and Cheng [65], from Figure 3.3 it becomes clear that this cross-dock under study is a single-stage cross-dock since the pallets are directly moved from the inbound truck to the outbound staging area.

3.3.4. Loading

Budbee's goal is to dispatch the trucks as soon as possible. Immediately after unloading an inbound truck, the inbound truck is available for loading and becomes an outbound truck. When loading pallets onto the outbound truck, an outbound scan is given to the pallet, which marks the pallet leaving the terminal. Budbee considers box routing to be more efficient than home delivery routing. As a result, box volume is prioritized over home volume when loading outbound trucks.

The goods transshipped are consolidated on pallets or in pallet boxes. These pallet boxes are often overpacked with either parcel boxes or parcel bags. Overpacked pallet boxes deform and lead to larger dimensions of the boxes. Typically, a pallet truck transports 33 pallets. However, due to the deformation of overpacked pallet boxes, the number of pallets per truck can sometimes decrease to 28 pallets per truck.

3.3.5. Dispatching

As soon as the truck finishes reloading, the truck is ready for dispatching. Based on the truck's due time, the truck is dispatched, and the actual departure time of the truck is marked. The result is a departed outbound truck with reloaded unit loads.

The A0-level presents the case study system at a high-level overview in Figure 3.4 by only depicting cross-docking at the case study as a whole, including the inputs and outputs.

3.3.6. IDEF0 A0-level representation

When zooming out on the cross-docking process depicted in Figure 3.3, the high-level overview of activities in the CDF is presented by Figure 3.4. This high-level overview helps with later defining the model scenarios and configurations.

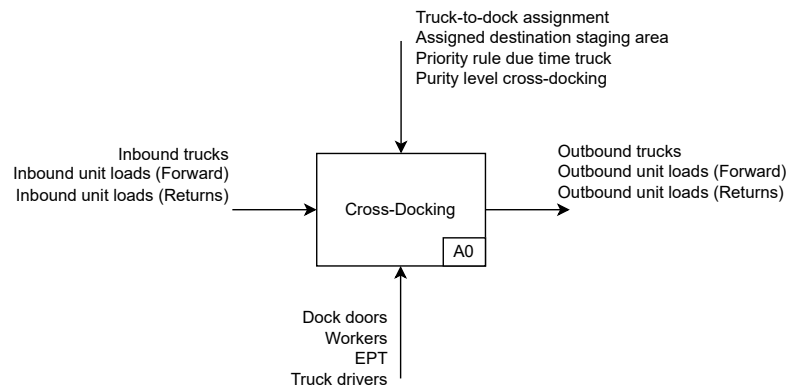


Figure 3.4: IDEF0 A0-level representation of the case study

3.4. System uncertainties and occurring problems

As e-commerce sales keep increasing and Budbee's operations are expanding, new regional terminals are opened, and additional line hauls are added to the network. This increase in volumes and the number of trucks to be cross-docked. Subsequently, this increase can lead to cross-docking becoming a bottleneck in operations. Among the problems are capacity issues due to growing volumes, storage capacity issues, unknown volumes and destinations loaded in the line haul, and fluctuating arrival times. To assess where the problems may lie, data on the total volumes, truck arrival times, and total cross-docking times are analyzed.

3.4.1. Transported volumes

When examining the transported volumes of Budbee it becomes clear that the daily transported volumes fluctuate significantly. The total transported volume fluctuates around a factor of 2 between roughly 100,000 and 200,000 parcels. The transported volumes are often prone to seasonal fluctuations (e.g. Black Friday, Christmas) and other temporary sales.

However, not all the transported freight but only cross-docked freight is of interest in this research. Moreover, the freight handled in cross-docking is consolidated on pallets. Therefore, the number of parcels is irrelevant, and only the number of pallets handled in cross-docking is assessed. The number of pallets handled in cross-docking operations is presented in Figure A.2. The corresponding numbers will be addressed in chapter 4.

Regional terminals do not produce or attract the same number of pallets. As a result, the capacities between the regional terminals and the cross-docking facility differ in the number of trucks used or the capacity of the truck(s). Managing the capacity is executed by a dedicated Budbee team that can upscale or downscale the capacities for transporting freight to the CDF. For some regional terminals, trucks can be deactivated, or additional trucks can be activated to match the line haul capacity to the demand. Seen from Table 3.4, most trucks can only carry 33 pallets. The case terminal under study mentions that occasionally, additional trucks are used to transport pallets in peak moments. Hence, higher demand volumes require more line hauls to transport pallets, which increases the total makespan.

Table 3.4: Number of line hauls per origin

Abbr.	Name	No. line hauls		Capacity [no. pallets]
		Active	Inactive	
AMS	Amsterdam	2	2	33
ANT	Antwerp	1	0	33
AHM	Arnhem	1	0	33
BRE	Breda	1	0	33
EIN	Eindhoven	1	0	33
GEL	Geleen	1	0	33
GRO	Groningen	1	0	18
HGL	Hengelo	1	0	33
HTB	Den Bosch	1	0	33
HAG	Leiderdorp/The Hague	1	0	33
LWD	Leeuwarden	1	0	33
ZWL	Zwolle	0	0	33
RTM	Rotterdam	1	2	33
UTR	Utrecht	n/a	n/a	n/a

Table 3.4 shows the number of active and inactive line hauls used in cross-docking. Only GRO uses an 18-pallet truck, whereas other line hauls use 33-pallet trucks. Direct line hauls between spokes are omitted in this overview since direct line hauls are not served during cross-docking. Table 3.4 shows no active nor inactive line hauls for ZWL since the line haul of LWD is used to pick up the freight at ZWL. The terminals in AMS and RTM have more (in)active line hauls at their disposal due to their larger shares in transported volumes.

As seen in Figure A.4, not all line hauls transport the same volume. Volume processed with cross-docking consists only of volume transported with line hauls. An origin-destination (OD) matrix is composed and shown in Table 3.5 to find the volume transported between regional terminals. This OD matrix is constructed using the total scanned parcels for a day of average volume on October 3rd, 2022. This is not the complete volume, but only the volume for which the origin is not the destination since only the volume for which the origin is not the destination is processed in cross-docking.

Table 3.5: Origin-destination matrix of scanned parcels for three months

	Destination														Grand Total	
	AHM	AMS	ANT	BRE	EIN	GEL	GRO	HAG	HGL	HTB	LWD	RTM	UTR	ZWL		
AHM		207	0	86	79	38	10	64	59	63	19	72	133	32	862	0.04%
AMS	3446		1272	3944	3121	1987	2660	5748	2114	2662	1750	6585	8974	4351	48614	2.11%
ANT	1	4		5	2	20	1	0	1	4	0	2	2	1	43	0.00%
BRE	248	1123	967		460	355	142	595	174	415	81	559	810	258	6187	0.27%
EIN	364	1578	11	559		194	213	674	299	605	138	762	1046	427	6870	0.30%
GEL	15	89	0	38	33		1	30	8	26	7	38	49	13	347	0.02%
GRO	85	364	357	167	170	194		183	135	101	50	239	209	116	2370	0.10%
HAG	83	484	0	231	241	78	23		48	138	29	184	250	73	1862	0.08%
HGL	108	665	2	210	193	108	38	179		185	59	230	413	106	2496	0.11%
HTB	32957	208916	16451	22471	36970	16193	3069	69199	6453		5691	95083	64453	13757	591663	25.66%
LWD	41	181	2	82	90	45	27	74	31	65		96	109	42	885	0.04%
RTM	124	7764	1	451	362	114	36	3808	70	209	46		3940	212	17137	0.74%
UTR	105285	381934	13997	132842	111818	76511	62779	173993	68049	100071	45090	230571		122249	1625189	70.49%
ZWL	47	222	0 119	103	41	10	67	17	94	23	91	126		960	0.04%	
Grand Total	142804	603531	33060	161205	153642	95878	69009	254614	77458	104638	52983	334512	80514	141637	2305485	100.00%
	6.19%	26.18%	1.43%	6.99%	6.66%	4.16%	2.99%	11.04%	3.36%	4.54%	2.30%	14.51%	3.49%	6.14%		

Table 3.5 shows the number of scanned parcels per OD-pair. The distribution of the origin and destination volumes is summarized in Table 3.6. However, this OD-matrix does not show the number of pallets the parcels are loaded on. Since there is no validated data available on the number of pallets handled in cross-docking per OD-pair, the averaged shares of origin (production) and destination (attraction) can be regarded as the average OD-distribution.

Due to the lack of functions in Budbee's WMS, the daily actual inbound and subsequently outbound volume for each line haul truck is unknown and the data shown in Table 3.5 is only from hindsight data. If the daily actual OD volume is known, truck scheduling can be an interesting option. For now, since the number of pallets and their destinations in the inbound trucks are unknown, each truck currently has to wait with loading until all trucks are unloaded to be sure that no pallets are left behind.

Table 3.6: Three-month average distribution per origin and destination

Terminal	Origin	Destination
AHM	0.04%	6.19%
AMS	2.11%	26.18%
ANT	0.00%	1.43%
BRE	0.27%	6.99%
EIN	0.30%	6.66%
GEL	0.02%	4.16%
GRO	0.10%	2.99%
HAG	0.08%	11.04%
HGL	0.11%	3.36%
HTB	25.66%	4.54%
LWD	0.04%	2.30%
RTM	0.74%	14.51%
UTR	70.49%	3.49%
ZWL	0.04%	6.14%
	100.00%	100.00%

From Table 3.6, it is clear that the majority of the volume originates in UTR and HTB. This is due to injections of the largest volume merchants at these two terminals. Significantly smaller volumes are injected in AMS and the other terminals. Destination-wise, the largest volumes are transported to AMS and RTM, which explains the larger number of (in)active line hauls for these two regional terminals.

When examining Table 3.6, it becomes clear that the cross-docking facility under study does not solely function as a cross-dock without internal storage. Therefore, the cross-docking terminal under study cannot be seen as a pure cross-docking terminal. Instead, the cross-docking terminal under study will be regarded as a hybrid cross-docking terminal. It could be interesting to see how the performance of the terminal changes when the facility acts in a pure cross-docking mode in which the demand injection is removed from the cross-docking facility and instead demand is injected throughout the network. Following Buijs, Vis, and Carlo [14], this injection decision can be considered a network management decision on the strategic level, hence a network design choice.

3.4.2. Arrival and departure times

As mentioned, all cross-docking trucks are required to be available at the terminal at a specific time. Table 3.7 provides an overview of the arrival times, due times, and return times of the line hauls per origin. For all line hauls, the return time at the regional terminal is set to 10:00 AM for sorting and route loading to begin on time. Based on the travel time from UTR to the regional terminals, the due time at UTR is calculated per line haul. In general, each line haul truck is planned to be available for cross-docking at 6:15 AM. Since UTR functions as both a regional injection terminal and cross-dock, there is no travel time from and to UTR which means that arrival times, due times, and return times are not applicable to UTR.

Table 3.7: Departure and arrival times trucks at terminal

Origin terminal	Travel time	Departure to UTR	Arrival at UTR	Due time UTR	Return time origin
AHM	80 min	4:55 AM	6:15 AM	8:40 AM	10:00 AM
AMS	60 min	5:15 AM	6:15 AM	9:00 AM	10:00 AM
ANT	100 min	4:35 AM	6:15 AM	8:20 AM	10:00 AM
BRE	60 min	5:15 AM	6:15 AM	9:00 AM	10:00 AM
EIN	75 min	5:00 AM	6:15 AM	8:45 AM	10:00 AM
GEL	130 min	4:05 AM	6:15 AM	7:50 AM	10:00 AM
GRO	160 min	3:35 AM	6:15 AM	7:20 AM	10:00 AM
HAG	60 min	5:15 AM	6:15 AM	9:00 AM	10:00 AM
HGL	110 min	4:25 AM	6:15 AM	8:10 AM	10:00 AM
HTB	35 min	5:40 AM	6:15 AM	9:25 AM	10:00 AM
LWD	140 min	3:55 AM	6:15 AM	7:40 AM	10:00 AM
RTM	60 min	5:15 AM	6:15 AM	9:00 AM	10:00 AM
UTR	n/a	n/a	n/a	n/a	n/a
ZWL	110 min	4:25 AM	6:15 AM	8:10 AM	10:00 AM

Table 3.7 shows the planned arrival times. The earliest due time is set at 7:20 AM for the line haul departing to GRO. This means that the shortest total cross-docking time is for GRO at 01:05:00. Budbee aims to limit cross-docking operations to around 45 minutes per truck. However, the line haul with the first due time (GRO) allows for more than 45 minutes of cross-docking. Arrival times can vary due to traffic, delays at the regional terminals, delays during cross-docking, or drivers being late. An example of the data on the Expected Time of Arrival (ETA), Expected Time of Departure (ETD), Actual Time of Arrival (ATA), and Actual Time of Departure (ATD) of the trucks is shown in Table 3.8. The courier names are anonymous and the data covers from January 2021 to November 2022.

Table 3.8: Example of data on arrival times trucks

Date	Route	Route driving time	Courier	Capacity	ETA	ETD	ATA	ATD
12/9/2022	AMS	1:00:00	Courier1	33	06:15:00	07:00:00	06:00:00	06:40:00
12/9/2022	AMS	1:00:00	Courier2	33	06:15:00	07:00:00	05:40:00	06:30:00
12/9/2022	EIN	1:00:00	Courier3	33	06:15:00	07:00:00	07:00:00	07:28:00
12/9/2022	HTB	0:35:00	Courier4	33	06:15:00	07:00:00	07:00:00	07:19:00
12/9/2022	GEL	2:15:00	Courier3	33	07:00:00	07:30:00	07:20:00	07:40:00
12/9/2022	RTM	1:15:00	Courier7	33	06:15:00	07:00:00	06:45:00	07:15:00
12/9/2022	BRE	1:45:00	Courier4	33	06:15:00	07:00:00	05:30:00	06:45:00
12/9/2022	ZWO	1:45:00	Courier5	33	06:15:00	07:00:00	04:45:00	06:45:00
12/9/2022	HGL	2:00:00	Courier5	33	04:45:00	05:30:00	06:00:00	06:52:00
12/9/2022	GRO	2:45:00	Courier6	18	06:15:00	07:00:00	05:45:00	06:50:00
12/9/2022	ANT	1:45:00	Courier4	33	06:15:00	07:00:00	05:15:00	06:45:00
:	:	:	:	:	:	:	:	:

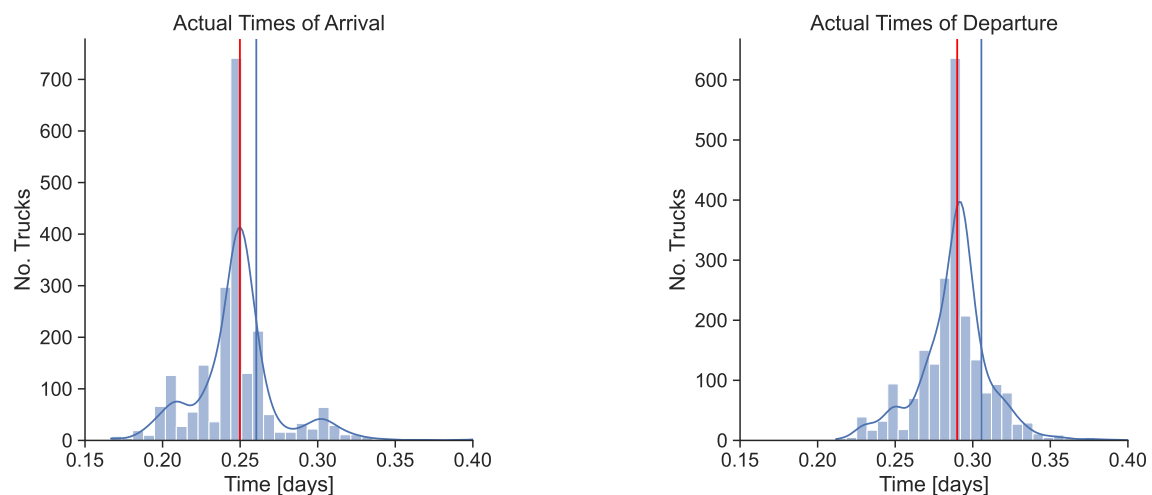
Table 3.8 shows that the ETA in UTR is not 6:15 AM for every courier. This is remarkable since the arrival time for each line hauls is set to be 6:15 AM as shown in Table 3.7. Similarly, Table 3.8 shows that the ETD does not match the planned arrival and due times shown in Table 3.7 and the ETD varies per line haul. Similarly, the actual time of arrival varies a lot per line haul. Since Table 3.8 shows that the ETA does not match the stated arrival times in Table 3.7, the difference between the ETA and ATA, the difference between the ETD and ATD, as well as the total cross-docking times are calculated. The descriptive statistics for these calculated times are obtained and depicted in Table 3.9.

Table 3.9: Descriptive statistics empirical data

	Capacity	ETA [hh:mm:ss]	ETD [hh:mm:ss]	ATA [hh:mm:ss]	ATD [hh:mm:ss]	Individual makespan [min]
Count	1188	1188	1188	1188	1187	1185
mean	31.5606	06:15:40	06:59:24	06:00:35	07:01:09	60.719
std	4.4199	00:38:12	00:35:38	00:29:16	00:26:36	16.5932
min	18	04:45:00	05:30:00	04:18:00	03:30:00	5
25%	33	06:15:00	07:00:00	06:00:00	07:00:00	60
50%	33	06:15:00	07:00:00	06:00:00	07:00:00	60
75%	33	06:15:00	07:00:00	06:00:00	07:00:00	60
max	33	09:45:00	10:15:00	10:50:00	11:25:00	165

Table 3.9 shows that on average, the difference between the expected and actual arrival times is 00:09:48, indicating that on average trucks arrive almost 10 minutes later than expected. However, the difference between the expected and actual departure time is (00:04:21) which indicates that the trucks on average leave 4 minutes and 21 seconds earlier than expected. In some extreme cases, a truck arrived over 5 hours earlier than expected (05:15:00) and a truck departed almost 5 hours prior to the expected departure time (04:45:00). The average total time used for cross-docking is 00:57:45. The minimum value of total cross-docking time is -20 minutes, which must be due to faulty input data.

When plotting the times in a histogram, the times are treated as continuous values before translating the values back to a 'HH:MM:SS' representation. The data shows the following distributions for the ATAs and ATDs depicted in Figure 3.5. Additionally, a Kernel Density Estimation (KDE) is plotted to depict the smoothed distribution of the histogram.



(a) Actual Times of Arrival (ATA)

(b) Actual Times of Departure (ATD)

Figure 3.5: Distribution of actual arrival and departure times

Figure 3.5 shows in red that the mean of the ATAs is 0.249856 days, which corresponds to 05:59:48. The mean of the ATDs is 0.289963 days which corresponds to 06:57:33 AM. The vertical blue lines in Figure 3.5a and Figure 3.5b respectively depict the first scheduled arrival time and the first scheduled due time of the cross-docking operations as defined in Table 3.7. Both figures show that on average the arrival time and departure time (in red) are both well before the scheduled arrival time and first due times (in blue).

Moreover, Figure 3.5a shows that the actual arrival times do not only center around 0.25 days (06:00:00 AM) but also show two quite small spikes around 0.21 days and 0.30 days respectively. This shows that trucks can not only arrive later but also around an hour earlier than the stated cross-docking time of 6:15:00 AM. Figure 3.5b shows a more centered distribution of the ATDs which averages at 06:57:33 AM, more than 20 minutes earlier than the earliest due time of the line haul to GRO.

Since the example of the data in Table 3.8 shows that not every line haul truck was expected at 06:15:00 AM, the difference between a truck's ETA and ATA, and its ETD and ATD is calculated and plotted in Figure 3.6.

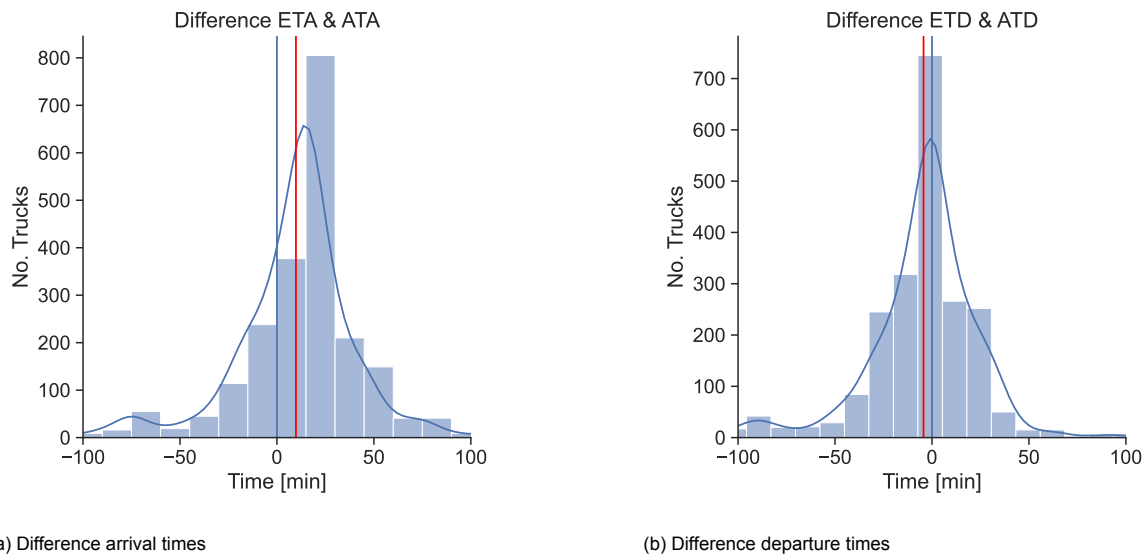


Figure 3.6: Distribution of difference in expected and actual arrival and departure times

Figure 3.6a shows a blue vertical line at 0.00 days. All counts to the left of this blue line are for trucks arriving earlier than expected. Subsequently, all counts to the right are for trucks arriving later than expected. The red vertical line represents the average time of 9 minutes and 48 seconds that trucks actually arrive later than expected (0.006808 days). Similarly, Figure 3.6b shows the difference between the ETD and the ATD. The mean difference is depicted in red at -0.003022 days to the left-hand side of the blue vertical line. This indicates that on average the trucks actually depart (00:04:21) earlier than expected. In both Figure 3.6a and Figure 3.6b a small spike is visible on the left-hand side of the figure.

As seen in subsection 3.4.2, the differences between the Expected Time of Arrival (ETA) & Actual Time of Arrival (ATA) and Expected Time of Departure (ETD) & Actual Time of Departure (ATD) show how the uncertainty in arrival times and the effect on the departure times. To identify outliers of the arrival times in Figure 3.6a, a box plot is constructed and presented in Figure 3.7. Additionally, the specific values of the box plot statistics are shown in Table 3.10.

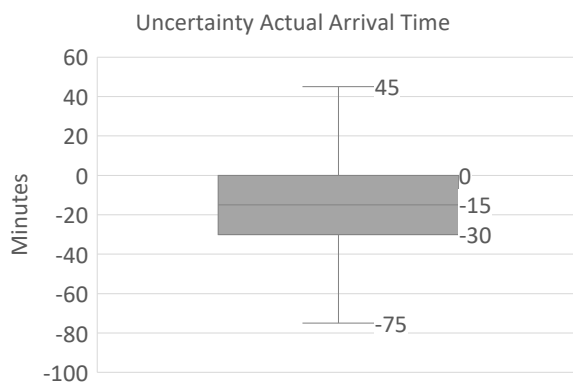


Table 3.10: Box plot values of the difference in ETAs and ATAs in the data set

Statistic	Value [minutes]
Max.	45
Q3	0
Median	-15
Q1	-30
Min.	-75

Figure 3.7: Box plot of the difference in ETAs and ATAs in the data set

It becomes clear that not only fluctuating demand levels influence the total cross-docking time, but

also truck availability which is uncertain due to fluctuations in actual arrival times. Figure 3.6a shows that on average, trucks arrive later than expected. Additionally, Figure 3.9 shows that the average time spent during cross-docking is higher than the preset 45 minutes. However, interestingly enough, Figure 3.6 depicts that on average the trucks depart earlier than expected. This could indicate that cross-docking is faster than expected or there is (unintended) buffer time in place.

Often the planned arrival times which are shown in Table 3.7 are not met when examining Figure 3.6a. Since Budbee makes use of subcontractors for line haul transportation (Fourth Party Logistics Provider (4PL)), the company is not only affected by processes within its own terminals but also relies on the performance of its subcontractors. Managing the subcontractors' processes and timely operations is considered difficult due to the management of numerous parties outside of their own company influences, lack of monitoring, and operational influence. Naturally, Budbee considers it easier to control its own operations than its subcontractors' operations. Additionally, a mismatch between a truck its ETA and ATA induces uncertainty in scheduled operations and processes. Budbee plans to be able to track the truck its actual position using GPS and subsequently can track and update the ETA of the truck. When this is the case, a predefined optimal truck schedule can be re-optimized to obtain a new optimal schedule. This leads to less uncertainty in arrival times.

3.4.3. Dock door availability

Figure 3.9 shows that the majority of the trucks already take more time for cross-docking than the predefined 45 minutes. This could be due to not accounting for waiting times. When keeping an average of 45 minutes for one truck to cross-dock and adding more line hauls to the network, the number of dock doors can be insufficient for all trucks to cross-dock in time. As a result, trucks would need to wait more often and or would need to arrive earlier or depart later. This increases the time spent cross-docking, which in turn costs money. Although waiting times are considered to be inevitable during cross-docking operations, logistical service providers try to minimize the waiting times.

Not only are the dock doors limited, but also the temporary storage capacity is limited during cross-docking to a number of pallets per storage area as shown in Table 3.1. This limited storage capacity can lead to some temporary capacity spilling over or even make the cross-dock scheduling problem infeasible. Preferably, this is never the case since spill-overs can cause the paths to become too narrow and decrease the maneuverability of the pallet trucks. Moreover, the number of pallets stored varies over time as stored pallets are transshipped to outbound trucks and leave the facility. However, since all trucks must be completely unloaded before reloading, the terminal floor capacity determines the total throughput of the cross-docking facility.

3.4.4. Internal handling time

Internal handling time is an important performance measure in cross-docking operations, as it directly impacts the efficiency and productivity of the operation. Internal handling time refers to the time it takes to move goods from the inbound dock to the outbound dock, including the time spent on sorting and consolidating shipments. The internal handling time is proportional to the internal distance covered and depends on the destinations assigned to the unit loads.

The internal handling time in cross-docking can be influenced by several factors, including:

- Size and complexity of the cross-docking operation: The larger and more complex the operation, the longer the internal handling time is likely to be. For example, if there are multiple inbound and outbound docks and a high volume of shipments, the sorting and consolidation process can take longer.
- Number of workers: The number of workers available to handle the shipments can also affect the internal handling time. If there are not enough workers, the process can become a bottleneck, resulting in longer handling times.
- Equipment availability: The availability of equipment such as pallet trucks can also impact the internal handling time. If the equipment is not available or is not functioning properly, the process can slow down, resulting in longer handling times.
- Product characteristics: The characteristics of the products being handled can also impact the internal handling time. For example, if the products are bulky, heavy, or difficult to handle, internal transportation can take longer.

Budbee could consider switching from pallets to roll containers as a consolidation method. Although pallets can hold more products and heavier loads than roll containers, pallets can only be moved using pallet trucks. Moving pallets using EPTs is quicker than using MPTs and roll-carts. However, EPTs can only be operated by trained workers, whereas both trained and untrained workers can move roll-carts.

When removing outliers from the data set as described in subsection 3.4.1 a correlation between the number of pallets handled in cross-docking and the total makespan is found as shown in Figure 3.8.

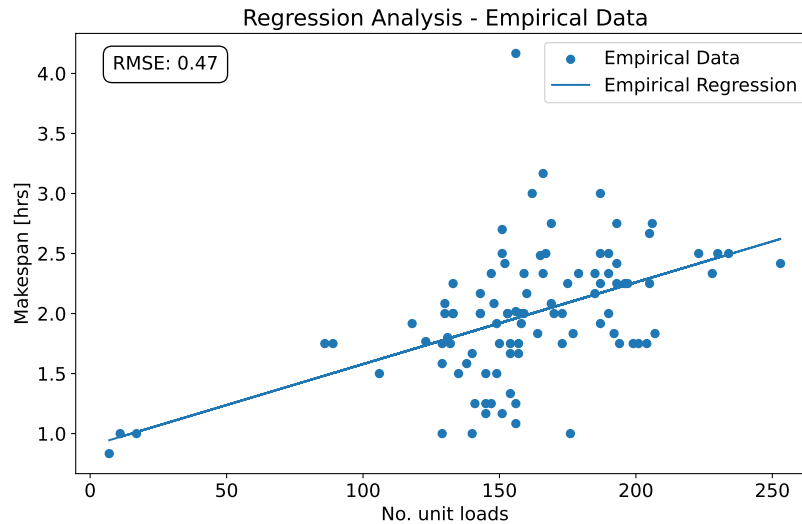


Figure 3.8: Correlation and regression of total unit loads and total makespan

Figure 3.8 shows that the total makespan differs for similar total pallets cross-docked. This can be the effect of uncertainty in arrival times, which can cause delays in cross-docking operations and therefore an increase in the total makespan. The R^2 of 0.47 is not considered to be a reasonable goodness of fit since a R^2 of 1 resembles a perfect fit of the correlation and 0 resembles no fit at all [15]. The low goodness of fit for the regression shown in Figure 3.8 can be either due to uncertainties in the cross-docking system or due to the validity of the empirical data. In general, it is considered a combination of both.

3.5. Current performance

To assess the current performance, the theoretical makespan according to Mulcahy [40] is compared to the actual makespan. Instead of using Equation 2.1 to determine the number of dock positions needed, the equation is used to calculate the time needed for loading/unloading. This time is considered to be the theoretical makespan. When calculating, the number of dock positions available is 10 as shown in section 3.1. The number of trucks per day is 13. The time difference between the designed arrival time at the cross-docking terminal and the earliest due time is 65 minutes (also 1.083 hours) as seen from Table 3.7. This marks the available work hours per day. The safety factor as originally chosen by [40] remains 25%. The calculation of the theoretical makespan is shown in Equation 3.1.

D	No. dock positions needed	10
N	No. trucks per day	13
T_{load}	Hours needed for loading/unloading	<i>t.b.d.</i>
T_{work}	Work hours per day	1.08
SF	Safety factor of 25%	1.25

$$D = N \times \frac{T_{load}}{T_{work}} \times SF \quad (3.1)$$

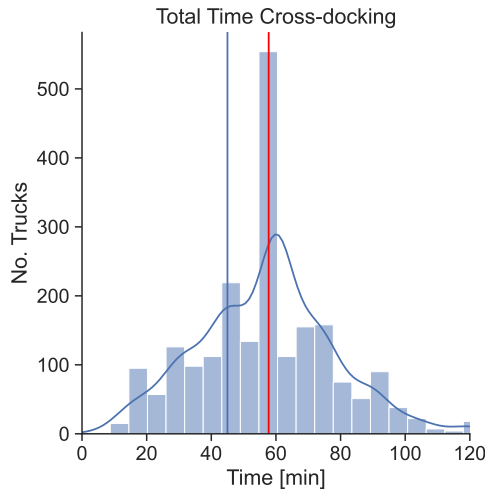
$$10 = 13 \times \frac{T_{load}}{1.08} \times 1.25 \quad (3.2)$$

$$T_{load} = \frac{10 \times 1.08}{13 \times 1.25} \quad (3.3)$$

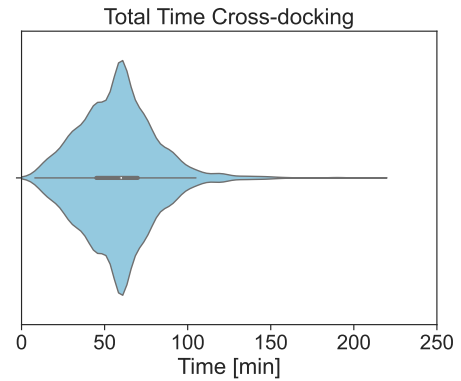
$$T_{load} = 0.724 \text{ hours} \quad (3.4)$$

$$T_{load} = 43.465 \text{ minutes} \quad (3.5)$$

To compare the theoretical makespan to the current performance, the total cross-docking time distribution is plotted and shown in Figure 3.9a. The average of this distribution should be 43.465 minutes for the current makespan to match the theoretical makespan. Additionally, a violin plot of the total cross-docking times is depicted in Figure 3.9b to better visualize the outliers in the data set.



(a) Distribution of total cross-docking times



(b) Violin plot of total cross-docking time

Figure 3.9: Total cross-docking times

Figure 3.9 shows that the recorded mean of individual cross-docking times is 57.75 minutes, which is higher than the expected 45 minutes. This means that the majority of the trucks spend more time at the cross-docking terminal than the planned cross-docking time. This can be due to cross-docking operations taking on average longer than initially designed. However, the truck waiting times are not included in the data set and therefore it is unclear how long internal transportation exactly took.

However, Figure 3.9 represents the individual truck makespan of cross-docking. Therefore, the total makespan per day is presented using a box plot in Figure 3.10.

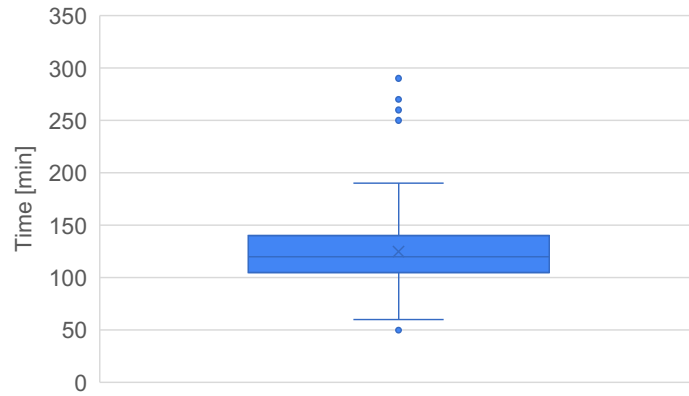


Figure 3.10: Makespan cross-docking operations

Figure 3.10 shows that on average, the total makespan of cross-docking operations is 125 minutes. Total makespans lower than 60 minutes or higher than 190 minutes are considered outliers.

In general, Budbee and its terminal operator want to minimize the makespan of their cross-docking operations to reduce costs. However, truck drivers often experience waiting times due to queuing at the terminal. Additionally, due to the interdependency of the trucks and the transported freight, late-arriving trucks cause delays in dispatching other trucks. When often experiencing waiting times, a truck driver could be more inclined to arrive at the cross-docking terminal later than agreed, causing delays for other trucks. Although waiting times are inevitable in cross-docking, terminal operators want to minimize the makespan whereas truck drivers want to minimize their own waiting time. Late-arriving trucks are often prioritized. Hence arriving later generally benefits truck drivers. This can be considered a conflict of interest.

3.6. Design alternatives identification

Now that the system at hand is analyzed, the literature research performed in chapter 2 is used to determine appropriate design solutions. Since this case under study assesses an existing cross-docking facility, moving to a completely newly designed and sized facility is out of scope. Additionally, due to circumstances at the cross-docking terminal under study, it is not possible to increase the number of dock doors available. Similarly, redesigning the cross-docking facility is limited. Therefore, only limited infrastructural changes can be made.

Moreover, due to the current inability to monitor live updates on truck arrival times and expected volumes in inbound trucks, truck scheduling optimization is currently inapplicable. Similarly, truck-to-dock assignment is performed by optimizing to know data and is therefore unsuitable for this problem. Hence, in the problem under study, truck-to-dock assignment can be performed based on either a "First In, First Out" policy or based on priority according to the truck its due time. Truck-to-dock assignments based on optimization can not be studied due to the requirement of monitoring real-time truck arrival times.

Storage area sizing is considered to be a potential bottleneck for operations since the total transported demand can overflow the limited storage area. However, given the indirect loading policy, all freight must be stored on the terminal floor and this limits the total throughput capacity of the terminal and therefore the network in general. Since the goal of this research is to study intra-terminal cross-docking operations, bottlenecks in the total network throughput are not of interest. Therefore, storage area sizing is not considered to be a potential design alternative.

Additionally, research shows that the use of mixed-mode dock doors improves cross-docking performance. Although the additional benefit of more than 60% mixed-mode dock doors is insignificant, operating with only mixed-mode dock doors proved to give the lowest makespan in research [11, 50, 10]. Since the cross-docking terminal under study operates with mixed-mode dock doors only, it can be concluded that operating with fewer mixed-mode dock doors will decrease the performance of cross-docking but not significantly as long as more than 60% of the doors is in mixed-mode. Hence, researching the effect of using fewer mixed-mode dock doors will not be

interesting.

One of the relevant potential design alternatives identified is the implementation of two-stage cross-docking. As seen from the IDEF0 system presentation in Figure 3.3, the current cross-docking process is single-stage, and although Gue and Kang [23] mention that a single-stage produces higher throughput, Yang, Balakrishnan, and Cheng [65] mention that a two-stage cross-dock can reduce congestion and therefore be more suited for systems with higher chance of congestion. This congestion can take place when many unit loads are moved simultaneously and pallet trucks block the pathways. It could be interesting to investigate how a two-stage cross-dock could be used to speed up operations in combination with re-allocating pallet trucks in case of late arriving trucks.

The second potential design alternative is the use of a different internal transportation method [40, 9]. EPTs are considered to be the quickest transportation method for pallets available to the company under study. However, EPTs can only be operated by trained operators who are not always available. There are MPTs available, but not in use due to their limited handling speed. Hence, when keeping pallets as a consolidation method the internal transportation method cannot be changed directly. However, the use of roll containers is also common practice in logistics. Roll containers have the benefit of allowing overtaking and passing due to their smaller dimensions than pallets and EPTs. Additionally, all workers can transport roll containers making internal transshipment independent of the number of trained workers available. The two identified design options are itemized below.

- Reallocating resources using two-stage cross-docking - Two-stage cross-dock [23]
- Roll containers as consolidation method - Internal transportation methods [40, 9]

The list above gives a clear identification of potential design alternatives that will be considered in this research. The modeling of the two design alternatives will be described and specified in chapter 4.

3.7. Conclusion problem description

To conclude the current state of the cross-docking operations in the case under study, the second sub-question of the main research question is answered.

The terminal layout is mainly characterized by staging areas that differ in capacities. Trucks are pre-assigned to dock doors based on historical volumes. Due to the differences in staging area capacity, trucks are statically assigned to docks. Additionally, due to a lack of data monitoring, truck scheduling optimization is not enabled.

The cross-docking terminal also functions as a regional terminal. Since the cross-docking terminal also allows injections that account for most of the demand volume, the terminal under study cannot be viewed as a pure cross-docking facility.

Activities in the cross-docking terminal are receiving inbound trucks, unloading the inbound trucks and simultaneously moving pallets to the staging area, reloading outbound trucks, and dispatching the outbound trucks. Each dock door has its own assigned EPT, which performs all unloading, moving, and reloading of the related door. Due to the lack of a well-functioning WMS, all pallets must be unloaded and stored in the corresponding staging area before reloading can commence.

The resources used in cross-docking are EPTs, MPTs, hand scanners, and picker phones. Dock doors are also considered resources since the dock doors are scarce, and trucks compete for available doors. However, under the current policy of first unloading all pallets before reloading, the staging area's capacity determines the system's maximum throughput.

The current state of the cross-docking terminal under study is embedded in the uncertainty of demand volumes and truck arrival times. Uncertainty in demand to be cross-docked leads to varying internal transshipment time. Additionally, the mode of transportation influences the internal handling time. The type of load carrier mainly determines the mode of transport. This case study mainly uses pallets and roll containers as load carriers. Pallets are transported using EPTs or MPTs, whereas terminal personnel can manually move roll containers without additional equipment. EPTs are considered to reach a relatively high traveling speed, but picking-up and dropping-off pallets take time. Moreover, EPTs can cause congestion by blocking terminal pathways due to the EPT's dimensions. Additionally, only trained workers can efficiently operate EPTs. Moving roll containers is considered to yield lower traveling speeds, and due to the smaller dimensions of roll containers, more roll containers are needed to carry the same load when using pallets. However, due to the smaller

dimensions of a roll container, workers moving roll containers can pass and overtake each other more easily, and less congestion on terminal pathways occurs.

The truck drivers want to minimize the waiting times and therefore consider their arrival times not as important as the terminal operator. In contrast, the terminal operator wants to minimize the total makespan. Therefore, the truck drivers and the terminal operator have conflicting interests.

Finally, potential design alternatives are defined to be changing the internal transportation method or redesigning the staging level strategy of the CDF.

4

Model formulation

This chapter elaborates on the model formulation by first discussing the model scope of the cross-docking facility and the relevant interdependencies in section 4.1. Next, the identified theories are identified, and design alternatives will be specified in section 4.2. Additionally, the requirements of the model and the assumptions made during modeling are elaborated on in section 4.3. After stating the requirements and assumptions, the key performance indicators are discussed in section 4.4. Then, the input parameters and variables are specified in section 4.5.

4.1. Model goal and scope

The main objective of the simulation model is to analyze how different load carrier types and staging level design options impact the performance of cross-docking operations in the real world, as explained in chapter 3. It aims to capture the effects of these design options mentioned in both chapter 2 and chapter 3. This study mainly focusses on change in the performance metrics rather than detailed cost modeling. However, the importance of incorporating costs when designing a CDF is acknowledged and will be briefly touched upon when interpreting the results.

Therefore, defining the system's scope before modeling is crucial, which involves excluding irrelevant complexities and focusing on relevant system dynamics for desired performance improvement. Firstly, the modeling method of choice is addressed. Next, the decisions on scoping the model regarding local and network choices are discussed in subsection 4.1.1. Finally, the layout and processes that have been scoped are presented in subsection 4.1.2.

4.1.1. Network versus local scope

This research focuses on local cross-dock management. However, after analysis in chapter 3, the cross-docking terminal under study does not function as a pure cross-docking terminal since roughly 70% of the total parcels are injected at the cross-docking terminal instead of throughout the network. As a result, only a relatively small number of pallets is cross-docked, and most of the volume is loaded on incoming trucks. One can argue that the terminal under study functions as a hybrid cross-docking terminal, making generalizations to pure cross-docking terminals difficult. Therefore, it is important to consider network demand distributions as seen in the case under study and network demand distributions expected at a pure cross-docking facility.

Figure 4.1 depicts the local scope of this study within the network scope. In this system, network design choices, such as the share of *pre-injected ULs* impact the local scope. However, changes in network choices are considered out of the scope of this research.

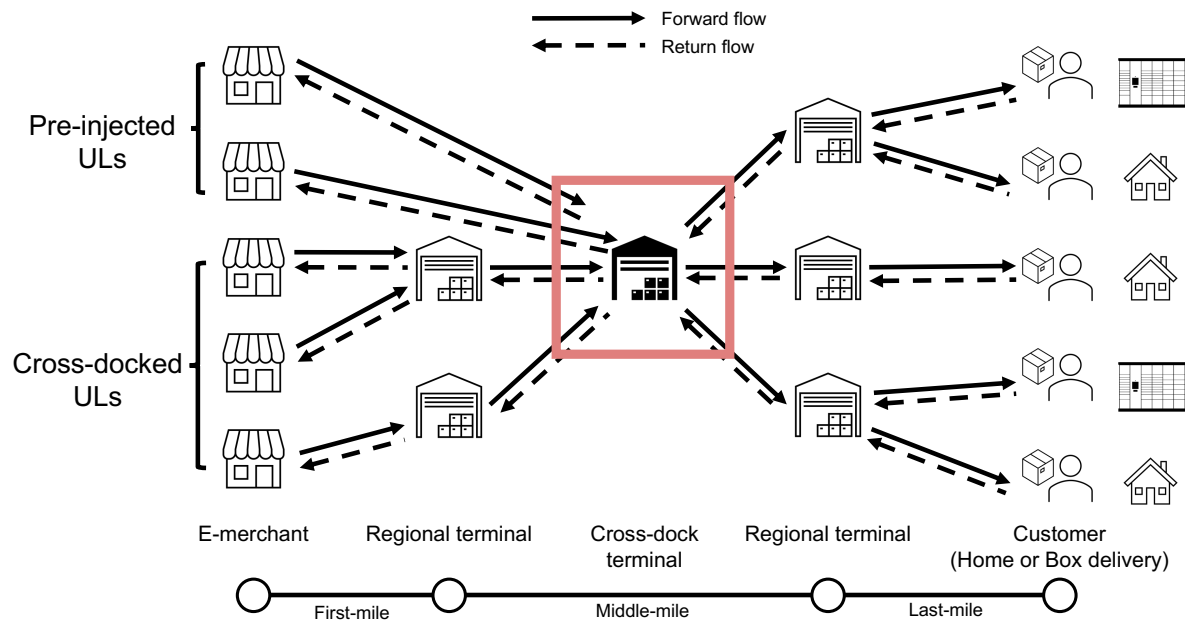


Figure 4.1: Local scope within network scope

4.1.2. Layout and processes

To better understand the problem under study, the model boundaries in terms of layout, and processes of the case study cross-docking terminal are shown in Figure 4.2. Since one of the design alternatives entails a different consolidation method, the pallet freight is henceforth called Unit Loads (ULs). A UL can be a pallet, container, postal bag, roll cart, or alternative consolidation method. This layout accurately represents the shape of the system being studied, showcasing the mixed-mode service dock doors connected by a terminal pathway where EPTs move the ULs. The distance between the dock doors and the staging area is considered insignificant and is therefore disregarded.

Although the system analysis shows that the staging areas have varying capacities, the current policy is not to reload until all ULs are stored on the terminal floor. As a result, the maximum capacity of the terminal area floor determines the cross-docking terminal's total throughput and operations as a whole. If demand exceeds the maximum terminal floor capacity, the modeled cross-docking system will stall under the current policy.

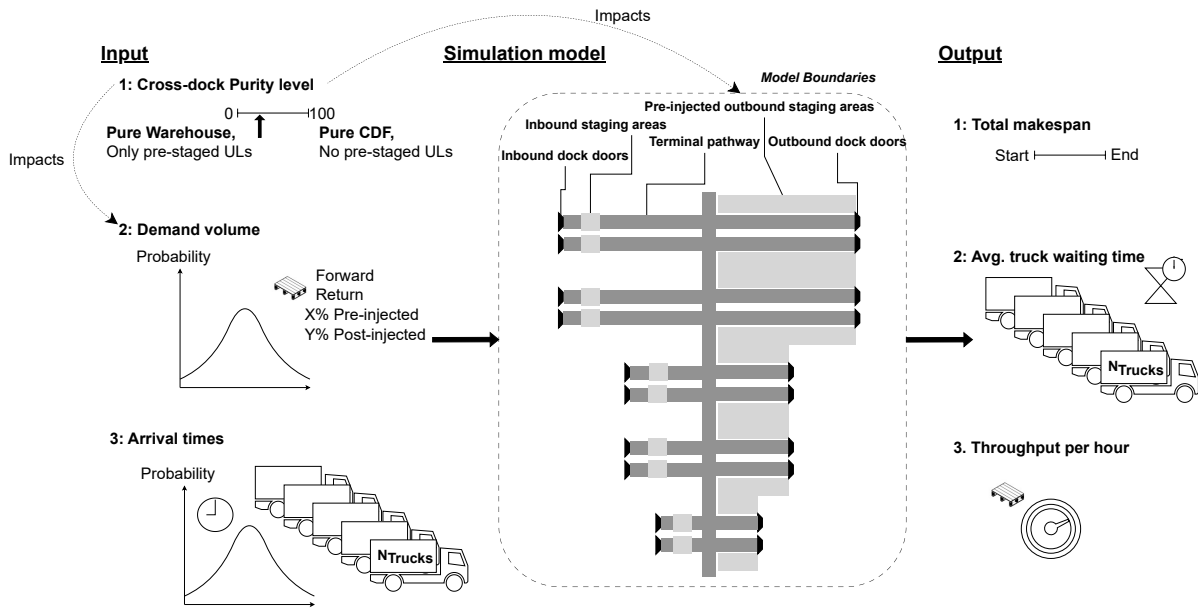


Figure 4.2: System scope and boundaries CDF

When a truck arrives at the cross-dock, the truck is first received at the assigned dock door. However, the arrival time of the truck is uncertain. Only available dock doors can receive trucks. The truck is loaded with ULs, which can be one of two UL types: Forward Unit Loads and Return Unit Loads (hereafter called Forward and Return entities). The number of ULs per truck is uncertain, and only estimations exist. After receiving the truck, the incoming truck is docked and still loaded with its initial freight.

After being received, the docked loaded truck is ready for unloading and sorting. Each staging area is assigned to a certain destination. As a result, the EPTs perform unloading and sorting simultaneously, which move the UL directly to the staging area corresponding to the ULs' destination. If all the ULs are unloaded, the truck is unloaded and ready for reloading. The truck can be swapped for the next truck assigned to the door to be received.

Until all trucks are unloaded and reloading can commence, the unloaded ULs are stored at their staging area. The ULs are stored based on the priority of the product type. Forward ULs are prioritized over Return ULs.

As soon as all trucks are finished unloading, reloading commences. The EPTs reload the outbound trucks with ULs (Forward and/or Returns) based on the loading priority and the truck's due date sequence. Occupied doors cannot simultaneously load more than one truck. After loading, the trucks are reloaded with ULs corresponding to their destination and ready to dispatch. For a truck to leave, it must either have reached its capacity, or there are no ULs left to be reloaded for the truck's destination. Trucks cannot leave before the unloading of the incoming trucks is completed. Hence trucks can depart late.

Dispatching the truck marks the departure time of the truck. Since the goal of cross-docking is not to have any inventory after cross-docking operations have ended, the ULs not reloaded to a truck are disregarded. Now the performance of the cross-docking operation can be reviewed by assessing the truck waiting times, total makespan, and throughput per hour.

Based on the previous system description in chapter 3 and the additional description in Appendix A, the factors under uncertainty are identified to be the truck arrival times, the demand volumes (Forward and Returns), the number of trucks, and the internal processing time due to skill of the worker.

4.1.3. Modeling method: simulation

The exploration of a cross-docking system in chapter chapter 3 revealed that the inbound and outbound trucks, as well as the allocation of resources, were interdependent with the staging-level design. Uncertainties significantly impact the performance of the CDF, as found in the same chapter. The literature review in chapter chapter 2 indicated that simulation is the best method for studying

such systems because it captures their unique characteristics. Therefore, simulation was chosen as the modeling method in this research, as discussed in chapter subsection 4.5.1, where the software used for simulation modeling will be thoroughly examined.

4.2. Theories and alternatives design

The system's conceptual model under study is presented in Figure 3.3 based on the system description provided in chapter 3. Hence, the relevant design alternatives identified in chapter 2 can now be specified for implementation in the model of the system at hand. The design alternatives to be modeled are two-stage cross-docking and two-stage cross-docking while using a different internal transportation method. Both designs are discussed in detail in subsection 4.2.2 and subsection 4.2.3.

4.2.1. Alternative design 0: Keep as is

This alternative does not entail any changes to the system. The function of modeling this alternative is for verification and validation purposes. Since this alternative does not entail any change, the current state of the system can be modeled, which functions as a modeling starting point for other alternatives. The IDEF0-representation of this alternative is by default identical to the depicted system in Figure 3.3 in chapter 3.

4.2.2. Alternative design 1: Two-stage cross-docking

Each dock door has its own EPT assigned to it, and although a single-stage cross-dock currently in place is said to produce a higher throughput, the total handling time can decrease when using two-stage cross-docking. Only one EPT can unload or reload a truck due to limited maneuvering space inside the trucks. Therefore, when instigating a two-stage cross-docking policy as shown in Figure 2.4b using pallets and EPTs, the pallets are staged 'First In, First Out' in front of the inbound dock and are then pulled by several EPTs to the designated staging area. Since the terminal pathways are too narrow for EPTs to pass or take over, this policy could increase the chance of congestion. Additionally, since there are only 10 EPTs available for 10 dock doors, each EPT is in use when simultaneous arrival occurs. Therefore, a two-stage cross-dock would not be suitable for the case under study during simultaneous arrivals. However, since uncertain arrival time is one of the system's prerequisites, two-stage cross-docking could be an interesting emergency procedure for a certain share of the last arriving trucks to speed up operations temporarily. The altered process during this emergency procedure is depicted in Figure 4.3. The redesign of the cross-docking process shown in Figure 3.3 is highlighted.

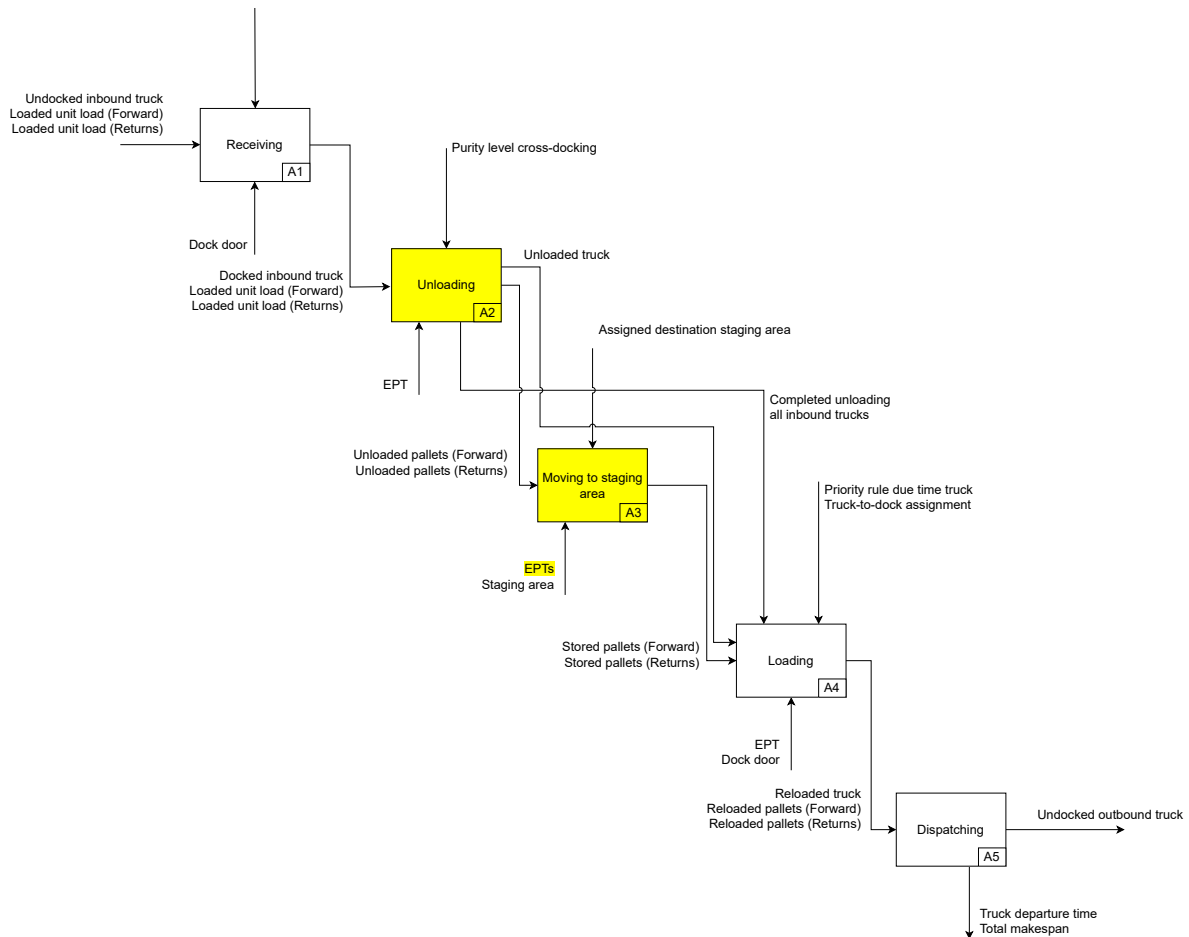


Figure 4.3: Alternative 1: Two-stage cross-docking

4.2.3. Alternative design 2: Two-stage cross-docking with roll containers

Roll containers offer a flexible consolidation method that allows workers to move them easily without needing equipment at an average walking speed of 5 km/h. This could make unloading and reloading UL faster. The smaller dimensions of roll containers also enable them to pass and overtake other containers in terminal pathways, reducing the risk of congestion. Roll containers come in various sizes but generally have a lower freight capacity than pallets. As a result, more roll containers are needed to transport the same amount of freight, increasing the number of UL per truck and the number of UL to be handled. While a truck can carry up to 33 pallets, it can easily transport up to 55 roll containers [36]. However, the downside of roll containers is their lower freight capacity. Weight constraints are not a concern for the case company, which mainly ships clothing and electronics, not heavy or bulky freight. To estimate the potential time saved using roll containers, the average unloading time per UL is presented in Table 4.1.

Table 4.1: Hypothetical performance per consolidation method and mode

Unit load	Euro pallet	Euro pallet	Roll container
Transportation method	EPT	MPT	Manual
Truck capacity [units]	33	33	55
Est. avg. unloading time [min/unit]	1	3	0.33
Total (un)loading time full truck [min]	33	99	18.15
Max. internal transportation speed [km/h]	6	3	5
Passing on terminal pathways	No	No	Yes
Chance of causing congestion	High	High	Low
Pick-up/Drop-off time [min]	1	0.5	2

From Table 4.1 it becomes clear that the total unloading time of unloading a truck full of roll containers is significantly smaller than the total unloading time of the same truckload with pallets. However, the average internal transportation speed is lower for roll containers. Therefore, for smaller average covered distances roll containers may be more efficient than transporting pallets using EPTs. To address the break-even distance for the three selected mode methods, the total cross-docking time of a Full Truckload (FTL) is calculated for different average internal distances covered using the data from Table 4.1. The average cross-docking times per mode are presented in Figure 4.4.

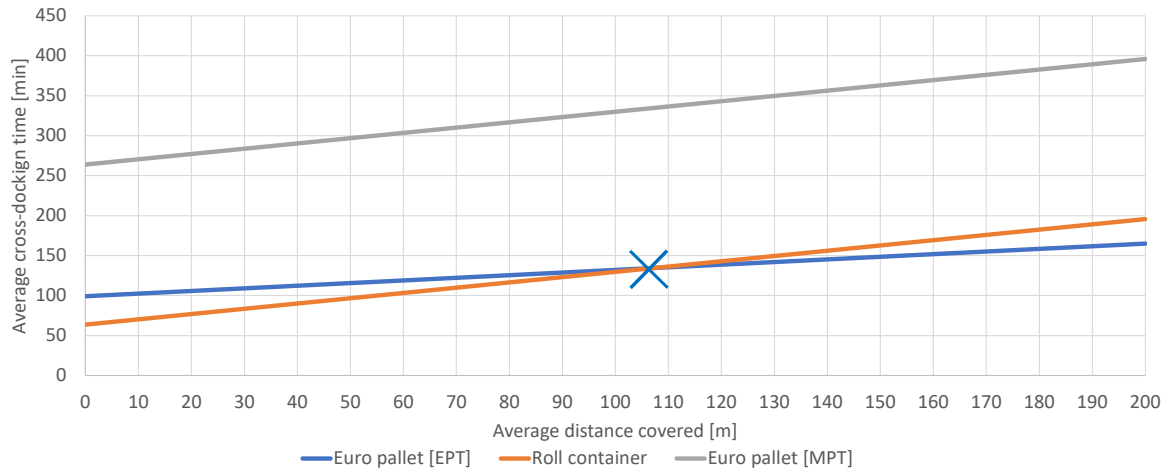


Figure 4.4: Average cross-docking time per FTL per mode over distance

From Figure 4.4 it is seen that for an average distance lower than 105 meters, moving roll containers manually outperform moving euro pallets with EPTs. The use of MPTs does not become beneficial. Since Table 3.3 shows that even the largest distances in the cross-docking terminal are smaller than 105 meters, roll containers could be a suitable alternative consolidation method to decrease the operational time. However, using roll containers does not mitigate the chance of late arriving trucks. Since the main benefit of roll containers is less risk for congestion due to passing on the terminal pathways and two-stage cross-docking can cause congestion on terminal pathways, the second design combines two-stage cross-docking and the use of roll containers. The conceptual model for this second design alternative is shown in Figure 4.5. The redesign of the cross-docking process concerning Figure 3.3 is highlighted.

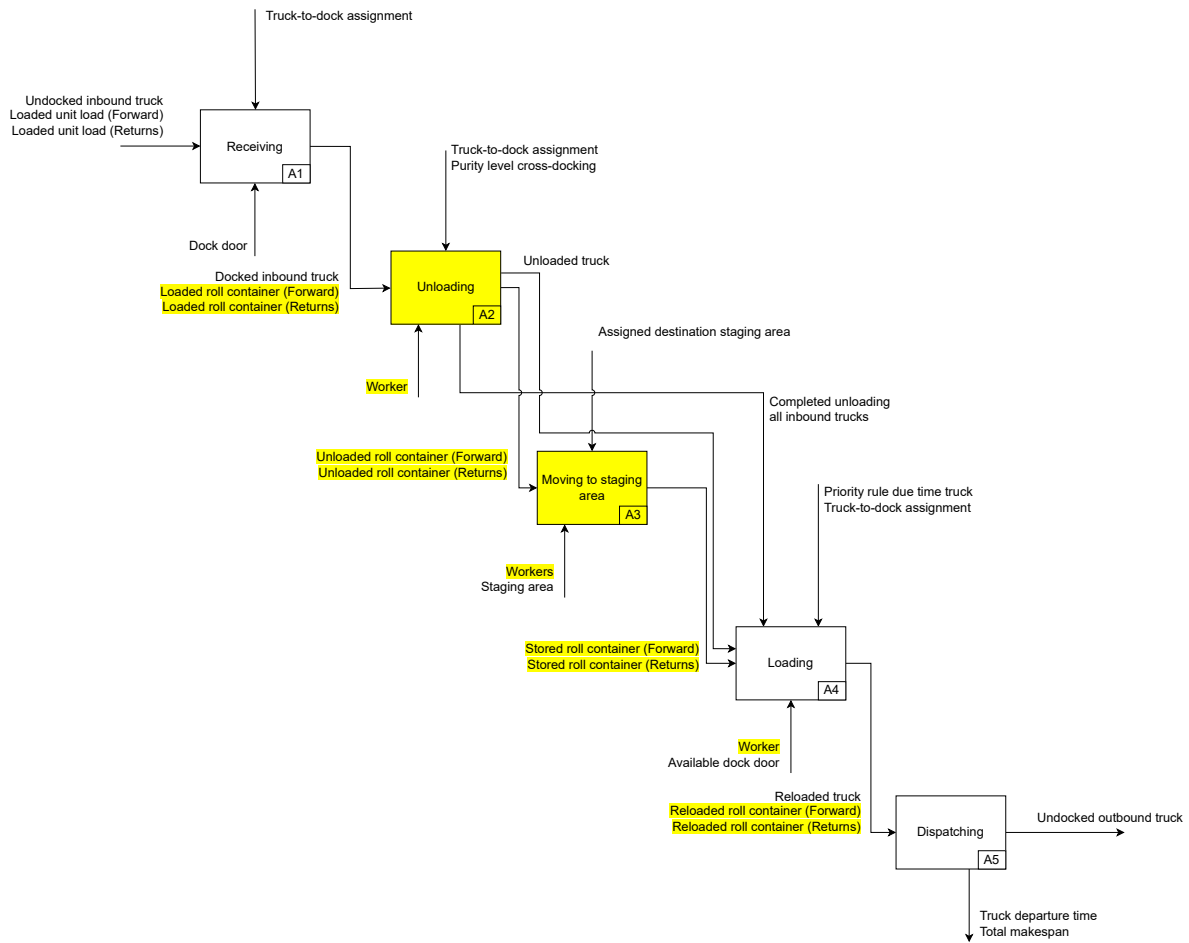


Figure 4.5: Alternative 2: Combination of reallocating resources and roll container usage

4.3. Requirements and assumptions

After identifying the two design alternatives and describing the potential benefits of the design alternatives, the requirements of the model and subsequently the assumptions are elaborated in subsection 4.3.1 and subsection 4.3.2 respectively.

4.3.1. Requirements

When translating the conceptual model of the system under study and the design alternatives to a simulation model, the requirements of the model must be clear to ensure reaching the end goals of the study. In this research, the simulation model must be capable of measuring the performance of a cross-docking facility in terms of total makespan (1), average truck waiting time (2), and throughput per hour (3). The model must also be capable of switching between ULs and staging-level designs. To this end, the model must generate transported demand with assigned destinations originating from regional terminals. The data for generated demand and arrival times must come from the case study to ensure the possibility of model validation. The model used must be verifiable. The model must also facilitate scaling demand volumes and truck arrival times for research purposes and verification and validation purposes. The model must always generate enough trucks to facilitate transportation with scaling the demand volumes. Regarding processes and activities, the model must be capable of realistically representing the processing and activities as described in the conceptual models shown in Figure 3.3, Figure 4.3, and Figure 4.5 respectively.

4.3.2. Assumptions

A model can never be 100% accurate, and simplification of models is needed to isolate the problem to study [44]. By simplifying the model, assumptions are made. One of the assumptions made in this research is that the truck-to-dock assignment is fixed and does not alter. Inbound trucks can start unloading after docking at their assigned dock and after the driver has been registered, which takes a specified random time.

The demand generated at RTs follows the same distribution for each RT. Volume generated destined for the regional terminal of origin is discarded immediately since this volume is not cross-docked. All ULs of the reverse logistics are transported to the reverse CDF. Since the goal of cross-docking is to not have any inventory left, it is assumed that backlog is always transported by additional trucks not involved in cross-docking. Therefore, backlog does not affect cross-docking operations. Hence it is assumed that the cross-docking terminal starts empty every day and a warm-up period is not required.

Uncertainty in truck arrival times is identical for all trucks, and the arrivals are distributed following a random triangular function. Similarly, the number of ULs per truck follows a random triangular function, and weekly demand patterns are discarded. The number of forward and return ULs per truck follows a fixed probability distribution. All trucks can be loaded to their theoretical maximum of 33 pallets and 55 roll containers. The possibility of ULs exceeding their designed dimensions is omitted. All loading docks have the same characteristics and have identical performance.

Only EPTs (no MPTs) are used, and there are the same number of EPTs as dock doors. A maximum of one EPT can perform (un)loading per truck. There are always enough licensed workers to operate the EPTs, and the workers are available at the start of the simulation. EPTs cannot pass or overtake when traveling terminal pathways and are assumed to wait for oncoming EPTs at intersections. The movement speed of EPTs follows a random triangular function due to acceleration and deceleration. It is also assumed that all EPTs are ready for operation at the start of the simulation.

4.4. Key performance indicators

In this research, the focus is on mitigating the effect of late trucks due to arrival time uncertainty. Therefore, the most important KPIs are time and throughput based. The time-based KPIs are limited to the total makespan and individual truck waiting time since the terminal operator wants to minimize the total makespan and the truck driver wants to minimize their waiting times. To compare results based on the input demand, the throughput level per hour is also important to consider. The three chosen KPIs are presented and summarized in Table 4.2.

Table 4.2: KPIs in this research

KPI	Definition	Unit
Makespan	Total cross-docking time per day	[min]
Individual waiting time	Waiting time per truck per day	[min]
Throughput per hour	Total number of unit loads cross-docked per hour	[unit loads/hour]

4.5. Implementation and specifications

Now that the designs are known, as well as the requirements, assumptions, and KPIs, the process steps in the model can be described. First, the method for translation of the conceptual model to the simulation model is described in subsection 4.5.1. Next, the simulation model processing steps are described in subsection 4.5.2, subsection 4.5.3, subsection 4.5.4, and subsection 4.5.5 for generating demand, truck arrival, unloading and staging, and reloading respectively.

4.5.1. Model implementation

The implementation of the conceptual model into a computer model is described in this subsection. For this research, discrete event simulation (DES) software is used. Multiple DES software packages are suitable such as SIMIO, Arena, FlexSim, and Simul8. Based on the availability of simulation software at Delft University of Technology, SIMIO is the software of choice. SIMIO stands for **S**imulation **M**odeling framework based on Intelligent **O**bjects. This object-based methodology considers primary data and procedures secondary. Therefore, the functions are associated with related data [17]. Since in this system, the ULs and arriving trucks contain data regarding their origin, destination, and volume, the

data underlying the objects determines the process. Therefore, this object-oriented methodology is suitable for the modeling problem at hand. An overview of the object definition is shown in Figure 4.6.

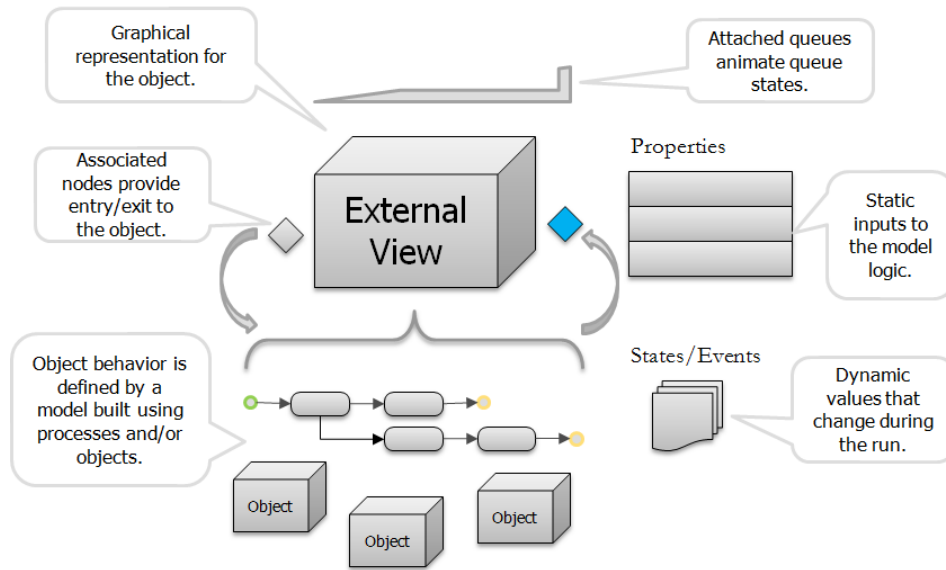


Figure 4.6: Object definition in SIMIO [56]

An object has five primary components: properties, states, events, external view, and logic [56]. Properties are static model inputs that do not dynamically change during simulation. States are dynamic values that can be altered during simulation. Similarly, events can occur dynamically and cause a certain process to start or stop. The external view of objects provides input and output possibilities for modeling. The logic of objects can be altered to satisfy the need of the modeler. Objects themselves can be either static or active. Static objects generally perform processes, whereas active objects undergo processes [56]. Active objects like entities or transporters are more dynamic and move during simulation. Static objects are used to perform operations on active objects.

In this research, the ULs and trucks are considered entities (active objects) that pass through the system and undergo operations, such as unloading, reloading, and transshipment. The EPTs and Workers are also active objects and are regarded as vehicles transshipping the entities inside of the CDF. The docks and staging areas are passive objects which perform actions on ULs and trucks, such as unloading, reloading, and storing.

Now that the model implementation method is described, the model specification describes the processes and steps of which the model consists. For specific model values, the reader is directed to the description in Appendix B.

4.5.2. Process step 1: Generating demand in trucks at regional terminals

The ULs and trucks created are modeled as entities. Per origin, both entities are generated according to the respective source's model demand level input and origin probability. ULs and trucks have data tables assigned to them storing properties. The most important properties are presented in Table 4.3 and Table 4.4. The more detailed model description, including actual data tables, is presented in Appendix B.

Table 4.3: Unit load properties

Unit load		
Unit load type mix		[Forward/Return]
Origin		
Destination		
Staging level	1 if single-stage, 0 if two-stage.	[Boolean]
Location in CDF	1 if staged at inbound door, 0 if staged at outbound door.	[Boolean]

Table 4.4: Truck properties

Truck		
Capacity		[# unit loads]
Origin		[-]
Destination		[-]
Staging level	1 if single-stage, 0 if two-stage.	[Boolean]

Based on the UL and truck entity data, sources are modeled from which the entities originate. Each origin has a Unit Load Source and a Truck Source. The most important properties of the Unit Load Sources and Truck Sources are shown in Table 4.5 and Table 4.6 respectively.

Table 4.5: Unit Load Source properties

Unit load source		
Entity types		Forward, Return
Arrival pattern	On event Unit Loads created	
Entities per arrival	DemandLevel $\times$ OriginProbability	[Case study]
	DemandLevel $\times$ PureCrossDocking.OriginProbability	[Pure cross-docking]

Table 4.6: Truck Source properties

Truck source		
Entity types		Truck
Arrival pattern	On event Unit Loads created	
Entities per arrival	Max. no. UL from/to RT	
	Capacity truck	

At the initialization of the simulation, the ULs and subsequent trucks will be created at the origin terminals. ULs can either be forward demand or return demand and have an origin assigned to them and a destination. Additionally, ULs have a staging level assigned for modeling purposes. The staging level is set up before the simulation run and therefore regarded as an input parameter. The number of trucks created depends on the number of ULs generated at the truck origin. The trucks have a fixed capacity, an origin that is their destination and an assigned staging level.

The number of ULs generated per origin is calculated by multiplying the total number of ULs created with the probability of the origin as specified in Table 4.7. Once a UL is created, the destination of the UL is assigned according to the probabilities also depicted in Table 4.7. The logic of the demand generation process is shown in Figure 4.7. Most importantly, ULs that are assigned an exactly matching origin and destination have their destination reassigned until their assigned destination differs from their respective origin.

Terminal nr.	Destination	Probability [%]	
		Origin case study	Origin pure cross-docking
RT1	6.19	0.04	8.33
RT2	26.18	2.11	8.33
RT3	1.43	0.00	8.33
RT4	6.99	0.27	8.33
RT5	6.66	0.30	8.33
RT6	4.16	0.02	8.33
RT7	2.99	0.10	8.33
RT8	11.04	0.08	8.33
RT9	3.36	0.11	8.33
RT10	4.54	25.66	8.33
RT11	8.44	0.08	8.33
RT12	14.51	0.74	8.33
RT13 (CDF)	3.49	70.49	n/a

Table 4.7: Demand volume distribution input parameters

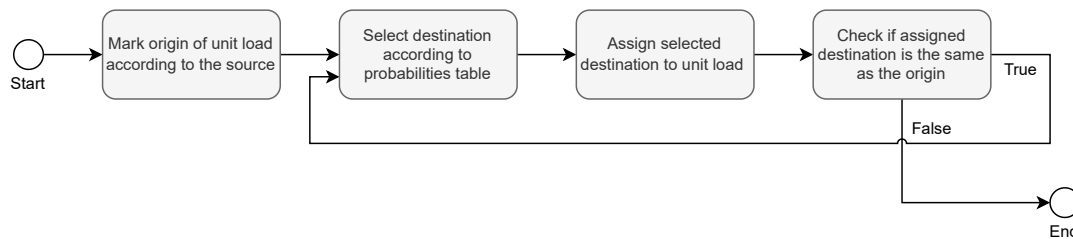


Figure 4.7: Destination assignment logic

Regarding the demand volume distributions shown in Table 4.7, the system at hand does not resemble a pure cross-docking system since the majority of ULs is directly injected at the CDF. Therefore, a hypothetically pure cross-docking demand scenario is also modeled in which the origin probabilities are evenly distributed over the regional terminals. The probability of originating is the same for each Regional Terminal (RT), except for RT13, which is the CDF and does not produce any demand in a pure cross-docking system. In this scenario, the destination probabilities are kept as previously defined in Table 4.7. The scenarios will be elaborated further in section 5.1.

After UL and truck generation, both entities enter a combiner in which the ULs are loaded (batched) into the trucks. Due to travel time between the truck origin and the cross-docking terminal, due times are associated with the trucks according to the times specified in Table 3.7. After creating and combining the entities, they enter a time path leading to the cross-docking facility. The time for crossing the time path is assigned randomly according to a triangular distribution, instigating the uncertainty of truck arrival time at the cross-docking facility with a first, average, and maximum arrival time. Since ULs are also injected directly in the cross-docking terminal for the case under study, the CDF itself generates ULs, which arrive instantly at the CDF after UL generation.

Process step 1 as described above is depicted in Figure 4.8. It is important to note that the travel time is set to a random triangular function to represent an arrival time window for truck arrivals. Other arrival patterns are also possible and can be modeled by using other travel time distributions.

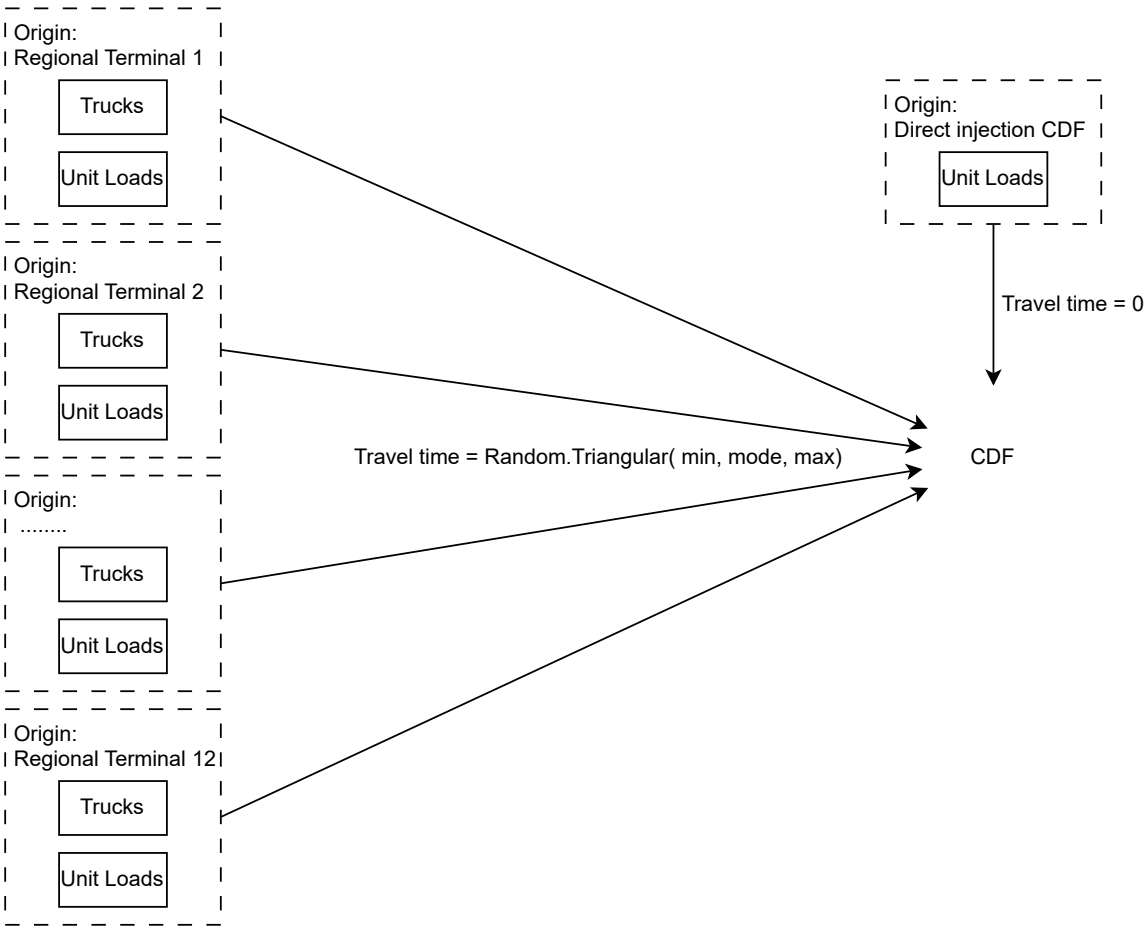


Figure 4.8: Step 1: Generation process of unit loads and trucks

The second process step commences once the loaded trucks are on their way to the CDF.

4.5.3. Process step 2: Truck arrival at unloading dock

Upon arrival at the CDF, the truck's actual arrival time (ATA) is marked, and the truck is assigned to a door to dock at. Additionally, the truck and its attached unit loads are marked to be unloaded using single-stage or two-stage cross-docking. Docking takes a specific time in which maneuvering the truck and registration of the driver is included. Each dock door has a maximum capacity of serving one truck at a time, at which trucks are served according to a FIFO policy. Therefore a queue emerges for trucks arriving while assigned to the same dock. The truck-to-dock assignment is based on the regional terminal (RT) from which the truck originated, as shown in Table 4.8. The corresponding truck-to-dock assignment process is depicted in Figure 4.9.

Table 4.8: Truck-to-dock assignment

Truck-to-dock assignment	
Unloading Dock 1	RT7
Unloading Dock 2	RT3
Unloading Dock 3	RT9
Unloading Dock 4	RT11
Unloading Dock 5	RT5
Unloading Dock 5	RT6
Unloading Dock 6	RT1
Unloading Dock 6	RT4
Unloading Dock 7	RT2
Unloading Dock 8	RT8
Unloading Dock 9	RT10
Unloading Dock 10	RT12

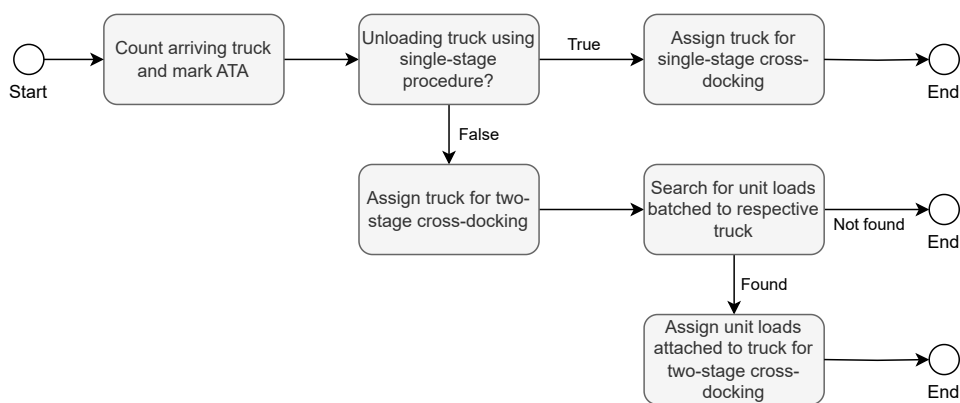


Figure 4.9: Step 2: Truck-to-dock assignment

Based on the assigned staging procedure, either single-stage or two-stage cross-docking is instigated as described in subsection 4.5.4.

4.5.4. Process step 3: Unloading and staging procedure

Once a truck arrives at the unloading dock object, the unloading procedure commences according to the object properties as shown in Table 4.9. Only one truck can dock per door simultaneously and trucks are served based on their time of arrival using First In First Out (FIFO) policy. Since the unit loads last loaded to the truck are closest to the trucks doors, the unit loads are unloaded (separated) from the truck entity using Last In First Out (LIFO) policy. The transfer-in time is the aggregated time estimate needed for docking the truck and registration of the truck driver. The processing time of the unloading dock object entails manoeuvring of the EPT inside of the inbound truck.

Table 4.9: Unloading dock specification

Unloading dock		
Capacity	1	[# trucks simultaneously served]
Ranking rule	First In First Out	
Separate logic	Last In First Out	
Transfer in time	Random.Triangular(5, 6, 7)	[Minutes]
Processing time	Random.Triangular(20, 30, 40)	[Seconds per unit load]

The EPT object or Worker object are both transporters that will be picking up the the unit loads have properties assigned to them as shown in Table 4.10 and Table 4.11 respectively. Both EPTs and Workers can only move one UL at a time and their task selection is based on the first in queue transport

request of the unit loads. The routing type of both transporters is on demand. The differences between EPTs and Workers lie in travel speed, pick-up and drop-off times and the ability for overtaking/passing on terminal pathways (links).

Table 4.10: EPT specification

<i>EPT</i>		
Ride capacity	1	[# unit loads]
Travel speed	Random.Triangular(3, 5, 6)	[km/h]
Pick-up/drop-off time	Random.Triangular(20, 30, 40)	[Seconds]
Task selection strategy	First in queue	
Routing type	On demand	
Overtaking/passing on links	False	

Table 4.11: Worker specification

<i>Worker</i>		
Ride capacity	1	[# unit loads]
Travel speed	Random.Triangular(2, 4, 5)	[km/h]
Pick-up/drop-off time	Random.Triangular(10, 15, 20)	[Seconds]
Task selection strategy	First in queue	
Routing type	On demand	
Overtaking/passing on links	True	

Now the unit loads can begin unloading. Unloading can only be performed by one transporter (EPT or Worker) at a time and only be the transporter assigned to the respective unloading dock door as shown in Table 4.12. First, a decision step is modeled to check whether the entity is an EPT or a worker for modeling purposes only, since - just like unit loads - EPTs and Workers are active object instances (entities) that can unintentionally trigger this process step. Next, the entity decides whether it is assigned to single-stage (true) or two-stage cross-docking (false). If assigned for single-stage cross-docking, the outbound staging area corresponding to the destination of the UL is searched for in the data table as shown in Table 4.13, and the UL is moved to the found inbound staging area. The intra-terminal transportation time is dependent on the pick-up time, drop-off time, travel speed, and the distance covered. The distances between the dock doors and between the staging areas and dock doors are presented in Table B.10 and Table B.11 respectively. If the UL is assigned for two-stage cross-docking, a decision step checks if the UL is marked as inbound staged. This states whether the UL is leaving the inbound staging area or leaving the truck. If the UL is leaving the truck, then the unloading staging area corresponding to the unloading dock is selected and the UL is moved to that inbound staging area. Additionally, the UL is marked as inbound staged. This modeling step makes the logic step as shown in Figure 4.10 applicable for both single-stage and two-stage cross-docking, since the UL leaving the inbound staging area can follow the same logic. Unit loads using a two-stage procedure requesting a move from the inbound staging area can be transported by all transporters, except from the one assigned to the Unloading Dock that the UL came through as specified in Table 4.12.

Transferring a UL into the Inbound staging area object is specified in Table 4.14. According to the inbound staging area properties, only one EPT or Worker can be simultaneously served, and queued transporters are served FIFO. The transfer in time is neglected since a worker is only picking-up or dropping off a UL which is included in the transporter object properties. Additionally, processing times are non-existent as the inbound staging area does not require any processing.

Table 4.12: EPT/Worker to door assignment

<i>EPT/Worker assignment</i>			
Unloading Dock 1	EPT1	Worker1	
Unloading Dock 2	EPT2	Worker2	
Unloading Dock 3	EPT3	Worker3	
Unloading Dock 4	EPT4	Worker4	
Unloading Dock 5	EPT5	Worker5	
Unloading Dock 6	EPT6	Worker6	
Unloading Dock 7	EPT7	Worker7	
Unloading Dock 8	EPT8	Worker8	
Unloading Dock 9	EPT9	Worker9	
Unloading Dock 10	EPT10	Worker10	

Table 4.13: Inbound staging area assignment

<i>Inbound staging area assignment</i>	
Unloading Dock 1	Inbound staging area 1
Unloading Dock 2	Inbound staging area 2
Unloading Dock 3	Inbound staging area 3
Unloading Dock 4	Inbound staging area 4
Unloading Dock 5	Inbound staging area 5
Unloading Dock 5	Inbound staging area 5
Unloading Dock 6	Inbound staging area 6
Unloading Dock 6	Inbound staging area 6
Unloading Dock 7	Inbound staging area 7
Unloading Dock 8	Inbound staging area 8
Unloading Dock 9	Inbound staging area 9
Unloading Dock 10	Inbound staging area 10

It is assumed that transporters place the pallets no more than 2 meters in front of the unloading dock in two-stage cross-docking to not obstruct transporters dropping-off unit loads at the door outbound staging area. Hence, the diagonal shows a distance of 2 meters from each door to its unloading staging area. Table B.11 does not show the distance from the CDF staging area to other doors, since the

Table 3.3 shows the distances between the dock doors as well as the distance to the staging area for unit loads destined for the CDF itself. The distances are derived from floor plans and measurements. It is assumed that unit loads are always moved along the terminal pathways depicted in Figure 4.2.

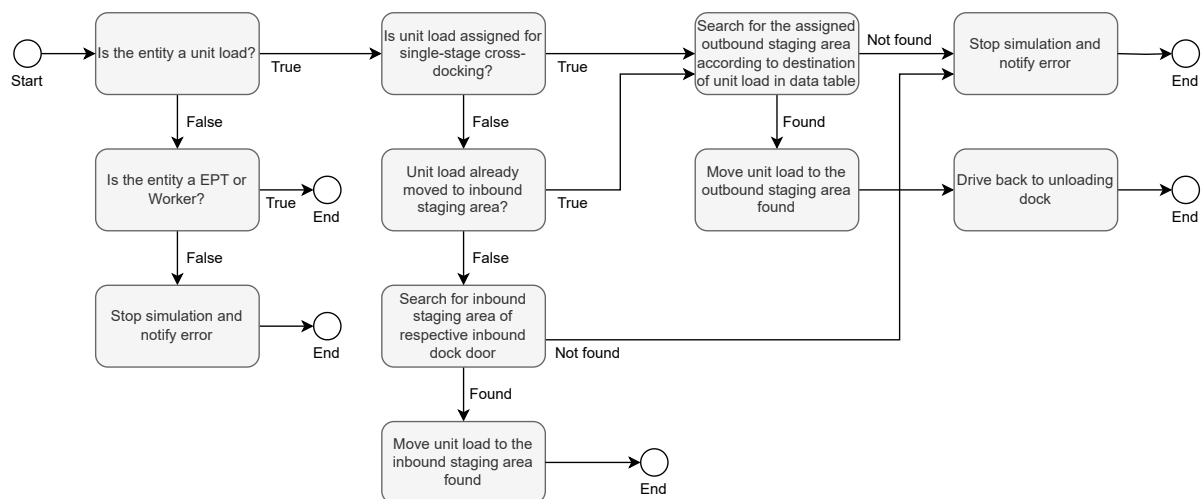


Figure 4.10: Step 3: Decision process Single-stage and Two-stage unloading procedure

Table 4.14: Inbound staging area specification

<i>Inbound staging area</i>		
Capacity	1	[# Unit loads simultaneously served]
Ranking rule	First In First Out	
Transfer in time	0.0	[Minutes]
Processing time	0.0	[Seconds per unit load]

4.5.5. Process step 4: Reloading truck and dispatching

Once all trucks are unloaded, the starting time of reloading is marked and the individual truck waiting time is recorded. Now reloading of the trucks commences at the reloading dock objects. The properties of the reloading dock objects are presented in Table 4.15. Only one truck can be docked at a time, therefore the capacity of the reloading dock is set to 1. Entering Truck entities are ranked on their due time so that early-due trucks are served first. The batch quantity is set to 33 when using pallets and 55 when using roll container unit loads. At the reloading dock object, the UL entities are batched to the truck parent entities based on their matching destination. The transfer in time for the truck entities is the total estimated time required for dispatching registration and optionally swapping the truck. In the case under study, multiple trucks are assigned to a door which can require a truck to be swapped if one of the truck entities in a queue has an earlier due time. Therefore, truck swapping time depends on the truck's due time and is checked before the truck entity is transferred in. The processing time for reloading is the sum of the pick-up and drop-off time of the transporter entity and is proportional to the number of unit loads batched.

Table 4.15: Reloading dock specification

<i>Reloading dock</i>		
Capacity	1	[# parent entities simultaneously served]
Ranking rule	Earliest Due Time First	
Batch Quantity	33	[# pallets per truck]
	55	[# roll containers per truck]
Batch logic	Matching entity destination	
Transfer in time	Random.Triangular(5, 6, 7)	[Minutes]
Processing time	$2 \times \text{Random.Triangular}(20, 30, 40) \times \text{Batch.UnitLoads}$	[Seconds for EPT]
	$2 \times \text{Random.Triangular}(10, 15, 20) \times \text{Batch.UnitLoads}$	[Seconds for Worker]

Once a truck is loaded, the truck is dispatched and the departure time is recorded. For the last departing truck, the departure time is recorded to calculate the total makespan. Additionally, the total throughput level is recorded to compare the results of the experiments. Now the KPIs of the simulation are recorded for the simulation replication.

After specifying all the process steps, the schematic overview of the simulation model is presented in Figure 4.11.

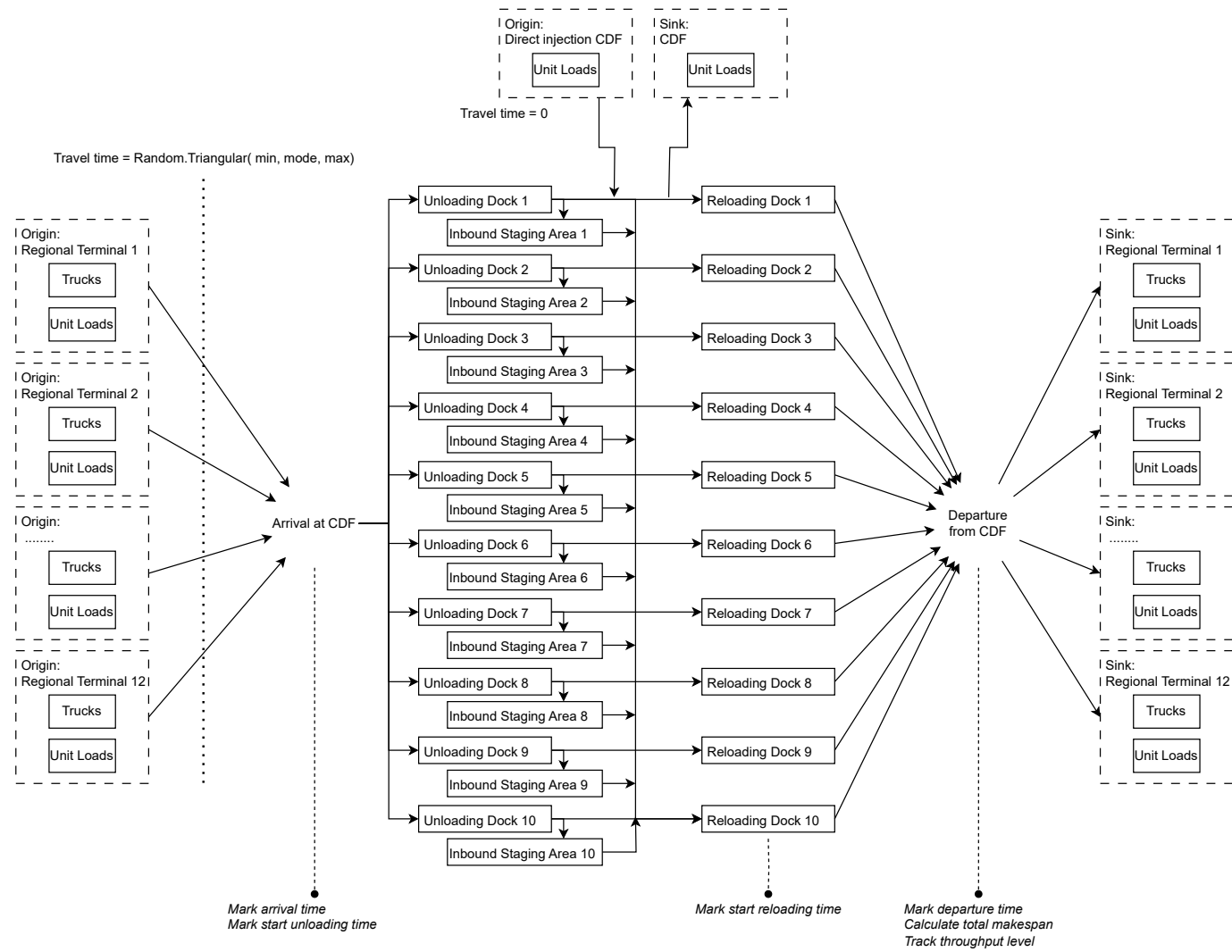


Figure 4.11: Full overview simulation model

4.6. Verification & validation

Following the line of reasoning of Robinson [51], verification and validation help to increase confidence in a model. However, the goal of verification and validation is not to demonstrate the absolute accuracy of the model [51]. The goal of verification is to ensure that the model is true to the conceptual model [51]. Robinson [51] regards this as assessing whether the model has been built right. On the other hand, the goal of validation is to ensure that the simulation model has sufficient accuracy for representing the problem at hand [51]. Robinson [51] regards this as assessing whether the right model has been built and constructed the diagram as shown in Figure 4.12 accordingly.

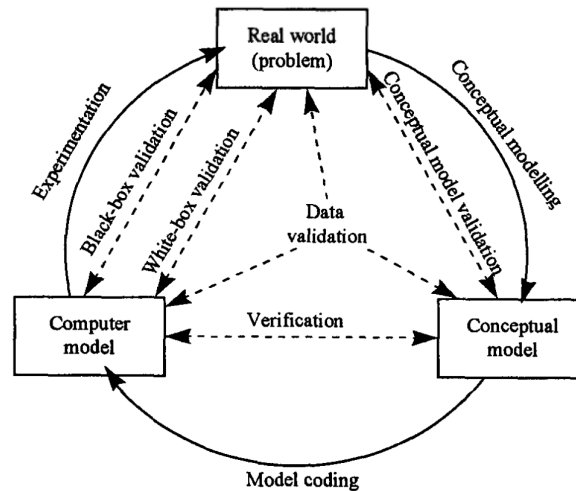


Figure 4.12: Verification and validation in the modelling process [51]

First, following Figure 4.12, the accuracy of translating the conceptual model into the computer model is addressed in subsection 4.6.1. Once verified, the model is calibrated in subsection 4.6.2 to ensure the proper working of the model parameters. Then, data validation is discussed in subsection 4.6.3 to address the confidence in the data set. Next, the accuracy in constructing the conceptual model is addressed in subsection 4.6.4. Furthermore, white-box validation is performed in subsection 4.6.5 by visually checking the simulation model and inspecting the output reports [51]. Next, in subsection 4.6.6, black-box validation is executed to validate the overall behavior of the model and increase confidence in the capability of the simulation model to replicate real-world output data sufficiently [51]. Finally, this section is concluded in subsection 4.6.7.

4.6.1. Verification

The first verification test is a flow conservation test. This test ensures that the number of unit loads generated in the origins equals the number assigned to each destination. Seen in Figure 4.13, the total number of unit loads is the same for the origins and destinations. This means that no unit loads spontaneously appear or disappear in the model.

	Origin	Dest.
CDF	93	6
RT1	1	7
RT2	6	41
RT3	1	4
RT4	1	11
RT5	1	16
RT6	1	5
RT7	1	2
RT8	2	12
RT9	1	4
RT10	48	24
RT11	2	11
RT12	3	18
	161	161

Figure 4.13: Unit loads flow conservation verification (end of verification run)

Additionally, time conservation is verified by examining the total makespan. The total makespan must be equal to the total unloading time plus the total reloading time. As seen from Figure 4.14, the total unloading time (Time Unloading [min]) is equal to the recorded last finished unloading time minus the first started unloading time. Similarly, the reloading time (Time Reloading [min]) equals the time mark of the last reloaded truck minus the first reloaded truck. In turn, Figure 4.14 shows that the total makespan equals the total unloading time plus the total reloading time.

		Inbound Trucks Departed	14
		Inbound Trucks Arrived	14
Unit Loads to be Cross-Docked	68	Outbound Trucks to be Served	0
Unit Loads pre-Cross-Docked	93	Inbound Trucks to be Served	0
Total Throughput Unit Loads	161	Trucks in System	0
Unit Loads Left In System	0		
Time start first unloading	0.134...	Time start first reloading	113.39...
Time last unloaded	113.3...	Time last reloaded	159.48...
Time Unloading [min]	113.2...	Time Reloading [min]	46.083...
		Average waiting time Reloading [min]	54.409...
		Total makespan [min]	159.34...
		Average individual makespan [min]	118.79...

Figure 4.14: Time conservation (end of verification run)

Moreover, the total number of trucks left in the system equals zero as well as the total number of unit loads left in the system. This means that the system ends empty. Also the number of inbound trucks that arrived and the number of trucks departed is equal, this means that no truck is left behind.

Next, event tracing is performed for two of the main entities: unit loads and trucks. The comprehensive model trace as well as the analysis is found in Appendix B Table B.13. The analysis shows that the model trace is consistent with the conceptual model.

When inputting zero demand, the data shown in Figure 4.15 is generated. This means that no input is generated when no input is given.

			Inbound Trucks Departed	0	
			Inbound Trucks Arrived	0	
Unit Loads to be Cross-Docked	0				
Unit Loads pre-Cross-Docked	0		Outbound Trucks to be Served	0	
Total Throughput Unit Loads	0		Inbound Trucks to be Served	0	
Unit Loads Left In System	0		Trucks in System	0	
Time start first unloading	10000	Time start first reloading	10000	Average waiting time Reloading [min]	NaN
Time last unloaded	10000	Time last reloaded	10000	Total makespan [min]	0
Time Unloading [min]	0	Time Reloading [min]	0	Average individual makespan [min]	NaN

Figure 4.15: Zero demand input flow conservation (end of verification run)

Additionally, if no unit loads are generated, no trucks are generated as well. This is according to the assumption that the number of trucks is always sufficient to transport the generated demand.

The verification methods above-provided insights into the quality of implementing the conceptual model into the simulation model. This increases the confidence of the model. Next, calibration of the model is assessed in subsection 4.6.2.

4.6.2. Calibration

Although not addressed by Robinson [51], Ahmed et al. [2] mentions that calibration plays an important part in modeling since model parameters are sometimes assumed and these parameters directly affect the simulation model. By performing calibration, these assumed parameters can be tuned such that the simulation outcomes are more realistic. However, the tuning of the parameters must be realistic to not decrease the model's validity. Table 4.16 presents the input variables for calibration of the model parameters and the respective outcomes for KPI1 (Total Makespan) and KPI3 (Throughput). KPI2 (Avg. Waiting Time per Truck) is not regarded since no empirical data on this metric is available to calibrate the model to.

Parameter	Value	Units	KPI1	KPI3
<i>TruckDockingTime</i>	Random.Triangular(3, 5, 7)	[min]	126.095	164.86
<i>PU_DO_EPT</i>	Random.Triangular(20, 30, 40)	[sec]		
<i>EPT_Speed</i>	Random.Triangular(3, 5, 6)	[km/h]		
<i>TruckArrivalTime</i>	Random.Triangular(0, 10, 20)	[min]		
<i>KPI1 = TotalMakespan, KPI2 = Throughput</i>				

Table 4.16: Parameter calibration

When inputting the parameters as specified in Table 4.16, the simulation results are shown in Figure C.7. Following the Student's T-Test, the observed differences in mean are not significant. This means that the specified parameters provide simulation results that do not differ significantly from the empirical data.

4.6.3. Data validation

As mentioned by Robinson [51] and seen from chapter 3, real-world data may be inaccurate. Table 3.9 shows that the individual total cross-docking time has a negative value. Negative individual cross-docking times (makespans) are impossible since a truck cannot depart before arriving. This shows that the data set available is not perfect, and flaws exist in the data set. The effects of inaccurate data are mitigated by removing outliers using box plots. This increases the validity of the data set. However, since the outlier removal discards outlier observations, the holistic view of the data set is reduced.

Additionally, the data set available consists of manually inputted data, and no input validation is performed on the inputs. This increases the probability of invalid data points which also can reduce the data validity.

As seen in Figure 3.8, the total makespan increases as the total number of pallets cross-docked increases. This is expected behavior since the total internal handling time is proportional to the total number of pallets cross-docked. Variance in the total makespan could be explained by uncertainty of arrival times.

Finally, the empirical data set available is only relevant to cross-docking operations similar to the case under study. This is a downside of case studies in general. For this case study, the cross-docking system is not considered a pure cross-docking facility since roughly 70% of the throughput is injected directly at the CDF. Therefore, the model cannot be validated for pure cross-docking systems. It is interesting for future research to compare different case studies. Although this simulation model is not validated for pure CDFs, the experiments

4.6.4. Conceptual model validation

Robinson [51] mentions that no formal methods exist for validating a conceptual model. However, outlining the project's objectives and establishing a core system understanding helps construct a conceptual model. A logic flow diagram is constructed to validate the conceptual model in this research, together with the CDF terminal managers, as shown in section A.7. This logic flow diagram forms the basis of processes and decisions in cross-docking operations of the case under study. Since the logic flow diagram of Figure A.5 is validated by discussions with the CDF managers, and the conceptual model shown in Figure 3.3 is constructed from the logic flow diagram, the conceptual model is simultaneously validated as well.

As seen in Figure A.5, terminal managers mentioned that in some cases, unit loads are moved to the backlog when the number of pallets exceeds the capacity of an outbound truck. However, the pallets in backlog are picked up by dedicated trucks during the same day and are not stored in the terminal until the next cross-docking moment. Therefore, the cross-docking system can be seen as starting and ending empty. Therefore, no warm-up period is modeled.

4.6.5. White-box validation

Appendix B is dedicated to describing the simulation model in a high level of detail. Using this detailed description, visual checks are performed to ensure white-box validation.

When regarding the simulation model description in Appendix B, it is seen that the model consists of the process steps as identified in Figure 4.3. Since the simulation model must be consistent with the conceptual model.

4.6.6. Black-box validation

In black-box validation, the overall behavior of the model is addressed by assessing whether the model outputs are sufficiently similar under the same inputs as the real-world system. The black-box validation process is shown in Figure 4.16.

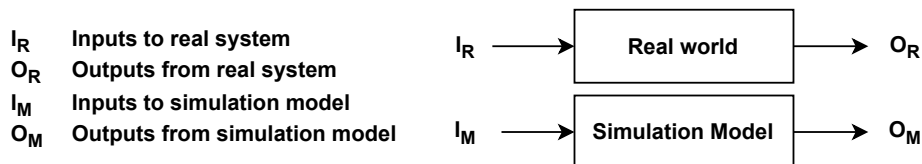


Figure 4.16: Black-box validation [51]

Since the real world and the simulation model are prone to the uncertainty of parameters, one-on-one comparison of input and output values is difficult. Hence, multiple replications are made for the base case scenario, and the outcomes' validity is assessed by performing statistical analysis. First, a statistical analysis is performed on the 'Throughput' and the 'Total Makespan'.

To examine whether the difference between the outcomes is statistically significant, a Student's T-Test with Bootstrapping and chi-squared goodness of fit test is performed and presented elaborately in section C.2 in Appendix C.

From the statistical analysis performed in section C.2, it is concluded with 95% confidence that the difference in means between the 'Throughput' and 'TotalMakespan' of the empirical data and the simulation generated data is not statistically significant. Regarding the goodness of fit, the 'TotalMakespan' of the simulation data shows a very good fit with the empirical data ($p = 0.8235$). Similarly, the 'Throughput' of the simulation data differs insignificantly from the empirical data

($p = 0.9282$). Also, visually inspecting the histograms in Figure 4.17, the throughput and total makespan for the empirical and simulation data set show significant overlap.

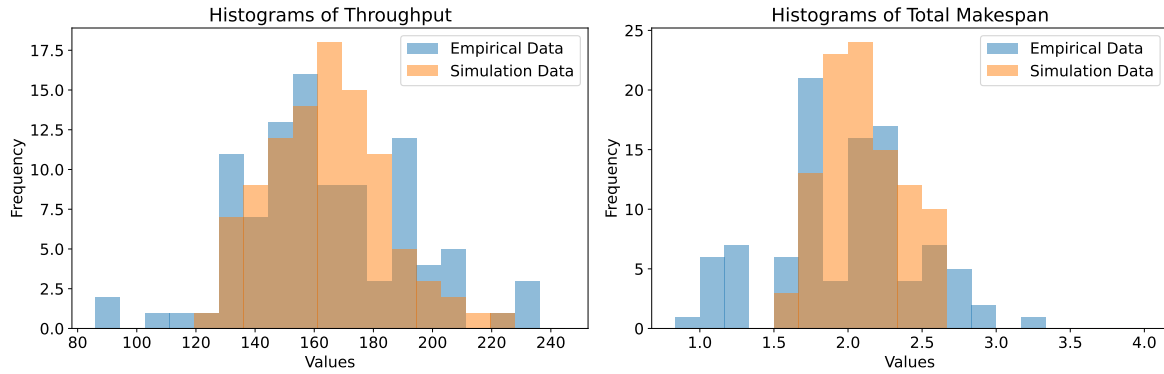


Figure 4.17: Histograms of empirical data and simulation generated base case data (copy from section C.2)

The number of replications executed for the statistical analysis in section C.2 is set to $N = 100$. Since this provided statistically insignificant differences for the 'TotalMakespan' and 'Throughput', this number of replications is also chosen for experimentation.

Spreading of the individual KPIs is not enough by itself, but the spread of the combinatory observations for 'TotalMakespan' and 'Throughput' is important. Therefore, Figure 4.18 presents the scatter plot for the empirical and simulation data.

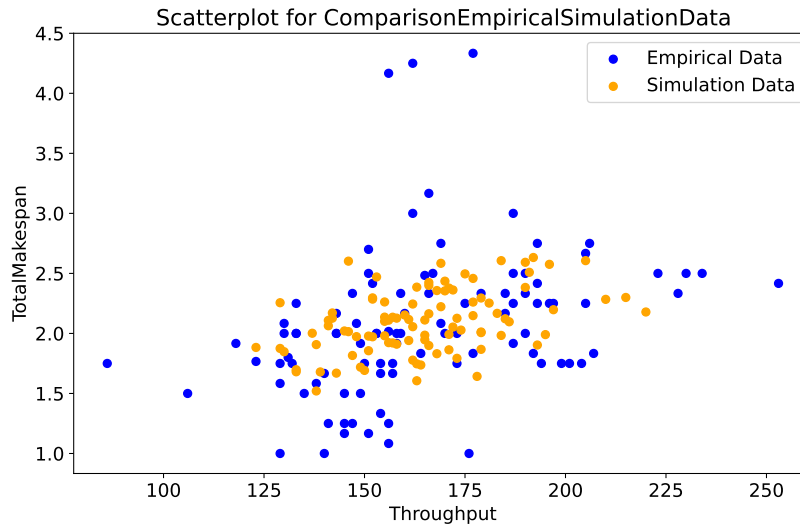


Figure 4.18: Scatter plot empirical data and simulation model data

Figure 4.18 shows that the simulation data falls well within the empirical data, which indicates that at least the mean for both 'Total Makespan' and 'Throughput' is similar. Although the spread of the simulation data is less wide than the empirical data, the simulation data does not show a linear correlation, which would be unexpected due to the system's stochastic nature.

4.6.7. Conclusion verification & validation

section 4.6 has shown that the constructed simulation model is sufficiently verified and validated for the case study to be used for experimentation. This means that the results obtained from the experiments can be regarded with sufficient confidence. Although the model cannot be validated for pure CDFs due to lacking pure CDF data, experiments will be run on pure CDF configurations for generalization.

4.7. Conclusion model formulation

To develop a model for testing the impact of the different design alternatives, a simulation model is developed. This type of modeling is best known for capturing the interdependencies in the system described in chapter 3 such as arriving trucks, occupancy of resources, and queuing. Additionally, DES can capture possible congestion to a great extent.

When determining the requirements of the model, the simulation model must be able to model two-stage cross-docking and roll containers as the two design alternatives. Hence, the processes and activities to be modeled identified in chapter 3 are extended to be truck and load carrier generation (1), receiving of incoming trucks (2), unloading (2), moving to stage areas (3), reloading (4), and dispatching (5).

Models are never 100% accurate due to the need to make assumptions while modeling. In this research, one of the assumptions is that the truck-to-dock assignment is fixed and doesn't change. Trucks can start unloading after docking at their assigned dock and after the driver has been registered, which takes a specified random triangular distributed time. Backlog is always transported by additional trucks not involved in cross-docking, so it doesn't affect cross-docking operations. The cross-docking terminal starts empty every day and a warm-up period is not required. The arrivals of trucks and the number of ULs per truck follow random triangular functions. All loading docks have the same characteristics and have identical performance. Only EPTs are used, and there are the same number of EPTs as dock doors. The movement speed of EPTs follows a random triangular function due to acceleration and deceleration. All EPTs and Workers are ready for operation at the start of the simulation.

Additionally, the model must measure the performance of the design alternatives by recording the key performance measures, which are identified to be the *Total Makespan*, *Individual Waiting Time*, and *Throughput per hour*.

Implementation of the simulation model is executed in SIMIO, which is an *object oriented* simulation modeling software. This method suits the problem since underlying data regarding the origin, destination, number of entities, travel time, and operational time underlying the model objects such as ULs, trucks, Transporters, and Servers ((un)loading docks and staging areas). Next, the implementation is performed according to the previously determined processes and activities.

After verification and validation, the conceptual model and the computer model are verified. Additionally, the model parameters are calibrated to ensure proper functioning, and the validity of the input data is assessed. Next, the conceptual model is validated using visual checks and tracing. Additionally, white-box and black-box validation is performed. The black-box validation showed with 95% confidence that the difference in means for the '*Throughput*' and '*Total Makespan*' between the simulation data and the empirical data is statistically insignificant. Furthermore, statistical analysis showed insignificant differences in the distribution of the empirical and simulation data. These verification and validation steps increased the confidence placed in the simulation model. However, since pure CDF data is lacking, the model's validation for pure CDFs cannot be ensured. Nonetheless, experiments will be conducted on the case study and pure CDF configurations to achieve generalization.

Experimentation and results

In this chapter, the experiments and results are presented. First, the definition of the scenarios is discussed in section 5.1. Next, the design alternatives are specified in section 5.2, which leads to an experimental design in section 5.3. Furthermore, the experimentation results are presented in section 5.4. Finally, the results are discussed in section 5.5

5.1. Scenario definition

Scenarios must be defined to construct an experimental design for which the alternatives can be tested. Law, Kelton, and Kelton [34] mentions that 2^k *factorial designs* can be constructed to determine how factors affect the responses in simulation. Experimenting by varying all scenario factors is also called full-factorial experimentation design. For this research, a full-factorial experimentation design is chosen to assess the influence of the input factors.

The input variables are determined using the IDEF0 A-0 level representation as shown in chapter 4. The input variables chosen are depicted in Figure 5.1.

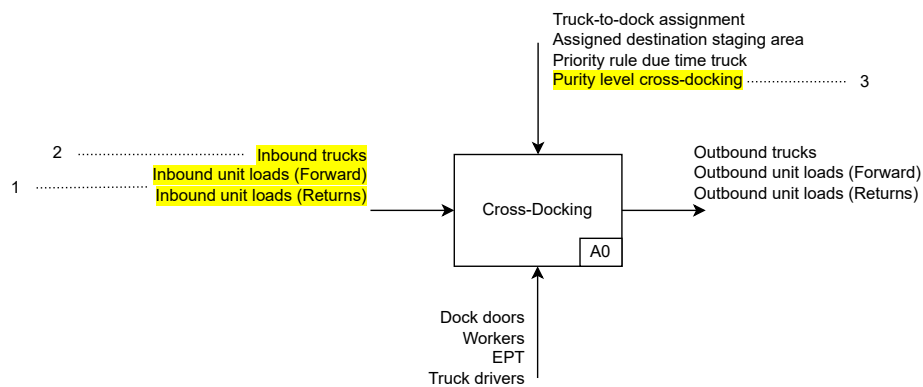


Figure 5.1: Scenario definition from IDEF0 A-0 level

The first input parameter identified is the number of *inbound unit loads* (1). Using this input parameter, multiple demand levels can be constructed for experimentation. The second input parameter is the *arrival time of the inbound trucks* (2), and subsequently, multiple arrival time windows can be constructed for experimentation.

Based on the throughput level data of the case study presented in subsection 3.4.1, three stochastic input parameter levels are chosen for the demand levels in the simulation runs. The demand levels can be either high, medium, or low. To find the distribution of the total number of pallets, statistical analysis is performed as described in Appendix C. From this analysis, the Logistic distribution was regarded to fit the data best. Moreover, the fitted distribution represents the medium unit load demand

scenario. Next, the demand distribution is multiplied by 1.5 and 0.5 respective scenarios to construct a hypothetical high-demand and a low-demand scenario. For all three demand-level scenarios, the distribution scale remains as shown in Table 5.1.

Next, as seen from subsection 3.4.2, the uncertainty of the arrival time can be either high, medium, or low. The stochastic arrival time input parameter represents the narrowness of the confidence of inbound trucks arriving simultaneously. The arrival times follow a random triangular distribution derived from Figure 3.7.

The medium uncertainty level input parameter is considered the base case. It is represented by the first quartile, median, and third quartile as the minimum value, median value, and maximum value, respectively. Naturally, travel times can only be positive values. Therefore, according to the distribution, the minimum travel time is considered the earliest possible arrival. Hence, the earliest arrival time is at $t = 0$, which naturally translates to the minimum value of the travel time being 0 minutes. Then, according to Figure 3.7, the average truck arrives at $t = 10$, and the latest trucks arrive at $t = 20$. This results in truck arrival times having an average arrival time surrounded by 10 minutes of arrival time possibilities in the time window.

Arbitrarily, widening and narrowing of the arrival time window is performed by multiplying the arrival time window by 0.5 and 1.5 for the narrow and wide scenario, respectively.

Table 5.1: Input parameter specification

Demand volume level		Uncertainty arrival time [min]		Pure cross-docking level	
High	$1.5 \times \text{Random.LogLogistic}(162.393..., 9.786...)$	Wide	$1.5 \times \text{Random.Triangular}(0, 10, 20)$	Yes	100%
Medium	$\text{Random.LogLogistic}(162.393..., 9.786...)$	Medium	$\text{Random.Triangular}(0, 10, 20)$	No	30%
Low	$0.5 \times \text{Random.LogLogistic}(162.393..., 9.786...)$	Narrow	$0.5 \times \text{Random.Triangular}(0, 10, 20)$		

When examining the input parameter specification in Table 5.1, the scenarios to be modeled for the 30% cross-docking purity level can be identified as follows:

- Scenario 1: High Demand, Wide ATAs
- Scenario 2: Medium Demand, Wide ATAs
- Scenario 3: Low Demand, Wide ATA
- Scenario 4: High Demand, Medium ATAs
- Scenario 5: Medium Demand, Medium ATAs
- Scenario 6: Low Demand, Medium ATAs
- Scenario 7: High Demand, Narrow ATAs
- Scenario 8: Medium Demand, Narrow ATAs
- Scenario 9: Low Demand, Narrow ATA

Scenario 10 to Scenario 18 follow the same pattern as stated above but are modeled for pure cross-docking (100% purity).

Next, the design alternatives are specified.

5.2. Design alternative specification

The complete alternative design modeling steps are elaborated on in Appendix B. This section briefly discusses the key parameter differences between the three design alternatives.

The three design alternatives consist of the referential state (Current State) and the two alternatives as designed in chapter 4. Hence, the three system alternatives to be modeled are the following:

- **A0: Single-stage CD with pallets (Current State)** - This alternative is the current state which imposes a single-stage cross-docking alternative with pallets as a consolidation method and EPTs as an internal transportation method.

UL_type = Pallets
 Capacity Truck = 33
 Transporter = EPTs
 Staging level design = Single-Stage

- **A1: Two-Stage CD with Pallets** - This alternative is the first alternative that imposes a two-stage cross-docking alternative with pallets as a consolidation method and EPTs as an internal transportation method.

UL_type = Pallets
 Capacity Truck = 33
 Transporter = EPTs
 Staging level design = Two-Stage

- **A2: Two-Stage CD with Roll Containers** - This alternative is the second alternative that imposes a two-stage cross-docking alternative with roll containers as a consolidation method and Workers as an internal transportation method.

UL_type = Roll Containers
 Capacity Truck = 55
 Transporter = Workers
 Staging level design = Two-Stage

Appendix B states the alternative specification in detail. Now that the alternatives are specified, the experimental design is concluded in section 5.3.

5.3. Experimental design

Since the input parameters and system alternatives are determined, the following experimental design can be constructed as shown in Table 5.2.

Table 5.2: Experimental Design

Scenario	Input			Response								
				A0			A1			A2		
	Demand	Time window	CD Purity	KPI1	KPI2	KPI3	KPI1	KPI2	KPI3	KPI1	KPI2	KPI3
1	High	Wide	30%									
2	Medium	Wide	30%									
3	Low	Wide	30%									
4	High	Medium	30%									
5	Medium	Medium	30%									
6	Low	Medium	30%									
7	High	Narrow	30%									
8	Medium	Narrow	30%									
9	Low	Narrow	30%									
10	High	Wide	100%									
11	Medium	Wide	100%									
12	Low	Wide	100%									
13	High	Medium	100%									
14	Medium	Medium	100%									
15	Low	Medium	100%									
16	High	Narrow	100%									
18	Low	Narrow	100%									
17	Medium	Narrow	100%									

A0 = Current State, A1 = Two-Stage Pallets, A2 = Two-Stage Roll Containers

KPI1 = Total Makespan, KPI2 = Avg. Waiting Time Truck, KPI3 = Throughput level

Due to the stochasticity of the input variables, replications are needed to assess the system's behavior. Since cross-docking operation systems typically start empty and end empty, no warm-up period is needed for the replications. The number of replications per experiment needed to ensure that the responses match the 95% confidence interval is addressed during verification in section 4.6 and was found to be 100 replications per scenario.

Next, the simulation experiments are run, and the results are presented in section 5.4.

5.4. Results

The simulations are run according to the experimentation plan discussed in section 5.1. The computational efforts are performed on a computer with the characteristics shown in Table 5.3.

Characteristic	Specification
Processor	Apple M1 (2020)
OS	macOS 13.4
Memory (RAM)	8 GB

Table 5.3: Computer characteristics

The computer characteristics shown in Table 5.3 provide an overview of the computer's hardware and software configuration for the scientific research thesis. They contribute to the system's overall computational capabilities and performance, impacting the execution of research tasks and data analysis.

After running all experiments, it is seen that not all experiments show enough data points. Since 100 replications are run per scenario, the number of observations per experiment is expected to be 1800 per modeled alternative. However, only the experiments for A3 show observations close to the expected number of replications. A0 and A1 show a significant drop in the number of successful replications. High congestion levels likely cause this drop in the number of successful replications. As mentioned by Sturrock [58], deadlock situations can occur in SIMIO but can be minimized using certain solutions described in Appendix B. However, even after using the suggestions on the SIMIO Forum, deadlock situations still occurred when looking at Table 5.4. Deadlock situations are difficult to avoid completely [58]. Moreover, deadlock situations tell the modeler something about the change for congestion. Therefore, the number of successful replications is important to analyze before interpreting the results. The number of successful replications per experiment is shown in Table 5.4.

Using the Central Limit Theorem [25] as shown in section D.3, scenarios comprising less than 30 replications are discarded in the analysis due to their small sample sizes. This number will be the threshold value since the number of observations of these scenarios is not large enough to reliably draw conclusions or make statistical inferences with a high degree of confidence. Less than 30 replications are marked in Table 5.4.

Table 5.4: Number of successful replications per scenario for the alternatives

Scenario	Input			No. replications		
	Demand	ATA window	CD purity level	A0	A1	A2
1	High	Wide	30%	100	93	99
2	Med			100	99	99
3	Low			100	100	99
4	High	Medium	30%	100	97	99
5	Med			100	93	100
6	Low			100	100	99
7	High	Narrow	30%	100	91	99
8	Med			99	95	99
9	Low			100	96	100
10	High	Wide	100%	2	17	99
11	Med			74	32	100
12	Low			90	48	100
13	High	Medium	100%	69	21	100
14	Med			75	28	100
15	Low			84	35	100
16	High	Narrow	100%	75	24	100
17	Med			79	29	100
18	Low			83	41	100

A0 = Current State, A1 = Two-Stage Pallets, A2 = Two-Stage Roll Containers

Since A0 and A1 show less than the threshold for scenario 10, only A2 has valid results, and the alternatives in this scenario cannot be compared. Therefore, scenario 10 is discarded for further analysis. For scenarios 13, 14, 16, and 17, only a comparison can be made between A0 and A1.

The 30-50 successful replications per scenario indicate weaker validity of the results and a relatively high chance of congestion. Overall, the A3 alternative shows more resilience to congestion with almost all successful replications.

After discarding scenario 10, the resulting KPI averages of the simulation replications are obtained as presented in Table 5.5. All results are presented fully in Appendix D using box plots and by performing statistical analysis.

Table 5.5: Results

Scenario	KPI1 [hours]			KPI2 [hours]			KPI3 [UL/hour]			
	A0	A1	A2	A0	A1	A2	A0	A1	A2*	
Non-pure CD (30%)	1	3.13	2.39	3.02	1.04	0.69	0.64	80.87	105.48	86.08
	2	2.21	1.73	2.24	0.76	0.53	0.52	76.07	96.44	78.05
	3	1.39	1.16	1.38	0.50	0.40	0.39	63.93	75.95	66.57
	4	3.00	2.19	2.89	0.98	0.60	0.59	84.45	115.01	89.76
	5	2.06	1.56	2.06	0.69	0.44	0.44	81.53	107.06	84.67
	6	1.21	0.96	1.19	0.41	0.30	0.29	73.18	91.81	77.71
	7	2.93	2.22	2.83	0.95	0.61	0.56	86.33	112.94	91.86
	8	1.99	1.54	1.99	0.65	0.44	0.41	84.12	108.26	87.82
	9	1.12	0.90	1.11	0.37	0.27	0.26	78.84	98.04	83.32
Pure CD (100%)	11	4.73	2.53	5.26	1.08	0.89	1.02	61.33	115.70	58.41
	12	2.58	1.63	2.95	0.62	0.52	0.61	56.55	89.32	53.39
	13	4.05	3.89	9.42	1.44	1.37	1.65	107.76	111.10	47.00
	14	2.79	2.61	5.53	1.00	0.91	1.02	104.63	113.32	54.57
	15	1.70	1.54	2.88	0.55	0.47	0.57	85.65	94.22	55.27
	16	4.04	3.82	9.46	1.42	1.33	1.64	107.97	111.90	46.81
	17	2.73	2.59	5.85	0.97	0.90	1.04	107.15	113.14	51.35
	18	1.64	1.55	2.95	0.52	0.47	0.56	88.41	93.32	52.52

A0 = Current State, A1 = Two-Stage Pallets, A2 = Two-Stage Roll Containers

KPI1 = Total Makespan, KPI2 = Avg. Waiting Time Truck, KPI3 = Throughput level

Total run time = 68,787 seconds

*number of unit loads proportional to match load capacity pallets $[x \times \frac{33}{55}]$

The averaged results of A1 and A2 are compared to the Current State alternative (A0) performance for the non-pure cross-docking scenarios (30% CD purity). For each KPI, the comparative performance is color coded to present the relative change in Table 5.6.

Table 5.6: Change in KPIs the for non-pure cross-docking scenarios relative to A0 (the Base Case)

Scenario	KPI1			KPI2			KPI3			
	A0	A1	A2	A0	A1	A2	A0	A1	A2*	
Non-pure CD (30%)	1	0%	-24%	-4%	0%	-34%	-38%	0%	30%	6%
	2	0%	-22%	2%	0%	-30%	-31%	0%	27%	3%
	3	0%	-16%	0%	0%	-20%	-22%	0%	19%	4%
	4	0%	-27%	-3%	0%	-38%	-40%	0%	36%	6%
	5	0%	-24%	0%	0%	-35%	-35%	0%	31%	4%
	6	0%	-20%	-2%	0%	-28%	-29%	0%	25%	6%
	7	0%	-24%	-4%	0%	-36%	-41%	0%	31%	6%
	8	0%	-23%	0%	0%	-33%	-37%	0%	29%	4%
	9	0%	-20%	-1%	0%	-27%	-31%	0%	24%	6%
Average		-22%	-1%		-31%	-34%		28%	5%	
Pure CD (100%)	11	0%	-46%	11%	0%	-18%	-6%	0%	89%	-5%
	12	0%	-37%	14%	0%	-17%	-3%	0%	58%	-6%
	13	0%	-4%	132%	0%	-5%	15%	0%	3%	-56%
	14	0%	-6%	98%	0%	-9%	2%	0%	8%	-48%
	15	0%	-10%	69%	0%	-15%	4%	0%	10%	-35%
	16	0%	-5%	134%	0%	-7%	16%	0%	4%	-57%
	17	0%	-5%	115%	0%	-8%	7%	0%	6%	-52%
	18	0%	-6%	80%	0%	-10%	8%	0%	6%	-41%
	Average		-15%	82%		-11%	5%		23%	-37%

A0 = Current State, A1 = Two-Stage Pallets, A2 = Two-Stage Roll Containers

KPI1 = Total Makespan, KPI2 = Avg. Waiting Time Truck, KPI3 = Throughput level

*number of unit loads proportional to match load capacity of pallets

For the non-pure CD (30%) scenarios in Table 5.6, it is seen that, on average, A1 outperforms A0 and A2. For the non-pure CD environment, A1 reduces the total makespan by 22% while increasing the average throughput by 28%. Also, A1 performs better than A2, as the total makespan of A2 differs insignificantly from the total makespan of A0 as seen in section D.2 in Appendix D.

Meanwhile, the *AverageWaitingTimePerTruck* decreases for both two-stage cross-docking alternatives. This is expected for the Two-Stage (EPT) alternative: since the total makespan decreases, the average waiting time per truck must also decrease if (un)loading is faster. Subsequently, since the average total makespan does not change when comparing A0 to A2 but the average waiting time per truck does, this decreasing average waiting time for A2 suggests that the unloading and reloading of trucks are more time-consuming while using roll containers than pallets.

Additionally, the total throughput per hour naturally decreases as the demand levels decrease. Therefore, it cannot be stated at what demand level the system is capacitated. It is important to note that for system A1, the total throughput per hour is higher for Medium ATA Time Window scenarios (Scenario 4, 5, and 6) compared to the High ATA Time Window scenarios and Narrow Time Window scenarios. Similarly, the total makespan of A1 compared to A0 is lower for the Medium ATA Time Window scenarios than for the Wide and Narrow Time Window Scenarios. This confirms the findings in the literature that suggest that spread in arrival times improves the allocation of resources and therefore the performance of a CDF (i.e. total throughput or makespan).

For the pure CD (100%) environment, scenarios 11 to 18 show an average reduction of the total makespan of 15% when using A1. The highest reduction in total makespan is seen for scenarios 11 and 12, which are both wide ATA window scenarios. On average, A2 results in a significant increase of 82% in KPI1 (total makespan). Therefore, A1 outperforms A0 and A2 in pure CD environments regarding the total makespan.

Regarding KPI2, the average waiting time per truck, A1 provides a reduction of 11% compared to A0, whereas A2 leads to an average increase of KPI2 of 5%. For wide ATA window scenarios 11 and 22, A2 performs better than A0 with a 6% decrease in KPI2 for the pure CD configurations.

When examining KPI3, A1 shows the largest increase (>58%) in throughput for wide ATA window scenarios 11 and 12. For scenarios 13 to 18, the throughput is less increased between 3% and 10%.

To pair-wise compare the results for the KPIs, scatter plots for the non-pure cross-docking scenarios are presented in Figure 5.2.

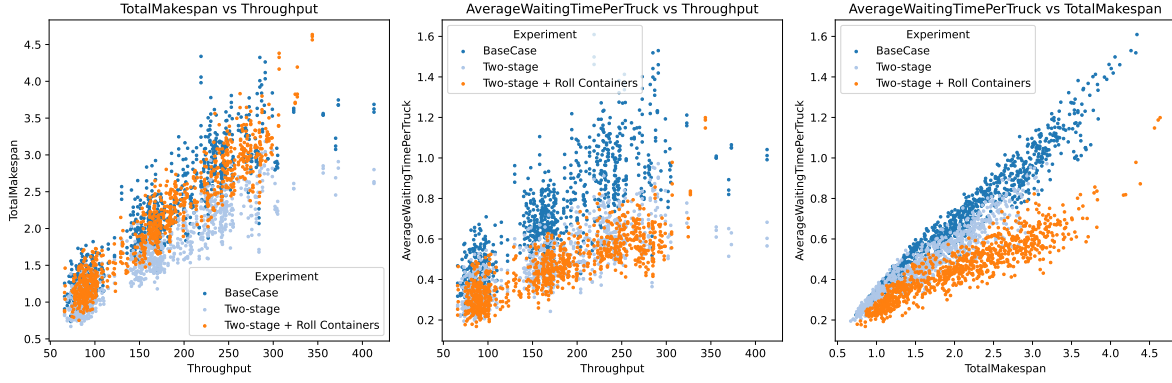


Figure 5.2: Scatter plots of results for non-pure CD

Interestingly, the total makespan is similar for the A0 and the A2 in all scenarios, whereas A1 is superior. Following the statistical analysis in section D.2, the difference in Total Makespan between the A0 and A2 is insignificant. However, the significance of differences between A0 and A2 decreases for higher throughput levels, indicating that two-stage cross-docking with roll containers can become more efficient for higher demand levels.

Regarding the average waiting time, Two-Stage cross-docking with roll containers yields lower average waiting times than the base case and two-stage cross-docking with EPTs. Since the total makespan of two-stage cross-docking with roll containers is not statistically different from the base case total makespan, the lower waiting times can be due to longer unloading and reloading times. Hence, under the given inputs, unloading and reloading roll containers is slower compared to (un)loading pallets.

5.4.1. Non-pure CD configurations

Since the main KPI in cross-docking is considered the makespan, the following analysis mainly concerns the behavior of the total makespan under the different input parameters (demand, arrival times, and purity of cross-docking). *Total makespan* and *throughput per hour* are considered to be the most important metrics, whereas *average truck waiting time* is a less significant metric.

Figure 5.3 depicts the histograms of total makespan found in the non-pure cross-docking scenarios (Scenario 1 to Scenario 9).

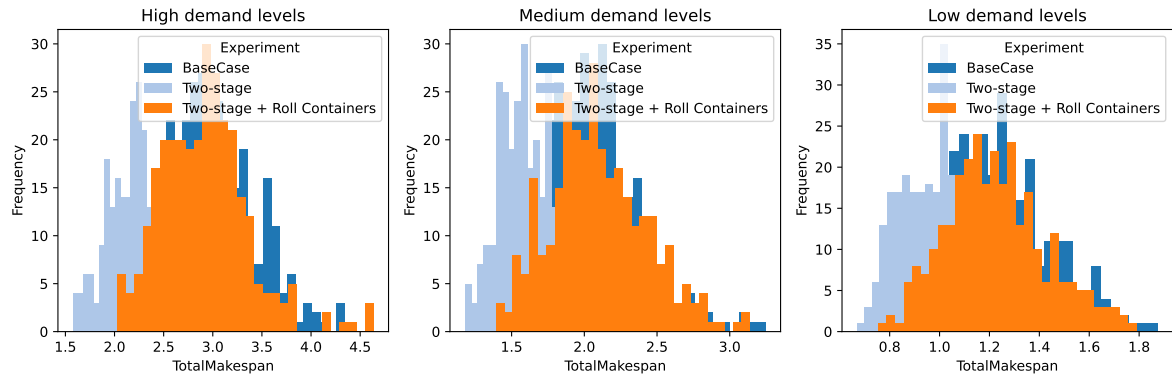


Figure 5.3: Histograms of total makespan for A0, A1, and A2 for non-pure CD

Referring to the experimental design in Table 5.2, the most left histogram presents the high-demand scenarios (Scenario 1, Scenario 4, and Scenario 7). The mid-positioned histogram depicts the medium

demand levels (Scenario 2, Scenario 5, and Scenario 8). Finally, the low demand levels (Scenario 3, Scenario 6, and Scenario 9) are presented in the right histogram.

When solely regarding the input parameter of the demand levels in Figure 5.3, it can be seen that the average total makespan decreases with decreasing demand as expected. This is also seen in Table D.1, where Scenario1, Scenario2, and Scenario3 show decreasing t-statistics for the two-stage cross-docking with pallets alternative under. This means that for higher throughput, the benefit of two-stage cross-docking increases.

Additionally, the variance in total cross-docking time increases under higher demand levels. This is seen in the bandwidth of the x-axis for the three histograms shown in Figure 5.3. Since each pallet induces uncertainty in internal handling time, the bandwidth of the total makespan will likely increase when the total demand increases. This is seen in all three alternatives.

Interestingly, two-stage cross-docking with roll containers performs marginally better (2%) than the base case. From section D.2 it is seen that the difference in total makespan for the roll-container alternative is statistically insignificant for all non-pure cross-docking scenarios. This can be because, in the case under study, a large share of the unit loads is pre-sorted at the CDF, which reduces the potential benefit of roll containers in cross-docking processes.

The second input parameter for which the KPI Total Makespan is analyzed is the Arrival Times (Wide, Medium, and Narrow).

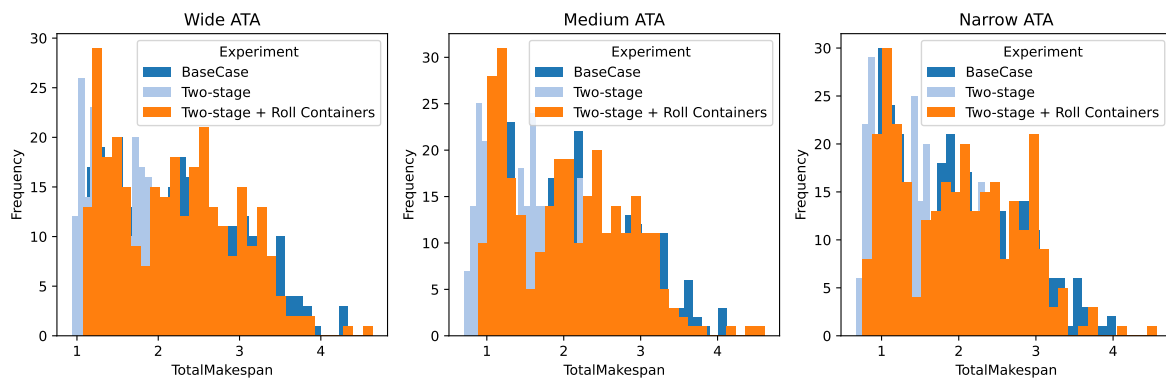


Figure 5.4: Histograms of total makespan for three experiments under varying width arrival time window

To address the behavior of the two input parameters on the total makespan, the relative influence of the input parameters on the total makespan is examined by comparing the two alternatives to the current state using a pair-wise Student's T-test as shown in Table 5.7. The Student's T-test addresses the difference in means which is represented by the T-test value (or t-statistic) and expresses the significance of the found difference using the p-value. All pure cross-docking results are discarded for analysis due to incomplete simulation results. Hence, the outcomes of the analysis only represent the case study.

Table 5.7: T-test values of Total Makespan for A0 (Current State) versus A1 (Two-Stage Pallets)

Demand level	Arrival time window		
	Wide	Med	Narrow
High	13.36659	15.15708	13.22521
Med	12.98546	14.51430	13.66831
Low	9.50148	12.31263	11.59134
<i>only significant values included</i>			

Table 5.7 and Table D.1 show that for all t-test values for comparing the base case to the two-stage cross-docking alternative, the differences in total makespan are found to be statistically significant.

Regarding the T-test values shown in Table 5.7, the difference in performance is larger for the medium wideness arrival time windows than for narrow or wide arrival time windows in the case of two-stage cross-docking with pallets. This is expected since trucks arriving simultaneously is inefficient

when the number of trucks to be served exceeds the number of dock doors available. This indicates that the spread in arrival times helps spread resource use and operate more efficiently. This is the theoretical basis for truck scheduling optimization.

As discussed in section D.2, the T-test values for comparing the total makespan results of the base case versus two-stage cross-docking with roll containers are insignificant and, therefore, the difference in means cannot be included in the analysis.

5.4.2. Pure CD configurations

Finally, the histograms for comparing the non-pure cross-docking scenarios to the pure cross-docking scenarios are presented in Figure 5.5. Regarding Table 5.4, the pure CD scenarios have shown to be more prone to congestion since the number of successful replications drop for A0 and A1. Figure 5.5 shows that for pure CD configurations, the total makespan increases drastically compared to non-pure CD configurations. Additionally, A2 (Two Stage + Roll Containers) shows a more increased total makespan than A0 and A1 in pure CD scenarios. This was also seen in Table 5.6, where A2 showed an average increase of the total makespan of 82%.

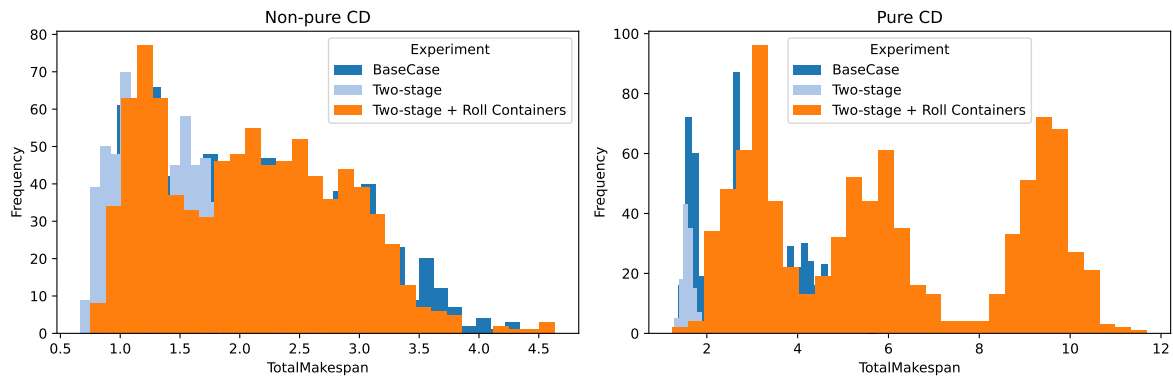


Figure 5.5: Histograms of total makespan for three experiments under varying cross-docking level

Now that the simulation results are presented, the results are discussed.

5.5. Discussion

A detailed case study was conducted at Budbee's cross-docking terminal to address the main research question. First, the results of the simulation testing are discussed. Next, the limitations and strengths of this research are discussed.

The study's findings are two-fold since non-pure cross-docking configurations and pure cross-docking configurations were examined. Regarding the non-pure cross-docking, this study demonstrates that the designed staging level significantly influences the performance of cross-docking operations in uncertain conditions. Specifically, when faced with uncertainty, the utilization of two-stage cross-docking with pallets can lead to a significant reduction in the total makespan by an impressive 22%. Moreover, this approach could also boost the average throughput per hour by an impressive 28% while reducing the waiting time of the truck drivers by 31%.

Regarding the pure cross-docking scenarios, the results show that two-stage cross-docking with pallets (A1) outperforms both single-stage cross-docking with pallets (A0) and two-stage cross-docking with roll containers (A2) by, on average, reducing the total makespan with 15%, reducing the average truck waiting time by 11%, and increasing the total throughput by 23%. However, the number of successful replications dropped for pure cross-docking configurations, likely due to congestion issues not being solved in the model. Therefore, it must be noted that the results obtained for pure cross-docking configurations are weak and biased.

Since this study focused on improving performance metrics instead of cost modeling, the costs for the alternatives can only be speculated. Regarding the costs of the alternatives, this study assumed that 55 roll containers or 33 Euro pallets fit into a truck. This means that 40% more roll containers are

needed to transport the same number of parcels compared to Euro pallets. Given the fact that roll containers are over 3 times more expensive than Euro pallets, the additional CAPEX (capital expenditures) costs of operating with roll containers are 4.2 times higher than operating with Euro pallets in similar conditions. However, this 4.2 times increase in CAPEX can break even in case the OPEX (operational expenditures) are decreased enough. This can be decreased by reducing the makespan of operations and therefore reducing the number of worker hours needed for operations. Two-stage cross-docking can require two times the square surface of the total single-staging area, increasing the CAPEX of two-stage cross-docking by a multiplier of 2.

The study's findings provide valuable insights, but the precise performance improvements observed are specific to the case study parameters. While results are based on one facility, the trends likely apply directionally to other cross-docking terminals dealing with uncertainty. However, expected benefits will vary based on factors like facility design, product types, demand patterns, automation level, etc.

Additionally, it is worth noting that consolidation methods have a notable impact on cross-docking performance, particularly on intra-terminal congestion. However, the purity of cross-docking operations must also be considered, as it can influence the number of intra-terminal transshipments and the likelihood of intra-terminal congestion. Therefore, the consolidation method of choice must be carefully selected, considering the purity of cross-docking. Overall, this study's findings provide valuable insights into improving cross-docking operations, especially when faced with uncertain conditions.

Next, the limitations and strengths of the research are discussed, including areas of interest for future research. Due to assumptions and simplification, the developed simulation model has limitations. The findings and results are explored by examining the decisions made during the research process. Future studies could focus on the following areas to address any limitations identified.

First and foremost, this study proved capable of testing for hybrid CDF configurations. However, the results obtained for pure cross-docking configurations are weak due to the reduced number of successful replications caused by the model's unsolvability of congestion effects. Hence, this research stipulated the importance of methods for modeling intra-terminal congestion. The conceptual model used in this research could be more effective when accounting for extensive procedures for coping with deadlock situations. Therefore, the results of this research are limitedly generalizable for pure cross-docking configurations. However, in practice, many CDFs operate as hybrid CDFs and few cross-docking terminals can be considered pure CDFs. Being a case study, this research comprised one single case and did not set out to study pure CDF configurations due to the lack of available data. Furthermore, since this research is conducted for a hybrid CDF and many real-world CDFs are considered hybrid CDFs, this model was capable of investigating the impact of internal transportation methods on the performance of non-pure cross-docking configurations. Incorporating data from other CDFs is necessary to guarantee the accuracy of this model in various situations.

This leads to the second point of discussion. Since this research is limited to one case only, generalization of the results is limited. Moreover, the case study operated on an arbitrarily low level of cross-docking purity. Since, in practice, no CDF operates according to pure cross-docking characteristics, it is hard to determine to what extent these research results apply to other CDFs. Future research can point out a framework or scale to which the cross-docking purity levels of CDFs are scaled and classify potential best practice strategies according to their cross-docking purity levels.

Interestingly, this research suggests that two-stage cross-docking outperforms single-stage cross-docking for this case study. However, Gue and Kang [23] found that single-stage cross-docking yielded higher throughput than the two-stage alternative. This could be because Gue and Kang [23] used exponential inter-arrival and service times are unrealistic for cross-docking situations in which arrival times concentrate at a given point in the day. The system studied by Gue and Kang [23] is M/M/1, whereas in this research, deterministic inter-arrival times are considered, and the operations center at one point in the day. Additionally, the case study CDF consisted of multiple servers in the form of dock doors.

This research used triangular distributions for truck docking times, pick-up and drop-off times, and transportation times. This is due to lacking information on parameter distributions. It is recommended that Budbee and future research address the actual distribution of processing times in their operations. To this end, monitoring of operations and data analysis must be conducted. By doing so,

more information on KPIs and processing times is gathered to identify bottlenecks in operations.

While this research focused on the total makespan and throughput per hour as the primary performance indicator, analyzing individual makespans can provide deeper insights into the performance of specific stages or processes within the cross-docking operation. To decrease the highest valued time factors, future research must be done on truck drivers' perceived hindrances in cross-docking operations. This research can be done using choice modeling. It could help identify perceived bottlenecks for truck drivers or areas of improvement within the cross-docking process, allowing for targeted optimization strategies to improve the perceived hindrance in operations for truck drivers.

This study revealed that not all pallets could be strictly classified as forward or return shipments in real-world cross-docking. Mixed flow pallets, which require additional sorting at the cross-dock, are standard in practice. Roll containers offer greater flexibility in handling such mixed flows, as they allow for easier separation of different types of shipments due to their reduced batch sizes compared to pallets. Unfortunately, this research did not yield good results in comparing roll containers with pallets in pure cross-docking scenarios directly. However, investigating the impact of roll containers on reducing the total makespan in systems involving sorting mixed pallets would be an exciting avenue for future research. This analysis could explore whether smaller batch sizes achieved through roll containers lead to a decrease in the total makespan when sorting mixed pallets is included. Comparing roll containers and pallets in scenarios with the complexities introduced by additional sorting tasks would provide insights into their suitability, efficiency, and potential advantages or limitations in pure cross-docking operations.

Additionally, the findings demonstrated a significant drop in average waiting time for the two-stage cross-docking with roll containers compared to the same scenarios with pallets. This observation suggests increased unloading and reloading time associated with roll containers. Further investigation is warranted to understand the underlying factors and dynamics that lead to this difference in waiting times between the two consolidation methods. Future studies could delve into the specific processes and operations that contribute to waiting time and explore strategies to optimize this aspect further.

Although this research did not incorporate due times for dispatching, it is important to recognize the significance of due times in real-world logistics operations. Incorporating truck due times in future research would provide valuable insights into the effect of various cross-docking strategies on meeting delivery deadlines at regional terminals and minimizing delays further in the supply chain.

During this research, failed simulation runs were encountered in pure cross-docking scenarios, indicating the presence of deadlock situations caused by congestion. Therefore, the conceptual model used in this research lacks processes or rules that can be instigated to cope with such deadlock situations. In future research, deadlock situations must be countered by effective processes. Additionally, understanding how different cross-docking strategies behave in such systems is essential, as congestion can significantly impact operational performance. Future research could focus on studying the behavior and performance of alternative consolidation methods in pure cross-docking situations with potential congestion issues. This investigation would provide valuable insights into the effectiveness and robustness of different strategies under challenging operational conditions.

Additionally, efficient scheduling plays a critical role in optimizing cross-docking operations. We found that ensuring a spread in arrival times is essential for maximizing resource utilization and minimizing congestion. These strategies could balance the arrival times of shipments while optimizing the utilization of cross-docking facilities and minimizing waiting times.

Incorporating due times, developing advanced scheduling strategies, considering mixed flow pallets and sorting, analyzing individual makespans, studying deadlock situations, and comparing roll containers with pallets in pure cross-docking scenarios are potential avenues for future research. Addressing these areas will enhance our understanding of cross-docking operations and improve transport, infrastructure, and logistics systems.

Now that the results are presented and discussed, the next chapter presents the conclusion of this research.

6

Conclusion

In this chapter, the key findings of this research are concluded by answering the main research question in section 6.1. By answering the main research question, the scientific relevance of this research is touched upon in section 6.2. Second, the scientific relevance of this research is emphasized in section 6.2. Next, recommendations for future research are discussed in section 6.3. Finally, recommendations for Budbee will be made by reflecting on the objective of the research in section 6.4.

6.1. Key findings and research question

This section presents the key findings of this study by answering the research questions stated in chapter 1. This research aimed to answer the following main research question:

“How to improve cross-docking operations under uncertainty?”

The key findings of this study suggest that the performance of cross-docking operations under uncertainty is highly affected by the designed staging level. For non-pure cross-docking configurations, two-stage cross-docking with pallets can reduce the total makespan by 22% and improve the average throughput per hour by 28 %. Design choices in load carrier type may enhance the performance of cross-docking performance due to effects on intra-terminal congestion.

For pure cross-docking configurations, two-stage cross-docking with pallets performs better than single-stage cross-docking with pallets and two-stage cross-docking with roll containers. It reduces total makespan by 15%, truck waiting time by 11%, and increases total throughput by 23%. However, congestion issues were not solved in the model, resulting in fewer successful replications. Hence the results for pure cross-docking configurations are weak and biased. Therefore, the results show that the purity of cross-docking operations impacts the number of intra-terminal transshipment and, subsequently, the probabilities of intra-terminal congestion, making the consolidation method dependent on the purity of cross-docking.

This study did not focus on cost modeling, so the costs of alternatives can only be approximated. Assuming 55 roll containers or 33 Euro pallets fit in a truck, it takes 40% more roll containers to transport the same parcels as Euro pallets. Roll containers are over three times more expensive, so operating costs are 4.2 times higher than with Euro pallets. But if operational costs decrease by reducing worker hours, the higher CAPEX cost can break even. Additionally, two-stage cross-docking requires double the space and increases CAPEX by 2.

Regarding non-pure configurations, the study suggests two-stage cross-docking with pallets can potentially reduce makespan and increase throughput under uncertainty. These principles may extend to other hybrid facilities, but practical gains depend on individual operations. For pure cross-docking, two-stage with pallets also showed gains, but congestion issues reduced replications, so these results are limited. Still, the challenges encountered provide learnings about modeling pure cross-docking scenarios.

Since monetary costs were not modeled, alternative expenses were approximated qualitatively. But actual costs and break-even points will differ across facilities. Overall, the research demonstrates how

cross-dock design can mitigate uncertainty, though exact improvements require custom simulation. Classifying facilities on a purity spectrum could aid in generalizing suitable practices.

As a single case study, the results are not broadly representative. Incorporating data from diverse operations would improve accuracy for various scenarios. Comparing staging results to past literature highlights the role of modeling assumptions. Extending the model to other contexts could better validate findings before implementation. Surveying multiple sites could supplement the simulation.

While focusing on overall makespan and throughput, individual process analyses could provide additional insights for targeted improvements. Exploring complexities like mixed pallets, congestion dynamics, and scheduling would also advance understanding of real-world operations.

The trends observed help guide cross-docking practices under uncertainty but require translation based on local constraints. This research provides a foundation for tailoring future simulation studies, surveying operations, and benchmarking across the purity spectrum to improve generalization.

To derive these results, first, the following sub-question was answered.

What is a cross-docking terminal, and how can uncertainty be mitigated in literature?

Cross-docking facilities (CDFs) are physical facilities used in supply chains to perform cross-docking, which aims to minimize point-to-point deliveries of goods. Unlike a warehouse, a CDF typically holds the goods for less than 24 hours. The goods are often consolidated on load carriers which determine the internal transportation method. The main activities taking place in cross-docking are unloading trucks, sorting goods, and (re)loading trucks. Cross-docking facilities are terminals used for direct freight transfer and therefor allow limited temporary storage [31]. Cross-docking is embedded in external and internal uncertainty factors [6].

Uncertainty in arrival times, internal transportation times, and freight levels impact the expected performance of cross-docking facilities [30]. Cross-docking facilities mitigate external uncertainty by monitoring freight levels and truck arrival times. However, CDFs not actively monitoring or acting upon monitored data lose potential performance improvements such as truck scheduling optimization and truck-to-dock assignment. When external factors of uncertainty are not monitored, and terminal design decisions cannot be altered, the means for improving cross-docking operations are limited to staging-level design [8] and internal transportation methods [8, 40].

To examine how the findings in the literature are reflected in the case study, the following research sub-question is answered:

What is the current state of cross-docking at the CDF of Budbee?

In this case study, the CDF can not anticipate to monitored arrival times or freight levels due to a non-functioning WMS. The CDF uses mainly pre-distribution since most pallets are already sorted to their flow and destination, but some pallets are post-distribution as pallets are combined due to low freight volumes which results in mixed pallet contents. Additionally, the CDF under study has over 70% of its total transported freight pre-injected at the terminal. This means that less freight is cross-docked, reducing truck interdependencies. Hence, the case study cannot be seen as a CDF operating under pure cross-docking principles. This is also found by Apte and Viswanathan [5], who state that most warehouses using cross-docking rather operate as hybrid warehouses. The staging-level design and consolidation method are addressed in improving the performance of cross-docking operations for the case study. Two staging-level designs addressed in this research are single-stage cross-docking and two-stage cross-docking for pallets and roll containers as consolidation methods.

After addressing the current state of the cross-docking terminal under study and potential design alternatives, the following sub-question is answered:

How to develop a model for testing the design alternatives for a cross-docking facility under uncertainty?

To this end, a Discrete Event Simulation (DES) model is constructed due to the capability of simulation models to easily capture interdependencies and stochasticity of parameters found in uncertain external factors. The simulation model is also run for pure cross-docking scenarios to ensure the generalization of results.

Finally, the results of the simulation model and experiments are discussed by answering the following sub-question:

What are the effects of different design alternatives under the case study?

For non-pure cross-docking configuration, the simulation results show that two-stage cross-docking using pallets enables the logistical service provider to reduce the *total makespan* by 22% and to reduce the *average truck waiting time* by 31% while increasing the *throughput* by 28%. Two-stage cross-docking using pallets outperforms single-stage cross-docking using pallets and two-stage cross-docking using roll containers under configurations similar to the case study.

For pure cross-docking, two-stage cross-docking with pallets outperforms single-stage cross-docking with pallets and two-stage cross-docking with roll containers. It reduces makespan by 15%, truck waiting time by 11%, and increases throughput by 23%. Two-stage cross-docking using pallets outperforms single-stage cross-docking using pallets and two-stage cross-docking using roll containers under hypothetical pure configurations. However, congestion issues were not solved in the model, so results for pure cross-docking configurations are weak and biased.

Regarding the costs of the alternatives, roll containers are over three times more expensive than Euro pallets, which significantly increases the CAPEX of A2. However, the CAPEX increase can break even if OPEX is decreased by reducing makespan and worker hours. Additionally, two-stage cross-docking requires twice the square surface of the total single-staging area, increasing CAPEX by a factor of 2.

Congestion is more likely to occur when using pallets due to the blocking of terminal pathways by EPTs. Single-stage cross-docking is less likely to cause congestion using pallets than two-stage cross-docking due to decreased EPTs traveling to the same locations. However, the results suggest that congestion is almost non-existent when using roll containers. Additionally, the results suggest that narrowing the window of truck arrival times is less effective due to the benefits of resource occupancy in more wide arrival time windows, which advocates for truck scheduling optimization.

6.2. Scientific relevance

Scientific research is performed to fill an identified knowledge gap by studying a particular topic. This study aims to fill two research gaps and investigate new topics. This section addresses the two gaps that are addressed in this research and stresses the scientific relevance of this research.

In chapter 2, the scientific gap mentioned by Van Belle, Valckenaers, and Cattrysse [59] was the absence of a comparison of internal transportation methods. This research conducted a pair-wise analysis of two load carriers (pallets & roll containers) in cross-docking operations for a logistics service provider. This new analysis stresses the importance of selecting the appropriate load carrier for logistical systems such as cross-docking facilities.

This research not only analyzed non-pure cross-docking configurations but also pure cross-docking configurations which increases the generalizability of the results.

Additionally, little research has been conducted before on the effect of staging-level design in cross-docking. Cross-docking operations are often concentrated at one time of day, which differs from the assumption of Gue and Kang [23]. This research suggests that decisions in staging-level design depend on queuing theory factors that can differ for other cross-docking facilities.

6.3. Recommendations for future research

In conclusion, this research contributes to understanding cross-docking operations by comparing single-stage and two-stage approaches and evaluating the performance of pallets versus roll containers as consolidation methods. This case study found that two-stage cross-docking is superior to single-stage cross-docking by decreasing waiting times by 31%, decreasing the total makespan by 22%, and increasing the throughput per hour by 28% in non-pure cross-docking configurations. Through the key findings, various areas for future research are identified.

Firstly, exploring the impact of roll containers on reducing the total makespan in scenarios involving sorting mixed pallets presents an interesting avenue for future research.

Secondly, analyzing individual makespans within cross-docking operations can offer a deeper understanding of specific stages or processes, enabling targeted optimization strategies.

Additionally, investigating the behavior and performance of alternative consolidation methods under congestion scenarios would enhance understanding of their effectiveness and robustness.

Furthermore, incorporating due times in cross-docking studies would provide valuable insights into meeting delivery deadlines and minimizing delays in the supply chain.

Finally, there is consensus on a continuum of cross-docking purity operations. Hence, developing a framework or scale to measure cross-docking purity levels and classifying best practice strategies accordingly would enhance the generalizability of this and other research findings.

6.4. Managerial insights

The purpose of deriving managerial insights from scientific research is to bridge the gap between theoretical knowledge and practical application in real-world business settings. By deriving managerial insights from scientific research, the aim is to translate the findings and implications of research studies into actionable recommendations and guidelines for managers and decision-makers. This research provides managers at logistics service providers with the following insights.

Firstly, this research showed that the performance of cross-docking operations under uncertainty is significantly affected by the designed staging level. Logistical service providers should consider the staging level design to improve operational performance.

Secondly, the choice of consolidation method substantially impacts cross-docking performance, particularly concerning intra-terminal congestion. However, the effectiveness of consolidation methods is influenced by the purity of cross-docking operations. Logistical service providers should assess the purity of their cross-docking operations and select the consolidation method that aligns with the level of cross-docking purity.

Thirdly, cross-docking facilities face external and internal uncertainty factors, including arrival times, transportation times, and freight levels. Logistical service providers should actively monitor and act upon these factors to mitigate uncertainty. Monitoring freight levels and tracking truck arrival times enable managers to implement sophisticated technologies such as truck scheduling optimization and truck-to-dock assignment.

Fourth, many CDFs operate as hybrid warehouses instead of pure cross-docking facilities. Logistical service providers should recognize that their operations may exhibit characteristics of hybrid warehouses and design their operations accordingly by considering terminal characteristics such as internal storage capacity.

Finally, discrete event simulation models can help to evaluate design alternatives and test cross-docking facility improvements by capturing the interdependencies and stochastic nature of uncertain external factors. Logistical service providers should leverage simulation models to analyze the effects of different design alternatives and optimize their cross-docking operations. However, this research showed that DES as used in this research lacks the capability to handle congestion accurately. Logistical service providers should consider this when selecting the appropriate tool to model congestion on terminal pathways. Furthermore, logistical service providers should acknowledge the importance of optimizing optimal schedules or assignments by subjecting the schedules and assignments to simulation.

These insights provide logistical service providers with guidance on improving the performance of cross-docking operations under uncertainty, including considerations for staging level design, consolidation methods, uncertainty mitigation, hybrid warehouse characteristics, and the utilization of simulation modeling for optimization.

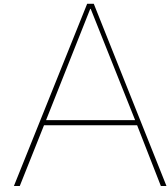
Bibliography

- [1] Korhan Acar, Ali Yalcin, and Daniel Yankov. "Robust door assignment in less-than-truckload terminals". In: *Computers and Industrial Engineering* 63.4 (Dec. 2012), pp. 729–738. ISSN: 03608352. DOI: 10.1016/j.cie.2012.04.008.
- [2] Mukhtar Ahmed et al. "Models calibration and evaluation". In: *Systems modeling* (2020), pp. 151–178.
- [3] Edward I Altman. "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy". In: *The journal of finance* 23.4 (1968), pp. 589–609.
- [4] Alireza Amini and Reza Tavakkoli-Moghaddam. "A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks". In: *Computers and Industrial Engineering* 96 (June 2016), pp. 180–191. ISSN: 03608352. DOI: 10.1016/j.cie.2016.03.023.
- [5] Uday M Apte and S Viswanathan. "Effective cross-docking for improving distribution efficiencies". In: *International Journal of Logistics* 3.3 (2000), pp. 291–302.
- [6] Allahyar (Arsalan) Ardakani and Jiangang Fei. "A systematic literature review on uncertainties in cross-docking operations". In: *Modern Supply Chain Research and Applications* 2.1 (Jan. 2020), pp. 2–22. ISSN: 2631-3871. DOI: 10.1108/mscra-04-2019-0011.
- [7] Georges Arnaout et al. "Modeling Cross-Docking Operations using Discrete Event Simulation". In: *Proceedings of the 6th international workshop on enterprise & organizational modeling and simulation*. 2010, pp. 113–120.
- [8] John J Bartholdi, Kevin R Gue, and Keebom Kang. "Staging freight in a crossdock". In: *Proceedings of the International Conference on industrial engineering and production management*. Quebec City, Canada. 2001, p. 10.
- [9] John J. Bartholdi and Kevin R. Gue. "The best shape for a crossdock". In: *Transportation Science* 38.2 (2004), pp. 235–244. ISSN: 00411655. DOI: 10.1287/trsc.1030.0077.
- [10] Lotte Berghman et al. "The truck scheduling problem at crossdocking terminals-exclusive versus mixed mode". In: *4th international conference on operations research and enterprise systems (ICORES 2015)*. 2015, pp–247.
- [11] Peter Bodnar, René De Koster, and Kaveh Azadeh. "Scheduling trucks in a cross-dock with mixed service mode dock doors". In: *Transportation Science* 51.1 (2017), pp. 112–131. ISSN: 15265447. DOI: 10.1287/trsc.2015.0612.
- [12] Nils Boysen and Malte Fliedner. *Cross dock scheduling: Classification, literature review and research agenda*. Dec. 2010. DOI: 10.1016/j.omega.2009.10.008.
- [13] Paul Buijs, Hans W. Danhof, and J. Hans C. Wortmann. "Just-in-Time Retail Distribution: A Systems Perspective on Cross-Docking". In: *Journal of Business Logistics* 37.3 (Sept. 2016), pp. 213–230. ISSN: 21581592. DOI: 10.1111/jbbl.12135.
- [14] Paul Buijs, Iris F.A. Vis, and Héctor J. Carlo. *Synchronization in cross-docking networks: A research classification and framework*. Dec. 2014. DOI: 10.1016/j.ejor.2014.03.012.
- [15] A Colin Cameron and Frank AG Windmeijer. "An R-squared measure of goodness of fit for some common nonlinear regression models". In: *Journal of econometrics* 77.2 (1997), pp. 329–342.
- [16] Stephanie Chevalier. *Global retail e-commerce sales 2026*. Sept. 2022. URL: <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/>.
- [17] Larry L Constantine. "Object-oriented and structured methods: toward integration". In: *American Programmer* 2.7/8 (1989), pp. 34–40.

- [18] Augusto da Cunha Reis et al. "Warehouse design: a systematic literature review". In: *Brazilian Journal of Operations & Production Management* 14.4 (2017), pp. 542–555.
- [19] Steve Dennis. "The ticking time bomb of E-commerce returns". In: *Forbes* (2018).
- [20] Statista Research Department. *Warehouse Rent Netherlands 2022*. Apr. 2023. URL: <https://www.statista.com/statistics/527845/warehouse-primary-rent-cost-logistics-market-netherlands-europe/>.
- [21] Philip T Evers and Xiang Wan. "Systems analysis using simulation". In: *Journal of Business Logistics* 33.2 (2012), pp. 80–89.
- [22] Andrea Gallo et al. "Scheduling cross-docking operations under uncertainty: A stochastic genetic algorithm based on scenarios tree". In: *EURO Journal on Transportation and Logistics* 11 (2022), p. 100095. ISSN: 21924376. DOI: 10.1016/j.ejtl.2022.100095. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2192437622000206>.
- [23] Kevin R Gue and Keebom Kang. "STAGING QUEUES IN MATERIAL HANDLING AND TRANSPORTATION SYSTEMS". In: *Proceedings of the 2001 Winter Simulation Conference*. 2001, pp. 1104–1108.
- [24] Aldy Gunawan et al. "Vehicle Routing Problem with Forward and Reverse Cross-Docking: Formulation and Matheuristic Approach". In: *IEEE International Conference on Automation Science and Engineering*. Vol. 2021-August. IEEE Computer Society, Aug. 2021, pp. 1467–1472. ISBN: 9781665418737. DOI: 10.1109/CASE49439.2021.9551486.
- [25] Mohammad Rafiqul Islam. "Sample size and its role in Central Limit Theorem (CLT)". In: *Computational and Applied Mathematics Journal* 4.1 (2018), pp. 1–7.
- [26] LW Johnston. "Student's t-Test". In: *Journal of Quality Technology* 2.4 (1970), pp. 243–245.
- [27] Daisuke Kanama and Taketoshi Kido. "The innovation strategy of a non-food-industry company regarding the traditional food supply chain". In: *Agriculture & Food* 4 (2016), pp. 687–695.
- [28] Dinçer Konur and Mihalıs M. Golias. "Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty". In: *Computers and Industrial Engineering* 65.4 (2013), pp. 663–672. ISSN: 03608352. DOI: 10.1016/j.cie.2013.05.009.
- [29] Dinçer Konur and Mihalıs M. Golias. "Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach". In: *Transportation Research Part E: Logistics and Transportation Review* 49.1 (2013), pp. 71–91. ISSN: 13665545. DOI: 10.1016/j.tre.2012.06.007.
- [30] Yiyo Kuo. "Optimizing truck sequencing and truck dock assignment in a cross docking system". In: *Expert Systems with Applications* 40.14 (2013), pp. 5532–5541. ISSN: 09574174. DOI: 10.1016/j.eswa.2013.04.019.
- [31] Anne Laure Ladier and Gülgün Alpan. *Cross-docking operations: Current research versus industry practice*. July 2016. DOI: 10.1016/j.omega.2015.09.006.
- [32] Anne Laure Ladier and Gülgün Alpan. "Robust cross-dock scheduling with time windows". In: *Computers and Industrial Engineering* 99 (Sept. 2016), pp. 16–28. ISSN: 03608352. DOI: 10.1016/j.cie.2016.07.003.
- [33] Rim Larbi, Gülgün Alpan, and Bernard Penz. "Scheduling transshipment operations in a multiple inbound and outbound doorcrossdock". In: *2009 International Conference on Computers and Industrial Engineering, CIE 2009*. IEEE Computer Society, 2009, pp. 227–232. ISBN: 9781424441365. DOI: 10.1109/iccie.2009.5223927.
- [34] Averill M Law, W David Kelton, and W David Kelton. *Simulation modeling and analysis*. Vol. 3. McGraw-hill New York, 2007.
- [35] Qing Li and Yu-Liu Chen. "Idef0 function modeling". In: *Modeling and Analysis of Enterprise and Information Systems*. Springer, 2009, pp. 98–122.
- [36] Bert Mateboer. *Werken met Rolcontainers: Code Rood*. Sept. 2018. URL: <https://truckstar.nl/werken-met-rolcontainers-code-rood/#:~:text=In%20een%20volwassen%20oplegger%20passen,loopt%20risico%20op%20fysieke%20overbelasting..>

- [37] Reza Kiani Mavi et al. *Cross-docking: A systematic literature review*. June 2020. DOI: 10.3390/su12114789.
- [38] Douglas L. McWilliams, Paul M. Stanfield, and Christopher D. Geiger. "The parcel hub scheduling problem: A simulation-based solution approach". In: *Computers and Industrial Engineering* 49.3 (Nov. 2005), pp. 393–412. ISSN: 03608352. DOI: 10.1016/j.cie.2005.07.002.
- [39] Danyar Molavi, Amin Shahmardan, and Mohsen S. Sajadieh. "Truck scheduling in a cross docking systems with fixed due dates and shipment sorting". In: *Computers and Industrial Engineering* 117 (Mar. 2018), pp. 29–40. ISSN: 03608352. DOI: 10.1016/j.cie.2018.01.009.
- [40] David E Mulcahy. *Warehouse distribution and operations handbook*. McGraw-Hill Professional, 1994.
- [41] Mohammad Mahdi Nasiri et al. "A predictive-reactive cross-dock rescheduling system under truck arrival uncertainty". In: *Expert Systems with Applications* 188 (Feb. 2022). ISSN: 09574174. DOI: 10.1016/j.eswa.2021.115986.
- [42] Osman Özkaraca. "A REVIEW ON USAGE OF OPTIMIZATION METHODS IN GEOTHERMAL POWER GENERATION". In: *Mugla Journal of Science and Technology* (June 2018), pp. 130–136. ISSN: 2149-3596. DOI: 10.22531/muglajsci.437340.
- [43] David Peidro et al. "Quantitative models for supply chain planning under uncertainty". In: *International Journal of Advanced Manufacturing Technology* 43.3-4 (July 2009), pp. 400–420. ISSN: 02683768. DOI: 10.1007/s00170-008-1715-y.
- [44] Mike Pidd. "Model development and HCI". In: *Proceedings of the 28th conference on Winter simulation*. 1996, pp. 681–686.
- [45] Andréa Raffaella et al. *Integrative conceptual framework to support decisions on warehousing operations in forward and reverse flow*. Tech. rep. 1. 2022, pp. 53–74.
- [46] Ali Rahbari et al. "The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models". In: *Applied Mathematical Modelling* 70 (June 2019), pp. 605–625. ISSN: 0307904X. DOI: 10.1016/j.apm.2019.01.047.
- [47] S. Rahmanzadeh Tootkaleh, S. M.T. Fatemi Ghomi, and M. Sheikh Sajadieh. "Cross dock scheduling with fixed outbound trucks departure times under substitution condition". In: *Computers and Industrial Engineering* 92 (Feb. 2016), pp. 50–56. ISSN: 03608352. DOI: 10.1016/j.cie.2015.12.005.
- [48] Sai Manoj Rayapureddy et al. "Comparative Study of Combustion, Performance and Emission Characteristics of Hydrotreated Vegetable Oil–Biobutanol Fuel Blends and Diesel Fuel on a CI Engine". In: *Sustainability (Switzerland)* 14.12 (June 2022). ISSN: 20711050. DOI: 10.3390/su14127324.
- [49] Erhard Reschenhofer. "Generalization of the Kolmogorov-Smirnov test". In: *Computational statistics & data analysis* 24.4 (1997), pp. 433–441.
- [50] Arpan Rijal, Marco Bijvank, and René de Koster. "Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors". In: *European Journal of Operational Research* 278.3 (2019), pp. 752–771.
- [51] Stewart Robinson. "Simulation model verification and validation: increasing the users' confidence". In: *Proceedings of the 29th conference on Winter simulation*. Ed. by S. Andradottir et al. 1997, pp. 53–59.
- [52] Marieke de Ruiter and Daan Ballegeer. *Postnl neemt 4.100 pakketbezorgers in Loondienst na Kritiek Op Werkwijze*. Aug. 2022. URL: <https://www.volkskrant.nl/nieuws-achtergrond/postnl-neemt-4-100-pakketbezorgers-in-loondienst-na-kritiek-op-werkwijze~b96776a0/?referrer=https%5C%3A%5C%2F%5C%2Fwww.google.com%5C%2F>.
- [53] Laurence Saglietto, François Fulconis, and Gilles Paché. "4PL and models of strategic alignment". In: *Netcom* 21-3/4 (Dec. 2007), pp. 395–406. ISSN: 0987-6014. DOI: 10.4000/netcom.2294.

- [54] Christian Serrano, Xavier Delorme, and Alexandre Dolgui. "Scheduling of truck arrivals, truck departures and shop-floor operation in a cross-dock platform, based on trucks loading plans". In: *International Journal of Production Economics* 194 (Dec. 2017), pp. 102–112. ISSN: 09255273. DOI: 10.1016/j.ijpe.2017.09.008.
- [55] Mojtaba Shakeri et al. "An efficient incremental evaluation function for optimizing truck scheduling in a resource-constrained crossdock using metaheuristics". In: *Expert Systems with Applications* 45 (Mar. 2016), pp. 172–184. ISSN: 09574174. DOI: 10.1016/j.eswa.2015.09.041.
- [56] Jeffrey S Smith and David T Sturrock. *Simio and Simulation - Modeling, Analysis, Applications - 6th Edition*. 2021. ISBN: 9781938207037.
- [57] Juan Pablo Soto Zuluaga, Marcus Thiell, and Rosa Colomé Perales. "Reverse cross-docking". In: *Omega (United Kingdom)* 66 (Jan. 2017), pp. 48–57. ISSN: 03050483. DOI: 10.1016/j.omega.2016.01.010.
- [58] David Sturrock. *Non-blocking path with workers in both directions?* Apr. 2015. URL: <https://www.simio.com/simio-forum/topic/1812-non-blocking-path-with-workers-in-both-directions/>.
- [59] Jan Van Belle, Paul Valckenaers, and Dirk Cattrysse. *Cross-docking: State of the art*. Dec. 2012. DOI: 10.1016/j.omega.2012.01.005.
- [60] Jan Van Belle et al. "A tabu search approach to the truck scheduling problem with multiple docks and time windows". In: *Computers and Industrial Engineering* 66.4 (2013), pp. 818–826. ISSN: 03608352. DOI: 10.1016/j.cie.2013.09.024.
- [61] Iris F.A. Vis and Kees Jan Roodbergen. "Positioning of goods in a cross-docking environment". In: *Computers and Industrial Engineering* 54.3 (Apr. 2008), pp. 677–689. ISSN: 03608352. DOI: 10.1016/j.cie.2007.10.004.
- [62] Chia Nan Wang, Thanh Tuan Dang, and Ngoc Ai Thy Nguyen. "Outsourcing reverse logistics for e-commerce retailers: A two-stage fuzzy optimization approach". In: *Axioms* 10.1 (Mar. 2021). ISSN: 20751680. DOI: 10.3390/axioms10010034.
- [63] Jiana-Fu Wang and Amelia Regan. *Real-Time Trailer Scheduling for Crossdock Operations*. Tech. rep. 2. 2008, pp. 5–20. URL: <https://about.jstor.org/terms>.
- [64] Xiang Xi et al. "Two-stage conflict robust optimization models for cross-dock truck scheduling problem under uncertainty". In: *Transportation Research Part E: Logistics and Transportation Review* 144 (Dec. 2020). ISSN: 13665545. DOI: 10.1016/j.tre.2020.102123.
- [65] Kum Khiong Yang, Jaydeep Balakrishnan, and Chun Hung Cheng. "An Analysis of Factors Affecting Cross Docking Operations". In: *Journal of Business Logistics* 31.1 (Mar. 2010), pp. 121–148. ISSN: 21581592. DOI: 10.1002/j.2158-1592.2010.tb00131.x.
- [66] Vincent F. Yu, Parida Jewpanya, and A. A.N.Perwira Redi. "Open vehicle routing problem with cross-docking". In: *Computers and Industrial Engineering* 94 (Apr. 2016), pp. 6–17. ISSN: 03608352. DOI: 10.1016/j.cie.2016.01.018.
- [67] Yimeng Zhang, Bilge Atasoy, and Rudy R. Negenborn. "Preference-Based Multi-Objective Optimization for Synchromodal Transport Using Adaptive Large Neighborhood Search". In: *Transportation Research Record*. Vol. 2676. 3. SAGE Publications Ltd, Mar. 2022, pp. 71–87. DOI: 10.1177/03611981211049148.
- [68] Binghai Zhou, Yuanrui Lei, and Shi Zong. "Lagrangian relaxation algorithm for the truck scheduling problem with products time window constraint in multi-door cross-dock". In: *Journal of Industrial and Management Optimization* 18.6 (Nov. 2022), pp. 4129–4149. ISSN: 1553166X. DOI: 10.3934/jimo.2021151.



Case description

A.1. Case introduction

Budbee was established in Sweden in 2015 as a 4PL for e-commerce merchants and first started operating in the Netherlands in Vianen in September 2021. In over 1.5 years, Budbee's network has grown to 5 own and 9 partner terminals, which is still increasing. A 4PL provides logistical services without owning assets since it designs and sells supply chain solutions by coordinating the activities of carriers, storage operators, subcontractors, and packaging companies [53]. In the first mile of the supply chain, the parcels are picked up by Budbee or dropped off by the merchant at Budbee's regional terminal. This is also called the 'injection of parcels'. At the regional terminal, the parcels are sorted to their destinations and loaded on the line haul. Next, using the line hauls, the middle mile distributes the sorted goods to the destination terminals via the cross-docking terminal in Vianen (UTR). Once arrive at the destination terminal, the last-mile operations start, and the parcels are sorted to their assigned route and delivered to the customer. Delivery to the customer can either be by home delivery or delivery to a parcel box.

Budbee its operations are similar to parties such as PostNL and DHL except for the fact that Budbee exclusively uses subcontractors and operates fossil-free by requiring the subcontractors to refuel with HVO100, a fossil-free diesel alternative [48]. Companies like DHL and PostNL still own a substantial share of their fleet and rely on fossil fuels [52]. The goods Budbee transports are exclusively parceled goods consolidated on pallets, trolleys, or postbags.

Budbee has specified a time window each morning during which cross-docking must occur for the line hauls to arrive at the regional terminals in time for sorting. However, since volumes increase and more line hauls are added to their network, the cross-docking operations are under pressure. This raises the question of to what extent the current time window for cross-docking, the number of dock doors, and storage capacity are sufficient for the increase in volume and to what volume the cross-docking process will become a bottleneck.

A.2. Network

Like the example of a supply chain in Figure 1.1, Budbee's middle-mile centers around their cross-docking terminal UTR. This results in a hub-spoke network as shown in Figure A.1 in which the line hauls transport parcels from the regional terminals represented as the spokes via the cross-docking terminal UTR represented as the hub and back to the destination terminal. Line hauls between spokes are possible, but only in case the sorted volume at the regional terminal is large enough for a direct line haul to become economically beneficial.

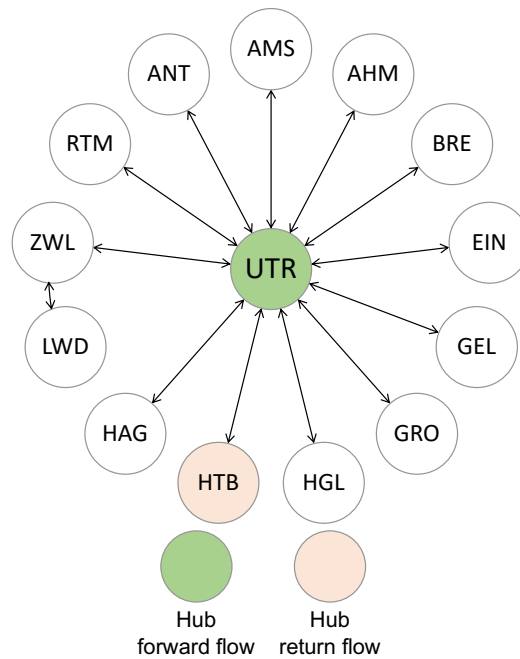


Figure A.1: Schematic representation of the hub-spoke network in the middle-mile

Figure A.1 shows that UTR serves as a cross-docking terminal for regional depots both in Belgium (BE) and the Netherlands (NL). UTR functions not only as a cross-dock but also as a regional terminal for Utrecht. As a result, parcels are also injected at UTR, sorted, and stored the day before cross-docking. The travel time to UTR varies per line haul, and to arrive at the cross-dock in time, some line hauls must depart earlier or later than others. Due to the transported volume, some routes are executed with multiple line hauls. Vice versa, one line haul can also service multiple destinations, as seen for ZWL and LWD, where one line haul departs from LWD, stops in ZWL, and continues to UTR.

Additionally, forward flows and return flows generated by the spokes are transported through the network shown in Figure A.1. All regional terminals generate return volume, cross-docked in UTR, and transported to HTB for sorting. As a result, an increase in return flow also affects the capacity of the cross-docking terminal UTR.

A.3. Numerical analysis

A.3.1. Number of total and cross-docked unit loads

Budbee mentions that the majority of the transported volume is injected at the CDF and that the CDF therefore does not function as a pure CDF.

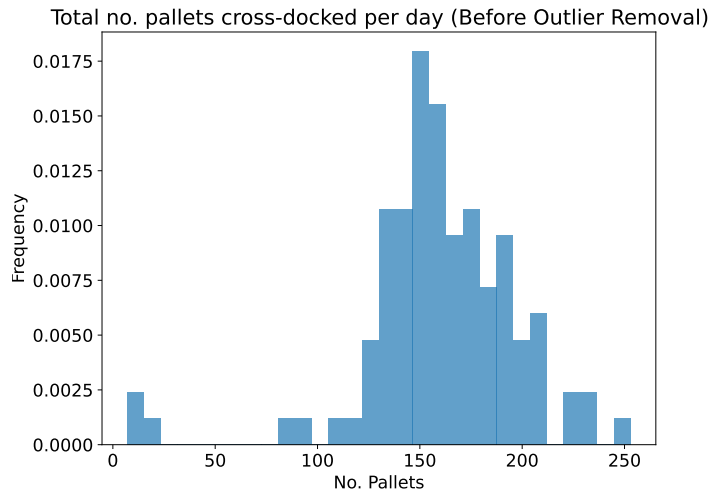


Figure A.2: Histogram of pallets handled in cross-docking

To examine outliers in the data set, a box plot of the observed values in the data set is presented in Figure A.3. Figure A.3 shows that on average, 158 pallets are handled during cross-docking. The first quartile and third quartile are 143.5 and 186.5 respectively.

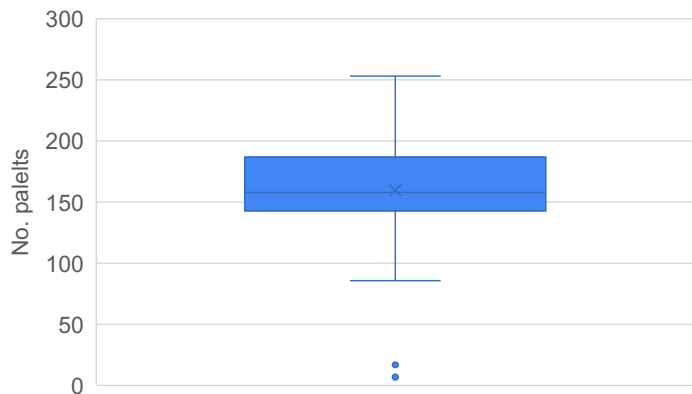


Figure A.3: Box plot of pallets handled in cross-docking

The number of pallets generated differs per regional terminal. For a three-week period, the cumulative number of pallets generated is presented using box plots in Figure A.4. Figure A.4 shows that the most significant shares of the volume are injected in two terminals, of which UTR is the cross-docking terminal itself. The volumes in Figure A.4 show that other terminals generate demand. These volumes, however, are considered by Budbee to be mostly return volumes. The data point associated with Figure A.4 are depicted in Table A.1.

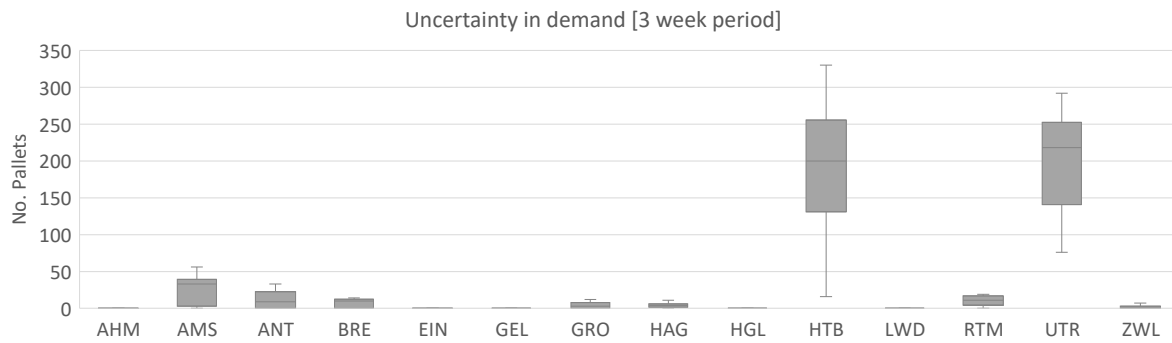


Figure A.4: Box plot of the generated pallet demand levels per origin in the data set

Table A.1: Box plot values of the pallet demand levels in the data set

Statistic	AHM	AMS	ANT	BRE	EIN	GEL	GRO	HAG	HGL	HTB	LWD	RTM	UTR	ZWL
Max.	1	56	33	14	2	3	12	11	2	330	1	19	292	7
Q2	1	40	23	13	2	2	8	6	2	251	1	17	252	4
Median	1	33	16	11	2	2	6	5	2	200	1	15	218	3
Q1	1	28	9	10	2	1	4	2	1	133	1	11	157	2
Min.	1	2	2	9	2	1	1	1	1	16	1	3	76	1

Since the internal transportation time is proportional to the transported demand, fluctuating demand levels lead to fluctuating cross-docking times.

A.4. Conclusion and scope system

In the case under study, the cross-docking network resembles a hub-spoke network, with UTR as the cross-docking terminal. Direct line hauls between spokes are possible but rarely used. Only for LWD, the line haul first stops at ZWL before arriving in LWD. GRO is the only line haul that uses an 18-pallet truck.

The largest shares of transported volumes are destined for AMS (29.01%) and RTM (22.24%), whereas the largest shares of volume originate from UTR (49.06%) and HTB (48.93%).

Originally, all line hauls are scheduled to be available for cross-docking in UTR at 6:15:00 AM. The return time to the origin terminal is at 10:00:00 AM to be in time for route sorting. The due times differ per line haul since the travel times differ per destination. Analysis shows that the trucks, on average, arrive later than expected. However, on average, the trucks also leave earlier than expected, which suggests that the scheduled time in the arrival and departure times data set is not yet at its limits. The mean total cross-docking time is roughly 57 minutes, more than the pre-scheduled 45 minutes. This seems remarkable but does not have to be a problem since the line haul with the first due time allows a theoretical least total cross-docking time of 01:05:00. Therefore, the 45 minutes of cross-docking can be wrongfully chosen. Additionally, no figures are available on waiting times nor a specified buffer time, making it unclear how long cross-docking took.

At Budbee's cross-docking terminal, the dock doors can be used for either unloading (inbound), reloading (outbound), or both unloading and reloading of arriving line haul trucks. The inbound trucks are the same as the outbound trucks, and the number of trucks to be serviced at the cross-dock is larger than the number of dock doors available, resulting in trucks competing for a scarce resource. Additionally, there is only limited temporary storage capacity at the dock door, which varies per dock. To increase complexity, the expected arrival times of the trucks are known, but drivers can arrive earlier or later than expected, which affects the cross-docking schedule. Moreover, currently, the number of pallets per destination loaded per truck is unknown, but Budbee expects to be able to track this in the near future. Similarly, Budbee plans for the line hauls to carry a GPS which then can be used to track and update the ETAs of the trucks.

Scoping the system is necessary to set the context of the problem. In this research, only activities inside the cross-docking terminal are included in the problem. Hence, activities outside the cross-docking terminal, such as in regional terminals or last-mile distribution, are omitted to focus on the

cross-docking operations only. Additionally, only the number of load units is relevant for the processing time at the cross-dock. Therefore, the exact number of packages per load unit is irrelevant, and the system will be scoped at the load unit level, not at the individual package level. The number of terminal workers and their pallet truck licenses are relevant for the handling time. They are therefore included as well as the physical resources available (EPT & MPT). Scanning of load units is considered a required activity and is included in the handling time of a load unit. The time horizon of the model is set to one week, running from Monday to Saturday. After the time horizon, the system should be emptied.

A.5. Goals and needs

This study aims to investigate the efficiency and effectiveness of cross-docking operations in the parcel delivery company's terminal under study. The problem is that cross-docking can be challenging due to its complexity, which involves coordinating different processes, activities, and resources. The objective is to identify the bottlenecks, inefficiencies, and opportunities for improvement in the cross-docking operations of Budbee.

The first objective is to develop a simulation model of the cross-docking process to investigate the impacts of different scenarios, such as different volume levels, resource allocation, scheduling, and (un)loading tactics. The second objective is to identify the critical factors (bottlenecks) that affect the performance of the cross-docking operations and develop guidelines for exploiting the bottlenecks to improve the system. The third objective is to evaluate the feasibility and validity of the simulation model as a tool for improving the efficiency and effectiveness of cross-docking operations.

A.6. Conclusion

The cross-docking terminal UTR serves 13 regional terminals directly, and one regional terminal indirectly excludes UTR. Each regional terminal is serviced by one or more trucks. Therefore, at least 13 trucks compete for one of the 10 dock doors, leading to queues and inducing waiting times. All dock doors are used for both unloading and loading, also called mixed-mode service, and all dock doors have a destination assigned to them. Cross-docking operations generally consist of truck unloading, freight sorting, and truck loading. The terminal layout influences the internal handling times, as does the size and complexity of the operations, the number of workers, equipment availability, and product characteristics. The cross-docking terminal UTR is not only used for forward flow but also processes return flow which is all directed to HTB. The destinations of the actual volume per line haul are unknown due to the lack of a WMS. Tracking the destinations of the freight volumes per truck could improve scheduling, which could decrease waiting times and subsequently decrease total cross-docking time or increase throughput. Uncertainty of truck arrival times leads to uncertainty about the total time cross-docking requires.

The system of the case under study is prone to uncertainty in arrival times of trucks, volume, and destination of the freight per truck, which influences the capacity and waiting times at the cross-docking terminal and the internal handling time. The next chapter revises the literature regarding appropriate system analysis tools to determine the most suited tool for analyzing the system.

A.7. Logic flow

In close collaboration with Budbee's CDF terminal manager, the logic flow diagram shown in Figure A.5 to facilitate a clear and concise understanding of the cross-docking process that was used in operations on October 21, 2022.

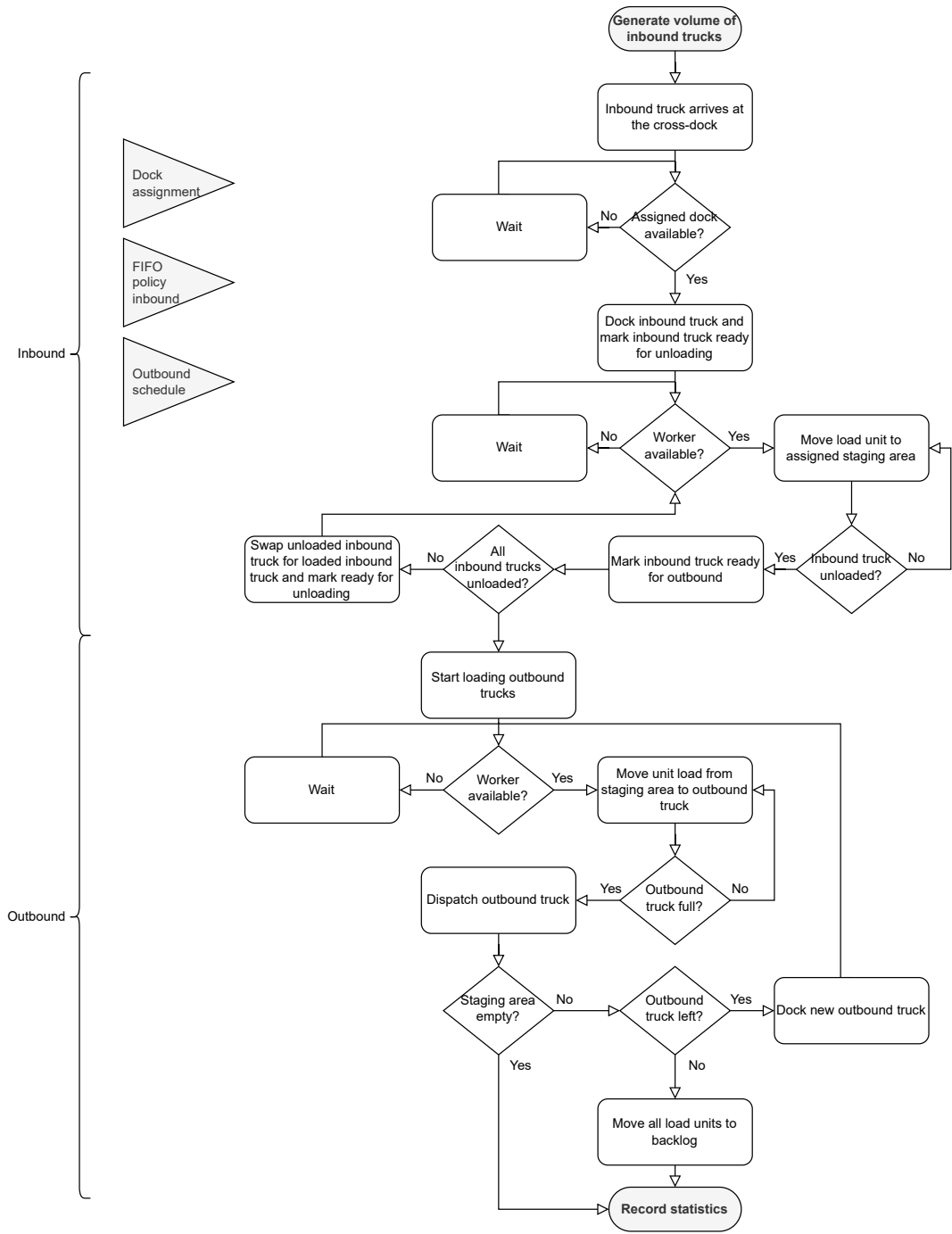
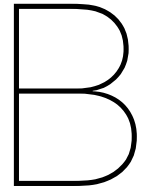


Figure A.5: Logic flow diagram



Simulation model

The simulation model overview as shown in Figure B.1, consists of mainly of the following parts from left to right. This chapter elaborated on the construction of the model by first discussing Entity Generation in section B.1. Next, the arrival and unloading steps are presented in section B.2. Then, the staging and internal transshipment processes are shown in section B.3. Additionally, the reloading and dispatching process is depicted in section B.4. Finally, the model trace is discussed in section B.5

1. Entity generation
2. Arrival and unloading
3. Staging and internal transshipment
4. Reloading and dispatching

All intra-terminal distances shown in Figure B.1 are scaled to reality, ensuring the proportionality of internal transshipment time to the covered distance.

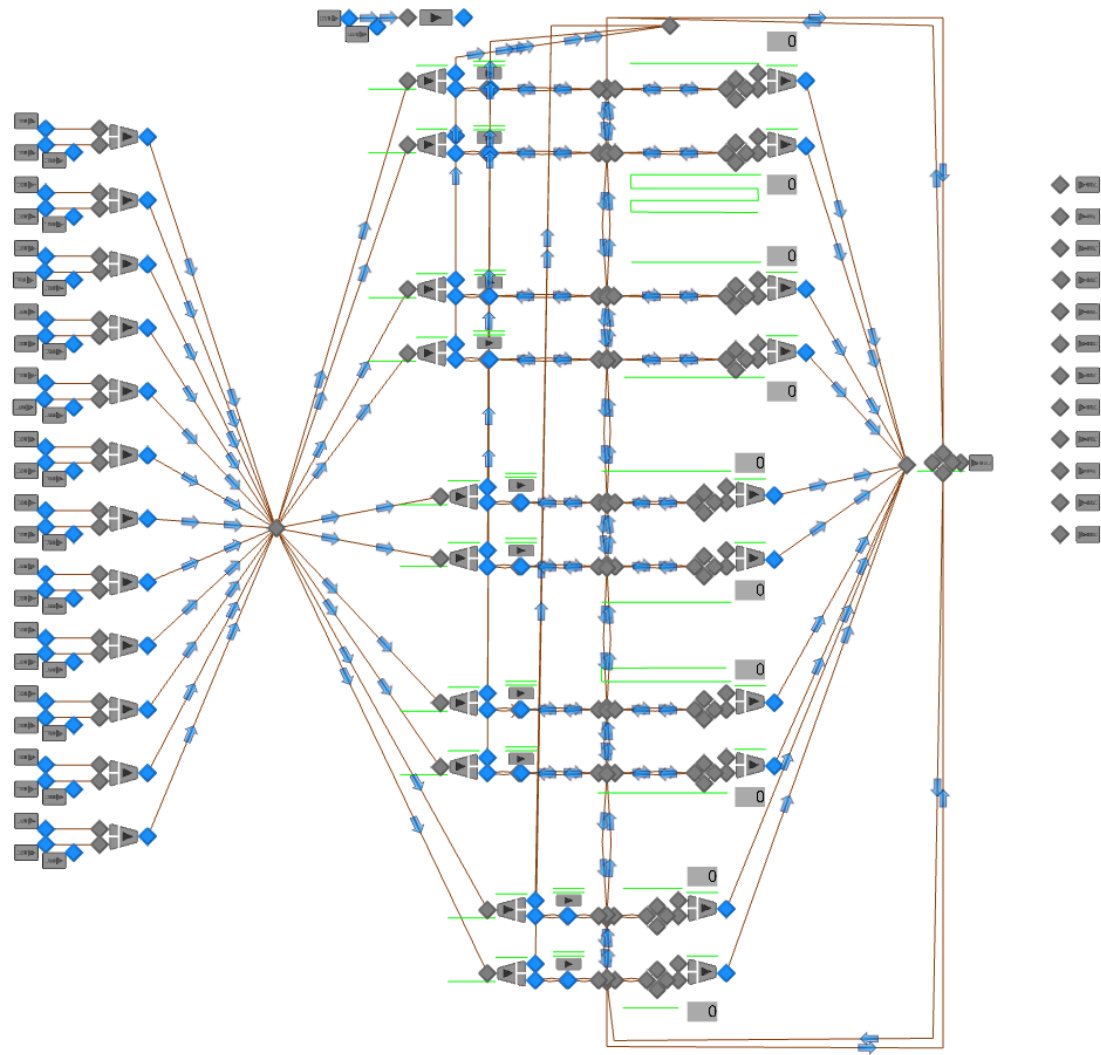


Figure B.1: Simulation model overview in SIMIO

B.1. Entity generation

Visualization of the Entities used in this model is shown in Figure B.2. One distinguishes Unit Load Entities (Forward & Return), Truck entities, and Transporter Entities (EPTs & Workers). There are 12 Truck Entities (one per Regional Terminal). The number of Transporters is equal to the number of dock doors (e.g. 10 EPTs & 10 Workers).

Unit Load Entities



Truck Entities



Transporters

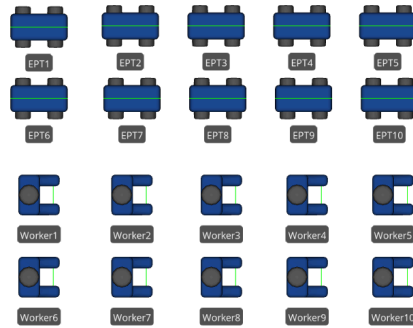


Figure B.2: Entity and transporter visualization

The transporters spawn at the Unloading docks at the start of the simulation. The Forward, Return and Truck entities are generated using sources for each regional terminal as shown in Figure B.3.

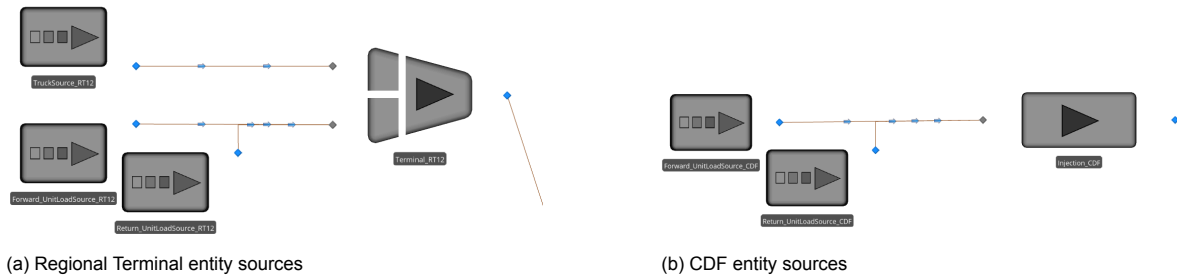


Figure B.3: Entity generation

Both Figure B.3a and Figure B.3b consists of two Unit Load sources (Forward and Return). Naturally, no trucks are generated for the CDF since the entities are spawned directly at the CDF, and transport to the CDF for direct injections is out of the scope. Therefore, only Regional Terminals have a Truck Source. The properties for the Forward, Return, and Truck sources are shown in Table B.1, Table B.2, and Table B.3 respectively.

Table B.1: Properties Forward unit load Source

Property	Value
<i>Entity Arrival Logic</i>	
Entity Type	Forward
Arrival Mode	Inter Arrival Time
Time Offset	0 [seconds]
Inter Arrival Time	1 [days]
Entities Per Arrival	Math.If(PureCrossDocking == True, Math.Ceiling(DemandLevel * VolumeGeneration.PureCross_Docking_Probability), Math.Ceiling(DemandLevel * VolumeGeneration.Volume))
<i>Add-On Process Triggers</i>	
Created Entity	AssignOriginDestination

Table B.2: Properties Return unit load Source

Property	Value
<i>Entity Arrival Logic</i>	
Entity Type	Return
Arrival Mode	Inter Arrival Time
Time Offset	0 [seconds]
Inter Arrival Time	1 [days]
Entities Per Arrival	Math.If(PureCrossDocking == True, Math.Ceiling(DemandLevel * VolumeGeneration.PureCross_Docking_Probability), Math.Ceiling(DemandLevel * VolumeGeneration.Volume))
<i>Add-On Process Triggers</i>	
Created Entity	AssignOriginDestination

Table B.3: Properties Truck Source

Property	Value
<i>Entity Arrival Logic</i>	
Entity Type	Truck_RT12
Arrival Mode	Inter Arrival Time
Time Offset	1 [minutes]
Inter Arrival Time	1 [days]
Entities Per Arrival	Math.Ceiling(Math.Max(UL_to_RT12 / CapacityTruck , UL_from_RT12 / CapacityTruck))
<i>Add-On Process Triggers</i>	
Created Entity	AssignOriginDestination

Both the Forward and Return unit load sources generate entities at the start of the simulation which occurs at most once a day max. The number of entities arriving depends on the purity of cross-docking, the demand level, and the table VolumeGeneration shown in Table B.4. If PureCrossDocking is instigated, then none of the freight is directly injected at the CDF and each truck is assumed to transport the same share of the total freight. The DemandLevel is as specified in section 4.5 and the VolumeGeneration.Volume is according to the distributions specified in Table 4.7.

For the Truck source, the number of Trucks generated depends on the number of Unit Loads to be transported. Since it is assumed that there are always enough trucks to transport the freight, the number of trucks generated equals the ceiling of the maximum value for the number of unit loads transported from and to the respective terminal. This ensures that there is always sufficient capacity to transport the generated unit loads.

Table B.4: DataTable: VolumeGeneration

Source	Volume	DemandMultiplier	PureCross_Docking_Probability
Forward_UnitLoadSource_RT1	0.0004 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT2	0.0211 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT3	0.0000 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT4	0.0027 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT5	0.0030 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT6	0.0002 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT7	0.0010 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT8	0.0008 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT9	0.0011 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT10	0.2566 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT11	0.0008 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_RT12	0.0074 * (1 - ReturnMix)	DemandLevel	1/12 * (1 - ReturnMix)
Forward_UnitLoadSource_CDF	0.7049 * (1 - ReturnMix)	DemandLevel	0
Return_UnitLoadSource_RT1	0.0619 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT2	0.2618 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT3	0.0143 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT4	0.0699 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT5	0.0666 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT6	0.0416 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT7	0.0299 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT8	0.1104 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT9	0.0336 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT10	0.0 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT11	0.0844 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_RT12	0.1451 * ReturnMix	DemandLevel	1/12 * (1 - ReturnMix)
Return_UnitLoadSource_CDF	0.0349 * ReturnMix	DemandLevel	0 ReturnMix)

On each created entity, the Add-On Process Trigger instigated is *AssignOriginDestination* depicted Figure B.4. This process step consists of two *Execute* steps that execute Figure B.5 and Figure B.6 respectively.

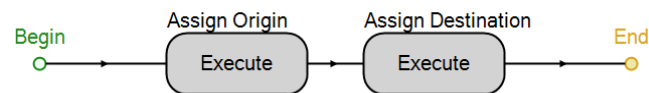


Figure B.4: Process step: AssignOriginDestination

First, the model checks whether the entity is a Truck or a Unit Load. Next, the Entity's origin is assigned by searching the current source location of the entity. Then, the respective UL origin counter is searched and the new origin counter value is the previous count incremented by 1. For Trucks, this process is almost the same. However, since the Trucks revisit their origin, the origin is also assigned as their destination.

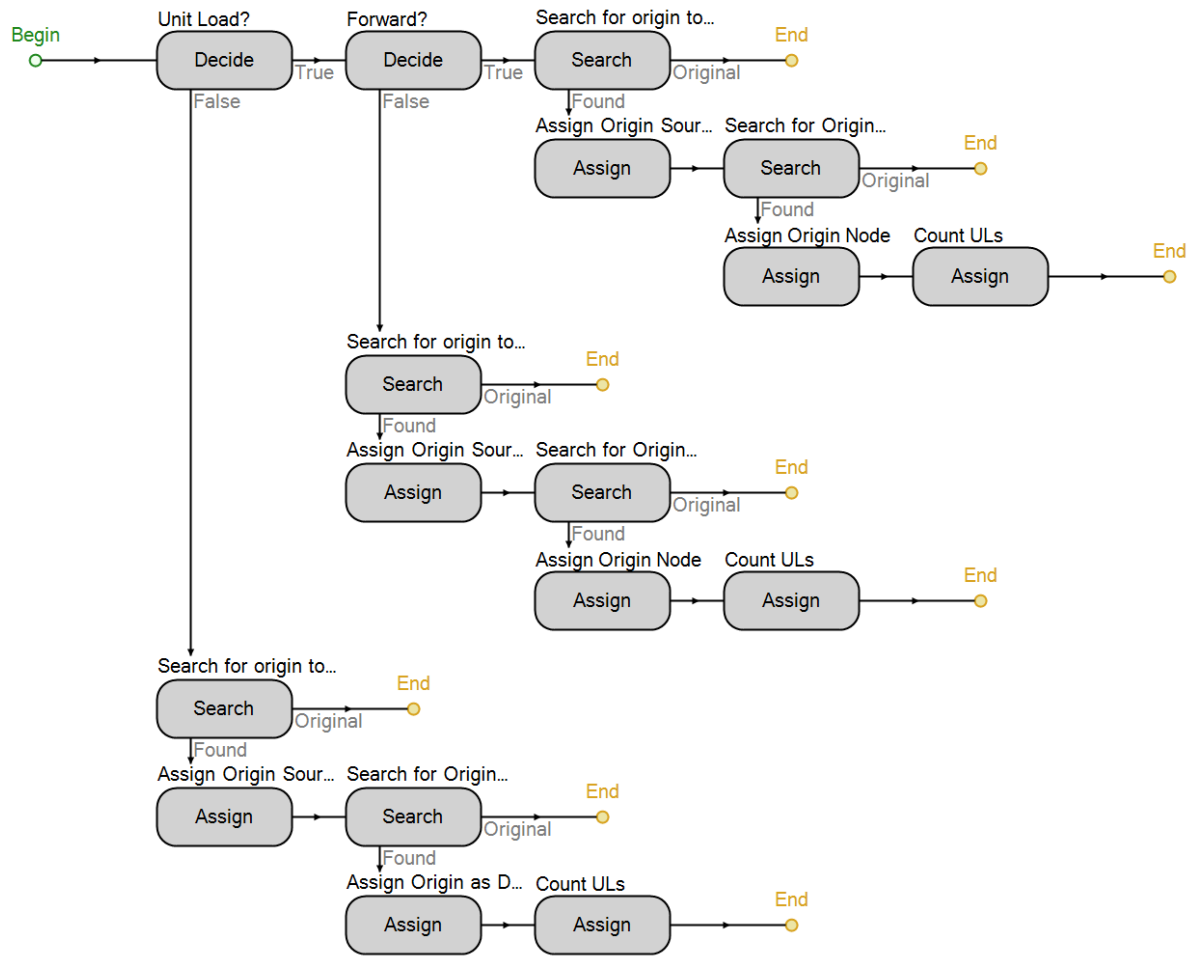


Figure B.5: Process step: AssignOrigin

Regarding the destination assignment, only Unit Loads need destination assignment. Unit Loads are assigned their destination according to the probability distribution in Table B.5. Forward Unit Loads randomly set destinations according to the probabilities specified in Table B.5b. Similarly, Return Unit Loads are assigned a destination according to Table B.5a. Although, due to the network of the study at hand, the destination probability of returns is 100% for RT10, since this is the return cross-docking hub as described in Appendix A.

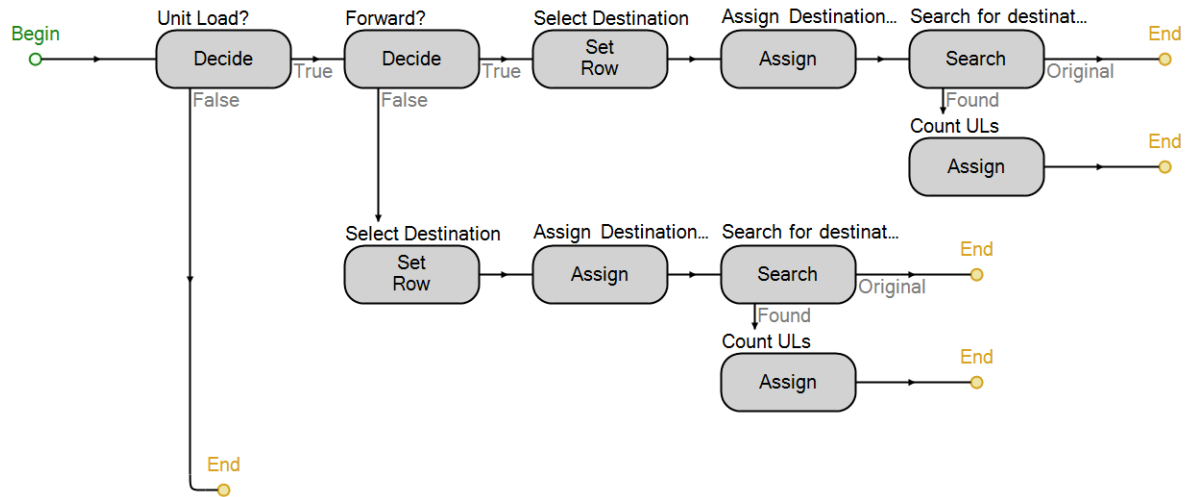


Figure B.6: Process step: AssignDestination

Table B.5: Probability tables

(a) DataTable: Probabilistic Destinations Return

Entity	DestinationNode	Probability
Return	Input@Sink_RT1	0.0
Return	Input@Sink_RT2	0.0
Return	Input@Sink_RT3	0.0
Return	Input@Sink_RT4	0.0
Return	Input@Sink_RT5	0.0
Return	Input@Sink_RT6	0.0
Return	Input@Sink_RT7	0.0
Return	Input@Sink_RT8	0.0
Return	Input@Sink_RT9	0.0
Return	Input@Sink_RT10	100
Return	Input@Sink_RT11	0.0
Return	Input@Sink_RT12	0.0
Return	Input@Sink_CDF	0.0

(b) DataTable: Probabilistic Destinations Return

Entity	DestinationNode	Probability
Forward	Input@Sink_RT1	6.19
Forward	Input@Sink_RT2	26.18
Forward	Input@Sink_RT3	1.43
Forward	Input@Sink_RT4	6.99
Forward	Input@Sink_RT5	6.66
Forward	Input@Sink_RT6	4.16
Forward	Input@Sink_RT7	2.99
Forward	Input@Sink_RT8	11.04
Forward	Input@Sink_RT9	3.36
Forward	Input@Sink_RT10	4.54
Forward	Input@Sink_RT11	8.44
Forward	Input@Sink_RT12	14.51
Forward	Input@Sink_CDF	3.49

Once the origins and destinations are assigned to the created entities, the Unit Loads and Trucks are batched in the Regional Terminal Combiner. The properties for the Regional Terminal Combiner are specified in Table B.6.

Table B.6: Properties Regional Terminal Combiner

Property	Value
<i>Batching Logic</i>	
Batching Quantity	CapacityTruck
Matching Rule	Any Entity
Release Batch Early Trigger	Location.Parent.TerminalCombiner.MemberInput.EntryQueue.NumberWaiting == 0

Table B.6 batches the Unit Loads to the Trucks according to the capacity of the Trucks (33 for pallets and 55 for roll containers, as specified in section 4.5). Not all trucks will be batched to the capacity limit. Therefore the trucks are released early in case the number of entities waiting for batching equals zero. The Unit Loads are batched to the Trucks and the Truck enters a TimePath to model the travel time to the CDF. Infinite Trucks can travel the TimePath and the Travel Time is randomly chosen according to the specified *ArrivalTime*, shown in section 4.5.

Table B.7: Properties Travel Time Path

Property	Value
<i>Travel Logic</i>	
Type	Unidirectional
Initial traveler capacity	Infinity
Entry Ranking Rule	First In First Out
Travel Time	ArrivalTime

Next, the arrival and unloading of the trucks commence.

B.2. Arrival and unloading

Once the randomly assigned TravelTime is finished, the trucks arrive at the terminal arrival node. At the terminal node, the number of inbound trucks that arrived is counted and the ATA as well as the start of the individual makespan is marked. The arrived truck is assigned an unloading dock using the data table Table 4.8. The properties of the arrival node are presented in Table B.8.

Table B.8: Properties Terminal Arrival Node

Property	Value
<i>State Assignments</i>	
On Entity Entering	NumInboundTrucksArrived == NumInboundTrucksArrived + 1
<i>Tally Statistics</i>	
On Entity Entering	Truck.ActualArrivalTime == TimeNow
On Entity Entering	Truck.IndividualMakespan == TimeNow
<i>Add-On Process Triggers</i>	
Entered	AssignUnloadingDock

From the terminal arrival node, the trucks move to the assigned unloading dock and start unloading. The unloading dock is modeled as a separator object seen in Figure B.7

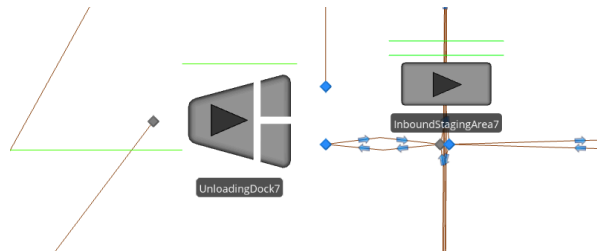


Figure B.7: Unloading dock object and inbound staging area object

The unloading object consists of one input node and two output nodes. One output node is for the parent entities and one is for the batch member entities. The unloading dock object has the properties shown in Table B.9.

Table B.9: Properties Unloading Dock

Property	Value
<i>Process Logic</i>	
Transfer-In Time	Math.If(Truck.BatchMembers > 0, TruckDockingTime, 0)
Processing Time	PU_DO_TimePerUnitLoad
<i>State Assignments</i>	
Before Parent	NumInboundTrucksToBeServed == NumInboundTrucksToBeServed - 1
Exiting	
Before Parent	NumOutboundTrucksToBeServed == NumOutboundTrucksToBeServed
Exiting	+ 1
<i>Add-On Process Triggers</i>	
Parent Exited	TrackLastUnloading

The truck docking time consists of the cumulative time for driver registration and docking at the door. The processing time equals the pick-up and drop-off time of the unit load and therefore differs for pallets or roll containers. At the member output nodes, the unit load entities exit the unloading dock and request a ride from the assigned transporter (Workers for roll containers and EPTs for pallets). Next, staging and moving commence.

B.3. Staging and internal transshipment

Once the UnitLoad entity requests a ride and the transport picks up the entity, the transporter checks if the entity is marked for the single-stage or two-stage procedure using the routing logic shown in Figure B.8. If the UnitLoad entity is assigned for two-stage cross-docking, the transporter moves the entity to the inbound staging area corresponding to the door at which the UL entity is picked up. The outbound staging area corresponding to the UnitLoad destination is searched, and the transporter brings the UL entity to the outbound staging area. UL entities destined for the CDF are directly moved to the CDF entity sink.

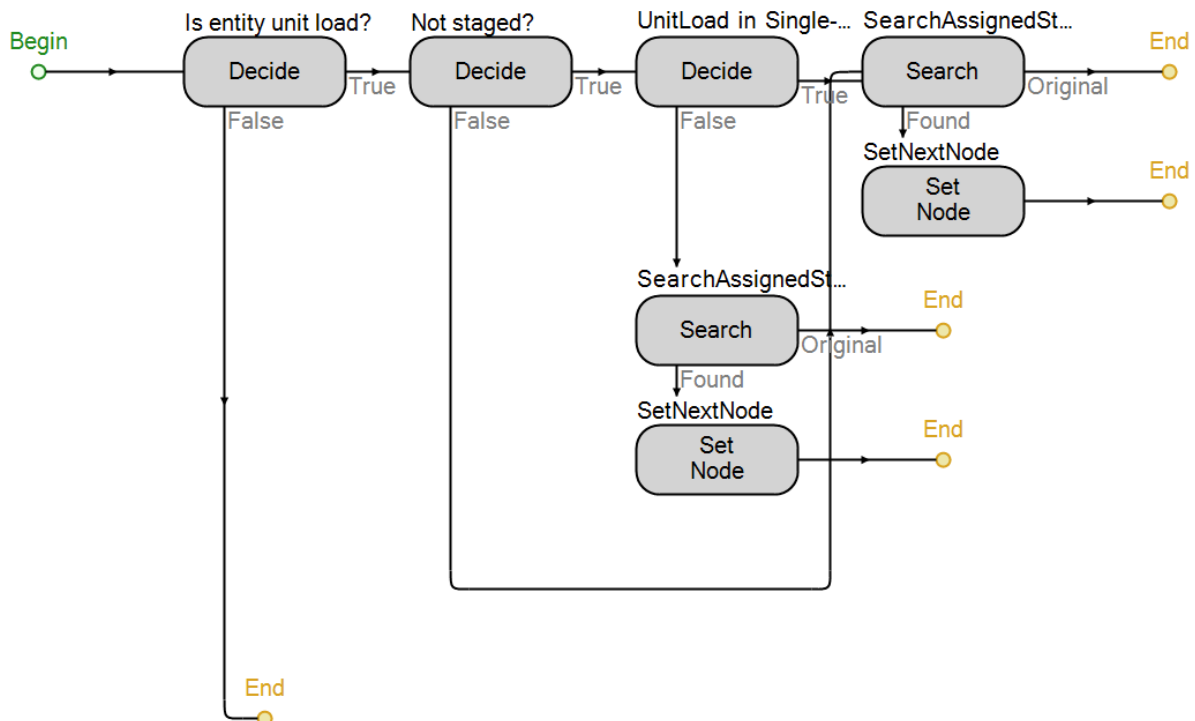


Figure B.8: Process Step: RoutingToStagingArea

For internal transshipment, the distance between objects is important to model according to real-

world measures. To this end, OD-matrices are obtained for distances between the inbound dock doors, inbound staging areas, outbound staging areas, and outbound dock doors.

Table B.10: OD matrix door-to-door distance [meters]

	RD1	RD2	RD3	RD4	RD5	RD6	RD7	RD8	RD9	RD10	CDF
UD1	-	12	24	28	37	41	53	57	66	70	50
UD2	12	-	20	24	33	37	49	53	62	66	62
UD3	24	20	-	18	27	31	43	47	60	60	74
UD4	28	24	18	-	23	27	39	43	52	56	78
UD5	37	33	27	23	-	18	30	34	43	47	87
UD6	41	37	31	27	18	-	26	30	39	43	93
UD7	53	49	43	39	30	26	-	24	33	37	87
UD8	57	53	47	43	34	30	24	-	29	33	83
UD9	66	62	60	52	43	39	33	29	-	24	74
UD10	70	66	60	56	47	43	37	33	24	-	60
CDF	50	62	74	78	87	93	87	83	74	60	-

Table B.11: OD matrix Inbound Staging Area (ISA) to Door [meters]

	RD1	RD2	RD3	RD4	RD5	RD6	RD7	RD8	RD9	RD10	CDF
ISA1	2	10	22	26	35	39	51	55	64	68	48
ISA2	10	2	18	22	31	35	47	51	60	64	60
ISA3	22	18	2	16	25	29	41	45	58	58	72
ISA4	26	22	16	2	21	25	37	41	50	54	76
ISA5	35	31	25	21	2	16	28	32	41	45	85
ISA6	39	35	29	25	16	2	24	28	37	41	91
ISA7	51	47	41	37	28	24	2	22	31	35	85
ISA8	55	51	45	41	32	28	22	2	27	31	81
ISA9	64	60	58	50	41	37	31	27	2	22	72
ISA10	68	64	58	54	45	41	35	31	22	2	58

An infinite number of EPTs can travel on the terminal pathways as long as the direction of travel matches since, due to limited terminal path widths, EPTs cannot overtake or pass each other. To solve possible deadlock situations where Transporters are stuck due to simultaneously entering links, as suggested by Sturrock [58], virtual roundabouts are instigated at which EPTs can pass, as shown in Figure B.9. Since the simulation model is drawn to scale, the transportation times are realistically proportional to the covered distance.

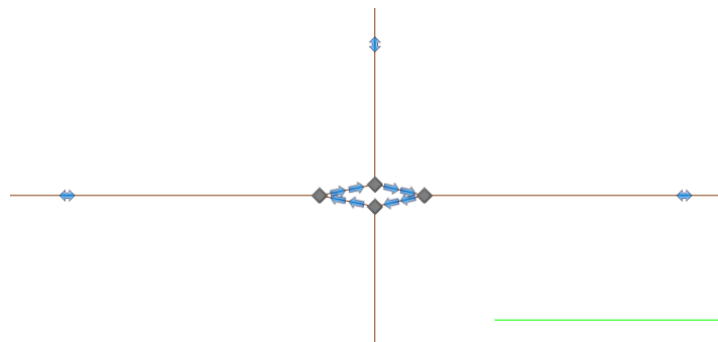


Figure B.9: Terminal Pathway Roundabout

Once the transporter arrives at the Loading Dock, the transported entity is dropped off and placed in the queue of the Loading Dock. This functions as the outbound staging area.

B.4. Reloading and Dispatching

Figure B.10 depicts the reloading dock as a combiner object. This object has two input nodes (one for Parent entities and one for Member entities) and one single output node where the batched entities leave from.

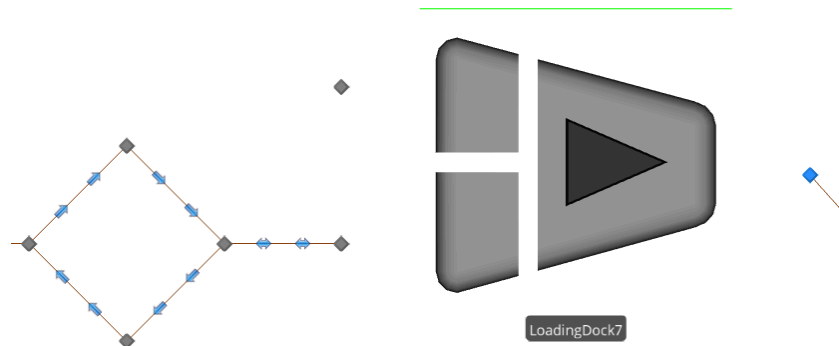


Figure B.10: Loading Dock

The properties of the Reloading Dock are shown in Table B.12. The quantity of the batch is determined by the maximum capacity of the truck and the entities are batched according to matching destinations. The *Release Batch Early Trigger* is based on the dispatching rule as soon as all unit loads for the respective truck are reloaded. Only one truck can dock at a time at the reloading dock and the docking time depends on whether the truck has to dock again or not. The processing time of a truck is proportional to the number of unit loads that are reloaded onto the truck. Upon exiting, the number of trucks left in the system is counted as well as the number of outbound trucks left. Finally, two Add-On Processes track the end of the individual makespan of the truck and route the truck to the sink corresponding to the truck's destination.

Table B.12: Properties Reloading Dock

Property	Value
<i>Batching Logic</i>	
Batch Quantity	CapacityTruck
Matching Rule	Match Members And Parent
Member Match	Unit_Load.EndDestination
Expression	
Parent Match	Truck. EndDestination
Expression	
Release Batch Early Trigger	
Trigger Type	Time Based
Wait Duration	0.0
Release Decision	Conditional
Type	
Release Condition	MemberInput@LoadingDock8.EntryQueue.NumberWaiting == 0
Or Probability	
<i>Process Logic</i>	
Capacity Type	Fixed
Initial Capacity	1
Parent Transfer-In	TruckDockingTime * TruckToUnloadingDockAssignment.
Time	TruckSwapping
Member Transfer-In	0.0
Time	
Process Type	Specific Time
Processing Time	(2 *PU_DO_TimePerUnitLoad) * Truck. BatchMembers.NumberWaiting
<i>State Assignments</i>	
Before Exiting	NumTrucksInSystem == NumTrucksInSystem - 1
Before Exiting	NumOutboundTrucksToBeServed == NumOutboundTrucksToBeServed - 1
<i>Add-On Process Triggers</i>	
After Processing	TrackLastReloading
Exited	RoutingToSink

To assess how the model behaves, the model trace is discussed in section B.5.

B.5. Model trace

Table B.13 shows the model trace for one Unit_Load entity (Forward.658) that is transported by one Truck (Truck_RT12) entity. By following the trace of these two entities, the model trace can be verified and the functionality of the model can be assessed.

Examining the trace shown in Table B.13, it becomes clear that at $t = 0$ the entity Forward.658 is generated. Unit load generation follows the following steps:

- Create unit load entity
- Assign origin to unit load entity state
- Assign destination that is not origin to unit load entity state [Forward.658, DestinationValue = 4]
- Batch member entity to truck parent entity that matches the origin state

The truck entity Truck_RT12.802 is generated at $t = 0.016667$ and follows the following steps:

- Create truck entity
- Assign origin to truck entity state
- Assign origin as destination to truck entity state
- Batch unit load entities truck parent entity for matching entity origin states

Next, the truck enters the *ArrivalTimePath* to the CDF, and a randomly generated travel time is assigned to the truck entity. At the end of the *ArrivalTimePath*, the arrival time is marked and the truck is assigned to UnloadingDock10 following the table 'TruckToUnloadingDockAssignment'. At the same time, the truck checks whether the single-stage cross-docking procedure is instigated and assigns this state to the batched entities accordingly.

At the unloading dock, the unloading time of the truck is marked and the batched unit load entities are removed from the truck one at a time. Pick-up and drop-off of the unit load entities are performed by one EPT (EPT10), which is assigned to the dock door.

Unit load entity Forward.658 checks if two-stage cross-docking is assigned to it. This is not the case, hence the next destination of the entity is set to the reloading dock corresponding to the entity destination state found in the table DestinationDockDoorAssignment. Forward.658 requests EPT10 to transport it to the found reloading dock and EPT10 transports the entity.

After completely unbatched (unloaded), Truck_RT12 marks the unloading time, is assigned a reloading door, leaves UnloadingDock10, and moves to the ParkingNode. Truck_RT12 waits until all trucks are unloaded and the start of the waiting time is marked.

Once all other trucks are unloaded, the waiting time is marked and the start of reloading is marked. Truck_RT12 then leaves the ParkingNode and proceeds to the LoadingDock10. Here, the following steps are taken:

- Batch all waiting unit load entities with matching entity state destinations
- Batching takes two times the pick-up/drop-off time of the EPT times the entity batch size
- The truck leaves if the batching capacity is reached or if the waiting queue is empty
- The end of reloading time is marked
- The truck is dispatched and the departure time is marked

Once Truck_RT12 is dispatched, the truck entity checks if all other trucks are already dispatched. If so, the run is ended, if not the run continues.

The described model trace is consistent with the model description stated in section 4.5. Therefore, the model is considered to be verified.

Table B.13: Model trace verification

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0	Forward.658	Forward_UnitLoadSource_RT12	13.OnEntityArrival	[Transfer] ToProcessing	Entity 'Forward.658' transferring from '[FreeSpace] Forward_UnitLoadSource_RT12' into station 'Forward_UnitLoadSource_RT12.Processing'.
0	Forward.658	Forward_UnitLoadSource_RT12	2.OnEnteredProcessing	[Begin]	Process 'Forward_UnitLoadSource_RT12.OnEnteredProcessing' execution started.
0	Forward.658	Forward_UnitLoadSource_RT12	2.OnEnteredProcessing	[EndTransfer] IntoProcessing	Entity 'Forward.658' ending transfer into station 'Forward_UnitLoadSource_RT12.Processing'.
0	Forward.658	Forward_UnitLoadSource_RT12	2.OnEnteredProcessing	[Execute] CreatedEntityAddOnProcess	Executing process 'AssignOriginDestination'.
0	Forward.658	Cross_Dock_EPTs	26.AssignOriginDestination	[Begin]	Process 'AssignOriginDestination' execution started.
0	Forward.658	Cross_Dock_EPTs	26.AssignOriginDestination	[Execute] Assign Origin	Executing process 'AssignOrigin'.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Begin]	Process 'AssignOrigin' execution started.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Decide] Unit Load?	Token branching on condition 'Math.If(Entity.EntityType.Is.Truck, False, True)'.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Decide] Unit Load?	Token sent to 'True' exit.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Decide] Forward?	Token branching on condition 'Unit_Load.EntityType.Is.Forward'.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Decide] Forward?	Token sent to 'True' exit.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Search] Search for origin to set	Searching collection '[ObjectList] OriginTerminals_UL' containing '26' item(s).
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Search] Search for origin to set	Searching collection in the forward direction for maximum of '1' item(s).
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Search] Search for origin to set	Index range of search is from starting index '1' to ending index '26'.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Search] Search for origin to set	Item 'Forward_UnitLoadSource_RT12' found at index '12' in the collection.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Search] Search for origin to set	'1' item(s) found by search.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[Search] Search for origin to set	Created token 'Token.8' associated with item 'Forward_UnitLoadSource_RT12' found at index '12' in the collection.
0	Forward.658	Cross_Dock_EPTs	18.AssignOrigin	[End]	Process 'AssignOrigin' execution ended.
0	Forward.658	Cross_Dock_EPTs	26.AssignOriginDestination	[Execute] Assign Destination	Executing process 'AssignDestination'.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Begin]	Process 'AssignDestination' execution started.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Decide] Unit Load?	Token branching on condition 'Math.If(Entity.EntityType.Is.Truck, False, True)'.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Decide] Unit Load?	Token sent to 'True' exit.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Decide] Forward?	Token branching on condition 'Unit_Load.EntityType.Is.Forward'.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Decide] Forward?	Token sent to 'True' exit.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[SetRow] Select Destination	Table reference for object or element 'Forward.658' set to row index '4' in table 'ProbabilisticDestinationsForward'.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Assign] Assign Destination to State	Assigning state variable 'Forward.658.EndDestination' the value 'Input@Sink_RT4'. Previous value was 'Nothing'.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Assign] Assign Destination to State	Assigning state variable 'Forward.658.DestinationValue' the value '4'. Previous value was '0'.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Search] Search for destination to set	Searching collection '[TableRows] ObjectDestinationReference' containing '13' item(s).
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Search] Search for destination to set	Searching collection in the forward direction for maximum of '1' item(s).
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Search] Search for destination to set	Index range of search is from starting index '1' to ending index '13'.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Search] Search for destination to set	Item 'ObjectDestinationReference[5]' found at index '5' in the collection.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Search] Search for destination to set	'1' item(s) found by search.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[Search] Search for destination to set	Created token 'Token.17' associated with item 'ObjectDestinationReference[5]' found at index '5' in the collection.
0	Forward.658	Cross_Dock_EPTs	17.AssignDestination	[Assign] Count ULs	Assigning state variable 'Cross_Dock_EPTs.UL_to_RT4' the value '1'. Previous value was '0'.
0	Forward.658	Cross_Dock_EPTs	17.AssignDestination	[End]	Process 'AssignDestination' execution ended.
0	Forward.658	Cross_Dock_EPTs	18.AssignDestination	[End]	Process 'AssignDestination' execution ended.
0	Forward.658	Cross_Dock_EPTs	26.AssignOriginDestination	[End]	Process 'AssignOriginDestination' execution ended.
0	Forward.658	Forward_UnitLoadSource_RT12	2.OnEnteredProcessing	[Transfer] ToOutputBuffer	Entity 'Forward.658' transferring from '[Station] Forward_UnitLoadSource_RT12.Processing' into station 'Forward_UnitLoadSource_RT12.OutputBuffer'.
0	Forward.658	Forward_UnitLoadSource_RT12	13.OnEntityArrival	[End]	Process 'Forward_UnitLoadSource_RT12.OnEntityArrival' execution ended.
0	Forward.658	Forward_UnitLoadSource_RT12	26.OnEnteredOutputBuffer	[Begin]	Process 'Forward_UnitLoadSource_RT12.OnEnteredOutputBuffer' execution started.
0	Forward.658	Forward_UnitLoadSource_RT12	26.OnEnteredOutputBuffer	[EndTransfer] IntoOutputBuffer	Entity 'Forward.658' ending transfer into station 'Forward_UnitLoadSource_RT12.OutputBuffer'.
0	Forward.658	Forward_UnitLoadSource_RT12	26.OnEnteredOutputBuffer	[Transfer] ToOutputNode	Entity 'Forward.658' transferring from '[Station] Forward_UnitLoadSource_RT12.OutputBuffer' into node 'Output@Forward_UnitLoadSource_RT12'.
0	Forward.658	Output@Forward_UnitLoadSource_RT12	13.OnEnteredFromAssociatedObject	[Begin]	Process 'Output@Forward_UnitLoadSource_RT12.OnEnteredFromAssociatedObject' execution started.
0	Forward.658	Output@Forward_UnitLoadSource_RT12	13.OnEnteredFromAssociatedObject	[Fire] EnteredEvent	Firing event 'Output@Forward_UnitLoadSource_RT12.Entered'.
0	Forward.658	Output@Forward_UnitLoadSource_RT12	13.OnEnteredFromAssociatedObject	[Transfer] ToOutboundLink	Entity 'Forward.658' transferring from '[Node] Output@Forward_UnitLoadSource_RT12' into node 'MemberInput@Terminal_RT12'.
0	Forward.658	MemberInput@Terminal_RT12	18.OnEnteredToAssociatedObject	[Begin]	Process 'MemberInput@Terminal_RT12.OnEnteredToAssociatedObject' execution started.
0	Forward.658	MemberInput@Terminal_RT12	18.OnEnteredToAssociatedObject	[Fire] EnteredEvent	Firing event 'MemberInput@Terminal_RT12.Entered'.
0	Forward.658	MemberInput@Terminal_RT12	18.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Forward.658' transferring from '[Node] MemberInput@Terminal_RT12' into object 'Terminal_RT12'.
0	Forward.658	MemberInput@Terminal_RT12	18.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Forward.658' transferring from '[Node] MemberInput@Terminal_RT12' into station 'Terminal_RT12.MemberInputBuffer'.
0	Forward.658	Forward_UnitLoadSource_RT12	2.OnEnteredProcessing	[End]	Process 'Forward_UnitLoadSource_RT12.OnEnteredProcessing' execution ended.
0	Forward.658	Terminal_RT12	8.OnEnteredMemberInputBuffer	[Begin]	Process 'Terminal_RT12.OnEnteredMemberInputBuffer' execution started.
0	Forward.658	Terminal_RT12	8.OnEnteredMemberInputBuffer	[Assign] OnMemberEnteringAssignments	Assigning state variable 'Cross_Dock_EPTs.NumUnitLoadsInSystem' the value '18'. Previous value was '17'.
0	Forward.658	Terminal_RT12	8.OnEnteredMemberInputBuffer	[Assign] OnMemberEnteringAssignments	Assigning state variable 'Cross_Dock_EPTs.NumUnitLoadsToBeCrossDocked' the value '18'. Previous value was '17'.
0	Forward.658	Terminal_RT12	8.OnEnteredMemberInputBuffer	[EndTransfer] IntoMemberInputBuffer	Entity 'Forward.658' ending transfer into station 'Terminal_RT12.MemberInputBuffer'.
0	Forward.658	Output@Forward_UnitLoadSource_RT12	2.OnExited	[Begin]	Process 'Output@Forward_UnitLoadSource_RT12.OnExited' execution started.
0	Forward.658	Output@Forward_UnitLoadSource_RT12	2.OnExited	[Fire] ExitedEvent	Firing event 'Output@Forward_UnitLoadSource_RT12.Exited'.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0	Forward.658	Output@Forward_UnitLoadSource_RT12	2.OnExited	[End]	Process 'Output@Forward_UnitLoadSource_RT12.OnExited' execution ended.
0	Forward.658	MemberInput@Terminal_RT12	2.OnExited	[Begin]	Process 'MemberInput@Terminal_RT12.OnExited' execution started.
0	Forward.658	MemberInput@Terminal_RT12	2.OnExited	[Fire] ExitedEvent	Firing event 'MemberInput@Terminal_RT12.Exited'.
0	Forward.658	MemberInput@Terminal_RT12	2.OnExited	[End]	Process 'MemberInput@Terminal_RT12.OnExited' execution ended.
0	Forward.658	Terminal_RT12	8.OnEnteredMemberInputBuffer	[Batch] Member	Entity 'Forward.658' waiting to be batched by batch logic 'Terminal_RT12.Batching'.
0	Forward.658	Forward_UnitLoadSource_RT12	26.OnEnteredOutputBuffer	[End]	Process 'Forward_UnitLoadSource_RT12.OnEnteredOutputBuffer' execution ended.
0	Forward.658	Output@Forward_UnitLoadSource_RT12	13.OnEnteredFromAssociatedObject	[End]	Process 'Output@Forward_UnitLoadSource_RT12.OnEnteredFromAssociatedObject' execution ended.
0	Forward.658	MemberInput@Terminal_RT12	18.OnEnteredToAssociatedObject	[End]	Process 'MemberInput@Terminal_RT12.OnEnteredToAssociatedObject' execution ended.
0.016666667	Truck_RT12.802	TruckSource_RT12	102.OnEntityArrival	[Transfer] ToProcessing	Entity 'Truck_RT12.802' transferring from '[FreeSpace] TruckSource_RT12' into station 'TruckSource_RT12.Processing'.
0.016666667	Truck_RT12.802	TruckSource_RT12	18.OnEnteredProcessing	[Begin]	Process 'TruckSource_RT12.OnEnteredProcessing' execution started.
0.016666667	Truck_RT12.802	TruckSource_RT12	18.OnEnteredProcessing	[EndTransfer] IntoProcessing	Entity 'Truck_RT12.802' ending transfer into station 'TruckSource_RT12.Processing'.
0.016666667	Truck_RT12.802	TruckSource_RT12	18.OnEnteredProcessing	[Execute] CreatedEntityAddOnProcess	Executing process 'AssignOriginDestination'.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	100.AssignOriginDestination	[Begin]	Process 'AssignOriginDestination' execution started.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	100.AssignOriginDestination	[Execute] Assign Origin	Executing process 'AssignOrigin'.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[Begin]	Process 'AssignOrigin' execution started.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[Decide] Unit Load?	Token branching on condition 'Math.If(Entity.EntityType.Is.Truck, False, True)'.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[Decide] Unit Load?	Token sent to 'False' exit.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[Search] Search for origin to set	Searching collection '[ObjectList] OriginTerminals_Truck' containing '12' item(s).
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[Search] Search for origin to set	Searching collection in the forward direction for maximum of '1' item(s).
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[Search] Search for origin to set	Index range of search is from starting index '1' to ending index '12'.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[Search] Search for origin to set	Item 'TruckSource_RT12' found at index '12' in the collection.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[Search] Search for origin to set	'1' item(s) found by search.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[Search] Search for origin to set	Created token 'Token.114' associated with item 'TruckSource_RT12' found at index '12' in the collection.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignOrigin	[End]	Process 'AssignOrigin' execution ended.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	100.AssignOriginDestination	[Execute] Assign Destination	Executing process 'AssignDestination'.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignDestination	[Begin]	Process 'AssignDestination' execution started.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignDestination	[Decide] Unit Load?	Token branching on condition 'Math.If(Entity.EntityType.Is.Truck, False, True)'.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignDestination	[Decide] Unit Load?	Token sent to 'False' exit.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	101.AssignDestination	[End]	Process 'AssignDestination' execution ended.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	100.AssignOriginDestination	[End]	Process 'AssignOriginDestination' execution ended.
0.016666667	Truck_RT12.802	TruckSource_RT12	18.OnEnteredProcessing	[Transfer] ToOutputBuffer	Entity 'Truck_RT12.802' transferring from '[Station] TruckSource_RT12.Processing' into station 'TruckSource_RT12.OutputBuffer'.
0.016666667	Truck_RT12.802	TruckSource_RT12	102.OnEntityArrival	[End]	Process 'TruckSource_RT12.OnEntityArrival' execution ended.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.016666667	Truck_RT12.802	TruckSource_RT12	100.OnEnteredOutputBuffer	[Begin]	Process 'TruckSource_RT12.OnEnteredOutputBuffer' execution started.
0.016666667	Truck_RT12.802	TruckSource_RT12	100.OnEnteredOutputBuffer	[EndTransfer] IntoOutputBuffer	Entity 'Truck_RT12.802' ending transfer into station 'TruckSource_RT12.OutputBuffer'.
0.016666667	Truck_RT12.802	TruckSource_RT12	100.OnEnteredOutputBuffer	[Transfer] ToOutputNode	Entity 'Truck_RT12.802' transferring from '[Station] TruckSource_RT12.OutputBuffer' into node 'Output@TruckSource_RT12'.
0.016666667	Truck_RT12.802	Output@TruckSource_RT12	102.OnEnteredFromAssociatedObject	[Begin]	Process 'Output@TruckSource_RT12.OnEnteredFromAssociatedObject' execution started.
0.016666667	Truck_RT12.802	Output@TruckSource_RT12	102.OnEnteredFromAssociatedObject	[Assign] OnEnteringAssignments	Assigning state variable 'Truck_RT12.802.InboundOrOutbound' the value '1'. Previous value was '0'.
0.016666667	Truck_RT12.802	Output@TruckSource_RT12	102.OnEnteredFromAssociatedObject	[Fire] EnteredEvent	Firing event 'Output@TruckSource_RT12.Entered'.
0.016666667	Truck_RT12.802	Output@TruckSource_RT12	102.OnEnteredFromAssociatedObject	[Transfer] ToOutboundLink	Entity 'Truck_RT12.802' transferring from '[Node] Output@TruckSource_RT12' into node 'ParentInput@Terminal_RT12'.
0.016666667	Truck_RT12.802	ParentInput@Terminal_RT12	101.OnEnteredToAssociatedObject	[Begin]	Process 'ParentInput@Terminal_RT12.OnEnteredToAssociatedObject' execution started.
0.016666667	Truck_RT12.802	ParentInput@Terminal_RT12	101.OnEnteredToAssociatedObject	[Fire] EnteredEvent	Firing event 'ParentInput@Terminal_RT12.Entered'.
0.016666667	Truck_RT12.802	ParentInput@Terminal_RT12	101.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Truck_RT12.802' transferring from '[Node] ParentInput@Terminal_RT12' into object 'Terminal_RT12'.
0.016666667	Truck_RT12.802	ParentInput@Terminal_RT12	101.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Truck_RT12.802' transferring from '[Node] ParentInput@Terminal_RT12' into station 'Terminal_RT12.ParentInputBuffer'.
0.016666667	Truck_RT12.802	TruckSource_RT12	18.OnEnteredProcessing	[End]	Process 'TruckSource_RT12.OnEnteredProcessing' execution ended.
0.016666667	Truck_RT12.802	TruckSource_RT12	114.OnExitedOutputBuffer	[Begin]	Process 'TruckSource_RT12.OnExitedOutputBuffer' execution started.
0.016666667	Truck_RT12.802	TruckSource_RT12	114.OnExitedOutputBuffer	[Execute] ExitedAddOnProcess	Executing process 'AssignDestination'.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	18.AssignDestination	[Begin]	Process 'AssignDestination' execution started.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	18.AssignDestination	[Decide] Unit Load?	Token branching on condition 'Math.If(Entity.EntityType.Is.Truck, False, True)'.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	18.AssignDestination	[Decide] Unit Load?	Token sent to 'False' exit.
0.016666667	Truck_RT12.802	Cross_Dock_EPTs	18.AssignDestination	[End]	Process 'AssignDestination' execution ended.
0.016666667	Truck_RT12.802	TruckSource_RT12	114.OnExitedOutputBuffer	[End]	Process 'TruckSource_RT12.OnExitedOutputBuffer' execution ended.
0.016666667	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Begin]	Process 'Terminal_RT12.OnEnteredParentInputBuffer' execution started.
0.016666667	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[EndTransfer] IntoParentInputBuffer	Entity 'Truck_RT12.802' ending transfer into station 'Terminal_RT12.ParentInputBuffer'.
0.016666667	Truck_RT12.802	Output@TruckSource_RT12	114.OnExited	[Begin]	Process 'Output@TruckSource_RT12.OnExited' execution started.
0.016666667	Truck_RT12.802	Output@TruckSource_RT12	114.OnExited	[Fire] ExitedEvent	Firing event 'Output@TruckSource_RT12.Exited'.
0.016666667	Truck_RT12.802	Output@TruckSource_RT12	114.OnExited	[End]	Process 'Output@TruckSource_RT12.OnExited' execution ended.
0.016666667	Truck_RT12.802	ParentInput@Terminal_RT12	114.OnExited	[Begin]	Process 'ParentInput@Terminal_RT12.OnExited' execution started.
0.016666667	Truck_RT12.802	ParentInput@Terminal_RT12	114.OnExited	[Fire] ExitedEvent	Firing event 'ParentInput@Terminal_RT12.Exited'.
0.016666667	Truck_RT12.802	ParentInput@Terminal_RT12	114.OnExited	[End]	Process 'ParentInput@Terminal_RT12.OnExited' execution ended.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.016666667	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄòTruck_RT12.802,Äò waiting to collect ,Äò33,Äò batch member(s) from queue state ,ÄòTerminal_RT12.Batching.MemberQueue,Äò using matching rule ,Äò[AnyEntity],Äò.
0.016666667	Truck_RT12.802	TruckSource_RT12	100.OnEnteredOutputBuffer	[End]	Process 'TruckSource_RT12.OnEnteredOutputBuffer' execution ended.
0.016666667	Truck_RT12.802	Output@TruckSource_RT12	102.OnEnteredFromAssociatedObject	[End]	Process 'Output@TruckSource_RT12.OnEnteredFromAssociatedObject' execution ended.
0.016666667	Truck_RT12.802	ParentInput@Terminal_RT12	101.OnEnteredToAssociatedObject	[End]	Process 'ParentInput@Terminal_RT12.OnEnteredToAssociatedObject' execution ended.
0.016666667	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄòTruck_RT12.802,Äò found ,Äò3,Äò currently available batch member(s) waiting in queue state ,ÄòTerminal_RT12.Batching.MemberQueue,Äò using matching rule ,Äò[AnyEntity],Äò.
0.016666667	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄòTruck_RT12.802,Äò collecting batch member ,ÄòReturn.641,Äò from queue state ,ÄòTerminal_RT12.Batching.MemberQueue,Äò using matching rule '[AnyEntity]'.
0.016666667	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄòTruck_RT12.802,Äò collecting batch member ,ÄòReturn.642,Äò from queue state ,ÄòTerminal_RT12.Batching.MemberQueue,Äò using matching rule '[AnyEntity]'.
0.016666667	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄòTruck_RT12.802,Äò collecting batch member ,ÄòForward.658,Äò from queue state ,ÄòTerminal_RT12.Batching.MemberQueue,Äò using matching rule '[AnyEntity]'.
0.016666667	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄòTruck_RT12.802,Äò found ,Äò0,Äò currently available batch member(s) waiting in queue state ,ÄòTerminal_RT12.Batching.MemberQueue,Äò using matching rule ,Äò[AnyEntity],Äò.
0.016666667	Forward.658	Terminal_RT12	8.OnEnteredMemberInputBuffer	[End]	Process 'Terminal_RT12.OnEnteredMemberInputBuffer' execution ended.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Batch] Parent	Release batch early trigger fired for entity ,ÄòTruck_RT12.802,Äò waiting to collect batch member(s) from queue state ,ÄòTerminal_RT12.Batching.MemberQueue,Äò. Wait duration '0.0166666666666667' Hours expired.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄòTruck_RT12.802,Äò found ,Äò0,Äò currently available batch member(s) waiting in queue state ,ÄòTerminal_RT12.Batching.MemberQueue,Äò using matching rule ,Äò[AnyEntity],Äò.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄòTruck_RT12.802,Äò cancelling request to collect any further batch members from queue state ,ÄòTerminal_RT12.Batching.MemberQueue,Äò.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Seize] Combiner	Object 'Truck_RT12.802' waiting to seize '1' resource(s) of resource type '[ParentObject] Terminal_RT12'.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Seize] Combiner	Object 'Truck_RT12.802' checking availability to seize '1' resource(s) of resource type '[ParentObject] Terminal_RT12'.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Seize] Combiner	Object 'Truck_RT12.802' checking availability to seize '1' capacity unit(s) of resource 'Terminal_RT12,Äò.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Seize] Combiner	Object 'Truck_RT12.802' found ,Äò1,Äò currently available capacity unit(s) of resource 'Terminal_RT12,Äò.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Seize] Combiner	Object 'Truck_RT12.802' found '1' currently available resource(s) of resource type '[ParentObject] Terminal_RT12'.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Seize] Combiner	Object 'Truck_RT12.802' seized '1' capacity unit(s) of resource 'Terminal_RT12'.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Seize] Combiner	Object 'Truck_RT12.802' completed seize of '1' resource(s) of resource type '[ParentObject] Terminal_RT12'.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[Transfer] ToProcessing	Entity 'Truck_RT12.802' transferring from '[Station] Terminal_RT12.ParentInputBuffer' into station 'Terminal_RT12.Processing'.
0.033333333	Truck_RT12.802	Terminal_RT12	51.OnEnteredProcessing	[Begin]	Process 'Terminal_RT12.OnEnteredProcessing' execution started.
0.033333333	Truck_RT12.802	Terminal_RT12	51.OnEnteredProcessing	[EndTransfer] IntoProcessing	Entity 'Truck_RT12.802' ending transfer into station 'Terminal_RT12.Processing'.
0.033333333	Truck_RT12.802	Terminal_RT12	51.OnEnteredProcessing	[Decide] IfProcessTypeSpecificTime	Token branching on condition 'ProcessType==Enum.TaskProcessType.SpecificTime'.
0.033333333	Truck_RT12.802	Terminal_RT12	51.OnEnteredProcessing	[Decide] IfProcessTypeSpecificTime	Token sent to 'True' exit.
0.033333333	Truck_RT12.802	Terminal_RT12	51.OnEnteredProcessing	[Transfer] ToOutputBuffer	Entity 'Truck_RT12.802' transferring from '[Station] Terminal_RT12.Processing' into station 'Terminal_RT12.OutputBuffer'.
0.033333333	Truck_RT12.802	Terminal_RT12	103.OnEnteredParentInputBuffer	[End]	Process 'Terminal_RT12.OnEnteredParentInputBuffer' execution ended.
0.033333333	Truck_RT12.802	Terminal_RT12	154.OnEnteredOutputBuffer	[Begin]	Process 'Terminal_RT12.OnEnteredOutputBuffer' execution started.
0.033333333	Truck_RT12.802	Terminal_RT12	154.OnEnteredOutputBuffer	[EndTransfer] IntoOutputBuffer	Entity 'Truck_RT12.802' ending transfer into station 'Terminal_RT12.OutputBuffer'.
0.033333333	Truck_RT12.802	Terminal_RT12	154.OnEnteredOutputBuffer	[Release] Combiner	Object 'Truck_RT12.802' attempting to release '1' resource(s) of resource type '[ParentObject] Terminal_RT12'.
0.033333333	Truck_RT12.802	Terminal_RT12	154.OnEnteredOutputBuffer	[Release] Combiner	Object 'Truck_RT12.802' released '1' capacity unit(s) of resource 'Terminal_RT12'.
0.033333333	Truck_RT12.802	Terminal_RT12	154.OnEnteredOutputBuffer	[Assign] BeforeExitingAssignments	Assigning state variable 'Cross_Dock_EPTs.NumTrucksInSystem' the value '2'. Previous value was '1'.
0.033333333	Truck_RT12.802	Terminal_RT12	154.OnEnteredOutputBuffer	[Assign] BeforeExitingAssignments	Assigning state variable 'Cross_Dock_EPTs.NumInboundTrucksToBeServed' the value '2'. Previous value was '1'.
0.033333333	Truck_RT12.802	Terminal_RT12	154.OnEnteredOutputBuffer	[Transfer] ToOutputNode	Entity 'Truck_RT12.802' transferring from '[Station] Terminal_RT12.OutputBuffer' into node 'Output@Terminal_RT12'.
0.033333333	Truck_RT12.802	Output@Terminal_RT12	103.OnEnteredFromAssociatedObject	[Begin]	Process 'Output@Terminal_RT12.OnEnteredFromAssociatedObject' execution started.
0.033333333	Truck_RT12.802	Output@Terminal_RT12	103.OnEnteredFromAssociatedObject	[Fire] EnteredEvent	Firing event 'Output@Terminal_RT12.Entered'.
0.033333333	Truck_RT12.802	Output@Terminal_RT12	103.OnEnteredFromAssociatedObject	[Execute] EnteredAddOnProcess	Executing process 'FireEventStartInbounding'.
0.033333333	Truck_RT12.802	Cross_Dock_EPTs	3.FireEventStartInbounding	[Begin]	Process 'FireEventStartInbounding' execution started.
0.033333333	Truck_RT12.802	Cross_Dock_EPTs	3.FireEventStartInbounding	[Fire] FireInbounding	Firing event 'StartInbound'.
0.033333333	Truck_RT12.802	Cross_Dock_EPTs	3.FireEventStartInbounding	[End]	Process 'FireEventStartInbounding' execution ended.
0.033333333	Truck_RT12.802	Output@Terminal_RT12	103.OnEnteredFromAssociatedObject	[Transfer] ToOutboundLink	Entity 'Truck_RT12.802' transferring from '[Node] Output@Terminal_RT12' onto link 'ArrivalTimePath1'.
0.033333333	Truck_RT12.802	ArrivalTimePath1	3.OnEntered	[Begin]	Process 'ArrivalTimePath1.OnEntered' execution started.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.033333333	Truck_RT12.802	ArrivalTimePath1	3.OnEntered	[Assign] EntityMovementRate	Assigning state variable 'ArrivalTimePath1.Token.3.ReturnValue' the value '0.0633769973848012'. Previous value was '1'.
0.033333333	Truck_RT12.802	ArrivalTimePath1	3.OnEntered	[Assign] EntityMovementRate	Assigning state variable 'Truck_RT12.802.MovementRate' the value '408.507835150436' Meters per Hour. Previous value was 'â' Meters per Hour.
0.033333333	Truck_RT12.802	ArrivalTimePath1	3.OnEntered	[EndTransfer] OntoTimePath	Entity 'Truck_RT12.802' ending transfer onto link 'ArrivalTimePath1'.
0.033333333	Truck_RT12.802	ArrivalTimePath1	3.OnEntered	[End]	Process 'ArrivalTimePath1.OnEntered' execution ended.
0.033333333	Truck_RT12.802	Terminal_RT12	51.OnEnteredProcessing	[End]	Process 'Terminal_RT12.OnEnteredProcessing' execution ended.
0.033333333	Truck_RT12.802	Terminal_RT12	154.OnEnteredOutputBuffer	[End]	Process 'Terminal_RT12.OnEnteredOutputBuffer' execution ended.
0.033333333	Truck_RT12.802	Output@Terminal_RT12	103.OnEnteredFromAssociatedObject	[End]	Process 'Output@Terminal_RT12.OnEnteredFromAssociatedObject' execution ended.
0.03639325	Truck_RT12.802	Output@Terminal_RT12	172.OnExited	[Begin]	Process 'Output@Terminal_RT12.OnExited' execution started.
0.03639325	Truck_RT12.802	Output@Terminal_RT12	172.OnExited	[Fire] ExitedEvent	Firing event 'Output@Terminal_RT12.Exited'.
0.03639325	Truck_RT12.802	Output@Terminal_RT12	172.OnExited	[End]	Process 'Output@Terminal_RT12.OnExited' execution ended.
0.096710331	Truck_RT12.802	ArrivalTimePath1	171.OnReachedEnd	[Begin]	Process 'ArrivalTimePath1.OnReachedEnd' execution started.
0.096710331	Truck_RT12.802	ArrivalTimePath1	171.OnReachedEnd	[Transfer] FromTimePath	Entity 'Truck_RT12.802' transferring from '[EndOfLink] ArrivalTimePath1' into node 'BasicNode3'.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[Begin]	Process 'BasicNode3.OnEntered' execution started.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[Assign] OnEnteringAssignments	Assigning state variable 'Cross_Dock_EPTs.NumInboundTrucksArrived' the value '2'. Previous value was '1'.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[Tally] OnEnteringTallies	Recording observation value(s) of type 'Expression' for tally statistic 'Truck_RT12.802.ActualArrivalTime'.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[Tally] OnEnteringTallies	Recording observation value '0.0967103307181345' Hours for tally statistic 'Truck_RT12.802.ActualArrivalTime'.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[Tally] OnEnteringTallies	Recording observation value(s) of type 'Expression' for tally statistic 'Truck_RT12.802.IndividualMakespan'.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[Tally] OnEnteringTallies	Recording observation value '0.0967103307181345' Hours for tally statistic 'Truck_RT12.802.IndividualMakespan'.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[Fire] EnteredEvent	Firing event 'BasicNode3.Entered'.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[Execute] EnteredAddOnProcess	Executing process 'AssignUnloadingDock'.
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	[Begin]	Process 'AssignUnloadingDock' execution started.
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	[Assign] Mark ATA	Assigning state variable 'Truck_RT12.802.ArrivalCDF' the value '0.0967103307181345'. Previous value was '0'.
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	[Search] SearchAssignedUnloadingDock	Searching collection '[TableRows] TruckToUnloadingDockAssignment' containing '12' item(s).
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	[Search] SearchAssignedUnloadingDock	Searching collection in the forward direction for maximum of '1' item(s).

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	[Search]	Index range of search is from starting index '1' to ending index '12'.
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	SearchAssignedUnloadingDock	Item 'TruckToUnloadingDockAssignment[10]' found at index '10' in the collection.
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	[Search]	'1' item(s) found by search.
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	SearchAssignedUnloadingDock	
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	[Search]	Created token 'Token.2' associated with item 'TruckToUnloadingDockAssignment[10]' found at index '10' in the collection.
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	2.AssignUnloadingDock	[SetNode] SetNextNode	Destination node of entity 'Truck_RT12.802' set to value 'Input@UnloadingDock10'.
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	2.AssignUnloadingDock	[End]	Process 'AssignUnloadingDock' execution ended.
0.096710331	Truck_RT12.802	Cross_Dock_EPTs	152.AssignUnloadingDock	[End]	Process 'AssignUnloadingDock' execution ended.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[Transfer] ToOutboundLink	Entity 'Truck_RT12.802' transferring from '[Node] BasicNode3' onto link 'InstantTimePath21'.
0.096710331	Truck_RT12.802	InstantTimePath21	152.OnEntered	[Begin]	Process 'InstantTimePath21.OnEntered' execution started.
0.096710331	Truck_RT12.802	InstantTimePath21	152.OnEntered	[Assign] EntityMovementRate	Assigning state variable 'InstantTimePath21.Token.152.ReturnValue' the value '0'. Previous value was '1'.
0.096710331	Truck_RT12.802	InstantTimePath21	152.OnEntered	[Assign] EntityMovementRate	Assigning state variable 'Truck_RT12.802.Movement.Rate' the value 'â' Meters per Hour. Previous value was 'â' Meters per Hour.
0.096710331	Truck_RT12.802	InstantTimePath21	152.OnEntered	[EndTransfer] OntoTimePath	Entity 'Truck_RT12.802' ending transfer onto link 'InstantTimePath21'.
0.096710331	Truck_RT12.802	InstantTimePath21	152.OnEntered	[End]	Process 'InstantTimePath21.OnEntered' execution ended.
0.096710331	Truck_RT12.802	ArrivalTimePath1	171.OnReachedEnd	[End]	Process 'ArrivalTimePath1.OnReachedEnd' execution ended.
0.096710331	Truck_RT12.802	BasicNode3	154.OnEntered	[End]	Process 'BasicNode3.OnEntered' execution ended.
0.096710331	Truck_RT12.802	BasicNode3	154.OnExited	[Begin]	Process 'BasicNode3.OnExited' execution started.
0.096710331	Truck_RT12.802	BasicNode3	154.OnExited	[Fire] ExitedEvent	Firing event 'BasicNode3.Exited'.
0.096710331	Truck_RT12.802	BasicNode3	154.OnExited	[End]	Process 'BasicNode3.OnExited' execution ended.
0.096710331	Truck_RT12.802	InstantTimePath21	154.OnReachedEnd	[Begin]	Process 'InstantTimePath21.OnReachedEnd' execution started.
0.096710331	Truck_RT12.802	InstantTimePath21	154.OnReachedEnd	[Transfer] FromTimePath	Entity 'Truck_RT12.802' transferring from '[EndOfLink] InstantTimePath21' into node 'Input@UnloadingDock10'.
0.096710331	Truck_RT12.802	Input@UnloadingDock10	171.OnEnteredToAssociatedObject	[Begin]	Process 'Input@UnloadingDock10.OnEnteredToAssociatedObject' execution started.
0.096710331	Truck_RT12.802	Input@UnloadingDock10	171.OnEnteredToAssociatedObject	[Fire] EnteredEvent	Firing event 'Input@UnloadingDock10.Entered'.
0.096710331	Truck_RT12.802	Input@UnloadingDock10	171.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Truck_RT12.802' transferring from '[Node] Input@UnloadingDock10' into object 'UnloadingDock10'.
0.096710331	Truck_RT12.802	Input@UnloadingDock10	171.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Truck_RT12.802' transferring from '[Node] Input@UnloadingDock10' into station 'UnloadingDock10.InputBuffer'.
0.096710331	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Begin]	Process 'UnloadingDock10.OnEnteredInputBuffer' execution started.
0.096710331	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Assign] OnEnteringAssignments	Assigning state variable 'Truck_RT12.802.AssignedUnloadingDock' the value 'UnloadingDock10'. Previous value was 'Nothing'.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.096710331	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Delay] TransferInTime	Delaying token for '0.105514929667165' Hours until time '0.2022252603853' Hours. Delay multiplier was '1'.
0.20222526	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[EndTransfer] IntoInputBuffer	Entity 'Truck_RT12.802' ending transfer into station 'UnloadingDock10.InputBuffer'.
0.20222526	Truck_RT12.802	Input@UnloadingDock10	180.OnExited	[Begin]	Process 'Input@UnloadingDock10.OnExited' execution started.
0.20222526	Truck_RT12.802	Input@UnloadingDock10	180.OnExited	[Fire] ExitedEvent	Firing event 'Input@UnloadingDock10.Exited'.
0.20222526	Truck_RT12.802	Input@UnloadingDock10	180.OnExited	[End]	Process 'Input@UnloadingDock10.OnExited' execution ended.
0.20222526	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Seize] Separator	Object 'Truck_RT12.802' checking availability to seize '1' resource(s) of resource type '[ParentObject] UnloadingDock10'.
0.20222526	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Seize] Separator	Object 'Truck_RT12.802' checking availability to seize '1' capacity unit(s) of resource 'UnloadingDock10.Åö'.
0.20222526	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Seize] Separator	Object 'Truck_RT12.802' found 'Åö1,Åö currently available capacity unit(s) of resource 'UnloadingDock10.Åö'.
0.20222526	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Seize] Separator	Object 'Truck_RT12.802' found '1' currently available resource(s) of resource type '[ParentObject] UnloadingDock10'.
0.20222526	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Seize] Separator	Object 'Truck_RT12.802' seized '1' capacity unit(s) of resource 'UnloadingDock10'.
0.20222526	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Seize] Separator	Object 'Truck_RT12.802' completed seize of '1' resource(s) of resource type '[ParentObject] UnloadingDock10'.
0.20222526	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[Transfer] ToProcessing	Entity 'Truck_RT12.802' transferring from '[Station] UnloadingDock10.InputBuffer' into station 'UnloadingDock10.Processing'.
0.20222526	Truck_RT12.802	InstantTimePath21	154.OnReachedEnd	[End]	Process 'InstantTimePath21.OnReachedEnd' execution ended.
0.20222526	Truck_RT12.802	Input@UnloadingDock10	171.OnEnteredToAssociatedObject	[End]	Process 'Input@UnloadingDock10.OnEnteredToAssociatedObject' execution ended.
0.20222526	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Begin]	Process 'UnloadingDock10.OnEnteredProcessing' execution started.
0.20222526	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[EndTransfer] IntoProcessing	Entity 'Truck_RT12.802' ending transfer into station 'UnloadingDock10.Processing'.
0.20222526	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Execute] ProcessingAddOnProcess	Executing process 'CommenceUnloading'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	171.CommenceUnloading	[Begin]	Process 'CommenceUnloading' execution started.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	171.CommenceUnloading	[Execute] Mark Arrival Unloading Dock	Executing process 'Mark_Arrival_UnloadingDock'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.Mark_Arrival_UnloadingDock	[Begin]	Process 'Mark_Arrival_UnloadingDock' execution started.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.Mark_Arrival_UnloadingDock	[Assign] Mark Unloading Time	Assigning state variable 'Truck_RT12.802.UnloadingTime' the value '0.2022252603853'. Previous value was '0'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.Mark_Arrival_UnloadingDock	[Tally] Time before unloading	Recording 'Åö1,Åö observation value(s) of type 'ÅöExpression,Åö for tally statistic 'ÅöWaitingTimeBeforeUnloading,Åö.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.Mark_Arrival_UnloadingDock	[Tally] Time before unloading	Recording observation value '0.105514929667165' for tally statistic 'WaitingTimeBeforeUnloading'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.Mark_Arrival_UnloadingDock	[End]	Process 'Mark_Arrival_UnloadingDock' execution ended.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	171.CommenceUnloading	[Execute] CountArrivingTruck	Executing process 'CountArrivingTruck'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.CountArrivingTruck	[Begin]	Process 'CountArrivingTruck' execution started.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.CountArrivingTruck	[Assign] Count arrival sequence	Assigning state variable 'Cross_Dock_EPTs.ArrivalSequence' the value '3'. Previous value was '2'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.CountArrivingTruck	[Assign] Count arrival sequence	Assigning state variable 'Truck_RT12.802.ArrivalSequence' the value '3'. Previous value was '0'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.CountArrivingTruck	[Execute] Mark Truck Stage	Executing process 'MarkTruckStage'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	3.MarkTruckStage	[Begin]	Process 'MarkTruckStage' execution started.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	3.MarkTruckStage	[Decide] Single-Stage?	Token branching on condition 'TwoStage_CD == False'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	3.MarkTruckStage	[Decide] Single-Stage?	Token sent to 'True' exit.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	3.MarkTruckStage	[Assign] Assign Truck as SingleStage	Assigning state variable 'Truck_RT12.802.TwoStageCD' the value '0'. Previous value was '0'.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	3.MarkTruckStage	[End]	Process 'MarkTruckStage' execution ended.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	154.CountArrivingTruck	[End]	Process 'CountArrivingTruck' execution ended.
0.20222526	Truck_RT12.802	Cross_Dock_EPTs	171.CommenceUnloading	[End]	Process 'CommenceUnloading' execution ended.
0.20222526	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Decide] IfProcessTypeSpecificTime	Token branching on condition 'ProcessType==Enum.TaskProcessType.SpecificTime'.
0.20222526	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Decide] IfProcessTypeSpecificTime	Token sent to 'True' exit.
0.20222526	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Delay] ProcessingTime	Delaying token for '0.00422505264675229' Hours until time '0.206450313032052' Hours. Delay multiplier was '1'.
0.20222526	Truck_RT12.802	UnloadingDock10	152.OnEnteredInputBuffer	[End]	Process 'UnloadingDock10.OnEnteredInputBuffer' execution ended.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Decide] IfMakeCopies	Token branching on condition 'SeparationMode==List.SeparationModeValue.MakeCopies'.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Decide] IfMakeCopies	Token sent to 'False' exit.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Decide] IfCreateNew	Token branching on condition 'SeparationMode==List.SeparationModeValue.CreateNew'.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Decide] IfCreateNew	Token sent to 'False' exit.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[UnBatch] DesiredSplitQuantity	Searching for '3' batch member(s) to remove from parent entity 'Truck_RT12.802'.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[UnBatch] DesiredSplitQuantity	Removing entity 'Forward.658' from batch members of parent entity 'Truck_RT12.802'.
0.206450313	Forward.658	UnloadingDock10	171.OnEnteredProcessing	[Assign] ResourceState	Assigning state variable 'UnloadingDock10.ResourceState' the value '2'. Previous value was '1'.
0.206450313	Forward.658	UnloadingDock10	171.OnEnteredProcessing	[Transfer] ToMemberOutputBuffer	Entity 'Forward.658' transferring from '[FreeSpace] UnloadingDock10' into node 'MemberOutput@UnloadingDock10'.
0.206450313	Forward.658	MemberOutput@UnloadingDock10	152.OnEnteredFromAssociatedObject	[Begin]	Process 'MemberOutput@UnloadingDock10.OnEnteredFromAssociatedObject' execution started.
0.206450313	Forward.658	MemberOutput@UnloadingDock10	152.OnEnteredFromAssociatedObject	[Fire] EnteredEvent	Firing event 'MemberOutput@UnloadingDock10.Entered'.
0.206450313	Forward.658	MemberOutput@UnloadingDock10	152.OnEnteredFromAssociatedObject	[Execute] EnteredAddOnProcess	Executing process 'RoutingToStagingArea'.
0.206450313	Forward.658	Cross_Dock_EPTs	154.RoutingToStagingArea	[Begin]	Process 'RoutingToStagingArea' execution started.
0.206450313	Forward.658	Cross_Dock_EPTs	154.RoutingToStagingArea	[Decide] Is entity unit load?	Token branching on condition 'Entity.EntityType.Is.Forward Entity.EntityType.Is.Return'.
0.206450313	Forward.658	Cross_Dock_EPTs	154.RoutingToStagingArea	[Decide] Is entity unit load?	Token sent to 'True' exit.
0.206450313	Forward.658	Cross_Dock_EPTs	154.RoutingToStagingArea	[Decide] Not staged?	Token branching on condition 'Unit_Load.StagedAfterUnloading == False'.
0.206450313	Forward.658	Cross_Dock_EPTs	154.RoutingToStagingArea	[Decide] Not staged?	Token sent to 'True' exit.
0.206450313	Forward.658	Cross_Dock_EPTs	154.RoutingToStagingArea	[Decide] UnitLoad in Single-Stage?	Token branching on condition 'Unit_Load.TwoStageCD == False'.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.206450313	Forward.658	EPT10[1]	3.OnRiderReservationAccepted	[SelectVisit] ToAcceptedDestination	Entity 'EPT10[1]' searching its visit request queue containing '1' accepted visit request(s).
0.206450313	Forward.658	EPT10[1]	3.OnRiderReservationAccepted	[SelectVisit] ToAcceptedDestination	Entity 'EPT10[1]' selected visit destination 'MemberOutput@UnloadingDock10'.
0.206450313	Forward.658	EPT10[1]	3.OnRiderReservationAccepted	[Decide] IfInFreeSpace	Token branching on condition '(CurrentNode==Nothing)&&(CurrentLink==Nothing)&&(CurrentStation==Nothing)'.
0.206450313	Forward.658	EPT10[1]	3.OnRiderReservationAccepted	[Decide] IfInFreeSpace	Token sent to 'False' exit.
0.206450313	Forward.658	EPT10[1]	3.OnRiderReservationAccepted	[Decide] IfUnpark	Token branching on condition 'IsParked &&(OnVisitingNode.TokensInProcess==0)'.
0.206450313	Forward.658	EPT10[1]	3.OnRiderReservationAccepted	[Decide] IfUnpark	Token sent to 'True' exit.
0.206450313	Forward.658	EPT10[1]	3.OnRiderReservationAccepted	[UnPark] Vehicle	Entity 'EPT10[1]' transferring from '[Station] MemberOutput@UnloadingDock10.ParkingStation' into node 'MemberOutput@UnloadingDock10'.
0.206450313	Forward.658	EPT10[1]	25.OnRiderLoading	[Begin]	Process 'EPT10[1].OnRiderLoading' execution started.
0.206450313	Forward.658	EPT10[1]	25.OnRiderLoading	[Assign]	Assigning state variable 'EPT10[1].OnEvaluatingRiderReservation.Enabled' the value '0'. Previous value was '0'.
0.206450313	Forward.658	EPT10[1]	25.OnRiderLoading	[Assign]	Assigning state variable 'EPT10[1].OnEvaluatingRiderReservation.ReturnValue' the value '0'. Previous value was '0'.
0.206450313	Forward.658	EPT10[1]	25.OnRiderLoading	[End]	Process 'EPT10[1].OnRiderLoading' execution ended.
0.206450313	Forward.658	MemberOutput@UnloadingDock10	152.OnEnteredFromAssociatedObject	[Execute] RoutingOutLogic	Token waiting at Execute step for '1' other processes to complete.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[UnBatch] DesiredSplitQuantity	Removing entity 'Return.642' from batch members of parent entity 'Truck_RT12.802'.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[UnBatch] DesiredSplitQuantity	Removing entity 'Return.641' from batch members of parent entity 'Truck_RT12.802'.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[UnBatch] DesiredSplitQuantity	Removed '3' batch member(s) from parent entity 'Truck_RT12.802'.
0.206450313	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[UnBatch] DesiredSplitQuantity	Token waiting at UnBatch step for '3' other processes to complete.
0.218396288	Forward.658	EPT10[1]	15.OnRiderLoaded	[Begin]	Process 'EPT10[1].OnRiderLoaded' execution started.
0.218396288	Forward.658	EPT10[1]	15.OnRiderLoaded	[EndTransfer] OntoVehicle	Entity 'Forward.658' ending transfer into station 'EPT10[1].RideStation'.
0.218396288	Forward.658	MemberOutput@UnloadingDock10	151.OnExited	[Begin]	Process 'MemberOutput@UnloadingDock10.OnExited' execution started.
0.218396288	Forward.658	MemberOutput@UnloadingDock10	151.OnExited	[Fire] ExitedEvent	Firing event 'MemberOutput@UnloadingDock10.Exited'.
0.218396288	Forward.658	MemberOutput@UnloadingDock10	151.OnExited	[End]	Process 'MemberOutput@UnloadingDock10.OnExited' execution ended.
0.218396288	Forward.658	EPT10[1]	15.OnRiderLoaded	[End]	Process 'EPT10[1].OnRiderLoaded' execution ended.
0.218396288	Forward.658	UnloadingDock10	171.OnEnteredProcessing	[End]	Process 'UnloadingDock10.OnEnteredProcessing' execution ended.
0.218396288	Forward.658	MemberOutput@UnloadingDock10	154.RoutingOutLogic	[Decide] IfReserveTransporter	Token branching on condition ' '.
0.218396288	Forward.658	MemberOutput@UnloadingDock10	154.RoutingOutLogic	[Decide] IfReserveTransporter	Token sent to 'False' exit.
0.218396288	Forward.658	MemberOutput@UnloadingDock10	154.RoutingOutLogic	[End]	Process 'MemberOutput@UnloadingDock10.RoutingOutLogic' execution ended.
0.218396288	Forward.658	MemberOutput@UnloadingDock10	152.OnEnteredFromAssociatedObject	[End]	Process 'MemberOutput@UnloadingDock10.OnEnteredFromAssociatedObject' execution ended.
0.218396288	Forward.658	EPT10[1]	3.OnRiderReservationAccepted	[End]	Process 'EPT10[1].OnRiderReservationAccepted' execution ended.
0.228059118	Forward.658	EPT10[1]	154.OnRiderUnloading	[Begin]	Process 'EPT10[1].OnRiderUnloading' execution started.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.228059118	Forward.658	EPT10[1]	154.OnRiderUnloading	[Decide] IfVehicleKeptReserved	Token branching on condition 'Capacity.ReservedTo(Entity)>0 && RideStation.Contents==1'.
0.228059118	Forward.658	EPT10[1]	154.OnRiderUnloading	[Decide] IfVehicleKeptReserved	Token sent to 'False' exit.
0.228059118	Forward.658	EPT10[1]	154.OnRiderUnloading	[End]	Process 'EPT10[1].OnRiderUnloading' execution ended.
0.235025174	Forward.658	LoadingDock6	149.OnEnteredMemberInputBuffer	[Begin]	Process 'LoadingDock6.OnEnteredMemberInputBuffer' execution started.
0.235025174	Forward.658	LoadingDock6	149.OnEnteredMemberInputBuffer	[Assign] OnMemberEnteringAssignments	Assigning state variable 'Cross_Dock_EPTs.NumUnitLoadsInSystem' the value '66'. Previous value was '67'.
0.235025174	Forward.658	LoadingDock6	149.OnEnteredMemberInputBuffer	[EndTransfer] IntoMemberInputBuffer	Entity 'Forward.658' ending transfer into station 'LoadingDock6.MemberInputBuffer'.
0.235025174	Forward.658	LoadingDock6	149.OnEnteredMemberInputBuffer	[Batch] Member	Entity 'Forward.658' waiting to be batched by batch logic 'LoadingDock6.Batching'.
0.285334749	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Assign] BeforeParentExitingAssignments	Assigning state variable 'Cross_Dock_EPTs.NumInboundTrucksToBeServed' the value '10'. Previous value was '11'.
0.285334749	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Assign] BeforeParentExitingAssignments	Assigning state variable 'Cross_Dock_EPTs.NumOutboundTrucksToBeServed' the value '4'. Previous value was '3'.
0.285334749	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[Transfer] ToParentOutputBuffer	Entity 'Truck_RT12.802' transferring from '[Station] UnloadingDock10.Processing' into node 'ParentOutput@UnloadingDock10'.
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	138.OnEnteredFromAssociatedObject	[Begin]	Process 'ParentOutput@UnloadingDock10.OnEnteredFromAssociatedObject' execution started.
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	138.OnEnteredFromAssociatedObject	[Assign] OnEnteringAssignments	Assigning state variable 'Cross_Dock_EPTs.TruckNumberToBeServed' the value '4'. Previous value was '3'.
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	138.OnEnteredFromAssociatedObject	[Fire] EnteredEvent	Firing event 'ParentOutput@UnloadingDock10.Entered'.
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	138.OnEnteredFromAssociatedObject	[Execute] EnteredAddOnProcess	Executing process 'AssignReloadingDock'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	14.AssignReloadingDock	[Begin]	Process 'AssignReloadingDock' execution started.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	14.AssignReloadingDock	[Search] SearchAssignedReloadingDock	Searching collection '[TableRows] TruckToReloadingDockAssignment' containing '12' item(s).
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	14.AssignReloadingDock	[Search] SearchAssignedReloadingDock	Searching collection in the forward direction for maximum of '1' item(s).
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	14.AssignReloadingDock	[Search] SearchAssignedReloadingDock	Index range of search is from starting index '1' to ending index '12'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	14.AssignReloadingDock	[Search] SearchAssignedReloadingDock	Item 'TruckToReloadingDockAssignment[10]' found at index '10' in the collection.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	14.AssignReloadingDock	[Search] SearchAssignedReloadingDock	'1' item(s) found by search.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	14.AssignReloadingDock	[Search] SearchAssignedReloadingDock	Created token 'Token.146' associated with item 'TruckToReloadingDockAssignment[10]' found at index '10' in the collection.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	146.AssignReloadingDock	[SetNode] SetNextNode	Destination node of entity 'Truck_RT12.802' set to value 'ParentInput@LoadingDock10'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	146.AssignReloadingDock	[End]	Process 'AssignReloadingDock' execution ended.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	146.AssignReloadingDock	[End]	Process 'AssignReloadingDock' execution ended.
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	138.OnEnteredFromAssociatedObject	[Transfer] ToOutboundLink	Entity 'Truck_RT12.802' transferring from '[Node] ParentOutput@UnloadingDock10' onto link 'InstantTimePath1'.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.285334749	Truck_RT12.802	InstantTimePath1	14.OnEntered	[Begin]	Process 'InstantTimePath1.OnEntered' execution started.
0.285334749	Truck_RT12.802	InstantTimePath1	14.OnEntered	[Assign] EntityMovementRate	Assigning state variable 'InstantTimePath1.Token.14.ReturnValue' the value '0'. Previous value was '1'.
0.285334749	Truck_RT12.802	InstantTimePath1	14.OnEntered	[Assign] EntityMovementRate	Assigning state variable 'Truck_RT12.802.Movement.Rate' the value 'â' Meters per Hour. Previous value was 'â' Meters per Hour.
0.285334749	Truck_RT12.802	InstantTimePath1	14.OnEntered	[EndTransfer] OntoTimePath	Entity 'Truck_RT12.802' ending transfer onto link 'InstantTimePath1'.
0.285334749	Truck_RT12.802	InstantTimePath1	14.OnEntered	[End]	Process 'InstantTimePath1.OnEntered' execution ended.
0.285334749	Truck_RT12.802	UnloadingDock10	5.OnExitedProcessing	[Begin]	Process 'UnloadingDock10.OnExitedProcessing' execution started.
0.285334749	Truck_RT12.802	UnloadingDock10	5.OnExitedProcessing	[Decide] IfTokenAssociatedObject	Token branching on condition 'Token.AssociatedObject != Nothing'.
0.285334749	Truck_RT12.802	UnloadingDock10	5.OnExitedProcessing	[Decide] IfTokenAssociatedObject	Token sent to 'True' exit.
0.285334749	Truck_RT12.802	UnloadingDock10	5.OnExitedProcessing	[Release] Separator	Object 'Truck_RT12.802' attempting to release '1' resource(s) of resource type '[ParentObject] UnloadingDock10'.
0.285334749	Truck_RT12.802	UnloadingDock10	5.OnExitedProcessing	[Release] Separator	Object 'Truck_RT12.802' released '1' capacity unit(s) of resource 'UnloadingDock10'.
0.285334749	Truck_RT12.802	UnloadingDock10	19.OnCapacityReleased	[Begin]	Process 'UnloadingDock10.OnCapacityReleased' execution started.
0.285334749	Truck_RT12.802	UnloadingDock10	19.OnCapacityReleased	[End]	Process 'UnloadingDock10.OnCapacityReleased' execution ended.
0.285334749	Truck_RT12.802	UnloadingDock10	5.OnExitedProcessing	[Execute] ParentExitedAddOnProcess	Executing process 'TrackLastUnloading'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	19.TrackLastUnloading	[Begin]	Process 'TrackLastUnloading' execution started.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	19.TrackLastUnloading	[Decide] First last truck unloading?	Token branching on condition 'NumInboundTrucksToBeServed == 0'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	19.TrackLastUnloading	[Decide] First last truck unloading?	Token sent to 'False' exit.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	19.TrackLastUnloading	[End]	Process 'TrackLastUnloading' execution ended.
0.285334749	Truck_RT12.802	UnloadingDock10	5.OnExitedProcessing	[End]	Process 'UnloadingDock10.OnExitedProcessing' execution ended.
0.285334749	Truck_RT12.802	UnloadingDock10	180.OnEnteredProcessing	[End]	Process 'UnloadingDock10.OnEnteredProcessing' execution ended.
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	138.OnEnteredFromAssociatedObject	[End]	Process 'ParentOutput@UnloadingDock10.OnEnteredFromAssociatedObject' execution ended.
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	139.OnExited	[Begin]	Process 'ParentOutput@UnloadingDock10.OnExited' execution started.
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	139.OnExited	[Fire] ExitedEvent	Firing event 'ParentOutput@UnloadingDock10.Exited'.
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	139.OnExited	[Execute] ExitedAddOnProcess	Executing process 'EvaluateEndOfUnloading'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	140.EvaluateEndOfUnloading	[Begin]	Process 'EvaluateEndOfUnloading' execution started.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	140.EvaluateEndOfUnloading	[Assign] Mark Start Waiting Reloading	Assigning state variable 'Truck_RT12.802.StartWaitingTimeReloading' the value '0.285334748568294'. Previous value was '0'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	140.EvaluateEndOfUnloading	[Decide] Num Trucks in Sys == Num Outbound Trucks?	Token branching on condition 'NumInboundTrucksToBeServed == 0'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	140.EvaluateEndOfUnloading	[Decide] Num Trucks in Sys == Num Outbound Trucks?	Token sent to 'False' exit.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	140.EvaluateEndOfUnloading	[End]	Process 'EvaluateEndOfUnloading' execution ended.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
0.285334749	Truck_RT12.802	ParentOutput@UnloadingDock10	139.OnExited	[End]	Process 'ParentOutput@UnloadingDock10.OnExited' execution ended.
0.285334749	Truck_RT12.802	InstantTimePath1	139.OnReachedEnd	[Begin]	Process 'InstantTimePath1.OnReachedEnd' execution started.
0.285334749	Truck_RT12.802	InstantTimePath1	139.OnReachedEnd	[Transfer] FromTimePath	Entity 'Truck_RT12.802' transferring from '[EndOfLink] InstantTimePath1' into node 'Parking'.
0.285334749	Truck_RT12.802	Parking	140.OnEntered	[Begin]	Process 'Parking.OnEntered' execution started.
0.285334749	Truck_RT12.802	Parking	140.OnEntered	[Fire] EnteredEvent	Firing event 'Parking.Entered'.
0.285334749	Truck_RT12.802	Parking	140.OnEntered	[Execute] EnteredAddOnProcess	Executing process 'WaitForAllUnloaded'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	138.WaitForAllUnloaded	[Begin]	Process 'WaitForAllUnloaded' execution started.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	138.WaitForAllUnloaded	[Decide] All Unloaded?	Token branching on condition 'AllTrucksUnloaded == 1'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	138.WaitForAllUnloaded	[Decide] All Unloaded?	Token sent to 'False' exit.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	138.WaitForAllUnloaded	[Wait] Wait for Unloading Completed	Token waiting at Wait step for event 'StartOutbound'.
0.285334749	Truck_RT12.802	Cross_Dock_EPTs	138.WaitForAllUnloaded	[Wait] Wait for Unloading Completed	Token waiting at Wait step for any one of '1' specified event(s) to occur.
0.285334749	Truck_RT12.802	Parking	140.OnEntered	[Execute] EnteredAddOnProcess	Token waiting at Execute step for '1' other processes to complete.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	138.WaitForAllUnloaded	[Wait] Wait for Unloading Completed	Wait for event(s) ended. Releasing token from Wait step.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	138.WaitForAllUnloaded	[Tally] Tally Waiting Time before Reloading	Recording 'Ä01,Ä0 observation value(s) of type 'Ä0Expression,Ä0 for tally statistic 'Ä0WaitingTimeBeforeReloading,Ä0.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	138.WaitForAllUnloaded	[Tally] Tally Waiting Time before Reloading	Recording observation value '1.88523238123607' Hours for tally statistic 'WaitingTimeBeforeReloading'.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	138.WaitForAllUnloaded	[End]	Process 'WaitForAllUnloaded' execution ended.
1.88998796	Truck_RT12.802	Parking	140.OnEntered	[Transfer] ToOutboundLink	Assigning state variable 'Truck_RT12.802.CurrentTravelMode' the value '2'.
1.88998796	Truck_RT12.802	Parking	140.OnEntered	[Transfer] ToOutboundLink	Entity 'Truck_RT12.802' could not find a followable travel path from starting location '[Node] Parking' to destination location 'ParentInput@LoadingDock10' using assigned network 'Global'.
1.88998796	Truck_RT12.802	Parking	140.OnEntered	[Transfer] ToOutboundLink	Entity 'Truck_RT12.802' failed to initiate transfer from '[Node] Parking' to an outbound link.
1.88998796	Truck_RT12.802	Parking	140.OnEntered	[Transfer] ToFreeSpace	Entity 'Truck_RT12.802' initiating transfer from '[Node] Parking' into free space of facility 'Cross_Dock_EPTs'.
1.88998796	Truck_RT12.802	Parking	214.OnExited	[Begin]	Process 'Parking.OnExited' execution started.
1.88998796	Truck_RT12.802	Parking	214.OnExited	[Fire] ExitedEvent	Firing event 'Parking.Exited'.
1.88998796	Truck_RT12.802	Parking	214.OnExited	[Execute] ExitedAddOnProcess	Executing process 'MarkStartOutbounding'.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	181.MarkStartOutbounding	[Begin]	Process 'MarkStartOutbounding' execution started.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	181.MarkStartOutbounding	[Decide] All Unloaded?	Token branching on condition 'AllTrucksUnloaded == 1'.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	181.MarkStartOutbounding	[Decide] All Unloaded?	Token sent to 'True' exit.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	181.MarkStartOutbounding	[End]	Process 'MarkStartOutbounding' execution ended.
1.88998796	Truck_RT12.802	Parking	214.OnExited	[End]	Process 'Parking.OnExited' execution ended.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Begin]	Process 'Truck_RT12.802.OnEnteredFreeSpace' execution started.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Assign] OnEnteredFreeSpaceAssignments	Assigning state variable 'Truck_RT12.802.FreeSpaceSteeringToken.175.EnteredFreeSpaceFromNode' the value '1'. Previous value was '0'.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Assign] OnEnteredFreeSpaceAssignments	Assigning state variable 'Truck_RT12.802.FreeSpaceSteeringToken.175.DestinationNode' the value 'ParentInput@LoadingDock10'. Previous value was 'Nothing'.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Decide] IfDestinationNode	Token branching on condition 'FreeSpaceSteeringToken.DestinationNode != Nothing'.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Decide] IfDestinationNode	Token sent to 'True' exit.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[SetNode] NextDestination	Destination node of entity 'Truck_RT12.802' set to value 'ParentInput@LoadingDock10'.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Decide] IfDirectToDestination	Token branching on condition 'Math.If(FreeSpaceSteeringBehavior == List.FreeSpaceSteeringBehaviorType.DirectToDestination CurrentNetwork == Nothing !FreeSpaceSteeringToken.EnteredFreeSpaceFromNode, True, !CurrentNetwork.PathExists(PreviousNode, DestinationNode))'.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Decide] IfDirectToDestination	Token sent to 'True' exit.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Travel] DirectToDestination	Entity 'Truck_RT12.802' starting free space travel movement to coordinates 'X=19 Y=0 Z=-1' Meters.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Travel] DirectToDestination	Entity 'Truck_RT12.802' finished free space travel movement to coordinates 'X=19 Y=0 Z=-1' Meters.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Travel] DirectToDestination	Setting movement rate of entity 'Truck_RT12.802' to desired ending value '0' Meters per Hour.
1.88998796	Truck_RT12.802	InstantTimePath1	139.OnReachedEnd	[End]	Process 'InstantTimePath1.OnReachedEnd' execution ended.
1.88998796	Truck_RT12.802	Parking	140.OnEntered	[End]	Process 'Parking.OnEntered' execution ended.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[Transfer] ToDestinationNode	Entity 'Truck_RT12.802' transferring from '[FreeSpace] Cross_Dock_EPTs' into node 'ParentInput@LoadingDock10'.
1.88998796	Truck_RT12.802	ParentInput@LoadingDock10	140.OnEnteredToAssociatedObject	[Begin]	Process 'ParentInput@LoadingDock10.OnEnteredToAssociatedObject' execution started.
1.88998796	Truck_RT12.802	ParentInput@LoadingDock10	140.OnEnteredToAssociatedObject	[Fire] EnteredEvent	Firing event 'ParentInput@LoadingDock10.Entered'.
1.88998796	Truck_RT12.802	ParentInput@LoadingDock10	140.OnEnteredToAssociatedObject	[Execute] EnteredAddOnProcess	Executing process 'Mark_Arrival_ReloadingDock'.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	139.Mark_Arrival_ReloadingDock	[Begin]	Process 'Mark_Arrival_ReloadingDock' execution started.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	139.Mark_Arrival_ReloadingDock	[Assign] Mark Reloading Time	Assigning state variable 'Truck_RT12.802.ReloadingTime' the value '1.88998796037887'. Previous value was '0'.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	139.Mark_Arrival_ReloadingDock	[Assign] Mark Reloading Time	Assigning state variable 'Cross_Dock_EPTs.FirstTruckReloaded' the value '1.88998796037887' Hours. Previous value was '1.88998796037887' Hours.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	139.Mark_Arrival_ReloadingDock	[Tally] Time before reloading	Recording 'Å01,Å0 observation value(s) of type 'Å0Expression,Å0 for tally statistic 'Å0WaitingTimeBeforeReloading,Å0.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	139.Mark_Arrival_ReloadingDock	[Tally] Time before reloading	Recording observation value '0' Hours for tally statistic 'WaitingTimeBeforeReloading'.
1.88998796	Truck_RT12.802	Cross_Dock_EPTs	139.Mark_Arrival_ReloadingDock	[End]	Process 'Mark_Arrival_ReloadingDock' execution ended.
1.88998796	Truck_RT12.802	ParentInput@LoadingDock10	140.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Truck_RT12.802' transferring from '[Node] ParentInput@LoadingDock10' into object 'LoadingDock10'.
1.88998796	Truck_RT12.802	ParentInput@LoadingDock10	140.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Truck_RT12.802' transferring from '[Node] ParentInput@LoadingDock10' into station 'LoadingDock10.ParentInputBuffer'.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Begin]	Process 'LoadingDock10.OnEnteredParentInputBuffer' execution started.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[EndTransfer] IntoParentInputBuffer	Entity 'Truck_RT12.802' ending transfer into station 'LoadingDock10.ParentInputBuffer'.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
1.88998796	Truck_RT12.802	ParentInput@LoadingDock10	19.OnExited	[Begin]	Process 'ParentInput@LoadingDock10.OnExited' execution started.
1.88998796	Truck_RT12.802	ParentInput@LoadingDock10	19.OnExited	[Fire] ExitedEvent	Firing event 'ParentInput@LoadingDock10.Exited'.
1.88998796	Truck_RT12.802	ParentInput@LoadingDock10	19.OnExited	[End]	Process 'ParentInput@LoadingDock10.OnExited' execution ended.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄðTruck_RT12.802,Äð waiting to collect ,Äð33,Äð batch member(s) from queue state ,ÄðLoadingDock10.Batching.MemberQueue,Äð using matching rule ,Äð[MatchMembers] Unit_Load.EndDestination [AndParent] Truck.EndDestination [Matching Value = Input@Sink_RT12],Äð.
1.88998796	Truck_RT12.802	Truck_RT12.802	175.OnEnteredFreeSpace	[End]	Process 'Truck_RT12.802.OnEnteredFreeSpace' execution ended.
1.88998796	Truck_RT12.802	ParentInput@LoadingDock10	140.OnEnteredToAssociatedObject	[End]	Process 'ParentInput@LoadingDock10.OnEnteredToAssociatedObject' execution ended.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	Release batch early trigger fired for entity ,ÄðTruck_RT12.802,Äð waiting to collect batch member(s) from queue state ,ÄðLoadingDock10.Batching.MemberQueue,Äð. Wait duration '0' Hours expired.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄðTruck_RT12.802,Äð found ,Äð18,Äð currently available batch member(s) waiting in queue state ,ÄðLoadingDock10.Batching.MemberQueue,Äð using matching rule ,Äð[MatchMembers] Unit_Load.EndDestination [AndParent] Truck.EndDestination [Matching Value = Input@Sink_RT12],Äð.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	Evaluating release batch early condition for entity ,ÄðTruck_RT12.802,Äð waiting to collect batch member(s) from queue state ,ÄðLoadingDock10.Batching.MemberQueue,Äð.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	Release batch early condition is True for entity ,ÄðTruck_RT12.802,Äð waiting to collect batch member(s) from queue state ,ÄðLoadingDock10.Batching.MemberQueue,Äð.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄðTruck_RT12.802,Äð collecting batch member ,ÄðForward.662,Äð from queue state ,ÄðLoadingDock10.Batching.MemberQueue,Äð using matching rule '[MatchMembers] Unit_Load.EndDestination [AndParent] Truck.EndDestination [Matching Value = Input@Sink_RT12]'.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄðTruck_RT12.802,Äð collecting batch member ,ÄðForward.663,Äð from queue state ,ÄðLoadingDock10.Batching.MemberQueue,Äð using matching rule '[MatchMembers] Unit_Load.EndDestination [AndParent] Truck.EndDestination [Matching Value = Input@Sink_RT12]'.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	Entity ,ÄTruck_RT12.802,Ä collecting batch member ,ÄForward.762,Ä from queue state ,ÄLoadingDock10.Batching.MemberQueue,Ä using matching rule '[MatchMembers] Unit_Load.EndDestination [AndParent] Truck.EndDestination [Matching Value = Input@Sink_RT12]'. Entity ,ÄTruck_RT12.802,Ä collecting batch member ,ÄForward.756,Ä from queue state ,ÄLoadingDock10.Batching.MemberQueue,Ä using matching rule '[MatchMembers] Unit_Load.EndDestination [AndParent] Truck.EndDestination [Matching Value = Input@Sink_RT12]'. Entity ,ÄTruck_RT12.802,Ä cancelling request to collect any further batch members from queue state ,ÄLoadingDock10.Batching.MemberQueue,Ä. Object 'Truck_RT12.802' waiting to seize '1' resource(s) of resource type '[ParentObject] LoadingDock10'. Process 'LoadingDock6.OnEnteredMemberInputBuffer' execution ended. Object 'Truck_RT12.802' checking availability to seize '1' resource(s) of resource type '[ParentObject] LoadingDock10'. Object 'Truck_RT12.802' checking availability to seize '1' capacity unit(s) of resource 'LoadingDock10,Ä. Object 'Truck_RT12.802' found ,Ä1,Ä currently available capacity unit(s) of resource 'LoadingDock10,Ä. Object 'Truck_RT12.802' found '1' currently available resource(s) of resource type '[ParentObject] LoadingDock10'. Object 'Truck_RT12.802' seized '1' capacity unit(s) of resource 'LoadingDock10'. Object 'Truck_RT12.802' completed seize of '1' resource(s) of resource type '[ParentObject] LoadingDock10'. Object 'Truck_RT12.802' waiting to seize '1' resource(s) of resource type '[Specific] EPT10'. Object 'Truck_RT12.802' checking availability to seize '1' resource(s) of resource type '[Specific] EPT10'. Object 'Truck_RT12.802' checking availability to seize '1' capacity unit(s) of resource 'EPT10[1],Ä. Object 'Truck_RT12.802' found ,Ä1,Ä currently available capacity unit(s) of resource 'EPT10[1],Ä. Object 'Truck_RT12.802' found '1' currently available resource(s) of resource type '[Specific] EPT10'. Object 'Truck_RT12.802' seized '1' capacity unit(s) of resource 'EPT10[1]'. Process 'EPT10[1].OnCapacityAllocated' execution started. Assigning state variable 'EPT10[1].ResourceState' the value '1'. Previous value was '7'.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Batch] Parent	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] Combiner	
1.88998796	Forward.658	LoadingDock6	149.OnEnteredMemberInputBuffer	[End]	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] Combiner	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] Combiner	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] Combiner	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] Combiner	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] Combiner	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] Combiner	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] SecondaryResources	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] SecondaryResources	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] SecondaryResources	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] SecondaryResources	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] SecondaryResources	
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] SecondaryResources	
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[Begin]	
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[Assign] ResourceState	

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[Assign] OnCapacityAllocatedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingRiderReservation.Enabled' the value '0'. Previous value was '1'.
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[Assign] OnCapacityAllocatedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingRiderReservation.ReturnValue' the value '0'. Previous value was '1'.
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[Assign] OnCapacityAllocatedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingSeizeRequest.Enabled' the value '0'. Previous value was '1'.
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[Assign] OnCapacityAllocatedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingSeizeRequest.ReturnValue' the value '0'. Previous value was '1'.
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[Assign] OnCapacityAllocatedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingRiderAtPickup.Enabled' the value '0'. Previous value was '1'.
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[Assign] OnCapacityAllocatedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingRiderAtPickup.ReturnValue' the value '0'. Previous value was '1'.
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[Fire] AllocatedEvent	Firing event 'EPT10[1].Allocated'.
1.88998796	Truck_RT12.802	EPT10[1]	169.OnCapacityAllocated	[End]	Process 'EPT10[1].OnCapacityAllocated' execution ended.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Seize] SecondaryResources	Object 'Truck_RT12.802' completed seize of '1' resource(s) of resource type '[Specific] EPT10'.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[Transfer] ToProcessing	Entity 'Truck_RT12.802' transferring from '[Station] LoadingDock10.ParentInputBuffer' into station 'LoadingDock10.Processing'.
1.88998796	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Begin]	Process 'LoadingDock10.OnEnteredProcessing' execution started.
1.88998796	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[EndTransfer] IntoProcessing	Entity 'Truck_RT12.802' ending transfer into station 'LoadingDock10.Processing'.
1.88998796	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Decide] IfProcessTypeSpecificTime	Token branching on condition 'ProcessType==Enum.TaskProcessType.SpecificTime'.
1.88998796	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Decide] IfProcessTypeSpecificTime	Token sent to 'True' exit.
1.88998796	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Delay] ProcessingTime	Delaying token for '0.268098630148885' Hours until time '2.15808659052776' Hours. Delay multiplier was '1'.
1.88998796	Truck_RT12.802	LoadingDock10	139.OnEnteredParentInputBuffer	[End]	Process 'LoadingDock10.OnEnteredParentInputBuffer' execution ended.
2.158086591	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Execute] AfterProcessingAddOnProcess	Executing process 'TrackLastReloading'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	19.TrackLastReloading	[Begin]	Process 'TrackLastReloading' execution started.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	19.TrackLastReloading	[Decide] Last truck reloading?	Token branching on condition 'NumTrucksInSystem == 1'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	19.TrackLastReloading	[Decide] Last truck reloading?	Token sent to 'False' exit.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	19.TrackLastReloading	[End]	Process 'TrackLastReloading' execution ended.
2.158086591	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Release] SecondaryResources	Object 'Truck_RT12.802' attempting to release '1' resource(s) of resource type '[Specific] EPT10'.
2.158086591	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Release] SecondaryResources	Object 'Truck_RT12.802' released '1' capacity unit(s) of resource 'EPT10[1]'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Begin]	Process 'EPT10[1].OnCapacityReleased' execution started.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Assign] ResourceState	Assigning state variable 'EPT10[1].ResourceState' the value '0'. Previous value was '0'.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Assign] OnCapacityReleasedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingRiderReservation.Enabled' the value '1'. Previous value was '0'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Assign] OnCapacityReleasedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingRiderReservation.ReturnValue' the value '1'. Previous value was '0'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Assign] OnCapacityReleasedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingSeizeRequest.Enabled' the value '1'. Previous value was '0'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Assign] OnCapacityReleasedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingSeizeRequest.ReturnValue' the value '1'. Previous value was '0'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Assign] OnCapacityReleasedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingRiderAtPickup.Enabled' the value '1'. Previous value was '0'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Assign] OnCapacityReleasedAssignments	Assigning state variable 'EPT10[1].OnEvaluatingRiderAtPickup.ReturnValue' the value '1'. Previous value was '0'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[SetNode] ToClearDestination	Destination node of entity 'EPT10[1]' set to value 'No Destination'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Fire] RemainInPlaceEndedEvent	Firing event 'EPT10[1].RemainInPlaceEnded'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[Fire] ReleasedEvent	Firing event 'EPT10[1].Released'.
2.158086591	Truck_RT12.802	EPT10[1]	19.OnCapacityReleased	[End]	Process 'EPT10[1].OnCapacityReleased' execution ended.
2.158086591	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Assign] BeforeExitingAssignments	Assigning state variable 'Cross_Dock_EPTs.NumOutboundTrucksToBeServed' the value '8'. Previous value was '9'.
2.158086591	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Assign] BeforeExitingAssignments	Assigning state variable 'Cross_Dock_EPTs.NumTrucksInSystem' the value '8'. Previous value was '9'.
2.158086591	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[Transfer] ToOutputBuffer	Entity 'Truck_RT12.802' transferring from '[Station] LoadingDock10.Processing' into node 'Output@LoadingDock10'.
2.158086591	Truck_RT12.802	Output@LoadingDock10	19.OnEnteredFromAssociatedObject	[Begin]	Process 'Output@LoadingDock10.OnEnteredFromAssociatedObject' execution started.
2.158086591	Truck_RT12.802	Output@LoadingDock10	19.OnEnteredFromAssociatedObject	[Fire] EnteredEvent	Firing event 'Output@LoadingDock10.Entered'.
2.158086591	Truck_RT12.802	Output@LoadingDock10	19.OnEnteredFromAssociatedObject	[Transfer] ToOutboundLink	Entity 'Truck_RT12.802' transferring from '[Node] Output@LoadingDock10' into node 'BasicNode2'.
2.158086591	Truck_RT12.802	BasicNode2	248.OnEntered	[Begin]	Process 'BasicNode2.OnEntered' execution started.
2.158086591	Truck_RT12.802	BasicNode2	248.OnEntered	[Tally] OnEnteringTallies	Recording 'Å01,Å0 observation value(s) of type 'Å0TimeBetween,Å0 for tally statistic 'Å0Truck_RT12.802.IndividualMakespan,Å0.
2.158086591	Truck_RT12.802	BasicNode2	248.OnEntered	[Tally] OnEnteringTallies	Recording observation value '2.06137625980962' Hours for tally statistic 'Truck_RT12.802.IndividualMakespan'.
2.158086591	Truck_RT12.802	BasicNode2	248.OnEntered	[Fire] EnteredEvent	Firing event 'BasicNode2.Entered'.
2.158086591	Truck_RT12.802	BasicNode2	248.OnEntered	[Execute] EnteredAddOnProcess	Executing process 'RoutingToSink'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	142.RoutingToSink	[Begin]	Process 'RoutingToSink' execution started.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	142.RoutingToSink	[SetNode] SetNextNode	Destination node of entity 'Truck_RT12.802' set to value 'Input@Sink_RT12'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	142.RoutingToSink	[End]	Process 'RoutingToSink' execution ended.

Table B.13 continued from previous page

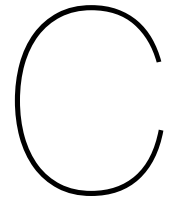
Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
2.158086591	Truck_RT12.802	BasicNode2	248.OnEntered	[Transfer] ToOutboundLink	Entity 'Truck_RT12.802' could not find a followable travel path from starting location '[Node] BasicNode2' to destination location 'Input@Sink_RT12' using assigned network 'Global'.
2.158086591	Truck_RT12.802	BasicNode2	248.OnEntered	[Transfer] ToOutboundLink	Entity 'Truck_RT12.802' failed to initiate transfer from '[Node] BasicNode2' to an outbound link.
2.158086591	Truck_RT12.802	BasicNode2	248.OnEntered	[Transfer] ToFreeSpace	Entity 'Truck_RT12.802' initiating transfer from '[Node] BasicNode2' into free space of facility 'Cross_Dock_EPTs'.
2.158086591	Truck_RT12.802	Output@LoadingDock10	142.OnExited	[Begin]	Process 'Output@LoadingDock10.OnExited' execution started.
2.158086591	Truck_RT12.802	Output@LoadingDock10	142.OnExited	[Tally] OnExitedTallies	Recording 'Å01,Å0 observation value(s) of type 'Å0TimeBetween,Å0 for tally statistic 'Å0Truck_RT12.802.IndividualMakespan,Å0.
2.158086591	Truck_RT12.802	Output@LoadingDock10	142.OnExited	[Tally] OnExitedTallies	Recording observation value '0' Hours for tally statistic 'Truck_RT12.802.IndividualMakespan'.
2.158086591	Truck_RT12.802	Output@LoadingDock10	142.OnExited	[Fire] ExitedEvent	Firing event 'Output@LoadingDock10.Exited'.
2.158086591	Truck_RT12.802	Output@LoadingDock10	142.OnExited	[Execute] ExitedAddOnProcess	Executing process 'Mark_Departure_CDF'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	9.Mark_Departure_CDF	[Begin]	Process 'Mark_Departure_CDF' execution started.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	9.Mark_Departure_CDF	[Assign] Mark ATD	Assigning state variable 'Truck_RT12.802.DepartureCDF' the value '2.15808659052776'. Previous value was '0'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	9.Mark_Departure_CDF	[Tally] Tally Truck Arrival Time CDF	Recording 'Å01,Å0 observation value(s) of type 'Å0Expression,Å0 for tally statistic 'Å0Makespan,Å0.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	9.Mark_Departure_CDF	[Tally] Tally Truck Arrival Time CDF	Recording observation value '2.06137625980962' Hours for tally statistic 'Makespan'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	9.Mark_Departure_CDF	[End]	Process 'Mark_Departure_CDF' execution ended.
2.158086591	Truck_RT12.802	Output@LoadingDock10	142.OnExited	[End]	Process 'Output@LoadingDock10.OnExited' execution ended.
2.158086591	Truck_RT12.802	BasicNode2	142.OnExited	[Begin]	Process 'BasicNode2.OnExited' execution started.
2.158086591	Truck_RT12.802	BasicNode2	142.OnExited	[Fire] ExitedEvent	Firing event 'BasicNode2.Exited'.
2.158086591	Truck_RT12.802	BasicNode2	142.OnExited	[End]	Process 'BasicNode2.OnExited' execution ended.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Begin]	Process 'Truck_RT12.802.OnEnteredFreeSpace' execution started.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Assign] OnEnteredFreeSpaceAssignments	Assigning state variable 'Truck_RT12.802.FreeSpaceSteeringToken.292.EnteredFreeSpaceFromNode' the value '1'. Previous value was '0'.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Assign] OnEnteredFreeSpaceAssignments	Assigning state variable 'Truck_RT12.802.FreeSpaceSteeringToken.292.DestinationNode' the value 'Input@Sink_RT12'. Previous value was 'Nothing'.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Decide] IfDestinationNode	Token branching on condition 'FreeSpaceSteeringToken.DestinationNode != Nothing'.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Decide] IfDestinationNode	Token sent to 'True' exit.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[SetNode] NextDestination	Destination node of entity 'Truck_RT12.802' set to value 'Input@Sink_RT12'.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Decide] IfDirectToDestination	Token branching on condition 'Math.If(FreeSpaceSteeringBehavior == List.FreeSpaceSteeringBehaviorType.DirectToDestination CurrentNetwork == Nothing !FreeSpaceSteeringToken.EnteredFreeSpaceFromNode, True, !CurrentNetwork.PathExists(PreviousNode, DestinationNode))'.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Decide] IfDirectToDestination	Token sent to 'True' exit.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Travel] DirectToDestination	Entity 'Truck_RT12.802' starting free space travel movement to coordinates 'X=38 Y=0 Z=6' Meters.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Travel] DirectToDestination	Entity 'Truck_RT12.802' finished free space travel movement to coordinates 'X=38 Y=0 Z=6' Meters.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Travel] DirectToDestination	Setting movement rate of entity 'Truck_RT12.802' to desired ending value '0' Meters per Hour.
2.158086591	Truck_RT12.802	LoadingDock10	142.OnExitedProcessing	[Begin]	Process 'LoadingDock10.OnExitedProcessing' execution started.
2.158086591	Truck_RT12.802	LoadingDock10	142.OnExitedProcessing	[Decide] IfTokenAssociatedObject	Token branching on condition 'Token.AssociatedObject != Nothing'.
2.158086591	Truck_RT12.802	LoadingDock10	142.OnExitedProcessing	[Decide] IfTokenAssociatedObject	Token sent to 'True' exit.
2.158086591	Truck_RT12.802	LoadingDock10	142.OnExitedProcessing	[Release] Combiner	Object 'Truck_RT12.802' attempting to release '1' resource(s) of resource type '[ParentObject] LoadingDock10'.
2.158086591	Truck_RT12.802	LoadingDock10	142.OnExitedProcessing	[Release] Combiner	Object 'Truck_RT12.802' released '1' capacity unit(s) of resource 'LoadingDock10'.
2.158086591	Truck_RT12.802	LoadingDock10	142.OnExitedProcessing	[Execute] ExitedAddOnProcess	Executing process 'RoutingToSink'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	9.RoutingToSink	[Begin]	Process 'RoutingToSink' execution started.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	9.RoutingToSink	[SetNode] SetNextNode	Destination node of entity 'Truck_RT12.802' set to value 'Input@Sink_RT12'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	9.RoutingToSink	[End]	Process 'RoutingToSink' execution ended.
2.158086591	Truck_RT12.802	LoadingDock10	142.OnExitedProcessing	[End]	Process 'LoadingDock10.OnExitedProcessing' execution ended.
2.158086591	Truck_RT12.802	LoadingDock10	169.OnEnteredProcessing	[End]	Process 'LoadingDock10.OnEnteredProcessing' execution ended.
2.158086591	Truck_RT12.802	Output@LoadingDock10	19.OnEnteredFromAssociatedObject	[End]	Process 'Output@LoadingDock10.OnEnteredFromAssociatedObject' execution ended.
2.158086591	Truck_RT12.802	BasicNode2	248.OnEntered	[End]	Process 'BasicNode2.OnEntered' execution ended.
2.158086591	Truck_RT12.802	Truck_RT12.802	292.OnEnteredFreeSpace	[Transfer] ToDestinationNode	Entity 'Truck_RT12.802' transferring from '[FreeSpace] Cross_Dock_EPTs' into node 'Input@Sink_RT12'.
2.158086591	Truck_RT12.802	Input@Sink_RT12	248.OnEnteredToAssociatedObject	[Begin]	Process 'Input@Sink_RT12.OnEnteredToAssociatedObject' execution started.
2.158086591	Truck_RT12.802	Input@Sink_RT12	248.OnEnteredToAssociatedObject	[Fire] EnteredEvent	Firing event 'Input@Sink_RT12.Entered'.
2.158086591	Truck_RT12.802	Input@Sink_RT12	248.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Truck_RT12.802' transferring from '[Node] Input@Sink_RT12' into object 'Sink_RT12'.
2.158086591	Truck_RT12.802	Input@Sink_RT12	248.OnEnteredToAssociatedObject	[Transfer] ToAssociatedObject	Entity 'Truck_RT12.802' transferring from '[Node] Input@Sink_RT12' into station 'Sink_RT12.InputBuffer'.
2.158086591	Truck_RT12.802	Sink_RT12	19.OnEnteredInputBuffer	[Begin]	Process 'Sink_RT12.OnEnteredInputBuffer' execution started.
2.158086591	Truck_RT12.802	Sink_RT12	19.OnEnteredInputBuffer	[EndTransfer] IntoInputBuffer	Entity 'Truck_RT12.802' ending transfer into station 'Sink_RT12.InputBuffer'.
2.158086591	Truck_RT12.802	Input@Sink_RT12	169.OnExited	[Begin]	Process 'Input@Sink_RT12.OnExited' execution started.
2.158086591	Truck_RT12.802	Input@Sink_RT12	169.OnExited	[Fire] ExitedEvent	Firing event 'Input@Sink_RT12.Exited'.
2.158086591	Truck_RT12.802	Input@Sink_RT12	169.OnExited	[End]	Process 'Input@Sink_RT12.OnExited' execution ended.
2.158086591	Truck_RT12.802	Sink_RT12	19.OnEnteredInputBuffer	[Execute] DestroyingEntityAddOnProcess	Executing process 'EndRun'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	169.EndRun	[Begin]	Process 'EndRun' execution started.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	169.EndRun	[Decide] All Trucks departed?	Token branching on condition 'NumTrucksInSystem == 0'.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	169.EndRun	[Decide] All Trucks departed?	Token sent to 'False' exit.
2.158086591	Truck_RT12.802	Cross_Dock_EPTs	169.EndRun	[End]	Process 'EndRun' execution ended.

Table B.13 continued from previous page

Time(Hours)	EntityID	ObjectName	ProcessID	StepName	Action
2.158086591	Truck_RT12.802	Sink_RT12	19.OnEnteredInputBuffer	[Tally] TimelnSystem	Recording observation value(s) of type 'Expression' for tally statistic 'Sink_RT12.TimelnSystem'.
2.158086591	Truck_RT12.802	Sink_RT12	19.OnEnteredInputBuffer	[Tally] TimelnSystem	Recording observation value '2.14141992386109' Hours for tally statistic 'Sink_RT12.TimelnSystem'.
2.158086591	Truck_RT12.802	Sink_RT12	19.OnEnteredInputBuffer	[Destroy] Entity	Destroying object 'Truck_RT12.802'.



Data analysis

In this chapter, the analysis of input data is fitted in section C.1 for both the uncertain arrival times and the uncertain demand. Next, the empirical data is compared to the simulation results using statistical analysis in section C.2.

C.1. Input parameter fitting

To assess the appropriate distributions for the demand volume and arrival times, distributions are fitted against the available data sets. First, however, outliers may be needed to remove from the data set. This is performed using the z-score method, which is a common technique to identify and remove data points that are significantly different from the mean of the data set. The z-score measures how many standard deviations a data point is away from the mean [3]. Altman [3] states that data points with a z-score greater than a certain threshold (e.g., 2 or 3) are considered outliers and can therefore be removed.

The three best-fitting distributions are plotted and tested to determine their goodness of fit. The goodness of fit is tested using the Kolmogorov-Smirnov (KS) test. The Kolmogorov-Smirnov (KS) test is a statistical test used to determine whether a sample distribution differs significantly from a theoretical distribution and its main outcomes are the KS statistic and the KS p-value [49].

The KS statistic measures the maximum difference between the empirical distribution and the theoretical distribution by representing the most significant discrepancy between the observed data and the expected distribution. The larger the KS statistic, the greater the difference between the sample and theoretical distributions [49].

The KS p-value is the probability of obtaining a KS statistic as observed while assuming the sample data follows the theoretical distribution [49]. A smaller p-value suggests that the sample data is unlikely to have come from the theoretical distribution. Commonly, a significance level of 0.05 (5%) is used. If the KS p-value is greater than the chosen significance level, the p-value suggests that the sample data is consistent with the theoretical distribution.

Seen in Listing C.1, the following code in Python is used for fitting the distributions to the data set containing the number of pallets and the ATAs.

Listing C.1: Code for input parameter fitting in Python

```
1 # Author: Toon Hofstee
2 # Date: 2023-03-21
3 # Description: This script fits the demand volume input parameters to a
  distribution and plots the distribution
4
5 import pandas as pd
6 import numpy as np
7 import matplotlib.pyplot as plt
8 from scipy import stats
9
```

```

10 plt.rcParams['font.size'] = 14 # Set the font size to 12 points globally
11
12 # Read the data from Excel file
13 data = pd.read_excel('Data.xlsx', sheet_name='Sheet1')
14
15 # Extract the No. Pallets from the data set
16 values1 = data['Sum of Total Pallets'].values
17 # Drop NaN values from values1
18 values1 = values1[~np.isnan(values1)]
19
20 # Extract the ATAs from the data set
21 values2 = data['ATA'].values * 24 # Convert ATAs from days to hours
22 # Drop NaN values from values2
23 values2 = values2[~np.isnan(values2)]
24
25 # Distributions to fit
26 distributions = [
27     'expon',
28     'gamma',
29     'erlang',
30     'norm',
31     'lognorm',
32     'triang',
33     'uniform',
34     'weibull_min',
35     'weibull_max',
36     'fisk',
37     'genextreme',
38     'johnsonsb',
39     'johnsonsu',
40     'betaprime'
41 ]
42
43 ### Perform Analysis
44 # Plot histogram of No. Pallets before outlier removal
45 plt.figure(figsize=(8, 6))
46 plt.hist(values1, bins=30, density=True, alpha=0.7)
47 plt.xlabel('No. Pallets')
48 plt.ylabel('Frequency')
49 plt.title('Total no. pallets cross-docked per day (Before Outlier Removal)')
50 plt.savefig('HistogramNoPalletsBeforeOutlierRemoval.pdf')
51 plt.show()
52
53 # Remove outliers based on z-scores
54 z_scores = np.abs(stats.zscore(values1))
55 threshold = 3 # Set the z-score threshold for outlier removal
56 filtered_values1 = values1[z_scores < threshold]
57
58 # Plot histogram of no. pallets after outlier removal
59 plt.figure(figsize=(8, 6))
60 plt.hist(filtered_values1, bins=30, density=True, alpha=0.7)
61 plt.xlabel('No. Pallets')
62 plt.ylabel('Frequency')
63 plt.title('Total no. pallets cross-docked per day (After Outlier Removal)')

```

```

64 plt.savefig('HistogramNoPalletsAfterOutlierRemoval.pdf')
65 plt.show()
66
67 # Fit continuous distributions
68 best_distributions = []
69 best_params = []
70 best_sse = [np.inf] * 3
71
72 # Iterate over distributions and find the three best fits
73 for distribution in distributions:
74     # Fit the distribution to the filtered values
75     dist = getattr(stats, distribution)
76     params = dist.fit(filtered_values1)
77
78     # Calculate the sum of squared errors (SSE) between the fitted
79     # distribution and the filtered values
80     pdf_values = dist.pdf(filtered_values1, *params)
81     sse = np.sum((pdf_values - filtered_values1) ** 2)
82
83     # Update the three best distributions based on SSE
84     for i in range(3):
85         if sse < best_sse[i]:
86             best_distributions.insert(i, distribution)
87             best_params.insert(i, params)
88             best_sse.insert(i, sse)
89             break
90
91 # Plot histogram of filtered values and the three best-fitting
92 # distributions
93 x = np.linspace(np.min(filtered_values1), np.max(filtered_values1), 1000)
94 plt.figure(figsize=(12, 8))
95 plt.hist(filtered_values1, bins=30, density=True, alpha=0.7, label='
96     Filtered Data')
97
98 for i in range(3):
99     distribution = getattr(stats, best_distributions[i])
100     params = best_params[i]
101     y = distribution.pdf(x, *params)
102     plt.plot(x, y, label=best_distributions[i])
103
104 plt.xlabel('No. Pallets', fontsize=16)
105 plt.ylabel('Probability Density', fontsize=16)
106 plt.title('Filtered Data of no. pallets cross-docked with Best-Fitting
107     Distributions', fontsize=18)
108 plt.legend(fontsize=16)
109 plt.savefig('Filtered Data of no. pallets cross-docked with Best-Fitting
110     Distributions.pdf', dpi=300)
111 plt.show()
112
113 # Print distribution parameters, KS statistic, and KS p-value for the
114 # three best-fitting distributions
115 print("Three Best-Fitting Distributions No. Pallets (Filtered Data):")
116 for i in range(3):
117     distribution = getattr(stats, best_distributions[i])
118     params = best_params[i]
119     ks_statistic, ks_pvalue = stats.ks_2samp(filtered_values1,

```

```

    distribution.rvs(size=len(filtered_values1), *params))
114 print("Distribution:", best_distributions[i])
115 print("Parameters:", params)
116 print("KS Statistic:", ks_statistic)
117 print("KS p-value:", ks_pvalue)
118 print()
119
120
121
122 # Plot histogram of ATAs before outlier removal
123 plt.figure(figsize=(8, 6))
124 plt.hist(values2, bins=30, density=True, alpha=0.7)
125 plt.xlabel('Time of day [hours]')
126 plt.ylabel('Frequency')
127 plt.title('ATA observations (Before Outlier Removal)')
128 plt.savefig('HistogramATABeforeOutlierRemoval.pdf')
129 plt.show()
130
131 # Remove outliers based on z-scores
132 z_scores = np.abs(stats.zscore(values2))
133 threshold = 3 # Set the z-score threshold for outlier removal
134 filtered_values2 = values2[z_scores < threshold]
135
136 values2 = values2[(values2 >= 5) & (values2 <= 8)]
137
138 # Plot histogram of ATAs after outlier removal
139 plt.figure(figsize=(8, 6))
140 plt.hist(filtered_values2, bins=30, density=True, alpha=0.7)
141 plt.xlabel('Time of day [hours]')
142 plt.ylabel('Frequency')
143 plt.title('ATA observations (After Outlier Removal)')
144 plt.savefig('HistogramATAsAfterOutlierRemoval.pdf')
145 plt.show()
146
147 # Perform same analysis for ATAs
148 # Fit continuous distributions
149 best_distributions2 = []
150 best_params2 = []
151 best_sse2 = [np.inf] * 3
152
153 # Iterate over distributions and find the three best fits
154 for distribution in distributions:
155     # Fit the distribution to the filtered values
156     dist = getattr(stats, distribution)
157     params = dist.fit(filtered_values2)
158
159     # Calculate the sum of squared errors (SSE) between the fitted
160     # distribution and the filtered values
161     pdf_values = dist.pdf(filtered_values2, *params)
162     sse = np.sum((pdf_values - filtered_values2) ** 2)
163
164     # Update the three best distributions based on SSE
165     for i in range(3):
166         if sse < best_sse2[i]:
167             best_distributions2.insert(i, distribution)
168             best_params2.insert(i, params)

```

```

168         best_sse2.insert(i, sse)
169         break
170
171 # Plot histogram of filtered values and the three best-fitting
    distributions
172 x = np.linspace(np.min(filtered_values2), np.max(filtered_values2), 1000)
173 plt.figure(figsize=(12, 8))
174 plt.hist(filtered_values2, bins=30, density=True, alpha=0.7, label='
    Filtered Data')
175
176 for i in range(3):
177     distribution = getattr(stats, best_distributions2[i])
178     params = best_params2[i]
179     y = distribution.pdf(x, *params)
180     plt.plot(x, y, label=best_distributions2[i])
181
182
183 plt.xlabel('Time of day [hours]', fontsize=16)
184 plt.ylabel('Probability Density', fontsize=16)
185 plt.title('Filtered Data of ATAs with Best-Fitting Distributions',
    fontsize=18)
186 plt.legend(fontsize=16)
187 plt.savefig('Filtered Data of ATAs with Best-Fitting Distributions.pdf',
    dpi=300)
188 plt.show()
189
190 # Print distribution parameters, KS statistic, and KS p-value for the
    three best-fitting distributions
191 print("Three Best-Fitting Distributions ATAs (Filtered Data):")
192 for i in range(3):
193     distribution = getattr(stats, best_distributions2[i])
194     params = best_params2[i]
195     ks_statistic, ks_pvalue = stats.ks_2samp(filtered_values2,
        distribution.rvs(size=len(filtered_values2), *params))
196     print("Distribution:", best_distributions2[i])
197     print("Parameters:", params)
198     print("KS Statistic:", ks_statistic)
199     print("KS p-value:", ks_pvalue)
200     print()

```

The outcomes of the code, as well as the results of fitting the demand volume data and ATA data, are discussed in subsection C.1.1 and subsection C.1.2 respectively.

C.1.1. Fitting distributions of the total pallets cross-docked

First, all data of the total number of pallets handled during cross-docking per day is shown in Figure C.1

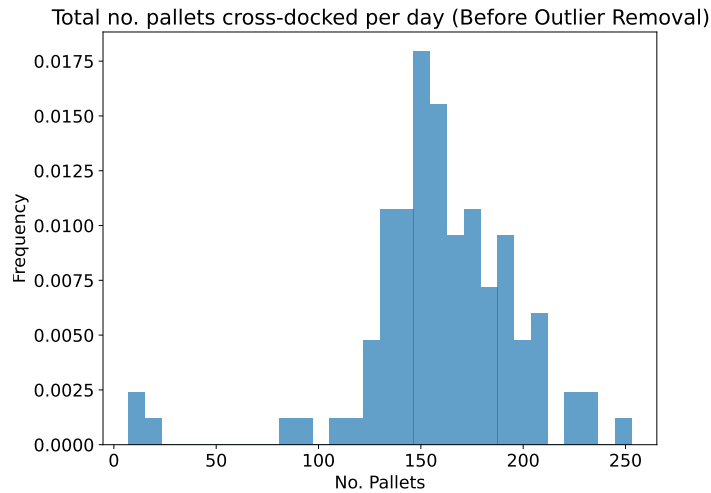


Figure C.1: Total number of pallets cross-docked per day (before outlier removal)

Depicted by Figure C.1 the data shows some outliers. Therefore, outlier removal is performed using the z-score method. After removing the outlier, the resulting data of the total number of pallets during cross-docking is shown in Figure C.2.

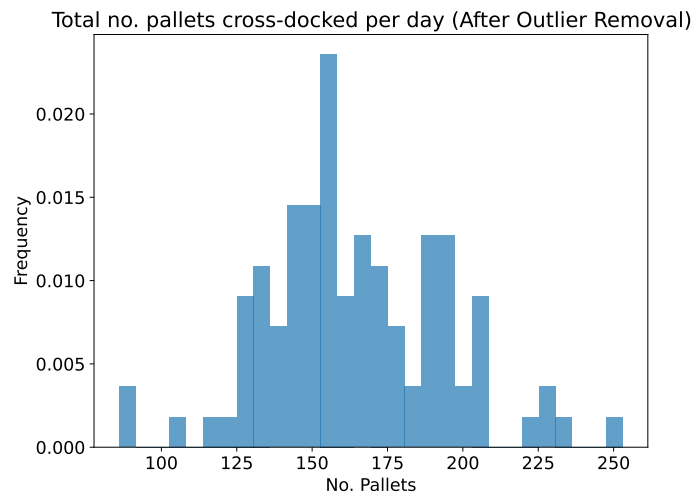


Figure C.2: Total number of pallets cross-docked per day (after outlier removal)

Now that the outliers are removed, the distributions available in SIMIO are fitted to the data. The three best-fitting distributions are shown in Figure C.3.

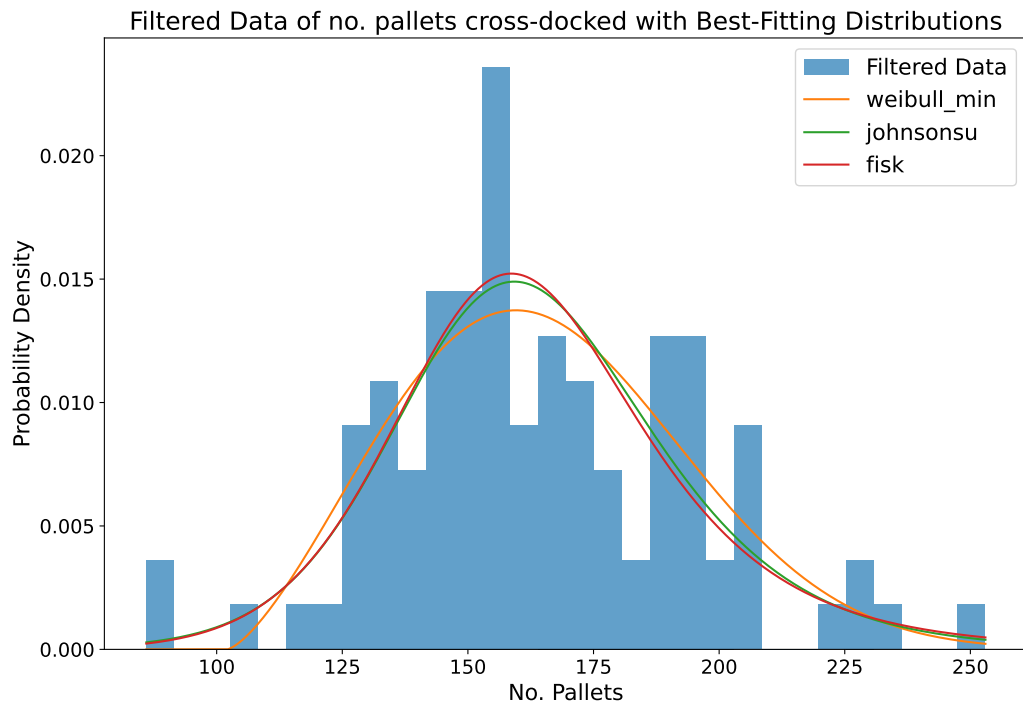


Figure C.3: Best fitting distributions to the total number of pallets cross-docked per day

The parameters of the three best-fitting distributions for the total number of pallets cross-docked are shown in Table C.1, Table C.2, and Table C.3.

Table C.1: Weibull Minimum Extreme Value Distribution Parameters

Parameter	Value
c	2.4200607775383887
$location$	102.05988855049631
$scale$	71.65644587497525
KS Statistic	0.0707070707070707
KS p-value	0.9669304174079811

Table C.2: JohnsonSU Distribution Parameters

Parameter	Value
a	-0.7346781698568297
b	2.2609129623897792
$location$	143.21367620188627
$scale$	57.95403853313774
KS Statistic	0.08080808080808081
KS p-value	0.9052627264915449

Table C.3: Fisk Distribution Parameters

Parameter	Value
<i>c</i>	162.39326613072024
<i>location</i>	9.785785970899354
<i>scale</i>	-0.398096889302725
KS Statistic	0.101010101010101
KS p-value	0.6961099797485474

Based on the KS statistic and the KS p-value, the *Fisk* distribution (also known as the LogLogistic distribution) is chosen for the demand volume distribution in this simulation model using the parameters presented in Table C.3.

For the filtered data on the total number of pallets cross-docked, the best-fitting distribution is the *Fisk* distributions, with parameters (9.785785970899354, -0.398096889302725, 162.39326613072024). These values indicate that the distribution fits the filtered data relatively well, but the p-value suggests some discrepancy.

The second best-fitting distribution for the filtered data is Johnson SU. The parameters are (-0.7346781698568297, 2.2609129623897792, 143.21367620188627, 57.95403853313774). The KS statistic is 0.15151515151515152, and the KS p-value is 0.20643393799415163. The p-value suggests a lower level of fit compared to the Weibull minimum distribution.

The third best-fitting distribution for the filtered data is *Weibull minimum*. The parameters for this distribution are (2.4200607775383887, 102.05988855049631, 71.65644587497525). The KS statistic is 0.12121212121212122, and the KS p-value is 0.4630101136905197. The KS statistic is 0.18181818181818182, and the KS p-value is 0.07568715714596005. The p-value indicates a relatively poor fit for the Fisk distribution.

In summary, the results suggest that the Weibull minimum distribution provides the best fit for the filtered data. The Johnson SU distribution performed moderately well and the Fisk distribution performed relatively poorly.

C.1.2. Fitting distributions of the arrival times

First, the arrival times are plotted before outlier removal in Figure C.4

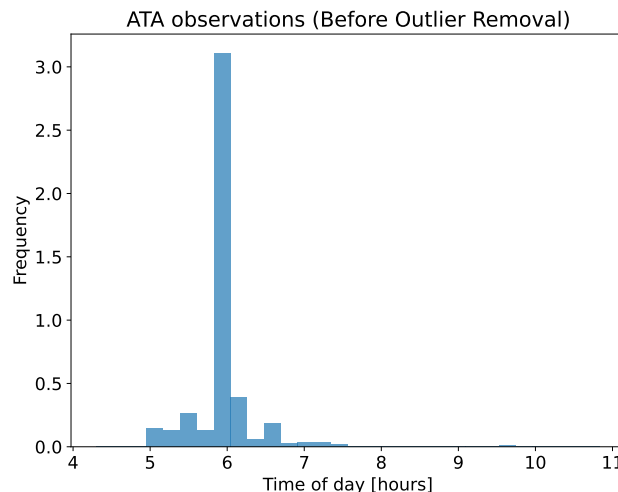


Figure C.4: Histogram ATA before outlier removal

After performing outlier removal, the data left is shown in Figure C.5.

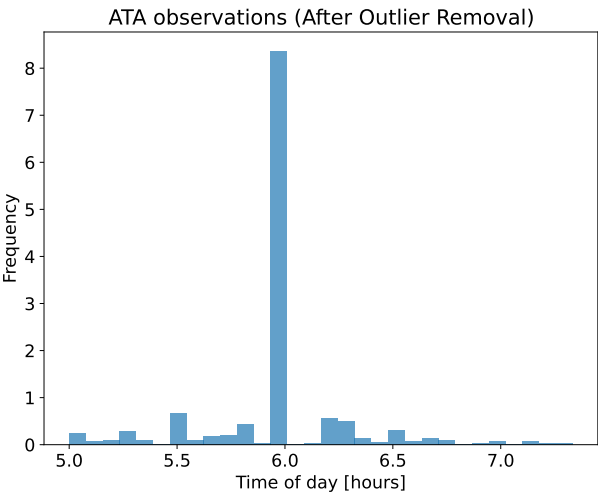


Figure C.5: Histogram ATA after outlier removal

For analysis of the arrival times distribution, the same code as shown in subsection C.1.1 is used and the data histogram plot including the three best-fitting distributions is presented in Figure C.6.

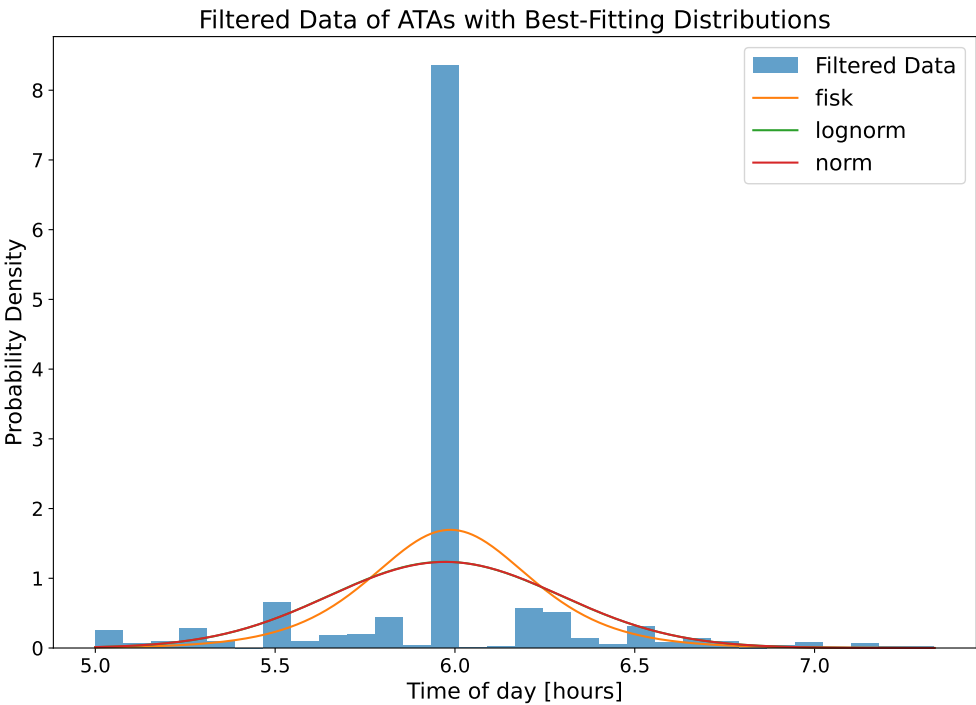


Figure C.6: Best fitting distributions on Actual Times of Arrival

The parameters of the three best-fitting distributions for the truck arrival times are shown in Table C.4, Table C.5, and Table C.6.

Table C.4: Fisk Distribution Parameters

Parameter	Value
<i>c</i>	74642.63922911385
<i>location</i>	-11002.506872763312
<i>scale</i>	11008.494258731065
KS Statistic	0.3264957264957265
KS p-value	$1.424990436067951 \times 10^{-55}$

Table C.5: Lognormal Distribution Parameters

Parameter	Value
<i>shape</i>	0.004083860053662496
<i>location</i>	-73.03415253300176
<i>scale</i>	79.00694496768216
KS Statistic	0.358974358974359
KS p-value	$2.3362471515870887 \times 10^{-67}$

Table C.6: Normal Distribution Parameters

Parameter	Value
<i>location</i>	5.973361823361824
<i>scale</i>	0.3230799064766255
KS Statistic	0.3435897435897436
KS p-value	$1.2659152730387234 \times 10^{-61}$

Based on the KS statistics and the KS p-values for the arrival time distributions, none of the distributions fit the empirical data sufficiently. Therefore, assumptions must be made for the arrival time distributions used in this research.

C.2. Analysis empirical and simulation data: Student's T-Test with Bootstrapping and Chi-Squared Test

Listing C.2: Code for performing Student's T-Test with Bootstrapping on simulation data and empirical data in Python

```
1 # Author: Toon Hofstee
2 # Date: 2023-05-27
3 # Description: This script compares the simulation data to the empirical
  data by plotting histograms and box plots. Additionally, it performs a
  t-test and bootstrapping to compare the difference in means of the two
  data sets. Finally, a chi-squared test is performed to compare the
  distributions of the two data sets.
4
5 import numpy as np
6 from scipy.stats import ttest_ind, zscore, ks_2samp, chisquare
7 import pandas as pd
8 import matplotlib.pyplot as plt
9
10
11 plt.rcParams['font.size'] = 14 # Set the font size to 12 points globally
12
13 # Read the empirical and simulation data
14 empirical_data = pd.read_csv('empirical_data.csv')
15 simulation_data = pd.read_csv('simulation_data3.csv')
16
17 # Filter simulation data for 'Scenario 5 - BaseCase'
18 simulation_data = simulation_data[simulation_data['Scenario'] == '
  Scenario5.2.3']
19
20 # Specify the column names
21 x_column = 'Throughput'
22 y_column = 'TotalMakespan'
23
24 # Extract the values for the specified columns
25 empirical_x_values = empirical_data[x_column].values
26 simulation_x_values = simulation_data[x_column].values
27
28 empirical_y_values = empirical_data[y_column].values
29 simulation_y_values = simulation_data[y_column].values
30
31 # Calculate Z-scores for each variable
32 z_scores_empirical_x = zscore(empirical_x_values)
33 z_scores_empirical_y = zscore(empirical_y_values)
34
35 z_scores_simulation_x = zscore(simulation_x_values)
36 z_scores_simulation_y = zscore(simulation_y_values)
37
38 # Set a threshold for Z-score outlier removal (e.g., 3 standard deviations
  )
39 z_score_threshold = 3
40
41 # Filter out outliers based on Z-scores
42 empirical_x_values_filtered = empirical_x_values[np.abs(
  z_scores_empirical_x) < z_score_threshold]
43 empirical_y_values_filtered = empirical_y_values[np.abs(
  z_scores_empirical_y) < z_score_threshold]
```

```

44
45 simulation_x_values_filtered = simulation_x_values[np.abs(
    z_scores_simulation_x) < z_score_threshold]
46 simulation_y_values_filtered = simulation_y_values[np.abs(
    z_scores_simulation_y) < z_score_threshold]
47
48 empirical_x_values = empirical_x_values_filtered
49 empirical_y_values = empirical_y_values_filtered
50
51 simulation_x_values = simulation_x_values_filtered
52 simulation_y_values = simulation_y_values_filtered
53
54 # Create a figure with three horizontally stacked subplots
55 fig, axs = plt.subplots(1, 2, figsize=(15, 5))
56
57 # Calculate the bin width based on the combined dataset
58 min_value_x = min(np.min(empirical_x_values), np.min(simulation_x_values))
59 max_value_x = max(np.max(empirical_x_values), np.max(simulation_x_values))
60 bin_width_x = (max_value_x - min_value_x) / 20 # Adjust the number of
    bins as needed
61
62 # Calculate the bin width for TotalMakespan
63 min_value_y = min(np.min(empirical_y_values), np.min(simulation_y_values))
64 max_value_y = max(np.max(empirical_y_values), np.max(simulation_y_values))
65 bin_width_y = (max_value_y - min_value_y) / 20 # Adjust the number of
    bins as needed
66
67 # Plot histograms for Throughput
68 #axs[0].hist(empirical_x_values, bins=20, alpha=0.5, label='Empirical Data
    ')
69 #axs[0].hist(simulation_x_values, bins=20, alpha=0.5, label='Simulation
    Data')
70 axs[0].hist(empirical_x_values, bins=np.arange(min_value_x, max_value_x,
    bin_width_x), alpha=0.5, label='Empirical Data')
71 axs[0].hist(simulation_x_values, bins=np.arange(min_value_x, max_value_x,
    bin_width_x), alpha=0.5, label='Simulation Data')
72 axs[0].set_xlabel('Values')
73 axs[0].set_ylabel('Frequency')
74 axs[0].set_title('Histograms of Throughput')
75 axs[0].legend()
76
77 # Plot histograms for TotalMakespan
78 #axs[1].hist(empirical_y_values, bins=20, alpha=0.5, label='Empirical Data
    ')
79 #axs[1].hist(simulation_y_values, bins=20, alpha=0.5, label='Simulation
    Data')
80 axs[1].hist(empirical_y_values, bins=np.arange(min_value_y, max_value_y,
    bin_width_y), alpha=0.5, label='Empirical Data')
81 axs[1].hist(simulation_y_values, bins=np.arange(min_value_y, max_value_y,
    bin_width_y), alpha=0.5, label='Simulation Data')
82 axs[1].set_xlabel('Values')
83 axs[1].set_ylabel('Frequency')
84 axs[1].set_title('Histograms of Total Makespan')
85 axs[1].legend()
86 plt.tight_layout()
87 plt.savefig('histograms_Empirical_Simulation.pdf')

```

```

88 plt.show()
89
90
91 # Create a figure with three horizontally stacked subplots
92 fig, axs = plt.subplots(1, 2, figsize=(15, 5))
93
94 # Create box plots for Throughput
95 axs[0].boxplot([empirical_x_values, simulation_x_values], labels=['
    Empirical Data', 'Simulation Data'])
96 axs[0].set_ylabel('Throughput')
97 axs[0].set_title('Comparison of Throughput')
98
99 # Create box plots for Total Makespan
100 axs[1].boxplot([empirical_y_values, simulation_y_values], labels=['
    Empirical Data', 'Simulation Data'])
101 axs[1].set_ylabel('Total Makespan')
102 axs[1].set_title('Comparison of Total Makespan')
103
104 # Adjust the layout and spacing between subplots
105 plt.tight_layout()
106 plt.savefig('boxplots_Empirical_Simulation.pdf')
107 plt.show()
108
109 # Perform bootstrapping for Throughput
110 num_bootstrap_samples = 1000
111 bootstrap_diffs_x = []
112 for _ in range(num_bootstrap_samples):
113     empirical_sample = np.random.choice(empirical_x_values, size=len(
        empirical_x_values), replace=True)
114     simulation_sample = np.random.choice(simulation_x_values, size=len(
        simulation_x_values), replace=True)
115     diff = np.mean(simulation_sample) - np.mean(empirical_sample)
116     bootstrap_diffs_x.append(diff)
117
118 # Perform bootstrapping for TotalMakespan
119 bootstrap_diffs_y = []
120 for _ in range(num_bootstrap_samples):
121     empirical_sample = np.random.choice(empirical_y_values, size=len(
        empirical_y_values), replace=True)
122     simulation_sample = np.random.choice(simulation_y_values, size=len(
        simulation_y_values), replace=True)
123     diff = np.mean(simulation_sample) - np.mean(empirical_sample)
124     bootstrap_diffs_y.append(diff)
125
126 # Calculate the observed difference in means for both variables
127 observed_diff_x = np.mean(simulation_x_values) - np.mean(
    empirical_x_values)
128 observed_diff_y = np.mean(simulation_y_values) - np.mean(
    empirical_y_values)
129
130 # Calculate the confidence intervals for both variables
131 confidence_level = 0.95
132 lower_bound_x = np.percentile(bootstrap_diffs_x, (1 - confidence_level) /
    2 * 100)
133 upper_bound_x = np.percentile(bootstrap_diffs_x, (1 + confidence_level) /
    2 * 100)

```

```

134 lower_bound_y = np.percentile(bootstrap_diffs_y, (1 - confidence_level) /
    2 * 100)
135 upper_bound_y = np.percentile(bootstrap_diffs_y, (1 + confidence_level) /
    2 * 100)
136
137 # Perform t-tests for both variables
138 stat_x, p_x = ttest_ind(simulation_x_values, empirical_x_values)
139 if p_x > 0.05:
140     significance_x = "not statistically significant"
141 else:
142     significance_x = "statistically significant"
143
144 stat_y, p_y = ttest_ind(simulation_y_values, empirical_y_values)
145 if p_y > 0.05:
146     significance_y = "not statistically significant"
147 else:
148     significance_y = "statistically significant"
149
150
151 # Create a figure with three horizontally stacked subplots
152 fig, axs = plt.subplots(1, 2, figsize=(15, 5))
153
154 # Plot bootstrap sampling distribution for both variables
155 axs[0].hist(bootstrap_diffs_x, bins=30, alpha=0.5)
156 axs[0].axvline(observed_diff_x, color='red', linestyle='--', label='
    Observed Difference')
157 axs[0].axvline(lower_bound_x, color='green', linestyle='--', label='Lower
    Bound')
158 axs[0].axvline(upper_bound_x, color='green', linestyle='--', label='Upper
    Bound')
159 axs[0].set_xlabel('Difference in Means')
160 axs[0].set_ylabel('Frequency')
161 axs[0].set_title('Bootstrap Sampling Distribution (Throughput)')
162 axs[0].legend()
163
164 axs[1].hist(bootstrap_diffs_y, bins=30, alpha=0.5)
165 axs[1].axvline(observed_diff_y, color='red', linestyle='--', label='
    Observed Difference')
166 axs[1].axvline(lower_bound_y, color='green', linestyle='--', label='Lower
    Bound')
167 axs[1].axvline(upper_bound_y, color='green', linestyle='--', label='Upper
    Bound')
168 axs[1].set_xlabel('Difference in Means')
169 axs[1].set_ylabel('Frequency')
170 axs[1].set_title('Bootstrap Sampling Distribution (TotalMakespan)')
171 axs[1].legend()
172
173 # Adjust the layout and spacing between subplots
174 plt.tight_layout()
175 plt.savefig('bootstrap_sampling_distributions.pdf')
176 plt.show()
177
178 # Print the results
179 print("Observed difference in means (Throughput):", observed_diff_x)
180 print("Confidence interval (Throughput):", (lower_bound_x, upper_bound_x))

```

```

181 print("The simulation results for Throughput are", significance_x, "
    compared to the empirical data.")
182
183 print("Observed difference in means (TotalMakespan):", observed_diff_y)
184 print("Confidence interval (TotalMakespan):", (lower_bound_y,
    upper_bound_y))
185 print("The simulation results for TotalMakespan are", significance_y, "
    compared to the empirical data.")
186
187
188 # Perform Kolmogorov-Smirnov test for Throughput
189 ks_stat_x, p_value_x = ks_2samp(empirical_x_values, simulation_x_values)
190
191 # Perform Kolmogorov-Smirnov test for TotalMakespan
192 ks_stat_y, p_value_y = ks_2samp(empirical_y_values, simulation_y_values)
193
194 # Print the results
195 print("Kolmogorov-Smirnov Test (Throughput):")
196 print("KS Statistic:", ks_stat_x)
197 print("p-value:", p_value_x)
198
199 print("\nKolmogorov-Smirnov Test (Total Makespan):")
200 print("KS Statistic:", ks_stat_y)
201 print("p-value:", p_value_y)
202
203
204 # Perform chi-squared goodness of fit test for Throughput
205 observed_freq_x, _ = np.histogram(simulation_x_values, bins=np.arange(
    min_value_x, max_value_x, bin_width_x))
206 expected_freq_x = np.histogram(empirical_x_values, bins=np.arange(
    min_value_x, max_value_x, bin_width_x))[0]
207
208 # Normalize the frequencies
209 observed_freq_x = observed_freq_x / np.sum(observed_freq_x)
210 expected_freq_x = (expected_freq_x + 1e-8) / np.sum(expected_freq_x + 1e
    -8) # Add a small constant to avoid zero divisions
211
212 chi2_stat_x, p_value_chi2_x = chisquare(f_obs=observed_freq_x, f_exp=
    expected_freq_x)
213
214 # Perform chi-squared goodness of fit test for TotalMakespan
215 observed_freq_y, _ = np.histogram(simulation_y_values, bins=np.arange(
    min_value_y, max_value_y, bin_width_y))
216 expected_freq_y = np.histogram(empirical_y_values, bins=np.arange(
    min_value_y, max_value_y, bin_width_y))[0]
217
218 # Normalize the frequencies
219 observed_freq_y = observed_freq_y / np.sum(observed_freq_y)
220 expected_freq_y = (expected_freq_y + 1e-8) / np.sum(expected_freq_y + 1e
    -8) # Add a small constant to avoid zero divisions
221
222 chi2_stat_y, p_value_chi2_y = chisquare(f_obs=observed_freq_y, f_exp=
    expected_freq_y)
223
224 # Print the results
225 print("Chi-squared Goodness of Fit Test (Throughput):")

```

```

226 print("Chi-squared Statistic:", chi2_stat_x)
227 print("p-value:", p_value_chi2_x)
228
229 print("\nChi-squared Goodness of Fit Test (Total Makespan):")
230 print("Chi-squared Statistic:", chi2_stat_y)
231 print("p-value:", p_value_chi2_y)

```

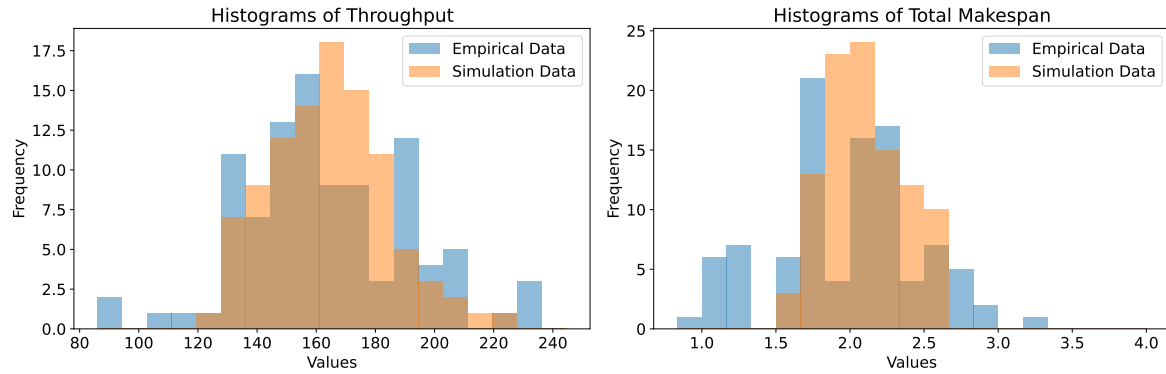


Figure C.7: Histograms of empirical data and simulation generated base case data

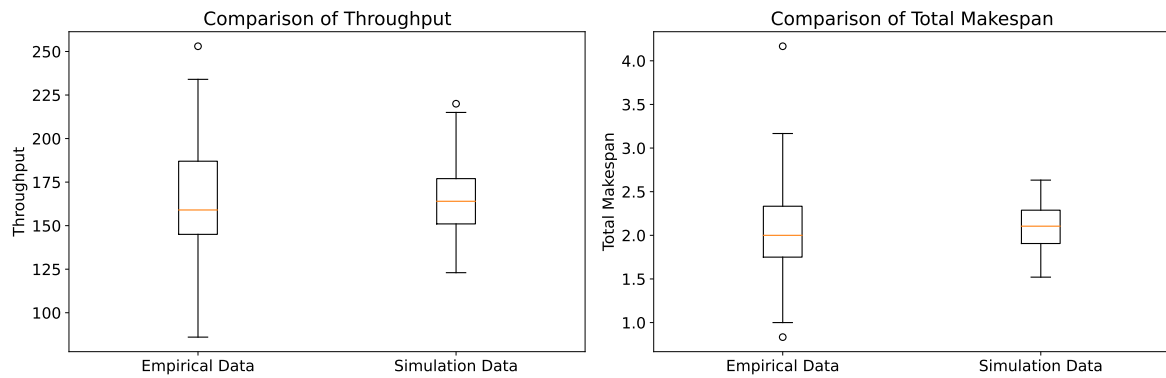


Figure C.8: Box plots of empirical data and simulation generated base case data

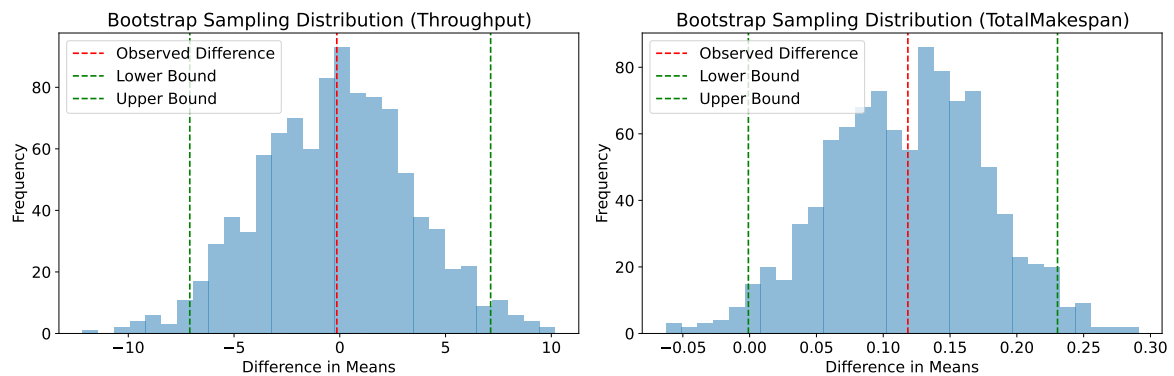


Figure C.9: Bootstrap plots of empirical data and simulation generated base case data

Table C.7: Student's T-Test Results

Variable	Observed Difference	Confidence Interval	Statistical Significance
Throughput [#]	-0.1414	(-7.0922, 7.1316)	Not Significant
TotalMakespan [h]	0.1184	(-0.0009, 0.2304)	Not Significant
<i>N_{empirical} = 102, N_{simulation} = 100</i>			

Table C.7 shows the observed difference in means for each variable, the confidence interval, and the statistical significance.

For the "Throughput" variable, the observed difference in means is -0.1414, indicating that the simulation results are slightly lower than the empirical data. However, the confidence interval (-7.0922, 7.1316) includes zero, which means that the difference is not statistically significant.

For the "TotalMakespan" variable, the observed difference in means is 0.1184, indicating that the simulation results are slightly higher than the empirical data. Again, the confidence interval (-0.0009, 0.2304) includes zero, indicating that the difference is not statistically significant.

Table C.8: Chi-squared Goodness of Fit Test Results

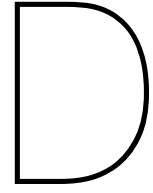
Variable	Chi-squared Statistic	p-value	Statistical Significance
Throughput	0.9022	0.9282	Not Significant
TotalMakespan	1.3408	0.8235	Not Significant
<i>N_{empirical} = 102, N_{simulation} = 100</i>			

Table C.8 shows the chi-squared statistic, p-value, and statistical significance for the Chi-squared Goodness of Fit Test.

For the "Throughput" variable, the chi-squared statistic is 0.9022, and the p-value is 0.9282, indicating an insignificant difference between the simulated and empirical data. Therefore, the simulation results for throughput are in good fit with the empirical data.

For the "TotalMakespan" variable, the chi-squared statistic is 1.3408, and the p-value is 0.8235, indicating that there is no significant difference between the simulated and empirical data. Therefore, the simulation results for total makespan are in good fit with the empirical data.

In both the Student's t-test and the Chi-squared goodness of fit tests, the results indicate that there is no statistically significant difference between the empirical and simulated data for both throughput and total makespan. The sample sizes for the empirical and simulated data are given as $N_{\text{empirical}} = 102$ and $N_{\text{simulation}} = 100$, respectively. However, it is essential to consider the specific data, analysis, and any limitations or assumptions made during the study when interpreting these results.



Results

The results from the experiments are presented using box plots in Figure D.1 and Figure D.2 for the KPIs *Makespan* and *Average Waiting Time Per Truck* respectively.

D.1. Box plots

For each experiment, the KPIs are presented using box plots. In Figure D.1, the box plots show that the variance of the results in total makespan for the two-stage alternative is lower than in the base case. Interestingly, the variance increases for using two-stage cross-docking with roll containers. The increase in the total number of unit loads transported can explain the increase in variance. Additionally, the mean and variances for the pure cross-docking scenarios (Scenario 10 to Scenario 18) are lower for the base case and the two-stage alternative. When examining Table 5.4, the decreases in successful replications indicate that the variance is unreliable and that the pure cross-docking scenarios cannot be used for comparison.

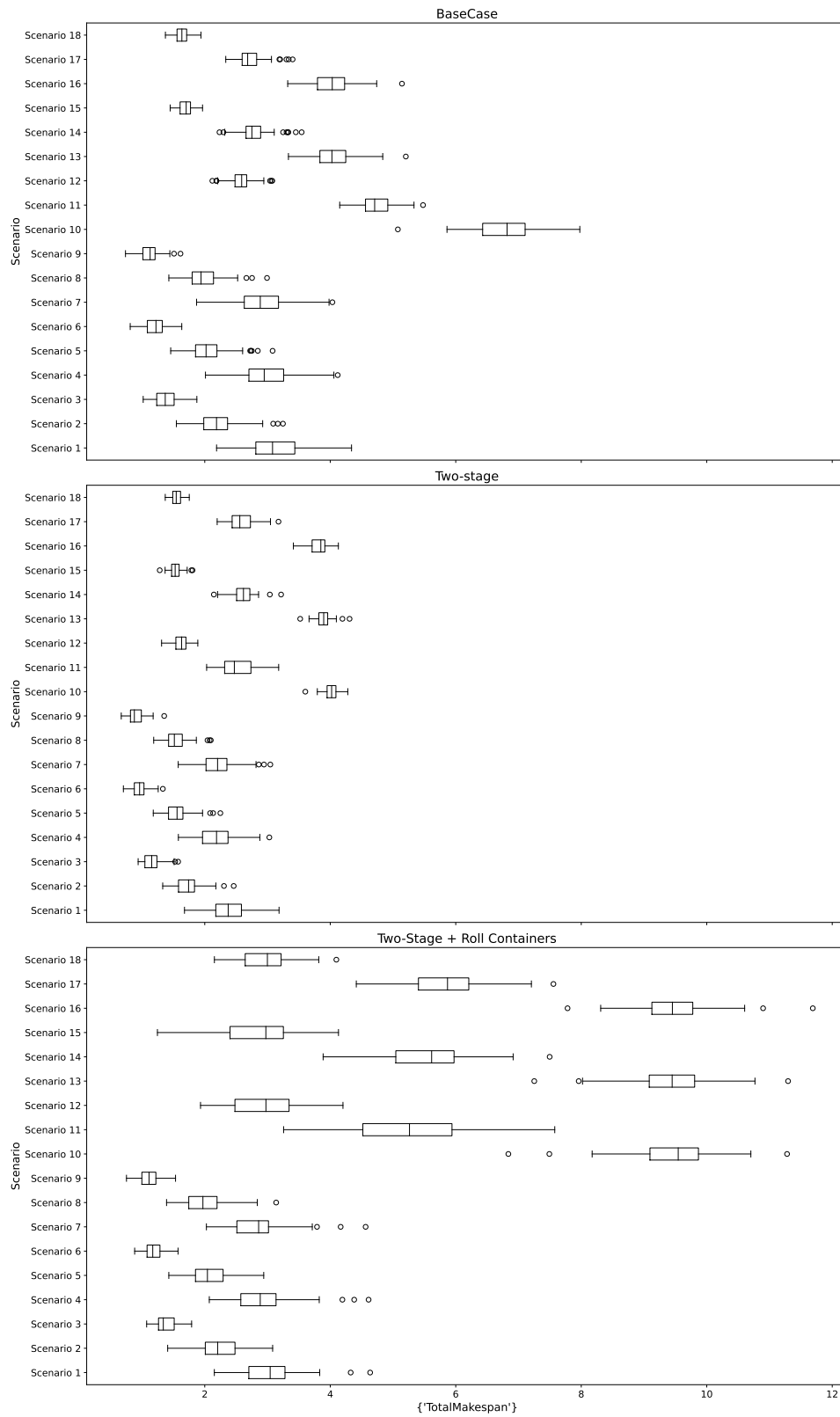


Figure D.1: Box plots Total Makespan

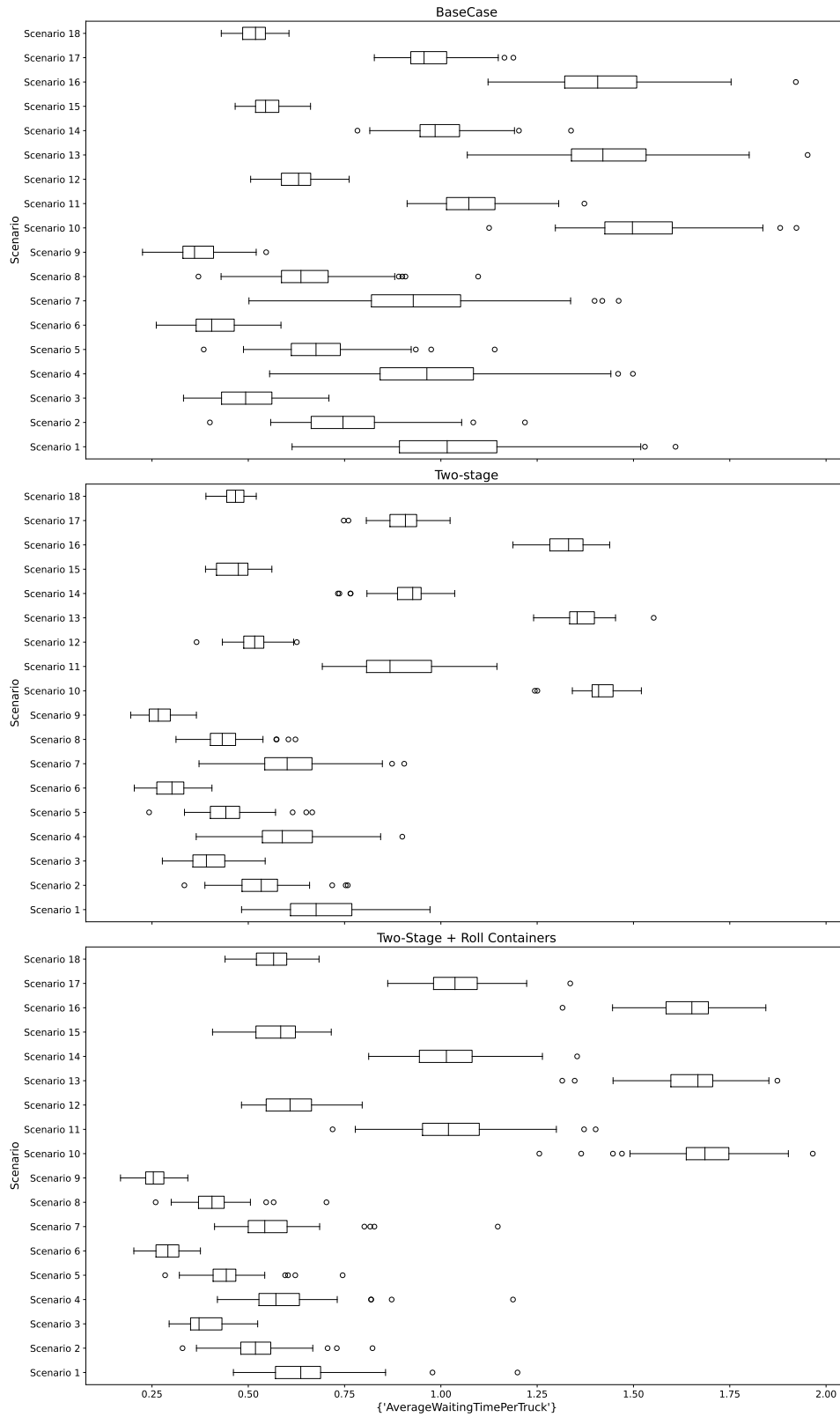


Figure D.2: Box plots Average Waiting Time Per Truck

D.2. Pair-wise Student's T-Testing

The Pair-wise Student's t-test is useful for analyzing simulation experiments for multiple scenarios. It quantitatively compares means, tests hypotheses, identifies significant differences, and quantifies effect size, aiding decision-making and scenario-specific analysis. The pair-wise Student's T-Test results for the Total Makespan are presented in Table D.1. The t-statistic and p-value are provided for each comparison. The t-statistic measures the difference between the means of the two groups [26], where a larger absolute value indicates a greater difference. The p-value represents the probability of observing the given t-statistic (or a more extreme value) under the null hypothesis that there is no significant difference between the means of the two groups [26]. A smaller p-value suggests stronger evidence against the null hypothesis and indicates a more significant difference between the means.

Upon analyzing the resulting Total Makespan for the experiments, several comparisons appear to be of significant interest due to their considerable differences. For the following scenarios, the pair-wise comparison presents statistically insignificant differences in Total Makespan:

- **Scenario1:** BaseCase vs. Two-Stage + Roll Containers (t-value = 1.7841 , p-value = 0.076)
- **Scenario2:** BaseCase vs. Two-Stage + Roll Containers (t-value = -0.7522 , p-value = 0.453)
- **Scenario3:** BaseCase vs. Two-Stage + Roll Containers (t-value = 0.1738 , p-value = 0.862)
- **Scenario4:** BaseCase vs. Two-Stage + Roll Containers (t-value = 1.6266 , p-value = 0.105)
- **Scenario5:** BaseCase vs. Two-Stage + Roll Containers (t-value = -0.1737 , p-value = 0.862)
- **Scenario6:** BaseCase vs. Two-Stage + Roll Containers (t-value = 1.0370 , p-value = 0.301)
- **Scenario7:** BaseCase vs. Two-Stage + Roll Containers (t-value = 1.7218 , p-value = 0.087)
- **Scenario8:** BaseCase vs. Two-Stage + Roll Containers (t-value = 0.1315 , p-value = 0.896)
- **Scenario9:** BaseCase vs. Two-Stage + Roll Containers (t-value = 0.5856 , p-value = 0.559)

These findings are interesting because the p-value indicates insignificant differences in total makespan when comparing the Base Case to the Two-stage cross-docking with Roll Containers alternatives. This means that the used inputs result in insignificant improvements when using roll containers compared to the base case. It could be interesting to investigate how this comparison would turn out if the throughput volume increases even more so.

Additionally, other results indicate statistically significant differences:

- **Scenario13:** Comparing Experiment 1 (BaseCase) with Experiment 2 (Two-Stage + Roll Containers) in Scenario13 shows a particularly striking difference. The t-statistic is -62.62514674, indicating a significantly lower mean for Experiment 2. The p-value is 6.7063E-118, which is exceptionally small, further supporting the significance of this result. This suggests that the addition of "Roll Containers" in Experiment 2 had a substantial impact, leading to significantly different outcomes compared to the BaseCase in Scenario13. This finding resembles that in pure cross-docking situations, the use of pallets can be preferred over roll containers.
- **Scenario16:** The comparison between Experiment 1 (BaseCase) and Experiment 2 (Two-Stage + Roll Containers) in Scenario16 also exhibits a highly notable difference. The t-statistic is -71.15821347, with an extremely small p-value of 4.825E-130. This indicates a substantially lower mean for Experiment 2 compared to Experiment 1. The magnitude of the t-statistic and the exceptionally small p-value suggest that the inclusion of "Two-Stage + Roll Containers" led to significantly different outcomes compared to the BaseCase in Scenario16. This finding means that in pure cross-docking situations, the use of pallets can be preferred over roll containers.

- **Scenario18:** In Scenario18, comparing Experiment 1 (BaseCase) with Experiment 2 (Two-Stage + Roll Containers) also reveals an interesting difference. The t-statistic is -27.98911585, and the p-value is 1.15374E-67. These values indicate a significantly lower mean for Experiment 2 compared to Experiment 1. The large absolute t-statistic and the extremely small p-value suggest that the addition of "Two-Stage + Roll Containers" had a substantial effect, leading to significantly different outcomes in Scenario18. This finding means that in pure cross-docking situations, the use of pallets can be preferred over roll containers.

Overall, these highlighted results demonstrate substantial differences between Experiment 1 and Experiment 2 in specific scenarios, indicating the potential impact of certain factors or interventions. Further investigation and analysis could help elucidate the underlying causes of these differences and provide valuable insights for decision-making, process optimization, or further experimentation in the respective scenarios.

Table D.1: Pair-wise Student's T-testing of the Total Makespan for the experiments per scenario

Scenario	Experiment 1	Experiment 2	t-statistic	p-value
Scenario1	BaseCase	Two-stage	13.366591	3.38E-29
Scenario1	BaseCase	Two-Stage + Roll Containers	1.7840511	0.075955
Scenario1	Two-stage	Two-Stage + Roll Containers	-11.278954	6.46E-23
Scenario2	BaseCase	Two-stage	12.985461	2.92E-28
Scenario2	BaseCase	Two-Stage + Roll Containers	-0.752166	0.452849
Scenario2	Two-stage	Two-Stage + Roll Containers	-13.149669	1E-28
Scenario3	BaseCase	Two-stage	9.5014813	7.06E-18
Scenario3	BaseCase	Two-Stage + Roll Containers	0.173832	0.862176
Scenario3	Two-stage	Two-Stage + Roll Containers	-9.9204838	4.53E-19
Scenario4	BaseCase	Two-stage	15.157082	8.41E-35
Scenario4	BaseCase	Two-Stage + Roll Containers	1.6266185	0.105417
Scenario4	Two-stage	Two-Stage + Roll Containers	-12.789164	1.47E-27
Scenario5	BaseCase	Two-stage	14.514303	1.17E-32
Scenario5	BaseCase	Two-Stage + Roll Containers	-0.1736834	0.862292
Scenario5	Two-stage	Two-Stage + Roll Containers	-13.723751	2.83E-30
Scenario6	BaseCase	Two-stage	12.31263	3.08E-26
Scenario6	BaseCase	Two-Stage + Roll Containers	1.0370071	0.301004
Scenario6	Two-stage	Two-Stage + Roll Containers	-11.333669	3E-23
Scenario7	BaseCase	Two-stage	13.225209	1.08E-28
Scenario7	BaseCase	Two-Stage + Roll Containers	1.7218356	0.086669
Scenario7	Two-stage	Two-Stage + Roll Containers	-11.190079	1.32E-22
Scenario8	BaseCase	Two-stage	13.668312	3.78E-30
Scenario8	BaseCase	Two-Stage + Roll Containers	0.1315003	0.895514
Scenario8	Two-stage	Two-Stage + Roll Containers	-12.115442	1.83E-25
Scenario9	BaseCase	Two-stage	11.591336	6.02E-24
Scenario9	BaseCase	Two-Stage + Roll Containers	0.58562	0.558798
Scenario9	Two-stage	Two-Stage + Roll Containers	-10.467529	1.3E-20
Scenario10	BaseCase	Two-stage	23.21357	8.5E-39
Scenario10	BaseCase	Two-Stage + Roll Containers	-28.979394	2.95E-67
Scenario10	Two-stage	Two-Stage + Roll Containers	-33.713982	3.98E-61
Scenario11	BaseCase	Two-stage	37.230086	6.12E-62
Scenario11	BaseCase	Two-Stage + Roll Containers	-4.6011324	8.13E-06
Scenario11	Two-stage	Two-Stage + Roll Containers	-15.866825	3.33E-32
Scenario12	BaseCase	Two-stage	30.146257	4.47E-62
Scenario12	BaseCase	Two-Stage + Roll Containers	-6.1315253	5.02E-09
Scenario12	Two-stage	Two-Stage + Roll Containers	-16.523405	3.05E-35
Scenario13	BaseCase	Two-stage	2.0591045	0.04244
Scenario13	BaseCase	Two-Stage + Roll Containers	-62.625147	6.7E-118
Scenario13	Two-stage	Two-Stage + Roll Containers	-38.579975	4.07E-69
Scenario14	BaseCase	Two-stage	3.3084093	0.001301
Scenario14	BaseCase	Two-Stage + Roll Containers	-32.383458	2.42E-75
Scenario14	Two-stage	Two-Stage + Roll Containers	-21.707836	2.13E-44
Scenario15	BaseCase	Two-stage	7.0414266	1.4E-10
Scenario15	BaseCase	Two-Stage + Roll Containers	-18.549995	8.15E-44
Scenario15	Two-stage	Two-Stage + Roll Containers	-13.7224	3.01E-27
Scenario16	BaseCase	Two-stage	3.1401959	0.002238
Scenario16	BaseCase	Two-Stage + Roll Containers	-71.158213	4.8E-130
Scenario16	Two-stage	Two-Stage + Roll Containers	-45.885181	8.35E-79
Scenario17	BaseCase	Two-stage	2.7933122	0.006192
Scenario17	BaseCase	Two-Stage + Roll Containers	-43.819741	5.99E-97
Scenario17	Two-stage	Two-Stage + Roll Containers	-28.516844	4.7E-57
Scenario18	BaseCase	Two-stage	4.3697838	2.63E-05
Scenario18	BaseCase	Two-Stage + Roll Containers	-27.989116	1.15E-67
Scenario18	Two-stage	Two-Stage + Roll Containers	-21.54465	3.84E-46

D.3. Central Limit Theorem

Using Central Limit Theorem (CLT), a sufficiently large sample size is around 30 samples [25]. Since the simulated data points are independent and homogeneous, the CLT can be applied. Therefore, only scenarios for which more than 30 replications have been completed are considered for analysis.

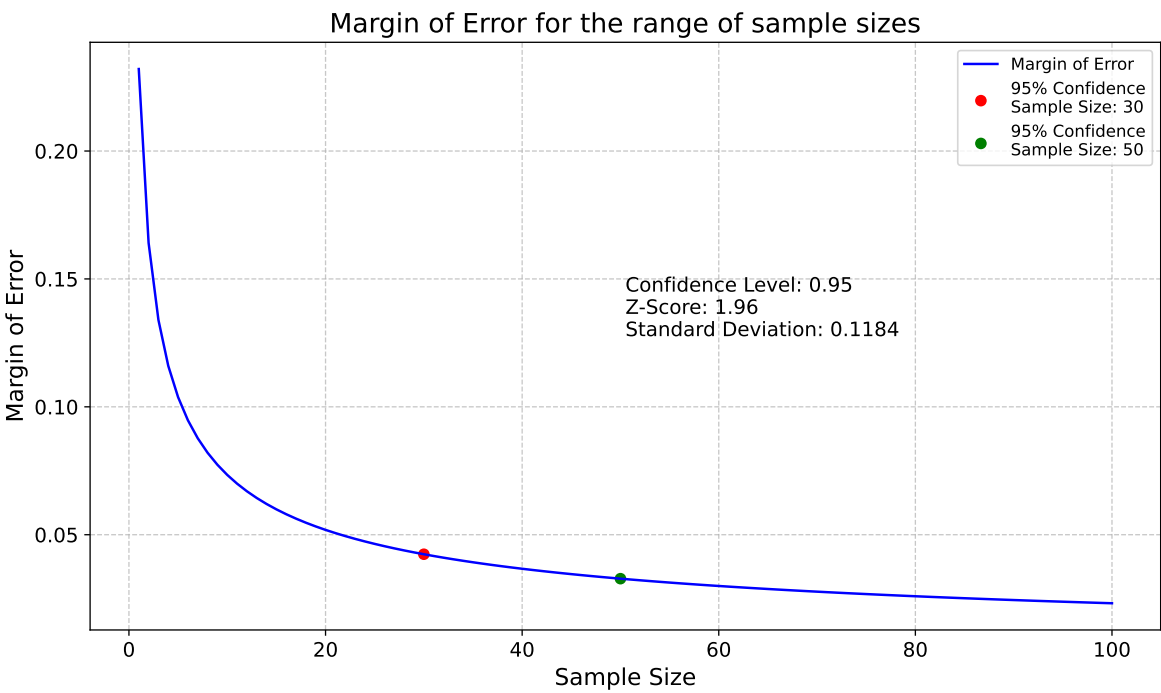


Figure D.3: Margin of error for sample sizes

E

Research paper

The Impact of Load Carrier Types and Staging-Level Designs on Cross-Docking Performance under Uncertainty

A.W.B. Hofstee^{a,b,1}, Prof.dr. R.R. Negenborn^a, Ir. M.M. Duinkerken^a, Dr. S. Fazi^b

^aTU Delft, Faculty of Mechanical, Maritime and Materials Engineering, Mekelweg 2, Delft, 2628 CD, The Netherlands

^bTU Delft, Faculty of Technology, Policy and Management, Jaffalaan 5, Delft, 2628 BX, The Netherlands

Abstract

This research paper focuses on improving the performance of cross-docking operations under uncertainty in the context of e-commerce logistics. The growth of e-commerce sales has increased product returns and complexity to supply chains. To address this issue, this study investigates how cross-docking operations can be improved under external and internal uncertainty factors. The research begins with a literature review to understand cross-docking facilities (CDFs) and measures to mitigate the effects of uncertainty. The current state of a CDF in a case study for a Fourth Party Logistics (4PL) provider is examined, and by reflecting on the literature overview, two potential means for decreasing the effects of uncertainty are identified: staging-level design and load carrier-type design.

A Discrete Event Simulation (DES) model is developed to test the effects of staging-level design and load carrier types on the performance of the CDF. The simulation model captures input factors such as truck arrivals, freight levels, and the purity level of cross-docking. The simulation model's performance is tested for different scenarios, and the effects of different design alternatives are analyzed.

The results demonstrate that two-stage cross-docking with pallets can significantly reduce the total makespan and improve operational efficiency compared to single-stage cross-docking with pallets. The results also show that using roll containers significantly decreases the chance of intra-terminal congestion but also results in longer unloading and reloading times.

The research contributes to the understanding of cross-docking operations under uncertainty, stresses the importance of staging-level and load carrier type design on CDF performance, and provides insights for logistics companies seeking to optimize their e-commerce supply chains.

Keywords: cross-docking, logistics, 4PL, simulation, modeling, staging-level design, load carrier type, cross-docking purity

1. Introduction

E-commerce sales have increased substantially in recent years, with a 290% growth reported between 2014 and 2020. This trend is set to continue, with an estimated \$8.1 trillion in e-commerce sales by 2026 (Chevalier, 2022). However, this growth has also led to a high product return rate of 30%, adding complexity to already complex supply chains. E-commerce merchants heavily rely on logistical companies for timely product delivery. These companies have established extensive supply chain networks, including warehouses, sorting terminals, transportation modes, and cross-docking facilities (CDFs). With the increase in e-commerce sales and transported volume, logistics companies must expand their assets and optimize their operations for greater efficiency.

According to Zhang, Atasoy and Negenborn (2022), optimal resource utilization and efficiency in logistics operations heavily depend on synchronization. This synchronization is crucial in cross-docking operations, where there is an absence or lim-

ited storage buffer. The primary purpose of a CDF is to transfer goods directly from inbound trucks to outbound trucks with minimal intermediate storage, achieving "just-in-time" (JIT) logistics (Ladier and Alpan, 2016b). However, synchronizing inbound and outbound trucks is often challenging due to delays and a limited number of dock doors. These delays lead to waiting times, inefficiencies, and reduced cost-effectiveness (Kuo, 2013). Effective synchronization of inbound and outbound operations in cross-docking, as Gallo et al. (2022) points out, leads to cost reduction and on-time deliveries. Uncertainties such as truck arrival times and product numbers can affect cross-docking operations.

This study aims to improve the efficiency of CDFs under uncertainty by addressing intra-terminal operations that limit CDFs' performance through a case study of a Fourth Party Logistics (4PL) provider.

2. Literature review

2.1. Cross-docking

A cross-docking terminal (CDF) is a facility used in supply chains to consolidate and redistribute goods without long-term

Email address: A.W.B.Hofstee@student.tudelft.nl (A.W.B. Hofstee)

¹Candidate of MSc Transport, Infrastructure & Logistics, at TU Delft

storage. Goods typically spend less than 24 hours at a CDF. The main activities at a CDF include unloading incoming trucks, sorting goods, and loading outbound trucks for delivery (Ladier and Alban, 2016a). Uncertainty in arrival times, processing times, and freight volumes can impact CDF performance (Arakani and Fei, 2020).

Cross-docking is a crucial logistics strategy to minimize handling and storage costs in distribution centers. The network structure of the facility locations and routes for freight volume transportation is determined through network design decisions based on expected demand. Network planning assigns volumes to routes, determining opportunities for consolidation at a cross-dock, while network scheduling considers capacity and time window constraints for vehicle routing.

Decision problems relevant to cross-docking terminals are listed from long-term strategic problems to short-term operational problems. These include the location of cross-docking terminals, layout of the cross-dock, assignment of destinations to dock doors, vehicle routing, truck scheduling, resource scheduling, and (un-)packing loads into (from) trucks (Van Belle et al., 2012).

Research on cross-docking aims to improve operations through optimization and simulation models that address scheduling, dock door assignment, transshipment, vehicle routing, product allocation, layout design, and network (Mavi et al., 2020). Warehouse management systems are also present in cross-docking operations, addressing inventory and data visibility, traceability, supply chain agility, and operational adaptability.

It is essential to ensure synchronization between inbound and outbound operations. This leads to cost reduction and increases in on-time deliveries. However, operations are sensitive to uncertain truck arrivals. Inbound and outbound truck sequencing can influence cross-docking efficiency, as delays occur when an inbound truck is waiting in the queue while its corresponding outbound truck is ready for loading. Cross-dock decisions include truck scheduling, assigning trucks to docks, and assigning products to locations.

2.2. CDF design

Strategic decision-making is necessary when designing a cross-dock terminal. Designing a cross-dock facility involves considering numerous aspects, as work by Boysen and Flidner (2010) stated. Once the cross-docking terminal locations are determined, the layout of the terminal is planned, including physical, operational, and flow characteristics, as Van Belle et al. (2012).

Bartholdi et al. (2001) mention that cross-docks can be designed in two ways. The first is pre-distribution, where pallets are sorted and labeled as soon as they arrive at the cross-dock. The second is post-distribution, which requires the cross-docking facility to sort the freight and assign shipment destinations. According to Yang et al. (2010), post-distribution requires more floor space than pre-distribution.

Staging methods for cross-docking operations can also vary. Single-stage cross-docking involves the direct staging of pallets into areas of their destination, while a two-stage cross-dock

permits the re-sorting of pallets but requires more floor space and additional travel time. A free-staging cross-dock is also possible, where pallets are placed at one end and pulled from the other without any queues, as explained by Gue and Kang (2001). Gue and Kang (2001) highlight that using flow racks over floor storage does not provide many benefits. According to Yang et al. (2010), additional handling by staging freight at the doors reduces congestion, so less handling may not always be better.

In addition to staging methods, cross-dock facilities must consider internal transportation and load carriers/consolidation methods. According to Van Belle et al. (2012), transportation methods can range from manual pallet jacks to automated guided vehicles (AGVs). The choice of transportation method can impact the time required to move freight within the facility, the level of automation, and associated costs.

Load carriers and consolidation methods are also important considerations. Boysen and Flidner (2010) note that load carriers can include pallets, containers, and even individual items, with each carrier type having advantages and disadvantages. Consolidation methods can range from manual sorting to automated sorting using conveyors and sorting systems (Bartholdi et al., 2001).

Yang et al. (2010) suggest using a combination of load carriers and consolidation methods to minimize handling and transportation time to optimize cross-docking operations. For example, using larger load carriers and sorting them using automated systems can reduce the required handling steps. Additionally, using standardized load carriers can simplify the consolidation process and reduce the need for manual sorting.

Overall, designing an efficient cross-dock facility requires careful consideration of internal transportation methods, load carrier/consolidation methods, staging methods, and terminal layout. By optimizing these factors, cross-dock facilities can reduce handling and transportation time, improve efficiency, and ultimately save costs.

Uncertainty surrounding truck arrival times is the main challenge in cross-docking systems, along with unpredictable demand and internal handling times. Optimization models can optimize problems to optimality and find optimal solutions to a given problem. Simulation models capture complex system behavior but require significant development time. Hybrid models integrate analytical and simulation models but can be complicated to coordinate. Dealing with uncertainty can be achieved through proactive, reactive, or predictive-reactive strategies. As proposed by Nasiri et al. (2022), a predictive-reactive system is designed to manage truck loading and unloading operations by rescheduling and considering delays. Simulation is better suited for modeling cross-docking operations with many uncertainties and interdependencies, while optimization is better for modeling truck-to-dock assignment, product-to-location assignment, and truck scheduling. However, the main shortcomings of discrete-event simulation are the limited generalizability of research findings and the inability to solve problems optimally.

To evaluate the effectiveness of cross-docking operations, a set of performance measures have been established by Ladier

Table 1: Characteristics of other studies versus this study

Reference	Scheduling problems		Physical characteristics		Process control		Product flow		Uncertainty factors			Method	Solving	
	Truck-to-dock assignment	Product-to-location assignment	Truck scheduling	Limited capacity staging area	Number of dock doors	Mixed-mode trucks	Mixed-mode dock doors	Reverse logistics	Forward logistics	Arrival times	Demand			Internal handling time
Boysen and Fliedner (2010)	✓			✓				✓					MILP	Exact
Bodnar et al. (2017)	✓	✓	✓		✓	✓	✓	✓	✓	✓			ILP	Exact & ALNS
Konur and Golias (2013a)			✓							✓			Bi-level MILP	GA
Konur and Golias (2013b)	✓									✓			Bi-level MILP	GA
Vis and Roodbergen (2008)			✓	✓	✓				✓				MILP	DP
Serrano et al. (2017)			✓	✓	✓		✓						MILP	Exact
Shakeri et al. (2016)	✓	✓	✓	✓					✓				DP	VNS, TS
Rahmanzadeh Tootkaleh et al. (2016)	✓		✓		✓				✓				MILP	Heuristic
Amini and Tavakkoli-Moghaddam (2016)	✓		✓		✓	✓			✓				Multi-objective MILP	NSGA-II, MOSA, MODE
Rahbari et al. (2019)	✓		✓	✓	✓					✓			Bi-level MILP	Exact
Xi et al. (2020)	✓		✓			✓	✓			✓		✓	MILP	C&CG
Larbi et al. (2009)	✓		✓	✓		✓	✓			✓			DP	Heuristic
Acar et al. (2012)	✓		✓		✓					✓			MIQP	Heuristic
Zhou et al. (2022)	✓		✓	✓		✓						✓	MILP	LR, GrA
Nasiri et al. (2022)	✓		✓	✓		✓				✓			MILP	Exact
Buijs et al. (2016)	✓		✓				✓			✓		✓	DES	n/a
McWilliams et al. (2005)	✓		✓		✓					✓			Simulation-optimization	GA
Yang et al. (2010)				✓		✓				✓		✓	DES	n/a
Buijs et al. (2016)	✓		✓		✓		✓			✓	✓	✓	DES	n/a
Arnaout et al. (2010)					✓	✓				✓	✓	✓	DES	n/a
Soto Zuluaga et al. (2017)								✓		✓	✓	✓	MILP	Exact
This study				✓	✓	✓	✓	✓	✓	✓	✓	✓	DES	n/a

ALNS = Adaptive Local Neighborhood Search, GA = Genetic Algorithm, DP = Dynamic Programming, VNS = Variable Neighborhood Search, TS = Tabu Search, NSGA-II = Non-dominated sorting genetic algorithm II, MOSA = Multi-Objective Simulated Annealing, MODE = Multi-Objective Differential Evolutionary, C&CG = Column and Constraint Generation, LR = Lagrangian Relaxation, GrA = Greedy Algorithm, MILP = Mixed Integer Linear Programming, ILP = Integer Linear Programming, MIQP = Mixed Integer Quadratic Programming, DES = Discrete Event Simulation, n/a = not applicable.

and Alpan (2016b), which are frequently included in optimization problem objective functions. The key indicators for cross-docking performance include monitoring inventory levels, measuring makespan, total loading or unloading time, waiting times, door utilization, travel distance, total pallets stay time, and pallets not loaded. These indicators help to identify inefficiencies or congestion at doors and bottlenecks closely related to total loading/unloading time. To achieve the primary objective of cross-docking, which is to minimize inventory, it is important to minimize the average time products spend in the facility. The number of pallets not loaded measures missed orders, corresponding to lost profit, and is referred to as a 'back-log'. Moreover, cross-docking performance is mainly evaluated through time-based and inventory indicators.

Simulation and optimization methods can be used to model cross-docking operations, but the most suitable method depends on the problem. The question raises how staging-level design and different load carriers perform under external and internal uncertainty factors in cross-docking operations.

3. Methodology

3.1. Case study

To study the effects of different load carriers under external and internal uncertainty factors in cross-docking operations, a case study at the CDF of a Dutch 4PL provider is conducted. The terminal layout comprises staging areas with varying capacities, and the truck-to-door assignment is fixed based on historical volumes. However, truck scheduling optimization is not possible due to a lack of data monitoring and a well-functioning warehouse management system (WMS).

The cross-docking terminal also functions as a regional terminal, allowing injections that account for most of the demand volume. Therefore, it cannot be considered a pure cross-docking facility. The activities in the terminal involve receiving inbound trucks, unloading them, moving the pallets to the staging area, reloading outbound trucks, and dispatching them. Each dock door has its assigned equipment, and all pallets must be unloaded and stored in the corresponding staging area before reloading can begin due to the lack of a well-functioning WMS.

The resources used in cross-docking operations include terminal workers, electric pallet trucks (EPTs), manual pallet trucks (MPTs), and dock doors. Additionally, the staging area's

capacity determines the system's maximum throughput under the current policy of unloading all pallets before reloading.

Euro pallets can hold up to 1,500 KG, while roll containers can only bear a mere 500 KG. Hence, the number of parcels that can fit inside roll containers is limited to a third of what Euro pallets can carry, multiplying the number of load carriers needed by three. In terms of cost, Euro pallets are the cheaper option, with prices ranging from 15-21 euros per pallet compared to over 70 euros per roll container. Moreover, since pallets can hold more volume and are cheaper than roll containers, they can be considered more economical. However, it's important to note that electric pallet trucks (EPTs) are considerably more expensive than manual pallet trucks (MPTs), with prices ranging from 1,200-4,000 euros for EPTs compared to just over 300 euros for MPTs on average. Finally, the current average price per square meter of warehousing surface in the Netherlands is 60-73 euros.

The uncertainty of demand volumes and truck arrival times affects the current state of the cross-docking terminal. Varying internal transshipment time is caused due to the uncertainty in demand to be cross-docked. The load carrier type determines the mode of transport, with pallets being transported using EPTs or MPTs, and roll containers being manually moved by terminal personnel.

When zooming out on the cross-docking process, the high-level overview of activities in the CDF is presented by Figure 1. This high-level overview helps with later defining the model scenarios and configurations.

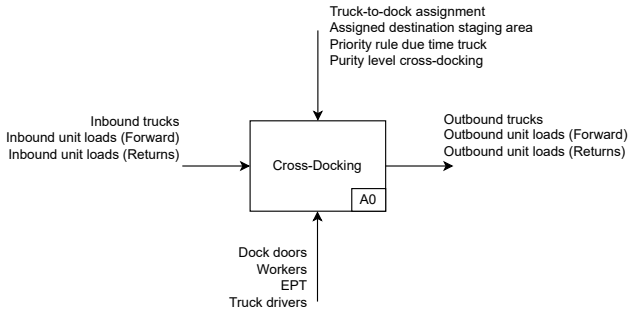


Figure 1: IDEF0: A-0 level

After analyzing the current system and having conducted an extensive literature review, the appropriate design solutions are determined. However, as we are dealing with an existing cross-docking facility, we cannot move to a completely new design or increase the number of dock doors due to limitations at the terminal under study. Therefore, only limited infrastructural changes can be made.

Although truck scheduling optimization and truck-to-dock assignment based on optimization are unsuitable for this problem due to the inability to monitor live updates on truck arrival times and expected volumes in inbound trucks, the truck-to-dock assignment can be performed based on either a "First In, First Out" (FIFO) policy or based on priority according to the truck's due time.

Research shows that using mixed-mode dock doors improves

cross-docking performance, and it can be concluded that operating with fewer mixed-mode dock doors will decrease performance but not significantly as long as more than 60% of the doors are in mixed-mode. Hence, researching the effect of using fewer mixed-mode dock doors will not be of interest.

Two potential design alternatives have been identified: implementing two-stage cross-docking and using roll containers as a consolidation method. The former can reduce congestion, while the latter allows for overtaking and passing due to the smaller dimensions of roll containers. Next, the two design alternatives are modeled and specified.

3.2. Simulation model

To establish the cross-docking system's model, we define the layout and processes involved. Trucks are received, unloaded, and sorted with EPTs. ULs are stored until all trucks are unloaded and reloading can commence. To evaluate performance, assess waiting times, makespan, and throughput. Uncertain factors include arrival times, demand volumes, and processing time. A logic flow diagram and IDEF0 diagram provide an overview of the process.

Unit Loads (UL) and trucks are modeled in this simulation model as entities, with data tables assigned to them storing properties. The most important properties of UL and trucks are presented in tables, and the Unit Load Sources and Truck Sources are studied based on their respective properties. At the initialization of the simulation, the UL and subsequent trucks are generated at the origin terminals, and the number of ULs generated per origin is calculated by multiplying the total number of ULs created with the probability of the origin. After UL and truck generation, both entities are loaded into the trucks and enter a time path leading to the cross-docking facility. The travel time is set to a random triangular function to represent an arrival time window for truck arrivals. Other arrival patterns are also possible and can be modeled by using other travel time distributions.

Upon arrival at the CDF, each truck is immediately marked with its Actual Time of Arrival (ATA) and assigned to a specific dock door. The truck and its attached unit loads are then designated for either single-stage or two-stage cross-docking, with docking including time for maneuvering and driver registration. It is important to note that each dock door can only serve one truck at a time, following a strict FIFO policy. Hence, a queue forms for trucks assigned to the same dock. The truck-to-dock assignment is fixed and determined by the regional terminal (RT) from which the truck originates. Depending on the assigned staging procedure, either single-stage or two-stage cross-docking is initiated.

The unloading process at the dock commences upon the arrival of a truck. Following the FIFO policy, it is crucial to note that only one truck can dock per door at a time. The last loaded unit on the truck is the first to be unloaded, using the last in, first out (LIFO) policy. The transfer-in time is the estimated time for docking and registering the truck driver, while the processing time involves moving the EPT inside the inbound truck.

To avoid obstructing transporters dropping off unit loads at the outbound staging area, they must place pallets in front of

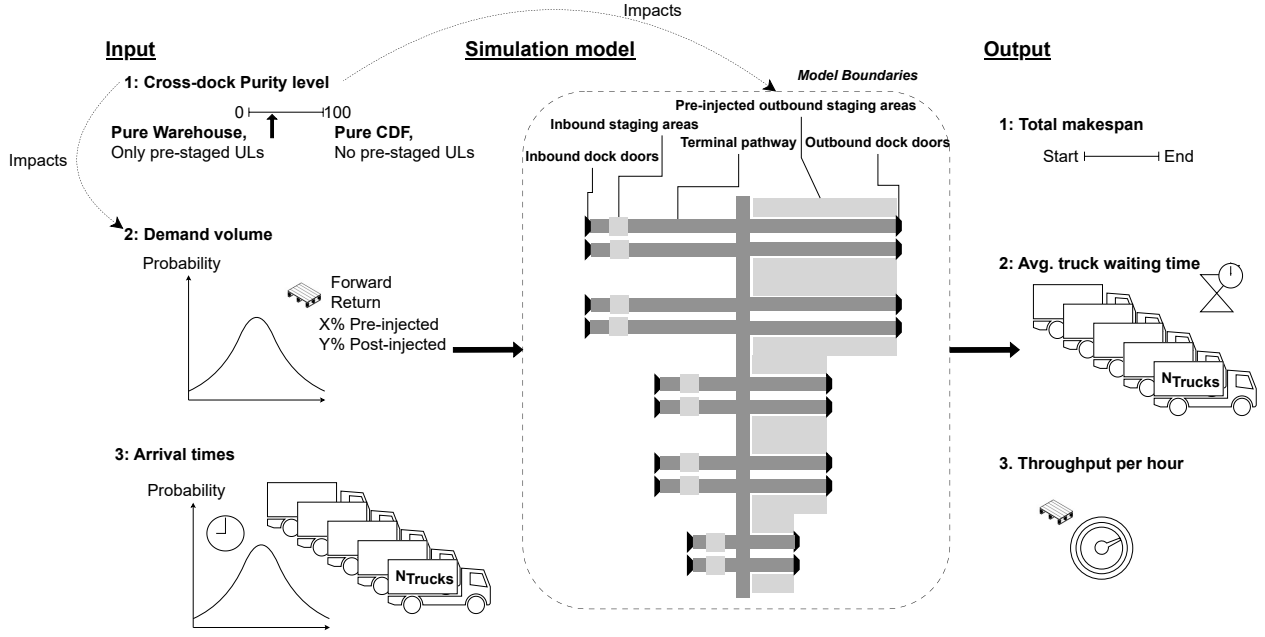


Figure 2: System scope and boundaries CDF

the unloading dock in two-stage cross-docking. The distances between the dock doors and the staging area for unit loads destined for the CDF are listed in data tables derived from the case study's floor plans and measurements. The unit loads are always moved along the terminal pathways.

After all the inbound trucks have been unloaded, the start time for reloading is noted, and the waiting time for each individual truck is recorded. The reloading of trucks takes place at the reloading dock objects, with a capacity of 1 truck at a time. Trucks are ranked by due time, with earlier-due trucks being served first. The batch quantity is either 33 pallets or 55 roll container ULs.

The UL entities are batched to the truck parent entities at the reloading dock object based on their matching destination. The transfer time for the truck entities includes dispatching registration and, if necessary, swapping the truck. Truck swapping time depends on the truck's due time and is checked before the transfer. The processing time for reloading is proportional to the number of unit loads batched.

Once a truck is loaded, it is dispatched, and the departure time is recorded. The departure time for the last departing truck is recorded to calculate the total makespan. The total throughput level is also recorded for comparison. Finally, the KPIs of the simulation are recorded for the simulation replication.

The computer model that incorporates the conceptual model has been successfully created using SIMIO, a Discrete Event Simulation (DES) object-oriented software. The primary focus in object-oriented modeling lies in the data associated with the objects, where the procedures play a secondary role (Constantine, 1989). For example, both the arriving trucks and ULs have relevant data such as their origin, destination, and volume, which determines the process. Hence, this object-oriented methodology is well-suited for the modeling problem at hand.

After conducting a thorough statistical analysis on the 'Throughput' and 'Total Makespan', including a Student's T-Test with Bootstrapping and chi-squared goodness of fit test, it has been determined with 95% confidence interval that there is no significant difference in means between the simulation-generated data and the empirical data. The 'Total Makespan' of the simulation data aligns well with the empirical data, and the 'Throughput' of the simulation data shows negligible differences from the empirical data. The histograms in Figure 3 reveal a significant overlap between the throughput and total makespan for the empirical and simulation data sets.

While the spread of the individual KPIs is not enough by itself, the spread of the combinatory observations for 'Total Makespan' and 'Throughput' is crucial. Therefore, the scatter plot presented in Figure 4 compares the empirical and simulation data and shows a visually good enough spread for the two metrics.

4. Results

4.1. Experimentation

For experimentation with the CDF's simulation model, scenarios are composed to test the selected design alternatives under different CDF configurations.

Constructing an experimental design is essential for testing various alternatives. A full-factorial experimentation design is effective in assessing the influence of input factors. Arrival time uncertainty can be high, medium, or low. The stochastic arrival time input parameter represents the confidence level of inbound trucks arriving simultaneously, and arrival times follow a random triangular distribution. Scenarios are constructed for the case study in which a 30% cross-docking purity level was found. Additionally, scenarios 10 to 18 are similar to scenarios

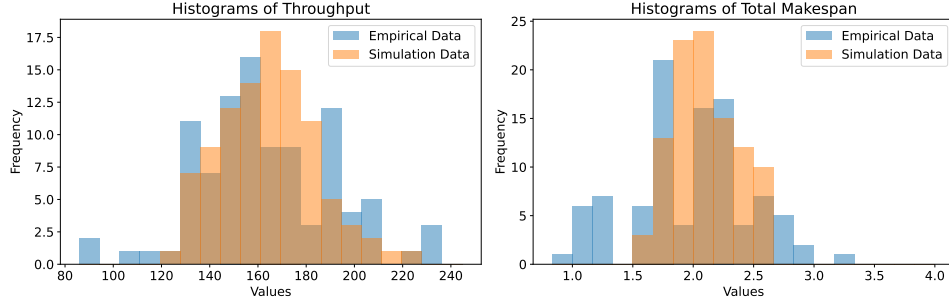


Figure 3: Verification using histograms of empirical and simulated data

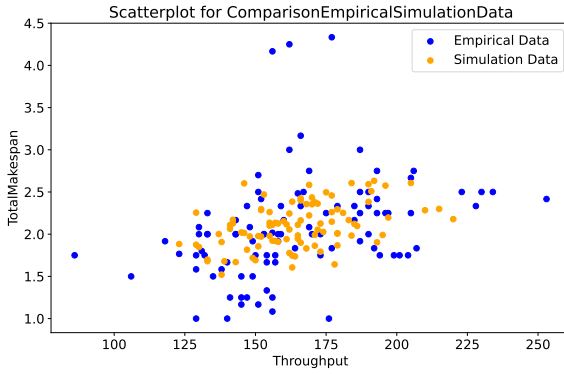


Figure 4: Scatter plot for assessing relative spread

1 to 9 but are constructed for pure cross-docking configurations (100% purity). The experimental plan is shown in Table 2.

Table 2: Full-factorial experimental design plan

Scenario	Demand	ATA	CD Purity	Scenario	Demand	ATA	CD Purity
1	High	Wide	30%	10	High	Wide	100%
2	Medium	Wide	30%	11	Medium	Wide	100%
3	Low	Wide	30%	12	Low	Wide	100%
4	High	Medium	30%	13	High	Medium	100%
5	Medium	Medium	30%	14	Medium	Medium	100%
6	Low	Medium	30%	15	Low	Medium	100%
7	High	Narrow	30%	16	High	Narrow	100%
8	Medium	Narrow	30%	18	Low	Narrow	100%
9	Low	Narrow	30%	17	Medium	Narrow	100%

To accurately analyze the behavior of the system, multiple replications are absolutely necessary due to unpredictable input variables. With cross-docking operations starting and ending with an empty system, there is no need for a warm-up period during these replications. It has been determined, through thorough verification, that a minimum of 100 replications per scenario are required to ensure results match the 95% confidence interval.

The three design alternatives consist of the referential state (Current State) and the two alternatives: Two-Stage Cross-Docking with Pallets and Two-Stage Cross-Docking with Roll Containers. Hence, the three system alternatives to be modeled are the following:

- **A0: Single-stage CD with pallets (Current State)** - This

alternative is the current state which imposes a single-stage cross-docking alternative with pallets as a consolidation method and EPTs as an internal transportation method. A0 is depicted in Figure .8.

UL Type = Pallets

Capacity Truck = 33

Transporter Type = EPTs

Staging level design = Single-Stage

- **A1: Two-Stage CD with Pallets** - This alternative is the first alternative. A1 imposes a two-stage cross-docking alternative with pallets as a consolidation method and EPTs as an internal transportation method. A1 is depicted in Figure .9.

UL Type = Pallets

Capacity Truck = 33

Transporter Type = EPTs

Staging level design = Two-Stage

- **A2: Two-Stage CD with Roll Containers** - This alternative is the second alternative that imposes a two-stage cross-docking alternative with roll containers as a consolidation method and Workers as an internal transportation method. A2 is depicted in Figure .10.

UL Type = Roll Containers

Capacity Truck = 55

Transporter Type = Workers

Staging level design = Two-Stage

The experiments are run using a computer with an Apple M1 processor, macOS 13.4 OS, and 8 GB RAM. After thoroughly analyzing all experiments, it is evident that some did not yield satisfactory data points. All experiments showed enough data points for the 30% cross-docking purity configuration. However, for the 100% cross-docking purity configurations, only A3 showed observations close to the expected number of replications. A0 and A1 experienced a significant drop in successful replications, which we attribute to high congestion levels. As noted by Sturrock (2015), SIMIO may encounter deadlock situations, but solutions such as roundabouts modeling can mitigate

these occurrences. Although minimal, a few deadlock situations still arose in some scenarios. While challenging to avoid deadlock situations altogether, the occurrence provides valuable insights into congestion changes. Therefore, the number of successful replications is vital to consider when interpreting the results.

To ensure reliable statistical inferences, scenarios with less than 30 replications are excluded from analysis using the Central Limit Theorem (Islam, 2018). This serves as the threshold value, as small sample sizes do not provide enough data to draw conclusions from with a high level of confidence. Scenarios with less than 30 replications are indicated in Table 3. The 30-50 successful replications per scenario indicate weaker validity of the results and a relatively high chance of congestion. Overall, the A3 alternative shows more resilience to congestion with almost all successful replications.

Table 3: Number of successful replications per scenario for the alternatives

Scenario	Demand	Input ATA window	CD purity level	No. replications		
				A0	A1	A2
1	High	Wide	30%	100	93	99
2	Med			100	99	99
3	Low			100	100	99
4	High	Medium	30%	100	97	99
5	Med			100	93	100
6	Low			100	100	99
7	High	Narrow	30%	100	91	99
8	Med			99	95	99
9	Low			100	96	100
10	High	Wide	100%	2	17	99
11	Med			74	32	100
12	Low			90	48	100
13	High	Medium	100%	69	21	100
14	Med			75	28	100
15	Low			84	35	100
16	High	Narrow	100%	75	24	100
17	Med			79	29	100
18	Low			83	41	100

A0 = Current State, A1 = Two-Stage Pallets, A2 = Two-Stage Roll Containers

Since A0 and A1 show less than the threshold for scenario 10, only A2 has valid results, and the alternatives in this scenario cannot be compared. Therefore, scenario 10 is discarded for further analysis. For scenarios 13, 14, 16, and 17, only a comparison can be made between A0 and A1. After discarding scenario 10, the resulting KPI averages of the simulation replications are obtained as presented in Table 4.

For non-pure CD configurations, according to the data in Table 5, A1 outperforms A0 and A2 with an average reduction of 22% in total makespan and an average increase of 28% in throughput. A2's total makespan is found to be insignificantly different from A0's. The Two-Stage (EPT) alternatives show a decrease in the Average Waiting Time Per Truck due to the reduction in total makespan and faster unloading and loading processes. However, A2's decreasing average waiting time indicates that roll containers are more time-consuming than pallets. It is uncertain at what demand level the system is capacitated, and the total throughput per hour decreases naturally as demand levels decrease. A1's total throughput per hour is

Table 4: Results

Scenario	KPI1 [hours]			KPI2 [hours]			KPI3 [UL/hour]			
	A0	A1	A2	A0	A1	A2	A0	A1	A2*	
Non-pure CD (30%)	1	3.13	2.39	3.02	1.04	0.69	0.64	80.87	105.48	86.08
	2	2.21	1.73	2.24	0.76	0.53	0.52	76.07	96.44	78.05
	3	1.39	1.16	1.38	0.50	0.40	0.39	63.93	75.95	66.57
	4	3.00	2.19	2.89	0.98	0.60	0.59	84.45	115.01	89.76
	5	2.06	1.56	2.06	0.69	0.44	0.44	81.53	107.06	84.67
	6	1.21	0.96	1.19	0.41	0.30	0.29	73.18	91.81	77.71
	7	2.93	2.22	2.83	0.95	0.61	0.56	86.33	112.94	91.86
	8	1.99	1.54	1.99	0.65	0.44	0.41	84.12	108.26	87.82
	9	1.12	0.90	1.11	0.37	0.27	0.26	78.84	98.04	83.32
Pure CD (100%)	11	4.73	2.53	5.26	1.08	0.89	1.02	61.33	115.70	58.41
	12	2.58	1.63	2.95	0.62	0.52	0.61	56.55	89.32	53.39
	13	4.05	3.89	9.42	1.44	1.37	1.65	107.76	111.10	47.00
	14	2.79	2.61	5.53	1.00	0.91	1.02	104.63	113.32	54.57
	15	1.70	1.54	2.88	0.55	0.47	0.57	85.65	94.22	55.27
	16	4.04	3.82	9.46	1.42	1.33	1.64	107.97	111.90	46.81
	17	2.73	2.59	5.85	0.97	0.90	1.04	107.15	113.14	51.35
	18	1.64	1.55	2.95	0.52	0.47	0.56	88.41	93.32	52.52

A0 = Current State, A1 = Two-Stage Pallets, A2 = Two-Stage Roll Containers

KPI1 = Total Makespan, KPI2 = Avg. Waiting Time Truck, KPI3 = Throughput level

Total run time = 68,787 seconds

*number of unit loads proportional to match load capacity pallets [$x \times \frac{33}{55}$]

Table 5: Change in KPIs the for non-pure cross-docking scenarios relative to A0 (the Base Case)

Scenario		KPI1			KPI2			KPI3		
		A0	A1	A2	A0	A1	A2	A0	A1	A2*
Non-pure CD (30%)	1	0%	-24%	-4%	0%	-34%	-38%	0%	30%	6%
	2	0%	-22%	2%	0%	-30%	-31%	0%	27%	3%
	3	0%	-16%	0%	0%	-20%	-22%	0%	19%	4%
	4	0%	-27%	-3%	0%	-38%	-40%	0%	36%	6%
	5	0%	-24%	0%	0%	-35%	-35%	0%	31%	4%
	6	0%	-20%	-2%	0%	-28%	-29%	0%	25%	6%
	7	0%	-24%	-4%	0%	-36%	-41%	0%	31%	6%
	8	0%	-23%	0%	0%	-33%	-37%	0%	29%	4%
	9	0%	-20%	-1%	0%	-27%	-31%	0%	24%	6%
Average			-22%	-1%		-31%	-34%		28%	5%
Pure CD (100%)	11	0%	-46%	11%	0%	-18%	-6%	0%	89%	-5%
	12	0%	-37%	14%	0%	-17%	-3%	0%	58%	-6%
	13	0%	-4%	132%	0%	-5%	15%	0%	3%	-56%
	14	0%	-6%	98%	0%	-9%	2%	0%	8%	-48%
	15	0%	-10%	69%	0%	-15%	4%	0%	10%	-35%
	16	0%	-5%	134%	0%	-7%	16%	0%	4%	-57%
	17	0%	-5%	115%	0%	-8%	7%	0%	6%	-52%
	18	0%	-6%	80%	0%	-10%	8%	0%	6%	-41%
Average			-15%	82%		-11%	5%		23%	-37%

A0 = Current State, A1 = Two-Stage Pallets, A2 = Two-Stage Roll Containers

KPI1 = Total Makespan, KPI2 = Avg. Waiting Time Truck, KPI3 = Throughput level

*number of unit loads proportional to match load capacity of pallets

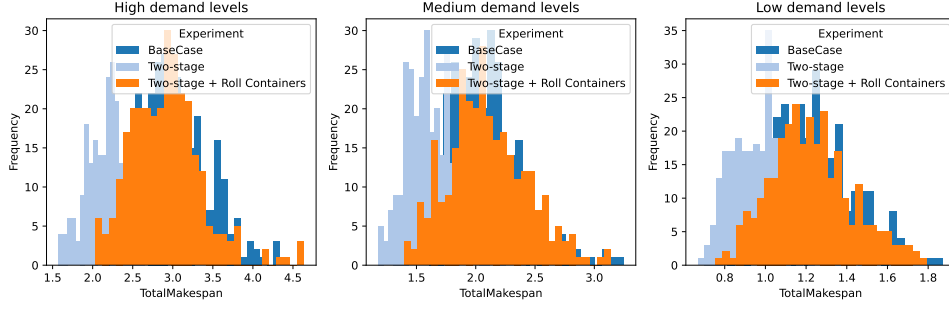


Figure 5: Histograms of total makespan for three experiments under varying demand levels

higher for Medium ATA Time Window scenarios than for High ATA Time Window and Narrow Time Window scenarios. A1's total makespan is lower for Medium ATA Time Window scenarios than for Wide and Narrow Time Window Scenarios. These findings confirm the literature's conclusions that the spread in arrival times enhances resource allocation and improves the CDF's performance.

In pure CD configurations, A1 reduces the total makespan by 15% on average, with the most significant reduction in wide ATA window scenarios 11 and 12. A2 leads to an 82% increase in total makespan on average. A1 also reduces the average waiting time per truck by 11%, while A2 increases it by 5%. A1 shows the most significant increase in throughput (58%) for wide ATA window scenarios 11 and 12. For scenarios 13 to 18, the throughput increases by less, ranging from 3% to 10%. Overall, A1 performs better than A0 and A2 regarding the total makespan in pure CD environments.

It is important to note that in all scenarios, A1 outperforms A0 and A2 in terms of total makespan. However, the total makespan is comparable between A0 and A2. After performing Pair-Wise Student's T-Tests, it was found that the difference in Total Makespan between A0 and A2 is not significant. Still, as the throughput levels increase, the significance of the differences between A0 and A2 decreases. This suggests that two-stage cross-docking with roll containers can become more efficient for higher demand levels.

Two-Stage cross-docking with roll containers results in lower waiting times compared to the base case and two-stage cross-docking with EPTs in terms of average waiting time. Although the total makespan of two-stage cross-docking with roll containers is not statistically different from the base case, the lower waiting times may be due to longer unloading and reloading times. Therefore, unloading and reloading roll containers is slower than loading and unloading pallets in the given inputs.

The primary focus of this analysis is on the behavior of the total makespan under different input parameters, such as demand, arrival times, and cross-docking purity, as makespan is the main KPI in cross-docking. The most important metrics are the total makespan and throughput per hour, while average truck waiting time is a less significant metric.

Figure 5 shows the histograms of total makespan in the non-pure cross-docking scenarios (Scenario 1 to 9).

The experimental design shows high, medium, and low de-

mand levels for Scenarios 1-9. The average makespan decreases as demand decreases. Throughput leads to increased benefits of two-stage cross-docking. Total cross-docking time variance increases as demand levels rise. Two-stage cross-docking with roll containers performs marginally better than the base case, but the difference is statistically insignificant for non-pure cross-docking scenarios. KPI Total Makespan for Arrival Times shows varying widths of arrival time windows and their total makespans are compared.

4.2. Discussion

This case study at a cross-docking terminal examined both non-pure and pure cross-docking configurations. For non-pure cross-docking, utilizing two-stage operations with pallets significantly improved performance under uncertainty, reducing total makespan by 22% and increasing hourly throughput by 28% compared to single-stage. Regarding pure configurations, two-stage cross-docking with pallets reduced makespan by 15% and increased throughput by 23% versus alternatives. However, pure cross-docking replications had a high failure rate, likely due to unresolved congestion issues.

Though this study focused on performance metrics and not costs, two-stage cross-docking would require double the space, increasing costs. But operational savings from a shorter makespan could offset this. Additionally, though roll containers enable flexibility for mixed loads, the 4.2 times higher capital costs may not balance cheaper operations. Further cost modeling is needed in future research.

While revealing clear improvements, the results are specific to the studied facility. Trends likely apply directionally to other cross-docks facing uncertainty, but expected benefits will vary based on factors like layout, product types, demand patterns, automation level, etc.

Several limitations provide avenues for future research. The model was limited to one case study, restricting generalization. Also, the low cross-docking purity level differs from pure configurations, limiting applicability. A framework classifying facilities by purity level could guide strategies. Failed replications indicated the model lacked congestion resolution procedures needed for realistic pure cross-docking. Comparing roll containers and pallets for sorting mixed pallets could provide insights into suitability for pure operations. Analyzing individual makespan stages could reveal process bottlenecks. Incorporating

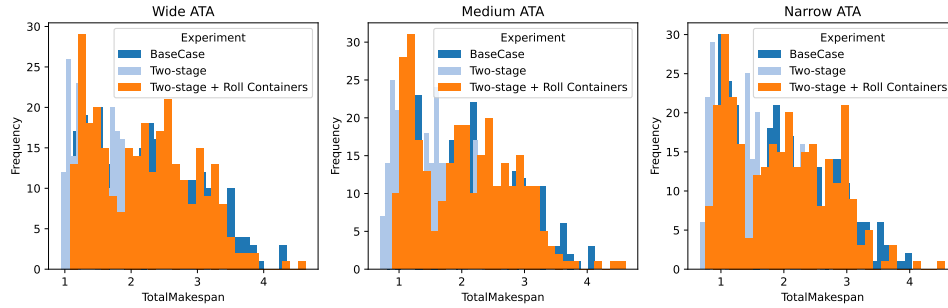


Figure 6: Histograms of total makespan for three experiments under varying width arrival time window

porating due times and scheduling optimization could improve realism. Finally, studying congestion and deadlock scenarios would aid congestion management.

While limited to a case study, these findings provide valuable insights for improving cross-docking operations facing uncertainty. Addressing limitations through congestion modeling, scheduling, cost analysis, and studying various facilities would strengthen the research. Overall, this study demonstrates the significant potential of two-stage cross-docking to enhance performance in uncertain conditions.

5. Conclusion

This study demonstrates that two-stage cross-docking with pallets can significantly improve performance under uncertainty for non-pure configurations, reducing makespan by 22% and increasing throughput by 28% versus single-stage. For pure cross-docking, two-stage pallets also showed gains over alternatives, but model congestion issues weakened these results.

The principles may apply directionally to other hybrid facilities, but exact improvements will depend on specific operations. Classifying facilities on a purity spectrum could aid the generalization of suitable practices.

Cost modeling was not performed, but qualitative estimates suggest roll containers have 4.2 times higher capital costs, which operational savings could offset. Actual costs will differ across sites.

As a single case study, the results are not broadly representative. Incorporating diverse facilities in future simulations would improve accuracy across scenarios. Comparing staging results to past literature highlights the influence of modeling assumptions.

Additional insights could come from analyzing individual process makespans and exploring complexities like mixed pallets, congestion dynamics, and scheduling. Surveying multiple sites could supplement the simulation.

Based on the findings, several recommendations for future research were identified. These include exploring the impact of roll containers on reducing makespan in scenarios involving mixed pallets, analyzing individual makespans within cross-docking operations, studying alternative consolidation methods under congestion situations, incorporating due times in cross-

docking studies, and developing a framework to measure cross-docking purity levels.

From a managerial perspective, the research provides insights for logistics service providers. It suggests considering staging level design to improve operational performance, selecting consolidation methods aligned with the level of cross-docking purity, actively monitoring and mitigating external and internal uncertainty factors, recognizing characteristics of hybrid warehouses in cross-docking operations, and leveraging simulation models to optimize design alternatives and evaluate improvements. Additionally, the importance of accurately modeling congestion and optimizing schedules or assignments is highlighted.

Overall, this research demonstrates the potential of two-stage cross-docking to mitigate uncertainty and provides a foundation for tailored follow-up across the purity spectrum. Findings guide implementation but require custom simulation and operational data before execution. With further validation, the observed trends can help optimize cross-docking practices facing uncertainty. Therefore, this research contributes to understanding cross-docking operations and offers practical recommendations for improving performance in real-world logistics settings.

References

- Acar, K., Yalcin, A., Yankov, D., 2012. Robust door assignment in less-than-truckload terminals. *Computers and Industrial Engineering* 63, 729–738. doi:10.1016/j.cie.2012.04.008.
- Amini, A., Tavakkoli-Moghaddam, R., 2016. A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks. *Computers and Industrial Engineering* 96, 180–191. doi:10.1016/j.cie.2016.03.023.
- Ardakani, A.A., Fei, J., 2020. A systematic literature review on uncertainties in cross-docking operations. *Modern Supply Chain Research and Applications* 2, 2–22. doi:10.1108/mscra-04-2019-0011.
- Arnaut, G., Rodriguez-Velasquez, E., Rabadi, G., Musa, R., 2010. Modeling Cross-Docking Operations using Discrete Event Simulation, in: *Proceedings of the 6th international workshop on enterprise & organizational modeling and simulation*, pp. 113–120.
- Bartholdi, J.J., Gue, K.R., Kang, K., 2001. Staging freight in a crossdock, in: *Proceedings of the International Conference on industrial engineering and production management*, Quebec City, Canada. p. 10.
- Bodnar, P., De Koster, R., Azadeh, K., 2017. Scheduling trucks in a cross-dock with mixed service mode dock doors. *Transportation Science* 51, 112–131. doi:10.1287/trsc.2015.0612.
- Boysen, N., Flidner, M., 2010. Cross dock scheduling: Classification, literature review and research agenda. doi:10.1016/j.omega.2009.10.008.

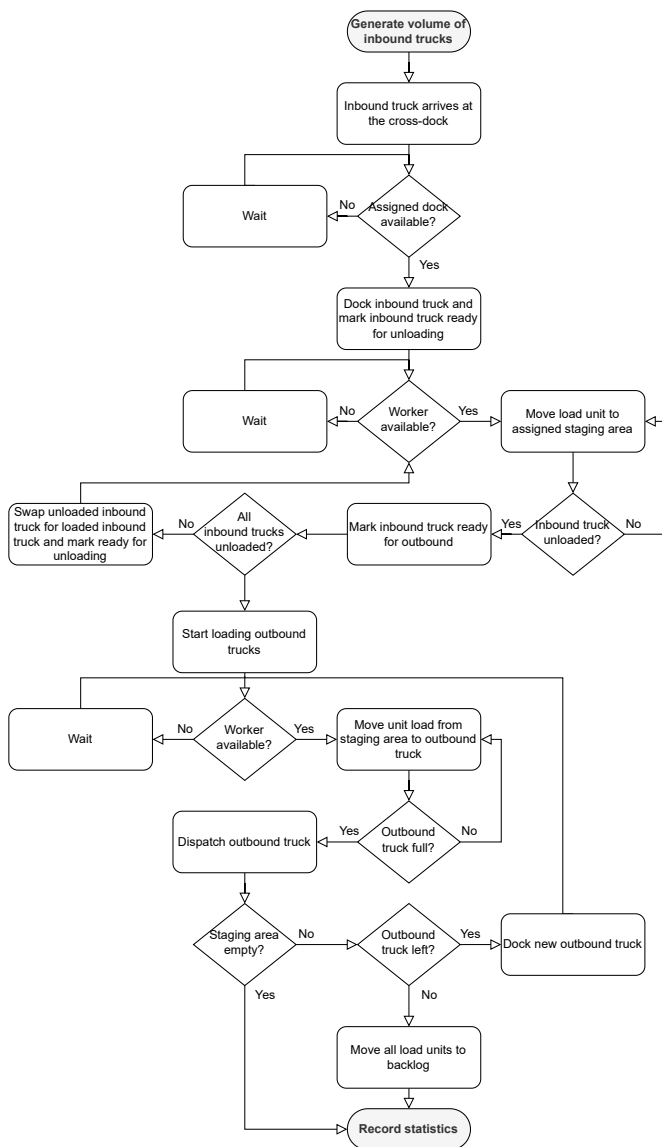


Figure .7: Logic flow diagram

- Buijs, P., Danhof, H.W., Wortmann, J.H.C., 2016. Just-in-Time Retail Distribution: A Systems Perspective on Cross-Docking. *Journal of Business Logistics* 37, 213–230. doi:10.1111/jbl.12135.
- Chevalier, S., 2022. Global retail e-commerce sales 2026. URL: <https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/>.
- Constantine, L.L., 1989. Object-oriented and structured methods: toward integration. *American Programmer* 2, 34–40.
- Gallo, A., Accorsi, R., Akkerman, R., Manzini, R., 2022. Scheduling cross-docking operations under uncertainty: A stochastic genetic algorithm based on scenarios tree. *EURO Journal on Transportation and Logistics* 11, 100095. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2192437622000206>, doi:10.1016/j.ejtl.2022.100095.
- Gue, K.R., Kang, K., 2001. STAGING QUEUES IN MATERIAL HANDLING AND TRANSPORTATION SYSTEMS, in: *Proceedings of the 2001 Winter Simulation Conference*, pp. 1104–1108.
- Islam, M.R., 2018. Sample size and its role in central limit theorem (clt). *Computational and Applied Mathematics Journal* 4, 1–7.
- Konur, D., Golias, M.M., 2013a. Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty. *Computers and Industrial Engineering* 65, 663–672. doi:10.1016/j.cie.2013.05.009.
- Konur, D., Golias, M.M., 2013b. Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach. *Transportation Research Part E: Logistics and Transportation Review* 49, 71–91. doi:10.1016/j.tre.2012.06.007.
- Kuo, Y., 2013. Optimizing truck sequencing and truck dock assignment in a cross docking system. *Expert Systems with Applications* 40, 5532–5541. doi:10.1016/j.eswa.2013.04.019.
- Ladier, A.L., Alpan, G., 2016a. Cross-docking operations: Current research versus industry practice. doi:10.1016/j.omega.2015.09.006.
- Ladier, A.L., Alpan, G., 2016b. Robust cross-dock scheduling with time windows. *Computers and Industrial Engineering* 99, 16–28. doi:10.1016/j.cie.2016.07.003.
- Larbi, R., Alpan, G., Penz, B., 2009. Scheduling transshipment operations in a multiple inbound and outbound doorcrossdock, in: *2009 International Conference on Computers and Industrial Engineering, CIE 2009, IEEE Computer Society*, pp. 227–232. doi:10.1109/iccie.2009.5223927.
- Mavi, R.K., Goh, M., Mavi, N.K., Jie, F., Brown, K., Biermann, S., Khanfar, A.A., 2020. Cross-docking: A systematic literature review. doi:10.3390/su12114789.
- McWilliams, D.L., Stanfield, P.M., Geiger, C.D., 2005. The parcel hub scheduling problem: A simulation-based solution approach. *Computers and Industrial Engineering* 49, 393–412. doi:10.1016/j.cie.2005.07.002.
- Nasiri, M.M., Ahmadi, N., Konur, D., Rahbari, A., 2022. A predictive-reactive cross-dock rescheduling system under truck arrival uncertainty. *Expert Systems with Applications* 188. doi:10.1016/j.eswa.2021.115986.
- Rahbari, A., Nasiri, M.M., Werner, F., Musavi, M.M., Jolai, F., 2019. The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models. *Applied Mathematical Modelling* 70, 605–625. doi:10.1016/j.apm.2019.01.047.
- Rahmanzadeh Tootkaleh, S., Fatemi Ghomi, S.M., Sheikh Sajadieh, M., 2016. Cross dock scheduling with fixed outbound trucks departure times under substitution condition. *Computers and Industrial Engineering* 92, 50–56. doi:10.1016/j.cie.2015.12.005.
- Serrano, C., Delorme, X., 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. doi:10.1016/j.ijpe.2017.09.008.
- Shakeri, M., Low, M.Y.H., Turner, S.J., Lee, E.W., 2016. An efficient incremental evaluation function for optimizing truck scheduling in a resource-constrained crossdock using metaheuristics. *Expert Systems with Applications* 45, 172–184. doi:10.1016/j.eswa.2015.09.041.
- Soto Zuluaga, J.P., Thiel, M., Colomé Perales, R., 2017. Reverse cross-docking. *Omega (United Kingdom)* 66, 48–57. doi:10.1016/j.omega.2016.01.010.
- Sturrock, D., 2015. Non-blocking path with workers in both directions? URL: <https://www.simio.com/simio-forum/topic/1812-non-blocking-path-with-workers-in-both-directions/>.
- Van Belle, J., Valckenaers, P., Cattrysse, D., 2012. Cross-docking: State of the art. doi:10.1016/j.omega.2012.01.005.
- Vis, I.F., Roodbergen, K.J., 2008. Positioning of goods in a cross-docking

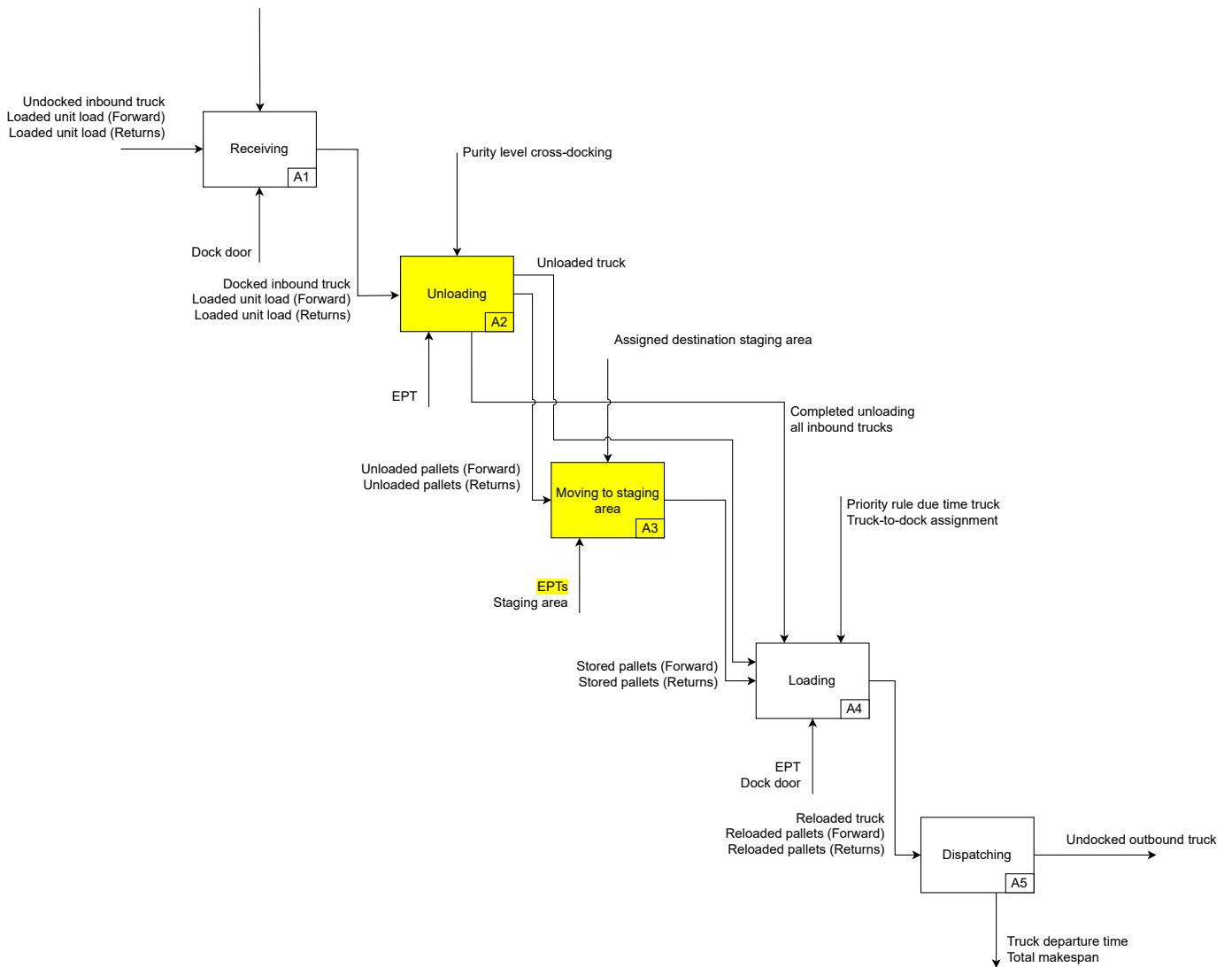


Figure .9: Alternative 1: Two-stage CD with pallets

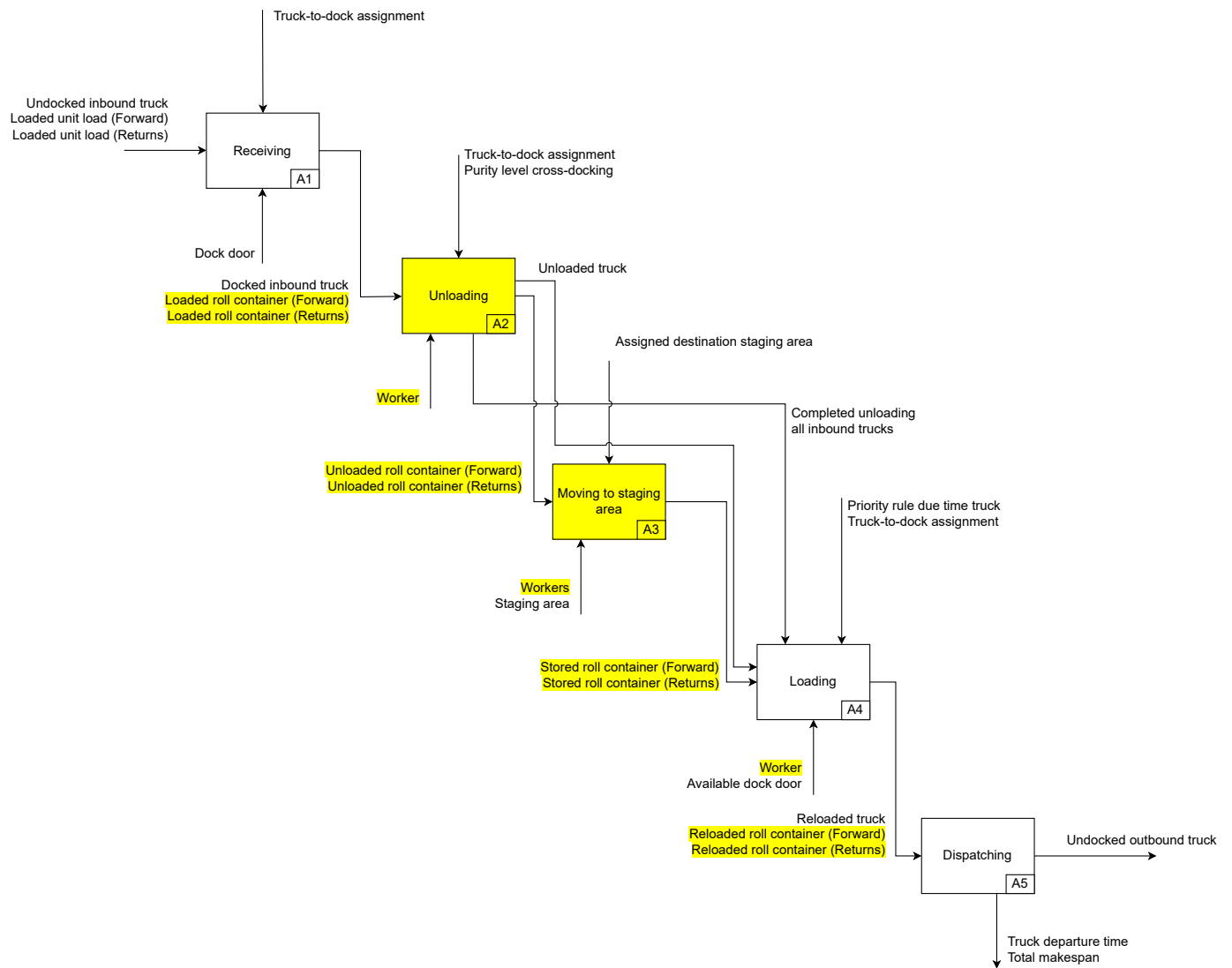


Figure .10: Alternative 2: Two-stage CD with roll containers

- environment. *Computers and Industrial Engineering* 54, 677–689. doi:10.1016/j.cie.2007.10.004.
- Xi, X., Changchun, L., Yuan, W., Loo Hay, L., 2020. Two-stage conflict robust optimization models for cross-dock truck scheduling problem under uncertainty. *Transportation Research Part E: Logistics and Transportation Review* 144. doi:10.1016/j.tre.2020.102123.
- Yang, K.K., Balakrishnan, J., Cheng, C.H., 2010. An Analysis of Factors Affecting Cross Docking Operations. *Journal of Business Logistics* 31, 121–148. doi:10.1002/j.2158-1592.2010.tb00131.x.
- Zhang, Y., Atasoy, B., Negenborn, R.R., 2022. Preference-Based Multi-Objective Optimization for Synchromodal Transport Using Adaptive Large Neighborhood Search, in: *Transportation Research Record*. SAGE Publications Ltd. volume 2676, pp. 71–87. doi:10.1177/03611981211049148.
- Zhou, B., Lei, Y., Zong, S., 2022. Lagrangian relaxation algorithm for the truck scheduling problem with products time window constraint in multi-door cross-dock. *Journal of Industrial and Management Optimization* 18, 4129–4149. doi:10.3934/jimo.2021151.