

FALKENBERG, SVEN FRANZ

THE USE OF NOVEL DATA AND OPTIMIZATION ALGORITHMS IN
WAREHOUSE MANAGEMENT

Dissertation
for obtaining the degree of Doctor of Business and Economics
(Doctor rerum politicarum - Dr. rer. pol.)

at WHU - Otto Beisheim School of Management

January 2 , 2023

First Advisor: Prof. Dr. Stefan Spinler

Second Advisor: Prof. Dr. Arne Strauss

Sven F. Falkenberg: *The use of novel data and optimization algorithms in warehouse management*,
© January 2 , 2023

*To my wife Laura
for her love and support.*

ABSTRACT

In 2020, approximately 151,000 warehouses were operating worldwide, with a total annual expenditure of €300 billion, representing roughly half of total supply chain costs. Optimized warehouse management may provide competitive advantages from both cost and customer service perspectives. One way to achieve these is to leverage the abundance of data collected in supply chains in combination with powerful algorithms.

This dissertation investigates how novel data sources and optimization algorithms, such as machine learning, can be applied in the context of warehouse advancement. In this research, we¹ analyze the warehouse environment from two perspectives. On the one hand, we examine two available resources in warehouses—equipment and employees—and explore how predicting breakdowns and productivity, respectively, may improve warehouse performance. On the other hand, we investigate whether the warehouse concept of cross-docking can be applied virtually to allow dynamic transfers between delivery vehicles.

In our first paper, we partner with one of the largest logistics service providers to examine how master, usage and sensor data on material handling equipment can be incorporated into a predictive maintenance model. Existing literature focuses on either time- or condition-based variables, whereas we show, in a comprehensive study of statistical learning methods, that both variable types can be included simultaneously. Our predictive maintenance model is able to capture the majority of breakdowns ($> 85\%$), while maintaining a low false-positive ratio ($< 7\%$).

In our second paper, we work with the same logistics service provider and apply Extreme Gradient Boosting to predict the productivity of new hires. We include operator, shift and product data to show that productivity can be predicted on an individual employee basis while substantially decreasing the forecasting error (50%), which translates into cost savings.

In our third paper, we look at dynamic and synchronized transshipments during direct deliveries. This concept uses transfers between vehicles, as carried out in cross-docking, but without the need for a physical warehouse. This reduces the proportion of empty return trips by increasing the proximity of vehicles to their location of origin. Our easy-to-implement multi-algorithm reduces the total distance by 15% on average compared with simple direct deliveries, and solves large problem instances within reasonable computational times.

This dissertation with its individual research contributions highlights how novel data sources and optimization algorithms can contribute to advancing warehouse management, and highlights the managerial implications of all three topics.

¹The term “we” refers to the authors of the respective chapters, as denoted at the beginning of each chapter.

ACKNOWLEDGMENTS

This dissertation was prepared at the Chair of Logistics Management at WHU - Otto Beisheim School of Management between 2019 and 2022.

First and foremost I thank my first advisor, Prof. Dr. Stefan Spinler, for providing me with the opportunity to complete my dissertation. His guidance on my initial research proposals, his networks connecting me with relevant organizations, expertise and knowledge to advance my research, and the freedom he granted me during difficult personal circumstances made this dissertation possible. When thinking back to these research years, I will always remember his dear and approachable personality.

Next, I thank my second advisor, Prof. Dr. Arne Strauss, for challenging my ideas, taking the time to offer in-depth knowledge, especially regarding my methodological approach, and improving my academic writing. I am grateful that he accepted my request when I was in search of a secondary advisor.

I would also like to thank those in the case study company for their kind support and dedication. Their provision of multiple datasets, which allowed me to apply my research to real cases, was invaluable for my dissertation.

Last but not least, I want to thank the backbone of my life: my parents, my friends and my wife. My parents, Sylvia and Michael Falkenberg provided me with the experiences and learning to become who I am. My friends, Adam Drensla, Philipp Sill and Andre Weckermann read what I put on paper, and Philipp, especially, advised me on mathematical questions. Our weekly reunions throughout the dissertation provided a great research-life balance.

My wife, Laura Falkenberg, supported me in my decisions, motivated me when I became discouraged, acquainted herself with my research to provide valuable ideas outside the box, and has made me enjoy life as I do with her.

CONTENTS

1	INTRODUCTION	1
1.1	The relevance of supply chain management	1
1.2	The role of warehouses in supply chains	1
1.3	Data and advanced analytics in supply chain management . . .	2
1.4	Contributions of this dissertation	3
2	PREDICTIVE MAINTENANCE FOR MATERIAL HANDLING EQUIP- MENT	5
2.1	Introduction	5
2.2	Related work and literature contribution	6
2.2.1	Preventive maintenance	6
2.2.2	Use of novel data sources and advanced algorithms . . .	7
2.2.3	Contribution to literature	8
2.3	From hypotheses to feature generation	8
2.3.1	Breakdown hypotheses	9
2.3.2	Primary data	10
2.3.3	Feature extraction	11
2.4	SLM-based breakdown predictions	12
2.4.1	Statistical learning models	12
2.4.2	Preprocessing, hyperparameter tuning and fitting	13
2.4.3	Baseline predictor	15
2.4.4	Insights from model comparison	15
2.5	Discussion and conclusion	22
2.5.1	Synthesis	22
2.5.2	Future research	23
3	PRODUCTIVITY PREDICTION OF WAREHOUSE EMPLOYEES	25
3.1	Introduction	25
3.2	Related work and literature contribution	26
3.3	Case introduction	26
3.3.1	Case partner and problem definition	26
3.3.2	Data overview	27
3.3.3	Baseline and simple predictor	32
3.4	Data preparation and method	33
3.4.1	Feature generation	33
3.4.2	Ensemble tree methods	35
3.4.3	Hyper-parameter tuning	36
3.5	Predicting new hires' productivity	37
3.5.1	Prediction results	37
3.5.2	Insights from variable analysis	39
3.5.3	Impact on staff planning	40
3.6	Discussion and conclusion	42
3.6.1	Synthesis	42

3.6.2	Future research	43
4	DYNAMIC TRANSFERS WITH PERFECT SYNCHRONIZATION	45
4.1	Introduction	45
4.2	Related work and literature contribution	47
4.3	Problem definition	49
4.3.1	Notation and cost evaluation	49
4.3.2	Problem complexity	50
4.3.3	Brute-force solution	51
4.4	Algorithm formulation	52
4.4.1	Sub-algorithm 1: Reducing the solution space	52
4.4.2	Sub-algorithm 2: Finding the optimal transshipment location	55
4.4.3	Sub-algorithm 3: Optimizing the combination of request pairs	57
4.5	Algorithm results	59
4.5.1	Numerical simulation	59
4.5.2	Food delivery case study	61
4.6	Discussion and conclusion	62
4.6.1	Synthesis	62
4.6.2	Future research	63
5	SUMMARY AND OUTLOOK	65
5.1	Synthesis, managerial implications and research contributions	65
5.2	Future research opportunities	68
A	APPENDIX TO CHAPTER 2	71
A.1	Data structure	71
A.2	Description of statistical learning methods	72
A.3	SLM hyperparameters	73
A.4	SLM prediction results for OEM2	74
A.5	Impact of implementing SLM	74
B	APPENDIX TO CHAPTER 3	75
B.1	EGB hyperparameter test values	75
C	APPENDIX TO CHAPTER 4	77
C.1	Complexity of request pairs and pair combinations	77
	BIBLIOGRAPHY	79

LIST OF FIGURES

Figure 2.1	Contractual relation between OEM and LSP	9
Figure 2.2	Primary data sources of Material Handling Equipment (MHE)	11
Figure 2.3	Feature correlation (OEM ₁)	14
Figure 2.4	Cost comparison for breakdown to maintenance cost ratio ($u_{TP} = 0$)	17
Figure 2.5	RFC metric scores for changing probability decision thresholds	18
Figure 2.6	Cost ratio and threshold dependent cost	18
Figure 2.7	RFC feature importance (OEM ₁)	19
Figure 3.1	Workforce planning	27
Figure 3.2	Productivity per month	29
Figure 3.3	Productivity per work day	29
Figure 3.4	Productivity with overtime, double-time and task switching	30
Figure 3.5	Picking and packing productivity in the first 40 days	30
Figure 3.6	Productivity by long-term stimulators	31
Figure 3.7	EGB permutation importance	39
Figure 3.8	Cost reduction potential of EGB	41
Figure 4.1	Total distance reduction through transfers	46
Figure 4.2	Example of transfer complexity with four requests	50
Figure 4.3	Exemplary request pairs with no improvement through transfers	52
Figure 4.4	Surrounding areas from the perspective of a request	53
Figure 4.5	All positions of request r' in relation to r	54
Figure 4.6	Transformation of transfer scenarios to determine relational positions	54
Figure 4.7	Costs of various transfer locations	56
Figure 4.8	Impact of multi-algorithm for each scenario	60

LIST OF TABLES

Table 2.1	Framework of MHE breakdown factors	9
Table 2.2	Feature extraction	12
Table 2.3	SLM prediction results (OEM1)*	16
Table 2.4	Confusion matrix	21
Table 2.5	Impact after SLM implementation	22
Table 3.1	Data overview	28
Table 3.2	Assumed ramp-up of new hires' productivity	32
Table 3.3	Framework of productivity impact factors	34
Table 3.4	Prediction results	38
Table 3.5	Recycling of hyperparameters	39
Table 3.6	Effect of impact category variables on EGB performance	40
Table 4.1	Research dealing with transfers of requests	48
Table 4.2	Numerical simulation results	60
Table 4.3	Algorithm results for food delivery data in Bordeaux . .	61

ABBREVIATIONS

BL	Bottom Left
BR	Bottom Right
CMA	Cumulative Moving Averages
DARP	Dial-a-Ride Problem
DARPT	Dial-a-Ride Problem with Transfers
EBA	Edmonds' Blossom Algorithm
EGB	Extreme Gradient Boosting
GB	Gradient Boosting
GNB	Gaussian Naïve Bayes
IO	Since Initial Operation
KKT	Karush-Kuhn-Tucker
KNN	K-Nearest-Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
LM	Since Last Maintenance
LR	Logistic Regression
LSP	Logistics Service Provider
L7D	Last 7 Days
L14D	Last 14 Days
MAE	Mean Absolute Error
MHE	Material Handling Equipment
ML	Middle Left
MLP	Multilayer Perceptron Neural Network
MR	Middle Right
NP	Naïve Predictor
OEM	Original Equipment Manufacturer
PCA	Principal Component Analysis
PDP	Pickup and Delivery Problem
PDPT	Pickup and Delivery Problem with Transfers
RF	Random Forest
RFC	Random Forest Classifier
RMSE	Root Mean Squared Error
RVM	Relevance Vector Machine
SGB	Stochastic Gradient Boosting

SLM	Statistical Learning Method
SLSQP	Sequential Least Squares Programming
SMA	Simple Moving Averages
SMOTE	Synthetic Minority Oversampling Technique
SVC	Support Vector Classifier
TL	Top Left
TR	Top Right
VRP	Vehicle Routing Problem

NOMENCLATURE

α	Angle between request vector and y-axis
α_t	Slope of function for optimal transfer location
BI_{t_i}	Breakdown indicator of MHE i at day t
$c(rp_{r,r'})$	Cost of a request pair without using a transfer location
$c(rp_{r,r'}, t_{r,r'})$	Cost of a request pair using a transfer location
C_{Base}	Base cost before SLM implementation
C_{BM}	Total cost after SLM implementation
c_M	Cost of a maintenance service
c_O	Cost of overestimating productivity
$C_{U,O}$	Cost of under and overestimating productivity
c_U	Cost of underestimating productivity
c_x	X-coordinate
c_y	Y-coordinate
CG	Coordinate grid
CMA	Cumulative Moving Average
$d(c, c')$	Euclidean distance between two coordinates
d_r	Delivery location for request r
D	Set of delivery locations
EF_{t_i}	Extracted features for MHE i at day t
E	Set of edges
$F1$	F1 Score
FN	False negatives
FP	False positives
G	Undirected graph
I_{EGB}	Impact of EGB model
I_{SLM}	Impact after SLM implementation
mc	Minority class
MD_i	Master data corresponding to MHE i
MHE_i	i^{th} Material Handling Equipment
M	Set of matchings between two vertices
m	Matching m
O_p	Overestimation of productivity
p_r	Pickup location for request r
p_t	Productivity at time t
pc_i	Request pair combination
PC	Set of pair combinations
pt	Probability threshold
P	Set of pickup locations
Q	Capacity constraint
r_{BM}	Ratio between breakdown and maintenance cost
$rp_{r,r'}$	Request pair consisting of request r and r'
RP	Set of requests pairs

R	Set of requests
r	Request r
SMA10	Simple Moving Average over 10 periods
SMA5	Simple Moving Average over 5 periods
s	Support probability of an over-productive employee
t_i	Observation day of i th MHE
t_{r,r'}	Transferlocation between request r and r'
TN	True negatives
TP	True positives
tw_r	Time window constraint for request r
TW	Set of time window constraints
T	Set of transfer locations
U_p	Underestimation of productivity
u_{TP}	Probability of breakdown despite a maintenance service
ud(0,1)	uniformed distribution between 0 and 1
UD_{t_i}	Usage Data of MHE i at day t
v_r	Vehicle responsible for request r
V	Set of vertices
wt_{v_r,v_{r'}}	Waiting time between v_r and $v_{r'}$

INTRODUCTION

1.1 THE RELEVANCE OF SUPPLY CHAIN MANAGEMENT

Supply chains can be described as a set of multiple entities that participate in supplying products, providing services, transferring finances and/or exchanging information both, up- and downstream. Supply chain management describes supply chain entities' tactics and strategies to improve their own long-term performance and that of the entire supply chain (Mentzer et al. (2001)).

Although the concept of supply chains was described early in the 20th century (e.g., Shaw (1915)), it has been part of human history for much longer. Examples include the organization of wars, trading between eastern and western continents along the silk route, and the flourishing sea-trade in ancient Southeast Asia and South India (Kumar (2001)).

Nowadays, supply chain management has become more than a simple necessity for businesses to operate, as optimizing supply chains' design and operation offers a competitive advantage (Amari et al. (2006)). In particular, the trend towards e-commerce requires supply chain strategies to be adapted to improve customer experience and, even more importantly, further reduce costs (Gartner (2021)).

Despite offering considerable upside potential through competitive advantage, supply chains also face immense downside risk if they cease to function properly, as shown by past and recent crises around the globe. The supply chain disruptions caused by the great earthquake in east Japan in 2011 are estimated to have led to a 0.47 percentage points decline in Japan's real GDP (Carvalho et al. (2021)). The more recent COVID-19 pandemic, resulting in economic shut-downs, caused shocks and disruptions to both supply and demand, and exposed the supply chain vulnerabilities of firms almost everywhere (Shih (2020)). The relevance of supply chains to mitigating the observed risk is reflected in a multitude of studies on forecasting (e.g., Nikolopoulos et al. (2021)) and the emerging topic of supply chain resilience (Ribeiro and Barbosa-Povoa (2018)). Although these crises have led to a rethinking of the role of supply chain management, some aspects will remain unchanged, such as customers' demand for low prices, even if production costs rise as a result of near-shoring suppliers (Shih (2020)).

1.2 THE ROLE OF WAREHOUSES IN SUPPLY CHAINS

Warehouses are an essential component of any supply chain. In 2020, there were approximately 151,000 warehouses globally, and this number is expected to reach 180,000 by 2025 (Statista (2021a)). The operational costs of warehouses worldwide amount to roughly €300 billion per annum, and these will continue to grow with the increasing complexity of globalized supply chains and the

prevalence of e-commerce (McKinsey (2019)). As the total warehouse cost accounts for roughly half of supply chain costs (Statista (2021b)), they offer great leverage in providing a competitive cost advantage. Given their critical intermediate role within supply chains, they not only provide a cost advantage, but also impact on service levels (Kiefer and Novack (1999)) by addressing customer expectations efficiently (Gu et al. (2007)), especially with the growth of e-commerce and changing consumer habits towards instant gratification (Dablanc et al. (2017)).

The main processes within a warehouse are inbound, storing, picking, packing and outbound (De Koster et al. (2007)). These are often carried out manually and can be described as follows (Gu et al. (2007)). In the inbound area, goods transported by a carrier arrive at the receiving dock and are unloaded. Before being accepted, the goods usually undergo some kind of quality inspection. In warehouses that only fulfill the function of a cross-dock, goods from various inbound carriers are consolidated into new shipments and immediately transferred to the outbound area. Traditional inventory-holding warehouses put the newly arrived goods into storage, which thus operates as a decoupling point between supply and demand. Once demand materializes, the goods are picked, mainly manually, from the storage area and sent to the packing area. The packing function then packs and labels the (consolidated) goods to make them ready for shipment. In the outbound area, the goods are then loaded onto carriers for final shipment. From a process perspective, the greatest complexity lies between picking, packing and outbound, as these need to be synchronized to operate efficiently. From a resource perspective, the executing entities (i.e., labor and equipment) must be allocated to the right warehouse function in the right quantity and at the right time.

1.3 DATA AND ADVANCED ANALYTICS IN SUPPLY CHAIN MANAGEMENT

Until recently, data posed a problem, not in terms of quality or scarcity, but because the skyrocketing volumes that emerged in the early 2000s could not be processed by computational power available at that time. With advancing technology, companies started to collect copious data, planning to extract insights that might provide some kind of competitive advantage (Russom et al. (2011)). Businesses' aim to leverage data is clearly not new, but the abundance of available data for decision-making marks a new area (Nikolopoulos et al. (2021)).

Today, one of the most pressing concerns of CEOs is to devise a road-map for the digital age (Hjartar et al. (2019)). Data, advanced analytics and machine learning are integral to digital roadmaps, and thus to companies' advancement (Kraus et al. (2020), Hastie et al. (2009)). Since many of these concepts are relatively new, companies are reluctant to make bold investment decisions, as success rates and value added are uncertain (Lavalle et al. (2011)).

An obvious area in which to start is the supply chain, for two reasons. First, the prerequisite of available and useful data is often met, as supply chains store large amounts of (transactional) data in a standardized way. Second, companies rely on supply chains, and especially warehouses, to fulfill more demanding

customer expectations, such as shorter lead times or better availability (Tiwari et al. (2017), Gu et al. (2007)). This is supported by multiple surveys by Gartner. These show that 50% of organizations plan to invest in digitizing their supply chains through advanced analytics and machine learning, which are deemed to be extremely important by 69% and 66% of respondents, respectively (Gartner (2022)). Although only 1% of supply chain leaders state that they have a digital ecosystem today, 23% aim to have one by 2025 (Gartner (2021)).

Fortunately, a strong body of research has recently developed exploring the incorporation of large and novel data sources into the supply chain decision-making process. Not only computational power, but also new methods in advanced analytics and machine learning are applied to handle the new data challenges. The topics cover the entire value chain, from initial investment, to demand predictions, warehouse operations and product returns. Using financial market data, Kim et al. (2019) apply deep learning methods to predict risky retail investors. Leveraging customer account details, Höppner et al. (2018) utilize decision trees to predict churn behavior. To predict customer purchases, Martínez et al. (2018) use time and value information from previous purchases in combination with gradient boosting, Cui et al. (2017) use social media information, and Steinker et al. (2017) use data from weather stations. In a warehouse environment, Büsch et al. (2017) test a neural network, support vector machine and decision tree to classify data. This enables them to determine the best storage place for data to optimize different access costs and times in the warehouse management system. Tufano et al. (2021) train classifier algorithms to improve warehouse design by predicting the best storage technology, material handling system, storage allocation strategy and picking policy. Product return volumes are predicted by Cui et al. (2019), using product characteristics with the Least Absolute Shrinkage and Selection Operator (LASSO).

Compared with the demand literature, which has a rich track record of applying data and machine-learning algorithms, the warehouse literature on the same topic is as yet sparse.

1.4 CONTRIBUTIONS OF THIS DISSERTATION

In the next three chapters of this dissertation, we¹ present individual research projects that contribute to the current literature.

In Chapter 2, we apply a comprehensive set of machine-learning algorithms to build a predictive maintenance model for Material Handling Equipment (MHE) which incorporates novel data sources to forecast breakdowns. In Chapter 3, we propose a prediction model for the productivity of newly hired warehouse employees using Extreme Gradient Boosting (EGB). In Chapter 4, we analyze the disruptive idea of virtual cross-docks that allow dynamic transfers between vehicles, and present a multi heuristic to solve this problem. These three chapters evolve as follows.

- In Chapter 2, we build a predictive maintenance model for material handling equipment that incorporates novel data sources to forecast break-

¹The term “we” refers to the authors of the respective chapters as noted at the beginning of each chapter.

downs. To this end, we develop a framework to structure and extract relevant predictor variables. We then conduct a comprehensive study of statistical learning methods for failure detection. We show that standard sensors in material handling equipment provide sufficient data to predict the majority of breakdowns ($> 85\%$). Our findings are confirmed using two independent datasets, and are thus transferable. We also provide a cost-based evaluation of these statistical learning methods, and find that K-Nearest-Neighbors (KNN) and Random Forest Classifiers (RFCs) are cost-optimal. While most current literature focuses on either time- or condition-based maintenance, we suggest a more robust approach. We demonstrate that time and condition are almost equally important. As a result, we present a prediction model that incorporates both variable types. From a managerial perspective, we provide recommendations on data collection, and highlight the importance of understanding the cost ratio between breakdowns and preventive maintenance services.

- In Chapter 3, we build a prediction model for the productivity of newly hired warehouse employees incorporating multiple data sources to support workforce planning. To this end, we develop a framework to identify relevant variables in four categories: warehouse, operator, shift and product. We show that using EGB with these variables can reduce the Root Mean Squared Error (RMSE) of predictions by more than 50%. A scenario analysis shows that improved productivity predictions translate into substantial cost savings.
- In Chapter 4, we look at direct deliveries to meet consumer demand for instant gratification. Delivery vehicles often drive empty when returning to their origin. Dynamic transshipments with perfect synchronization would allow loads to be transferred between these vehicles without the need for storage facilities. This would increase their proximity to their origin and reduce the proportion of empty rides. We present a problem-specific combination of three heuristics to solve this concept within a reasonable computational time. The first evaluates the relational position between two deliveries and, based on 36 structured cases, decides which pairs to exclude from the solution space. The second formulates the optimization problem to find the best transfer location while taking account of time-window constraints. Finally, we apply the request-pair combination problem to graph theory to determine the lowest overall distance. In a numerical simulation, we show that our heuristic reduces the initial distance of direct deliveries by 15%, and confirm the findings in a case study. We also demonstrate that the total distance is best reduced by relaxing time windows, rather than adding requests to increase the chance of transshipment. Our heuristic not only helps to reduce costs, but also improves sustainability, mitigates driver shortages and is easy to implement.

In the final chapter of this dissertation, we synthesize our three research projects, highlight their managerial implications, and suggest opportunities for future research.

THE ROLE OF NOVEL DATA IN MAINTENANCE PLANNING: BREAKDOWN PREDICTIONS FOR MATERIAL HANDLING EQUIPMENT

The following chapter is based on Falkenberg and Spinler (2022b).¹

2.1 INTRODUCTION

A digital strategy is one of the major topics on the agenda of most CEOs (Hjartar et al. (2019)). As part of that digital strategy, advanced analytics, machine learning or even artificial intelligence play an important role in the advancement of companies (Kraus et al. (2020), Hastie et al. (2009)). However, companies are hesitant to adapt as little is known about how these concepts are successfully employed or what their value add is (Lavallo et al. (2011)). An area with great potential for digital tools is supply chain management for the following two reasons. First, supply chains usually offer sufficient data availability, which is a necessity for most algorithms to work. Second, market competition requires supply chains, especially warehouses, to advance in order to fulfill increasing customer expectations like shorter lead-times or better availability (Gu et al. (2007)).

In warehouses, most time is spent on storage and picking. Both tasks are usually performed manually (Gu et al. (2007)). Hence, a common performance metric is the time spent per operation, e.g., orders picked per hour (Staudt et al. (2015a)). Reducing the time needed per warehouse operation shortens lead times and increases productivity. As cost per labor increases and availability of personnel decreases, the importance of automation through Material Handling Equipment (MHE) like forklifts or automated guided vehicles becomes more important (Mercier and Castro (2018)). The consequence is that poor functioning MHE may lead to a substantial loss in productivity (Srinivasan et al. (1994)). It also affects manufacturers of MHE. They are not only forced to deliver high quality products but also to provide after-sales service to deal with fierce competition (Wang et al. (2019)). It is therefore important to reduce breakdowns to avoid repair costs and productivity reduction (Valdez-Flores and Feldman (1989), McCall (1965)).

Most of the extensive literature on preventive maintenance is limited by the assumptions of the underlying data or by the scope of data sources it leverages (McCall (1965), Fouladirad et al. (2018)). The first limitation materializes as most preventive maintenance policies assume either time-based or condition-based factors to drive breakdowns. Additionally, the time to failure is often assumed to follow specific distributions (Ahmad and Kamaruddin (2012)). The second limitation is observed by the absence of research on novel data and

¹*This paper with the title "The role of novel data in maintenance planning: Breakdown predictions for material handling equipment" was written by Sven Falkenberg and Stefan Spinler, and has been published in the journal "Computers & Industrial Engineering".*

its impact on maintenance policies. Only few studies leverage sensor data and advanced algorithms to cover complex inter-dependencies in maintenance planning. The few existing ones are limited to one or two algorithms (e.g., Heng et al. (2009), Herzog et al. (2009), Tran et al. (2009), Caesarendra et al. (2010)).

In our research we shed light on the long established separation of time and condition-based maintenance, and whether a combination should receive greater attention in future research. Furthermore, we answer the question as to whether novel data that is available nowadays plays a vital role in improving preventive maintenance. We also investigate if an integration of various variables can be achieved through Statistical Learning Methods (SLMs) such as machine learning algorithms.

We show on two independent datasets from different Original Equipment Manufacturers (OEMs) that it is indeed crucial to combine time and condition-based maintenance. Our findings indicate that they are almost equally important for breakdown predictions. To circumvent the assumptions of most maintenance policies that data needs to follow a specific distribution, we show that SLMs make reliable predictions of MHE breakdowns, even if the distribution of failure times is not known. We further present a comprehensive study of SLMs and prove from a cost perspective, that depending on the cost structure either a Random Forest Classifier (RFC) or K-Nearest-Neighbors (KNN) is optimal. We subsequently provide recommendations for practitioners on data collection and feature extraction to improve prediction quality.

The remainder of this chapter is structured as follows. In Section 2.2 we review literature on preventive maintenance and the use of novel data, and point out our contribution to both fields. In Section 2.3 we introduce the dataset, develop a hypothesis-driven framework to understand breakdown factors of MHE, and extract independent variables based on that framework. In Section 2.4 we introduce the SLMs of our comparative study and the necessary preprocessing. We provide a detailed discussion on prediction results, insights gained and financial impact. Finally, in Section 2.5 we synthesize our contributions and present an overview of future research opportunities.

2.2 RELATED WORK AND LITERATURE CONTRIBUTION

Our research adds to two literature streams. First, the extensively researched field of preventive maintenance and secondly the use of novel data and its application through SLMs. We provide a short overview of previous literature in both streams and how our research contributes to them.

2.2.1 *Preventive maintenance*

Preventive maintenance is the process of conducting improvement activities to a system with the objective of increasing its availability (Usher et al. (1998)). Although unplanned equipment downtime can be dealt with by contractual penalties (Qin et al. (2019)), they are usually approached via preventive maintenance policies. Examples are simple surveillance plans (Derman and Solomon (1958)), experience based maintenance (Ahmad and Kamaruddin (2012)), time-

based maintenance (Sasieni (1956)) or condition-based maintenance (Baker and Christer (1994)). The two prominent literature streams focus on the latter two and have been consistently reviewed (Wang (2002), de Jonge and Scarf (2019)). Those policies deal with maintaining products of varying complexity, ranging from single parts (Barlow and Hunter (1960)) to sophisticated systems such as aircraft (Deng et al. (2019)). Policies often make assumptions about the system's condition, when that condition is observed, what condition levels exist and how they are improved. In most cases, systems are labeled as operative or inoperative (Derman (1963)). The inoperative state might be immediately detected (McCall (1965)), only when inspected (McCall (1963)) or not at all (Jorgenson and McCall (1963)). Once the deteriorated condition is observed, it can be improved through various maintenance services. Several studies look into the effect of minimal, perfect and other maintenance services (Klein (1962), Boland and Proschan (1982), Cl  roux et al. (1979), Nakagawa and Kowada (1983)). Thereby, imperfect repairs are assumed to be more realistic than perfect repairs (Mercier and Castro (2018)). Research on both streams, time and condition-based maintenance, are still relevant as recent studies by Cavalcante et al. (2018), Liu et al. (2019b) and Omshi et al. (2019) show.

The application of preventive maintenance policies can be difficult due to the required presence of a specific distribution of failure times (McCall (1965)). They also often prove relevant from a theoretical point of view rather than a practical one (Weiss (1962)). Less complex policies therefore benefit from ease of implementation and analytical simplicity (Manne (1960)). Consequently, they often assume that either time or condition-based factors determine the time to failure although it seems plausible that both contribute to machine breakdowns (Ahmad and Kamaruddin (2012)). It is clear that the equipment's environment affects it in multiple, complex ways which cannot be described by a single process (Shen et al. (2018)). Thus, advanced methodologies are required to appropriately address these complex environments (Linn  sson et al. (2019)). This becomes especially apparent when the system under investigation contains electrical parts (Ahmad and Kamaruddin (2012)) like MHE.

2.2.2 *Use of novel data sources and advanced algorithms*

The use of novel data has been the subject of multiple recent studies in operations where SLMs were used to incorporate data to improve predictions. The topics cover the entire value chain from initial investment, demand predictions to the return of products. Based on financial market data, Kim et al. (2019) use deep learning methods to predict risky retail investors. H  ppner et al. (2018) show how decision trees are used to predict churn behavior, using account details from customers. In predicting customer purchases, Mart  nez et al. (2018) leverage gradient boosting harnessing time and value information of previous purchases, whereas Cui et al. (2017) use social media information. Data from weather stations are used to improve the sales forecasts in e-commerce (Steinker et al. (2017)). Cui et al. (2019) apply Least Absolute Shrinkage and Selection Operator (LASSO) for predictions of product return volumes using product characteristics.

In the recent past, preventive maintenance evolved with the emergence of advanced prediction methods, and novel data found its way into maintenance literature (Ahmad and Kamaruddin (2012), Carvalho et al. (2019), Zonta et al. (2020)). In contrast to McCall (1965), advanced methodologies now handle large data inputs and dependencies in preventive maintenance without becoming analytically too complex or costly to implement. Herzog et al. (2009) estimated the residual life of components and machines based on neural networks. Heng et al. (2009) looked into how feed forward neural networks help to predict machine reliability. Regression trees were used for machine condition prognosis by Tran et al. (2009). Caesarendra et al. (2010) assessed the degradation of machines via relevance vector machines and logistic regressions. All of them restricted their studies to one or two algorithms instead of conducting a broad comparability study.

2.2.3 *Contribution to literature*

We contribute to both fields, preventive maintenance and the use of novel data. We demonstrate that time and condition-based maintenance are almost equally important factors driving breakdowns, and it is therefore essential that they are considered simultaneously in maintenance planning. For that task we show that SLMs exist which are capable of incorporating time and condition-based data without prior knowledge about the distribution of failure times. Our comprehensive overview of SLMs provides insights as to why some SLMs should be chosen over others. We further advance current literature on the use of novel data by leveraging equipment features, usage details and sensor data, and point out their relative importance. From there we derive guidance for practitioners in steering the data collection process based on a framework we developed.

2.3 FROM HYPOTHESES TO FEATURE GENERATION

Our research is motivated by a reliability problem of MHE in warehouses of a Logistics Service Provider (LSP). An OEM produces a broad range of forklifts and leases the MHE to the LSP. A service contract between the LSP and OEM defines the quantity of MHE, lengths of lease and a guaranteed uptime. To ensure the latter, the OEM conducts regular maintenance services, either as part of a preventative maintenance schedule or if an MHE breaks down. Dependent on fleet size, this is done via field service or by on-site technicians. The relation between OEM and MHE is shown in Figure 2.1. Despite maintenance services, MHE breaks down and often incurs high repair cost of approximately €1,000-1,500 per annum. Both parties, LSP and OEM, have an interest in reducing the number and length of breakdowns. The advantages for the LSP are twofold: First, a more stable warehouse operation is achieved which eventually drives warehouse performance. Second, cost is reduced as the LSP downsizes its leased fleet to achieve the same expected overall availability. The OEM on the other hand predicts when maintenance services are required and thus avoids incurring more expensive breakdown costs. Furthermore, cost effective field service

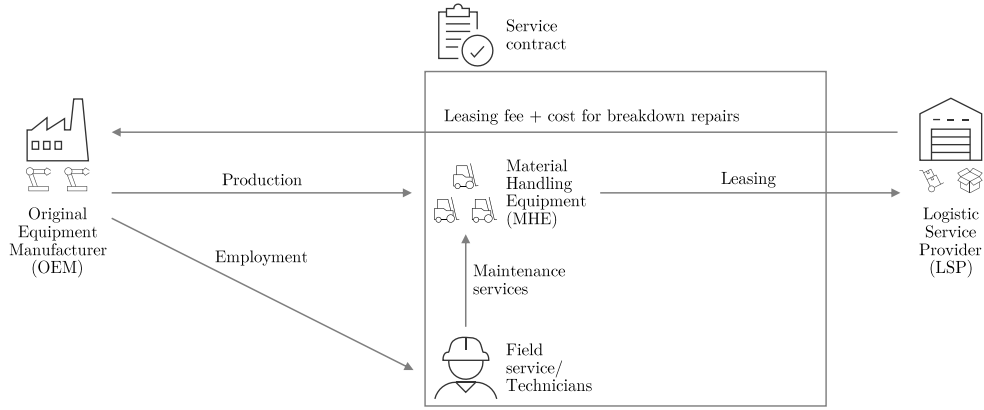


Figure 2.1: Contractual relation between OEM and LSP

is scheduled instead of costly on-site technicians. From a service perspective, the increased reliability also provides a competitive advantage.

In this section we first develop a hypothesis-driven framework to understand breakdown factors of MHE. Second, we provide an overview of the raw data available, and third, we extract independent variables by applying the framework to the raw data.

2.3.1 Breakdown hypotheses

An MHE, like any other machine, interacts with its environment. Multiple interfaces to that environment exist, e.g., the interaction between MHE and operator or between MHE and warehouse building. Those interfaces affect the performance of MHE and add to its deterioration and eventually to its failure. To the best of our knowledge no framework exists which provides a comprehensive overview of these interfaces. Therefore, we provide one (Table 2.1) with no pretense of exhaustiveness but the purpose of gaining an understanding of the possible impact factors. The framework also provides guidance as to which data is helpful for breakdown predictions.

Table 2.1: Framework of MHE breakdown factors

Environment interface	Interface dimensions	Hypothesis
What MHE is being used?	OEM*	Manufacturers differ in built quality
	MHE model*	MHE types differ in built quality
	Manufacturing year*	Older MHE break more often
	Dimensions (e.g., height)**	MHE of larger measurements are prone to collisions
	Performance (e.g., lifting capacity)*	MHE with greater workload wear down quicker
Where is the MHE operated?	Engine*	Some battery types create short circuits
	Warehouse location**	Temperature or humidity of location might affect electronics
Who uses the MHE?	Warehouse design (e.g., aisles width)**	Aisle design changes navigation difficulty and might lead to accidents
	Employee tenure**	Higher tenure leads to a better handling of MHE
	Employee qualifications**	Better qualifications lead to a better handling of MHE
When is the MHE operated?	Timing of the day (shifts)*	Reduced concentration during night shift
	Occurrences of operation*	Increase number of operations affect wear down
	Length of operation*	Short or long usages are adverse for MHE (rushes or inattentiveness)
How is the MHE treated?	Accidents (shocks)*	MHE accidents harm equipment parts
	Preventive maintenance*	Regular and/or thorough maintenance increases lifespan

* Data available in our dataset

** Unavailable data, relevant for future research

We do so by defining the main interfaces of an MHE to its environment (environment interface). Per environment interface we determine more granular interface dimensions. For each interface dimension we provide a hypothesis why it may impact breakdowns. The first environment interface is the MHE itself. It sounds intuitive that the inherent characteristics of an MHE, such as the manufacturer, the model, its age, purpose, dimensions and engine are related to the probability of a machine breakdown. The second environment interface deals with where the MHE is situated. This is the general location of the warehouse but also the physical warehouse design in which the MHE operates, e.g., the aisle specifications or floor conditions. The third interface is between the MHE and the operating employee. The operators mainly differ in their experience, tenure, and the training they underwent. The equipment is further impacted by two environment interfaces, its quantitative and qualitative use. Quantity (when the equipment is used) looks at the time of day, frequency and length of operation. Quality (how the equipment is used), sheds light on how many accidents occurred and how thoroughly the equipment is maintained. The derived framework can not only be applied to MHE but also to other equipment in similar environments.

2.3.2 *Primary data*

This section builds on two independent datasets. Each dataset comes from a different MHE manufacturer which are referred to as OEM1 and OEM2 hereafter. OEM1 is based in Europe and supplies its MHE to customers worldwide. The dataset of OEM1 is from a customer located in Europe. OEM2 is based in America and also serves customers globally. The dataset of OEM2 is from a warehouse location in North America. We use the dataset of OEM1 to build and validate our prediction model and to derive research insights. To show that both prediction model and insights are transferable, we repeat it on the dataset of OEM2. Each dataset has unique variables which may increase the prediction accuracy. However, these variables are not recorded by all OEMs. Ensuring that our findings are applicable beyond the scope of the two datasets, we build the prediction model based on those variables that are collected by most OEMs of MHE. An overview of the primary data available to us is given in Figure 2.2 with specific data points and their sources. The overall system of an MHE provides two types of data, master data and usage data. Master data describes information that is permanently associated with the equipment. It is characterized by the OEM, equipment model (e.g., low-level order picker or reach trucks), year of manufacture, performance specifications (e.g., lifting capacity) and engine type (e.g., combustion engine). Usage data on the other hand is data that is continuously recorded to reflect when and how the equipment is operated. This is done using a login terminal and sensor technology. The login terminal recognizes if an MHE is accessed and differentiates between operational use and maintenance services. For both, a timestamp for start and end time is recorded, providing info on date, time and length. The shock sensor records accidents as impact intensity. Impact intensity is measured horizontally (impacts from left/right) and vertically (impacts from front/rear). Especially shock sensors are

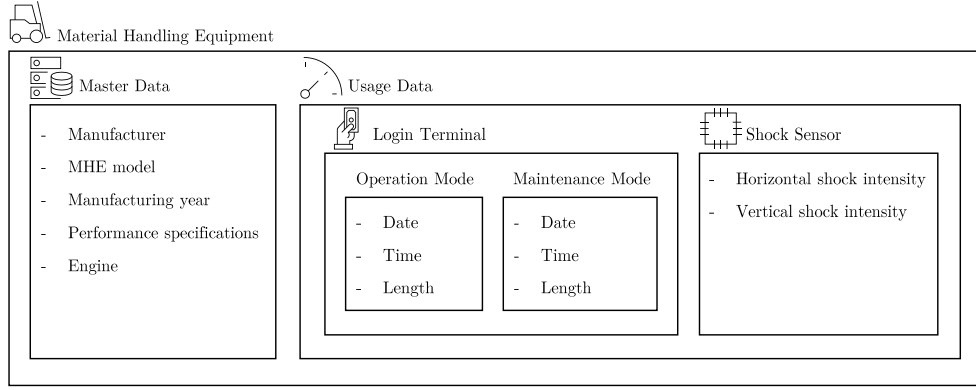


Figure 2.2: Primary data sources of MHE

common between OEMs and often used for condition-based maintenance (Ahmad and Kamaruddin (2012)). The dataset for OEM1 consists of 61 MHE units. Only those records are considered where all data points described in Figure 2.2 are on hand. As a result, 10,722 observation days are available with an average observation length of 175 days per MHE, 125 breakdowns and a mean time to breakdown of 57 days.

2.3.3 Feature extraction

The primary data from the previous section is used to extract features based on the introduced framework (Table 2.1). To understand the impact of operation quantity and quality on breakdowns, the mere time stamps captured in primary data are not sufficient. Features need to be extracted to capture relationships over time or between data points. As those features are problem-specific and relevance cannot be inferred a priori, we use multiple feature variations for our analysis (Table 2.2).

For example, we use the operation timestamps to calculate the number of short-term usages since the last maintenance service (similar to a diesel-engined automotive it might be possible that short-term usages are unfavorable for MHE). As the threshold for short-term usages is unknown, we define two (less than one and less than five minutes). The resulting data structure is shown in Appendix A.1. Each of the 10,722 observations is represented by a line item. For every MHE (MHE_i) the corresponding master data (MD_i) is provided. Observations per MHE are recorded daily (t_i). The gathered usage data for that day (UD_{t_i}) is used to generate the extracted features (EF_{t_i}). Eventually, each observation contains a breakdown indicator (BI_{t_i}) to classify if the MHE broke down or not. Contrary to Radner and Jorgenson (1963) we assume that only the entire MHE entity is observed as failed or not, not its parts individually, resulting in a binary classification.

Table 2.2: Feature extraction

Environment interface	Environment interface	Extracted feature
Quantity (When is the MHE operated?)	Operation date, time and length	- Days passed (IO)
		- Cum. time in operation (LM)
		- Cum. time in operation during morning shift (LM)
		- Cum. time in operation during afternoon shift (LM)
		- Cum. time in operation during late shift (LM)
		- Cum. count of operations (LM)
		- Cum. count of operations 1 min. (LM)
		- Cum. count of operations 5 min. (LM)
		- Cum. count of operations 60 min. (LM)
		Quality (How is the MHE treated?)
- Cum. count of maintenance services (IO)		
- Duration of last maintenance		
Horizontal shock intensity	- Cum. intensity of horizontal shock (LM)	
	- Max. intensity of horizontal shock (LM)	
Vertical shock intensity	- Cum. intensity of vertical shock (LM)	
	- Max. intensity of vertical shock (LM)	
Total shock intensity	- Cum. intensity of shock (LM)	
	- Cum. intensity of shock (L7D)	
	- Cum. intensity of shock (L14D)	
	- Cum. count of shocks (IO)	
	- Cum. count of shocks (LM)	
	- Cum. count of shocks (L7D)	
	- Cum. count of shocks (L14D)	

Notes: Cumulated (Cum) Since Initial Operation (IO) Since Last Maintenance (LM) Last 7 Days (L7D) Last 14 Days (L14D)

2.4 SLM-BASED BREAKDOWN PREDICTIONS

To solve the binary class prediction problem of the previously presented data structure we conduct a comprehensive study on SLMs for classification problems. The advances in SLMs provide multiple algorithms able to perform reliable classification tasks (Hastie et al. (2009)). Therefore, we first introduce the SLMs we investigate. Second, we discuss preprocessing steps of data and hyperparameter tuning for these SLMs. Third, we create a baseline against which we benchmark each SLM. Finally, we discuss their results and the insights they provide.

2.4.1 Statistical learning models

Solving classification problems is simple if classes are perfectly separable by linear functions or if breakdowns follow a specific distribution. Most real data do not fulfill this condition and it becomes computationally expensive to derive non-linear separators in high dimensional data space. This task is often approached using SLMs. Rather than focusing on a specific algorithm (like Herzog et al. (2009), Heng et al. (2009), Tran et al. (2009), Caesarendra et al. (2010)), we provide a comprehensive overview by investigating the following eight SLMs aimed at classification problems, which also include the methods used by the just mentioned authors. Gaussian Naïve Bayes (GNB) is a simple algorithm and uses the Bayes Theorem to determine a prediction class based on prior knowledge gained from features. K-Nearest-Neighbors (KNN) derives its prediction decision by measuring the weighted distance to already classified neighbors. Logistic Regression (LR) leverages a sigmoid function to separate classes using coefficients that provide better interpretability than many SLMs. Random Forest

Classifier (RFC) separately trains multiple decision trees and combines them to make predictions. Stochastic Gradient Boosting (SGB) also trains multiple decision trees, but uses an iterative approach to build a final model. Support Vector Classifier (SVC) uses kernels to construct a hyperplane that separates non-linear classes. Multilayer Perceptron Neural Network (MLP) builds a feedforward neural net using multiple layers of linear functions to generate a non-linear model. Relevance Vector Machine (RVM) generates predictions similar to SVC but provides probabilistic instead of point estimates. A more detailed description of the SLMs can be found in Appendix A.2. The interested reader might also refer to Hastie et al. (2009) and Hastie et al. (2015) for further details.

2.4.2 *Preprocessing, hyperparameter tuning and fitting*

For SLMs to yield optimal results, it is necessary to (1) preprocess the data and to (2) test different hyper parameter settings.

(1) Most data require preprocessing to deal with implicit data characteristics that affect presented SLMs adversely. For breakdown predictions, this means (a) upscaling of underrepresented dependent variables, (b) the standardization of data and (c) the reduction of multicollinearity between features. (a) Breakdowns are usually recorded less frequently than observations where a machine is operational. The resulting imbalance of the binary breakdown classifier, failure and non-failure respectively, leads some SLMs to underperform. A common method is to either up-sample the minority class or to down-sample the majority class. We used the Synthetic Minority Oversampling Technique (SMOTE) as presented by Chawla et al. (2002). Instead of simply duplicating minority observations which does not add additional information, SMOTE creates synthetic observations based on a regression. The regression in Equation 2.1,

$$mc_{j,SMOTE} = mc_i + ud(0, 1)(mc_i - mc_{k_i}), \quad (2.1)$$

takes an observation from the minority class mc_i , determines its k -nearest neighbors mc_{k_i} and uses a uniformed distribution ud between 0 and 1 to create a synthesized observation for the minority class $mc_{j,SMOTE}$ that lies between mc_i and mc_{k_i} . (b) Data often comes in different measuring units. For the MHE data in our research this is e.g., time in minutes, height in centimeters or shocks in impact acceleration. The resulting range of values might differ significantly. Various SLMs rely on unified data to calculate the distance between data points when fitting the separator function. As a result, data is standardized to take on similar values, e.g., by log transformations, min-max scaling or normal standardization. Our data was normal standardized with a mean of 0. (c) Features, especially when additional features were extracted from original ones, may show multicollinearity. This increases computational calculation time and leads to reduced interpretability of coefficients, e.g., in logistic regressions. To reduce multicollinearity, Principal Component Analysis (PCA) is used to combine correlated features via orthogonal transformations as well as eliminate low variance features. By combining features however, the interpretability of the resulting principal components becomes difficult. Alternatively, correlation between features is calculated and one of the two features with the highest

correlation is dropped. This process is repeated until a threshold of correlation among features is not exceeded. Compared to PCA, this loses marginally more information from the data, but maintains the interpretability of features. The latter was used with our data in which 22 out of 47 features showed multicollinearity (Figure 2.3) with $R^2 \geq 0.8$. The iterative elimination of factors led to a data set with 25 remaining features. (2) SLMs optimize parameters to

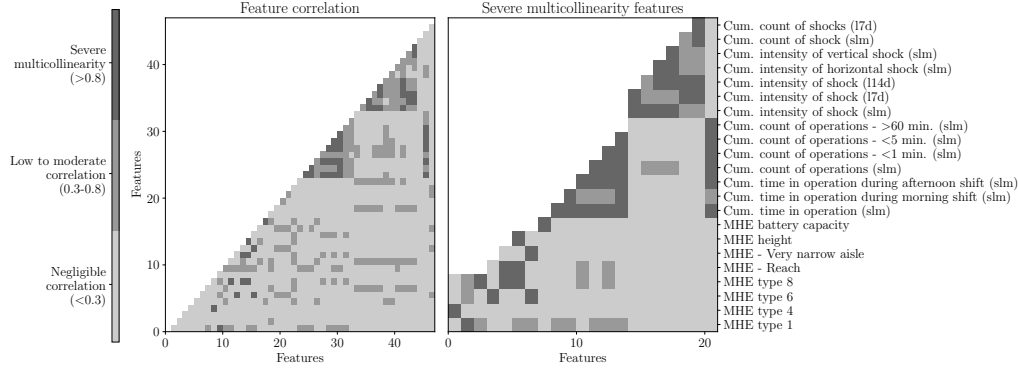


Figure 2.3: Feature correlation (OEM₁)

fit a given dataset. To ensure that the fitted model is generalizable to unseen data, 20% of the data is held out for a final performance test. Some parameters need to be provided to the models before fitting. These parameters are called hyperparameters and they are vital for the performance of SLMs. Ensuring that hyperparameters generalize as well, 10-fold stratified cross-validation was performed on the remaining 80% of data. A full enumeration of all parameter combinations was conducted using grid search. Appendix A.3 shows all tested hyperparameters with the indication which parameters yield the best predictive performance. To evaluate parameters and SLMs, we introduce four performance metrics, which are based on the elements of a confusion matrix. Those elements are true positives TP (an MHE was predicted to break down and actually broke down), true negatives TN (an MHE was predicted to stay operational and actually stayed operational), false positives FP (an MHE was predicted to break down but stayed operational) and false negatives FN (an MHE was predicted to stay operational but broke down). Precision in Equation 2.2,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (2.2)$$

determines correctly forecasted breakdowns divided by all forecasted breakdowns and answers the question: How many times do I order a service for the right reason? Recall in Equation 2.3,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (2.3)$$

determines correctly forecasted breakdowns divided by all true breakdowns and answers the questions: How many of the breakdowns do I detect? The F1 score in Equation 2.4,

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (2.4)$$

is the harmonic mean of Precision and Recall and thus provides a balanced view of over and under-prediction of breakdowns. Accuracy in Equation 2.5,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2.5)$$

provides insights on how many of all observations were predicted correctly. Although not particularly helpful for imbalanced data, Accuracy serves as comparison towards our baseline.

The best set of hyperparameters was selected based on the F1 score whereas all 4 metrics were used to evaluate the overall performance of SLMs on unseen test data. Due to highly imbalanced data and focus on breakdowns we do not report on specificity or receiver operating characteristics. The presented steps were implemented using Python v3.7 and its scikit-learn library for SLMs on a Windows machine with an 8-core CPU and 64 GB ram.

2.4.3 Baseline predictor

If nothing is known about the distribution of time to breakdown, it may be advantageous to never predict a breakdown as described in a minimax maintenance policy (McCall (1965)). In this case a maintenance service is ordered only after a breakdown occurs. In other words, MHE is always predicted to be functional. This prediction model is a Naïve Predictor (NP). It is used to show the impact of an SLM's presence versus its absence.

2.4.4 Insights from model comparison

The SLMs' results in Table 2.3 show two evident trends. First, prediction results increase with an increasing prediction window. This is intuitive as the prediction confidence of "Is the MHE going to break down within one day" is equal or smaller than "Is the MHE going to break down within seven days". Second, SLMs capturing non-linear relationships (SVC, KNN, SGB, RFC and MLP) outperform LR, GNB and RVM. Possible reasons are that LR, as applied by us, captures linear trends, GNB relies on the strong assumption of feature independence, and RVM could only be fitted with its default hyperparameters on a reduced training dataset (25%)².

The further discussion focuses on these five higher performing SLMs. They predict the majority of events correctly (Accuracy > 0.98) while dealing with the imbalance of breakdown versus non-breakdown events (F1 > 0.89). They consistently outperform NP in Accuracy, Precision and F1 by more than 70

²Calculations with the full dataset runs into memory errors, even on powerful machines with 64 processors and 256 gigabyte memory (ram).

Table 2.3: SLM prediction results (OEM1)*

Forecast	Metric	SVC	KNN	SGB	RFC	MLP	LR	GNB	RVM	NP
7 day	Accuracy	0.99	0.99	0.99	0.99	0.98	0.64	0.65	0.91	0.95
	Precision	0.91	0.89	0.93	0.93	0.81	0.13	0.13	0.37	0.00
	Recall	0.88	0.90	0.85	0.86	0.91	0.74	0.69	0.46	0.00
	F1	0.90	0.90	0.89	0.88	0.85	0.23	0.22	0.41	0.00
3 day	Accuracy	0.98	0.98	0.98	0.98	0.97	0.61	0.63	0.94	0.98
	Precision	0.73	0.75	0.72	0.72	0.61	0.05	0.05	0.19	0.00
	Recall	0.54	0.61	0.40	0.49	0.37	0.59	0.54	0.24	0.00
	F1	0.62	0.68	0.51	0.59	0.46	0.09	0.09	0.21	0.00
1 day	Accuracy	0.97	0.99	0.99	0.98	0.99	0.68	0.91	0.98	0.99
	Precision	0.08	0.21	0.00	0.00	0.20	0.02	0.03	0.09	0.00
	Recall	0.12	0.12	0.00	0.00	0.04	0.52	0.20	0.08	0.00
	F1	0.10	0.15	0.00	0.00	0.07	0.04	0.05	0.08	0.00

* Prediction results at probability decision threshold of 0.5 (See section 2.4.4.2 for more details)

percentage points at the expense of max. 15 percentage points lower Recall. The prediction results for the second independent dataset achieved similar results (Appendix A.4).

2.4.4.1 Cost-based SLM evaluation

As no SLM is strictly better than another (MLP outperforms at Recall, RFC and SGB at Precision, SVC and KNN at F1) it is difficult to determine which should be favored. To overcome this, we present a cost-based comparison in which we simulate the total cost C_{BM} (Equation 2.6) incurred by breakdowns and preventive maintenance services:

$$C_{BM} = c_M(TP + FP) + c_M r_{BM}(FN + u_{TP}TP) \quad 1 \leq r_{BM} \leq 10, \quad 0 \leq u_{TP} \leq 1. \quad (2.6)$$

Every predicted breakdown (TP and FP) results in a preventive maintenance service with cost c_M . The cost of a breakdown usually exceeds c_M (Chen and Feldman (1997)). Thus, each undiscovered breakdown (FN) yields the cost of c_M multiplied by the cost ratio r_{BM} . We provide a lower cost ratio boundary of 1 (breakdowns are as expensive as preventive maintenance) and an upper boundary of 10 (breakdowns are 10 times more expensive than preventive maintenance). Furthermore, there is a chance that an MHE breaks down despite the breakdown was correctly predicted and a maintenance service was ordered. The probability that the root cause of a correctly predicted breakdown stays undiscovered during maintenance is u_{TP} .

The cost comparison per SLM is shown in Figure 2.4. From a cost perspective, RFC or KNN should be chosen, depending if the cost ratio is below or above 2.2 respectively. For our dataset, r_{BM} is approximately 2.0 according to the LSP's experts.

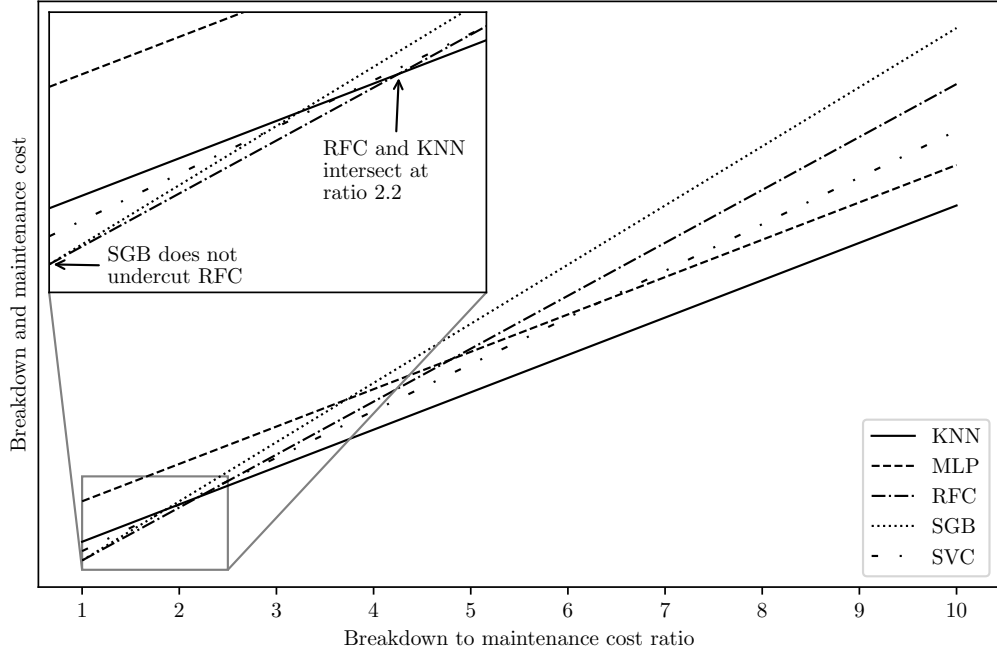


Figure 2.4: Cost comparison for breakdown to maintenance cost ratio ($u_{TP} = 0$)

2.4.4.2 Cost ratio dependent threshold optimization

Once an SLM is selected, further optimization is achieved to account for risk appetite, depending on whether undetected breakdowns or unnecessary maintenance services should be minimized. Each binary machine learning classifier calculates a probability threshold pt for unseen data to decide if it belongs to one or the other class. This default threshold is $pt = 0.5$, assigning failure if the probability belonging to that class is ≥ 0.5 , and assigning non-failure otherwise. This threshold can be changed and increasing it results in more conservative predictions of failures. Figure 2.5 shows the decision threshold-dependent change of metric scores for the RFC, which is the relevant SLM for our dataset. In the threshold corridor between 0.30 and 0.55, the F1 score stays relatively constant in a range of 0.88 – 0.90. In that corridor, Recall starts on top of Precision until a pt value of 0.42, from where the order reverses. In conclusion, multiple pairwise combinations of Recall and Precision are possible without sacrificing a reduction of F1.

To find the cost-optimized threshold, we calculate C_{BM} dependent on r_{BM} and pt (c_M set to 1 and u_{TP} set to 0). The resulting cost function assumes a semi-paraboloid shape (Figure 2.6) with a ruled surface as the function between r_{BM} and C_{BM} is linear. The function increases monotonically for gradient r_{BM} . For $pt < 0.15$, C_{BM} increases with decreasing pt values, independent of r_{BM} . This is because unreasonably many observations are labeled as breakdowns and each FP incurs cost c_M . Since c_M is constant, the increase of C_{BM} for $pt < 0.15$ is homogeneous. For $pt > 0.15$ and $r_{BM} < 1.5$, C_{BM} decreases with increasing pt . This is due to the fact that only high probability breakdowns should be predicted, as the gain of a TP is relatively low compared to the loss of an FP. In the extreme case of $r_{BM} = 1$, predicting breakdowns achieves no benefit and it

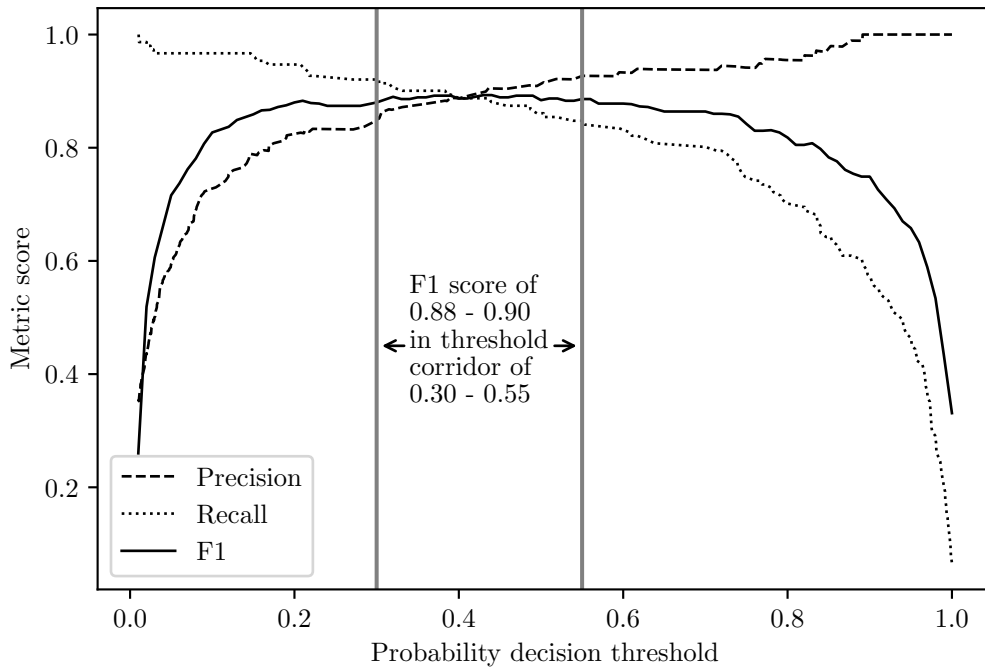


Figure 2.5: RFC metric scores for changing probability decision thresholds

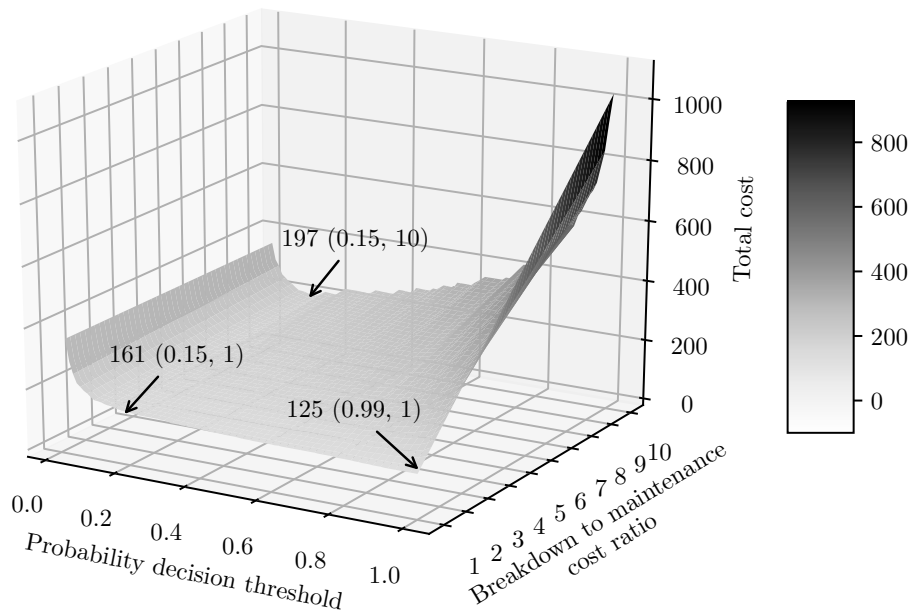


Figure 2.6: Cost ratio and threshold dependent cost

is always better to repair failures when they occur. For $pt > 0.15$ and $r_{BM} > 1.5$, C_{BM} increases with increasing pt . Since breakdowns get relatively expensive compared to maintenance services, a more liberal prediction of breakdowns should be followed since the gain of an TP is high compared to the loss of an FP. From a cost perspective we conclude that companies aiming at implementing breakdown predictions need to carefully study their breakdown to maintenance

cost ratio in order to choose the proper decision threshold. This is especially important if r_{BM} is in the area where the slope of C_{BM} changes for pt (In our case $r_{BM} \approx 1.3$)

2.4.4.3 Feature importance and implications

Additionally to a cost perspective it is important to understand what drives breakdowns. The insights from SLMs, however, are difficult to interpret as simple coefficients like in linear regressions are not available (Steinker et al. (2017)). From both cost-relevant SLMs, RFC still allows us to extract the importance of variables. The results not only indicate which factors drive breakdowns, but also what data should be collected by practitioners.

We use the Gini Impurity which measures the probability of an observation being wrongly classified. It serves as a criterion to construct the tree nodes which split data into branches so that these branches best contain a breakdown event or not (Hastie et al. (2009)). RFC feature importance provides information on how much each feature helps in decreasing the impurity. Focusing on those features whose importance exceeds 0.1 (Figure 2.7), we see that relevant features come from three categories. The first one covers time-based features (dark grey shading) and is the basis for literature on time-based maintenance. The second comprises condition-based features (medium grey shading) and is covered by literature on condition-based maintenance. Features in the third category contain static information, usually found in master data of MHE. Looking at individual features, time-based ones are the three most important. From a cumulative perspective, however, condition-based features exceed the other two categories, indicating that condition-related features are slightly more important than time-related ones. Given that time and condition features are both in a range of 40 – 50%, the conclusion is that it is crucial to leverage both data sources. This is particularly interesting as Amari et al. (2006) and Bloch and Geitner (2012) report that equipment breakdowns are mostly condition-driven, accounting for 80 – 99% of failures, which is not supported by our findings.

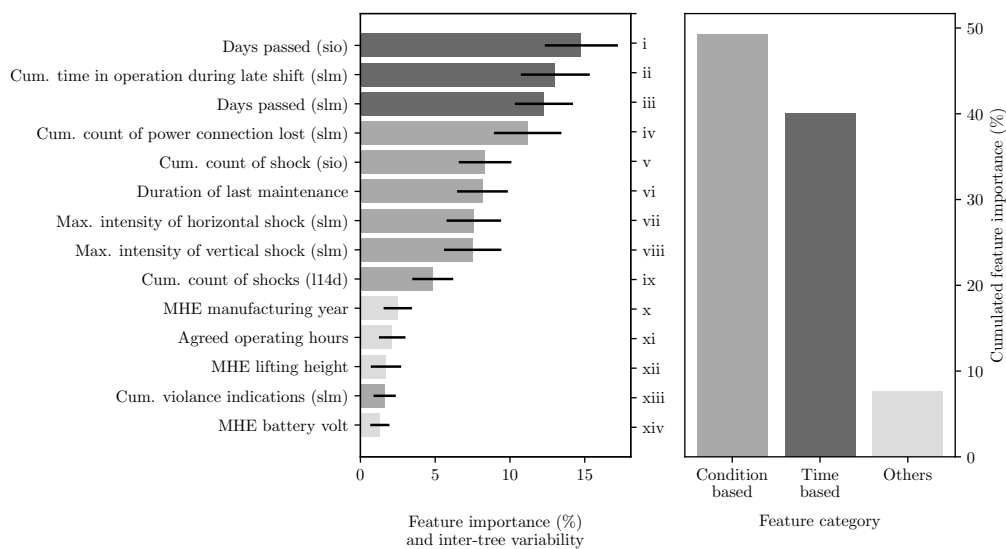


Figure 2.7: RFC feature importance (OEM₁)

When discussing the individual features from Figure 2.7 in their respective order (i the highest, xiv the lowest), we derive the following:

(i, iii) Time passed since an MHE was put into operation has the highest feature score. This indicates that the overall age of an MHE plays an important role, and that perfect repairs during maintenance, as covered by Klein (1962), are not apparent. Yet, days passed since last maintenance suggests that the distribution of time to failure is indeed affected by maintenance services. Hence, the repairs are not purely minimal repairs as described by Barlow and Hunter (1960) where equipment is just brought back to the state it was originally in before breakdown.

(ii) The cumulative operation time during late shifts ranks second. Although time-related, it is also an indication for an implicit condition-based feature as the operating behavior might change. For example, drivers during late shifts may be less attentive, thus contributing to the faster deterioration of MHE. Apparently, features for short (<1 and <5 minutes) and long (>60 minutes) term usages are not considered at all.

(iv) As electric parts in equipment are contributing factors towards breakdowns (Ahmad and Kamaruddin (2012)), it is not surprising that power cuts are a good predictor for failures as they indicate potential short circuits.

(v, vii, viii, ix, xiii) Furthermore, it is intuitive that shocks and other forms of violence contribute to breakdowns, and thus appear multiple times. Similar to the time-based features, the cumulative number of shocks is important from two angles. On one hand, the cumulative shocks since initial operation reflect a general wearing down of equipment parts. On the other hand, it seems to be especially relevant if there was a clustering of shocks in the recent past (last 14 days). Additionally, it is worthwhile noting that both, max. horizontal and vertical shock intensity, are among top predictors. They appear to complement the quantitative view on shocks by adding a level of magnitude as exceeding a certain threshold of impact increases the likelihood of failure. Although the violence indicator is among features with importance above 1%, its contribution to prediction accuracy is relatively low. It appears that manual observations of damage to MHE (e.g., dents) may be telling if equipment was treated inappropriately in the past, but only provide limited indicators about failures in the future.

(vi) The length of the last conducted maintenance service seems to serve as a proxy for MHE condition. A possible explanation is that longer maintenance times reflect equipment improvements close to a complete overhaul, whereas short maintenance times resemble minimal repairs.

(x, xi, xii, xiv) The last section of relevant features is comprised of MHE master data. Manufacturing year and agreed operating hours correlate with days passed since initial operation and cumulative time in operation respectively. Consequently, possible explanations for their contribution to prediction accuracy are similar. Nevertheless, both sets of information are valuable to organizations as they are known a priori, whereas the mentioned time-based features need to be calculated. Finally, MHE characteristics like lifting height and engine type are used for predictions, while manufacturers or MHE type are of no interest. A plausible explanation for the feature importance of lifting height and engine type is that they represent an MHE's working load. MHE with large

engines for heavy-duty or long-range to reach higher storage areas, tends to be subject to stress, which might affect breakdowns.

We conclude that both recording mechanisms, the login terminal and the sensors, are relevant data sources to improve prediction results. To our knowledge, OEMs of MHE already install both as default configurations. Although master data only contributes marginally, it should also be included as it is obtained at close to zero cost.

2.4.4.4 Impact of predictive maintenance

For impact calculation we compare the base cost C_{Base} of our dataset in Equation 2.7,

$$C_{Base} = c_M r_{BM} (TP + FN), \quad (2.7)$$

to the occurred cost after implementing the SLM (C_{BM}). The base cost is the total number of breakdowns ($TP + FN$) multiplied by the cost for breakdowns ($c_M \times r_{BM}$). The resulting impact after implementing an SLM (I_{SLM}) is shown in Equation 2.8,

$$I_{SLM} = \frac{C_{Base} - C_{BM}}{C_{Base}} \equiv \text{Recall} \left(1 - \frac{1}{r_{BM} \text{Precision}} - u_{TP} \right) \quad C_{Base}, r_{BM}, P > 0, \quad (2.8)$$

which is the percentage-wise cost reduction from C_{Base} to C_{BM} . It can be shown that I_{SLM} is a function of Precision, Recall, r_{BM} and u_{TP} (see Appendix A.5 for details). Thus, the impact is independent of breakdowns and non-breakdowns in an observation period for constant Recall and Precision.

Table 2.4: Confusion matrix

		Predicted class		
		Positive	Negative	Total
Actual class	Positive	108	17	125
	Negative	8	10,589	10,597
Total		116	10,606	10,722

The prediction results of our OEM1 dataset (Table 2.4) show that out of 10,722 observation days, a total of 125 breakdowns occurred. RFC correctly predicted 108 out of 125 breakdowns and missed 17. In total, the RFC predicted 116 breakdowns, thus misclassified 8 non-breakdowns as failures. A Recall of 0.86, Precision of 0.93, r_{BM} of ≈ 2.0 and u_{TP} of 0 yields an impact I_{SLM} of 40% (Table 2.5). Given that prediction results for OEM2 are similar to OEM1 (39%), the impact I_{SLM} is transferable. Based on the annual cost for breakdowns incurred by the 61 MHE units in our dataset which corresponded to €75,000, the cost savings is in the magnitude of €500 per MHE per year. As it is unlikely that all correctly predicted breakdowns can be avoided through maintenance

services, Table 2.5 shows the potential savings for varying values of u_{TP} . If 50% of correctly forecasted breakdowns cannot be avoided through maintenance, no savings can be achieved. Assuming a detection rate of 80% ($u_{TP} = 0.2$) and a worldwide MHE production quantity of 1.8 million units in 2016 (Bond (2017)), the overall impact adds up to a mid-three million digit cost saving only for the newly produced MHE.

Table 2.5: Impact after SLM implementation

	Scenarios for varying u_{TP}					
	0.0	0.1	0.2	0.3	0.4	0.5
I_{SLM}	40%	31%	23%	14%	5%	-3%
Savings (€/MHE/year)	489	383	277	172	66	-40

2.5 DISCUSSION AND CONCLUSION

2.5.1 *Synthesis*

In this chapter we build a predictive maintenance model for MHE which incorporates novel data sources (Figure 2.2) to forecast breakdowns, and test it on two separate and independent datasets from OEMs. Opposing to classical preventive maintenance models, our maintenance policy relies solely on the prediction of breakdowns and disregards standard maintenance intervals. If a breakdown is predicted, a maintenance service is scheduled which reduces the probability of the breakdown to occur.

To create meaningful features from raw data, we develop a framework (Table 2.1) to understand how an MHE interacts with its environment. The interfaces between MHE and environment provide guidelines for breakdown indicators and subsequent feature extraction (Table 2.2). A comprehensive investigation of SLMs demonstrates that five algorithms were capable of processing the extracted features by predicting breakdowns with a F1 score performance of > 0.85 (Table 2.3).

In detail we find that:

(1) SLMs incorporate complex data simultaneously, ranging from fixed master data to variable usage data such as operation times and sensor recordings. Accurate predictions are made without prior knowledge about the distribution function of time-to-failure.

(2) The extracted features are transferable as similar prediction performance is achieved on a second independent dataset (Appendix 3). This shows, in contrast to Kim et al. (2019), that the efficient recycling of features is viable.

(3) Two SLMs, RFC and KNN, minimize total cost. While RFC does so for $r_{BM} \leq 2.2$, KNN is cost-optimal for $r_{BM} > 2.2$. An annual cost reduction through breakdowns in the magnitude of €500 per MHE is accomplished and amounts to a mid-three million digit cost-saving worldwide.

(4) Given a constant SLM performance, it is shown that for a given decision threshold the total cost is a function of Recall, Precision and r_{BM} . The function is independent of the actual breakdown and non-breakdown observations.

(5) The probability threshold optimization for further cost reduction needs to be exercised with caution. Since C_{BM} , as a function of pt , r_{BM} and u_{TP} , assumes a semi-paraboloid shape with a ruled surface, it is of paramount importance to understand the value of r_{BM} . A one-directed change of pt increases or decreases C_{BM} , depending on r_{BM} .

(6) Where most literature focuses on either time or condition-based data, we not only show that it is pivotal to incorporate both data types, but also to leverage the equipment's master data. This also contradicts the findings of Amari et al. (2006) and Bloch and Geitner (2012) who state that the equipment's condition accounts for 80 – 99% of failures.

(7) The three data sources, master data, login terminal, and shock sensors, are sufficient to predict breakdowns. The prerequisites are given for most MHE as OEMs contacted by us use login terminals and shock sensors in their equipment by default. Thus, the limitations caused by difficult to obtain data (McCall (1965)), or data that is available but expensive to access (Ahmad and Kamaruddin (2012)), are no longer an issue.

2.5.2 Future research

Our research contributes and extends existing knowledge on the topic of preventive maintenance and provides a starting point for further research. We see mainly two areas for future exploration. The first area derives from the framework of impact factors (Table 2.1) we present in our research. Although we cover a wide range of possible factors, three environment interfaces are not examined. Therefore, potential research efforts might investigate the environment in which an MHE is located (warehouse location and design), the impact of operators (e.g., their tenure or qualifications), and additional sensor data (e.g., vibrations, temperature or humidity). The second area is concerned with the transferability of our research to other fields. Potential applications are all types of vehicles with similar complexity such as ride-on mowers, e-scooters, cranes, farming vehicles, bikes, cars or trucks, as they too undergo regular maintenance services.

From a resource perspective, there are two executing entities in the warehouse, i.e., equipment and labor. In this chapter, we conducted a comprehensive analysis of SLMs to predict breakdowns of MHE. In the following Chapter 3, we consider the other executing resource, the warehouse employees. Specifically, we investigate how Extreme Gradient Boosting (EGB) can be used to predict the productivity of operational employees to improve workforce planning.

INTEGRATING OPERATIONAL AND HUMAN FACTORS TO PREDICT DAILY PRODUCTIVITY OF WAREHOUSE EMPLOYEES USING EXTREME GRADIENT BOOSTING

The following chapter is based on Falkenberg and Spinler (2022a).¹

3.1 INTRODUCTION

Warehouses, as part of supply chains, may provide a substantial competitive advantage by enabling efficient satisfaction of customers' expectations (Gu et al. (2007)). The main processes in a warehouse are inbound, storage, picking, packing and outbound (De Koster et al. (2007)). These are often carried out manually (Gu et al. (2007)), and are affected by uncertainties from two sides. One is the workload (demand) that must be processed, which is subject to many external factors. The other relates to machinery and workforce (supply) issues caused, for example, by breakdowns or performance fluctuations. When engaging in workforce planning to match the workload with the workforce, it is important to reflect on these uncertainties to ensure effective and efficient processes (van Gils et al. (2017)). Demand-side forecasting has received considerable research attention (e.g., van Gils et al. (2015) and (2017), Schwarzkopf et al. (1988), De Gooijer and Hyndman (2006)), whereas supply-side predictions have been of secondary importance, even though accurately forecasting future personnel requirements is crucial for companies' success (Masud (1985)). In fact, the workforce is often used to offset poor demand forecasts (Sanders and Ritzman (2004)). Most studies focus on ways to improve productivity, such as skills allocation (Matusiak et al. (2017)), establishment of standard operating procedures (Mor et al. (2019)) and storage strategies (Chan and Chan (2011)). The few supply-side studies deal with highly aggregated predictions and make limited use of available data. Our study addresses this gap by providing a method that makes daily employee-specific productivity predictions incorporating a multitude of variables.

The remainder of this chapter is structured as follows. In Section 3.2 we review the existing literature and explain our contributions. In Sections 3.3 and 3.4 we introduce the dataset for our model, establish the baseline, and provide an overview of the technicalities of our method. In Section 3.5 we discuss the prediction results, highlight the insights derived, and present a cost-based impact calculation. In Section 3.6 we conclude by summarizing our study and suggesting avenues for future research.

¹*This paper with the title "Integrating operational and human factors to predict daily productivity of warehouse employees using extreme gradient boosting" was written by Sven Falkenberg and Stefan Spinler, and has been published in the journal "International Journal of Production Research".*

3.2 RELATED WORK AND LITERATURE CONTRIBUTION

Workforce productivity is driven by three sources of variability: employees, the operations that they conduct, and the environment in which they are placed (Doerr and Arreola-Risa (2000)). Most research focuses on a single factor within these categories. In the employee category, for example, studies investigate the effect of well-being (Gandy et al. (2014)), interview performance (Liziwe et al. (2008)), workers attitudes (Fletcher et al. (2006)), personality and demographics (Juran and Schruben (2004)), and differences between workers (Doerr and Arreola-Risa (2000)). Operations-specific effects are addressed in extensive research on line balancing problems (Sivasankaran and Shahabudeen (2014)), and on the relationship between visible backlogs and productivity (Powell and Schultz (2004)). Studies of the environmental effects on productivity examine the relationship between leadership and employees (Loi et al. (2011)), drug testing (Konovsky and Cropanzano (1991)) and general design factors.

Few studies involve actual predictions of productivity. Those that do are highly aggregated by time or warehouse, despite the fact that productivity is likely to vary by employee (Thompson and Goodale (2006)). Masud (1985) carries out productivity forecasting at a warehouse level and on a monthly basis using a Box-Jenkins modeling approach, while Islam et al. (2021) use swarm optimization. Both use only a single factor for their predictions. To our knowledge, only two studies leverage multiple variables to forecast productivity. Sonmez and Rowings (1998) use neural networks with several variables in a construction setting, while Thomas and Sakarcan (1994) apply this method in a warehouse context, but aggregated to productivity per square meter rather than at the employee level.

Reasons for the absence of research incorporating multiple variables include the increased complexity of relationships, the knowledge required about distributions, and the questionable cost-effectiveness of improved accuracy (Doerr and Arreola-Risa (2000)).

Our research addresses this problem and contributes in four ways. First, we provide a framework to identify the four main categories that impact on productivity. Second, we show that Extreme Gradient Boosting (EGB) is capable of capturing the complex relationship between multiple variables without prior knowledge of the distributions of these variables. Third, we show that the improvement to prediction accuracy using multiple variables is substantial. Fourth, we confirm that this can be achieved cost-effectively by showing that computational optimization efforts can be recycled, thus reducing the most expensive part of computations.

3.3 CASE INTRODUCTION

3.3.1 *Case partner and problem definition*

In this study, we consider a problem of a leading Logistics Service Provider (LSP) operating a large number of warehouses worldwide. A major task in managing these warehouses is operational workforce planning, which involves predicting

the upcoming workload (demand) and translating this into staff requirements (supply). This is a challenging task, as unknown variables come from both sides. On the one hand, demand is subject to seasons, trends, promotions and other factors that cause uncertainty. On the other hand, staff planning must account for no-shows of personnel, as well as the varying productivity of heterogeneous staff. To handle these uncertainties, the LSP conducts operational

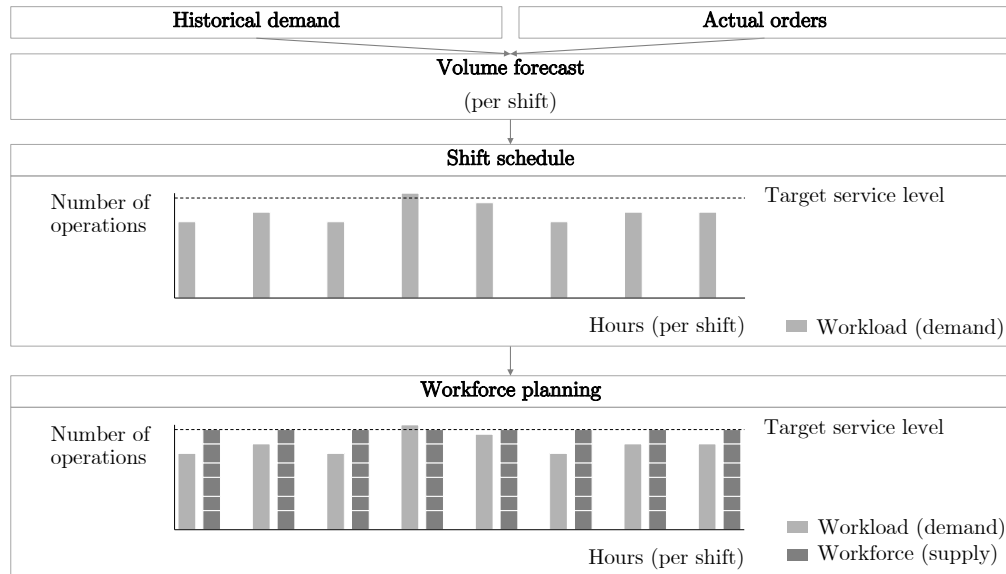


Figure 3.1: Workforce planning

workforce planning in the following steps (Figure 3.1). First, historical customer demand and actual orders are used to generate volume forecasts per shift. Second, the orders in a given shift are sequenced to create a detailed work schedule for each function (e.g., picking or packing). This schedule is translated into the number of operations required (e.g., number of picking units). Taking account of the target service level, the appropriate number of warehouse associates is scheduled. To determine the appropriate number of warehouse associates, each employee is assigned a tenure-based productivity rate expressed as operations per time unit (e.g., picking units per hour). Associates are added to the shift until the total productivity of all associates is sufficient to fulfill the given service level. This is particularly difficult owing to high churn rates among warehouse employees, as the productivity of new hires varies widely. Hence, the LSP wishes to implement a model that predicts productivity in the first eight weeks of employment to better reconcile workforce and workload.

3.3.2 Data overview

Our dataset covers the picking and packing process of a large warehouse with roughly 11 million recorded transactions. Aggregated by days, we have about 14,000 observations that occurred within the first eight weeks of employment. The recorded and aggregated variables v_i are shown in Table 3.1. Each observation can be identified by the date it occurred, the executing employee, and the function to which the employee was allocated.

Table 3.1: Data overview

Category	Variable	Variable name	Variable count [shape]
Identifier	v ₁	Date	365 [Timestamp]
	v ₂	Employee number	>600 [Employee ID]
	v ₃	Function	2 [Picking, Packing]
Long-term stimulators	v ₄	Job role	4 [Normal, seasonal, temporary and forklift operator]
	v ₅	Assigned shift	2 [Early, Late]
	v ₆	40 hour contract	2 [Yes, No]
	v ₇	Operational supervisor	11 [Operational supervisor ID]
	v ₈	Managing supervisor	4 [Managing supervisor ID]
Short-term stimulators	v ₉	Overtime	2 [Yes, No]
	v ₁₀	Double time	2 [Yes, No]
Operational productivity	v ₁₁	Hours worked	[≥ 1]
	v ₁₂	Orders processed	[≥ 1]
	v ₁₃	Quantity processed	[≥ 1]
	v ₁₄	Volume processed	[≥ 1]
	v ₁₅	Weight processed	[≥ 1]
Target variable	v ₁₆	Quantities processed per hour	[≥ 1]

Long-term stimulators relate to job roles, assigned shifts, contracted hours and supervisors. The job role determines whether someone is permanently hired for manual work (laborer) or specifically to handle forklifts (forklift operator). Non-permanent hires are hired either for a specific time period (seasonal labor) or to fill gaps with no specific timeline (temporary labor). Employees are assigned to one of two shifts, early or late, and some have a guaranteed 40-hour workweek. During shifts, they are supervised by two entities: the warehouse responsible (managing supervisor) and the shopfloor lead (operational supervisor).

Short-term stimulators of employees' productivity are extra payments and non-monotonous workdays. Extra payments are made if employees work more than eight hours a day (overtime) or on more than six consecutive days (double-time). Non-monotonous workdays refer to employees being allowed to switch between functions.

Operational productivity records warehouse associates' performance. The variables tracked include the daily number of orders processed, comprising individual units (quantity) and their respective volume and weight in cubic centimeters and grams. For each record, the processing time is logged.

The data reveal that productivity, expressed as operations per hour, peaks during summer and winter, which correlates with the LSP's promotion cycles (Figure 3.2). The two functions show similar patterns throughout the year with a stable 95% confidence interval.

At a weekly level, both functions have similar productivity from Monday to Sunday with a comparable spread (Figure 3.3). For both functions, the lowest performance is on Saturday, with regard to medians and outliers². A potential explanation is that motivation decreases at weekends, but may recover on Sunday when double-time payments apply.

²Outliers are observations beyond the whiskers that extend the inner quartiles by 1.5 times the interquartile range.

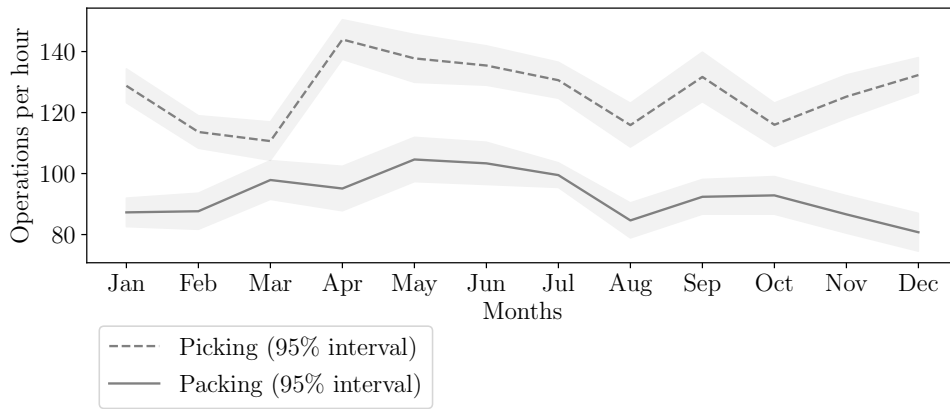


Figure 3.2: Productivity per month

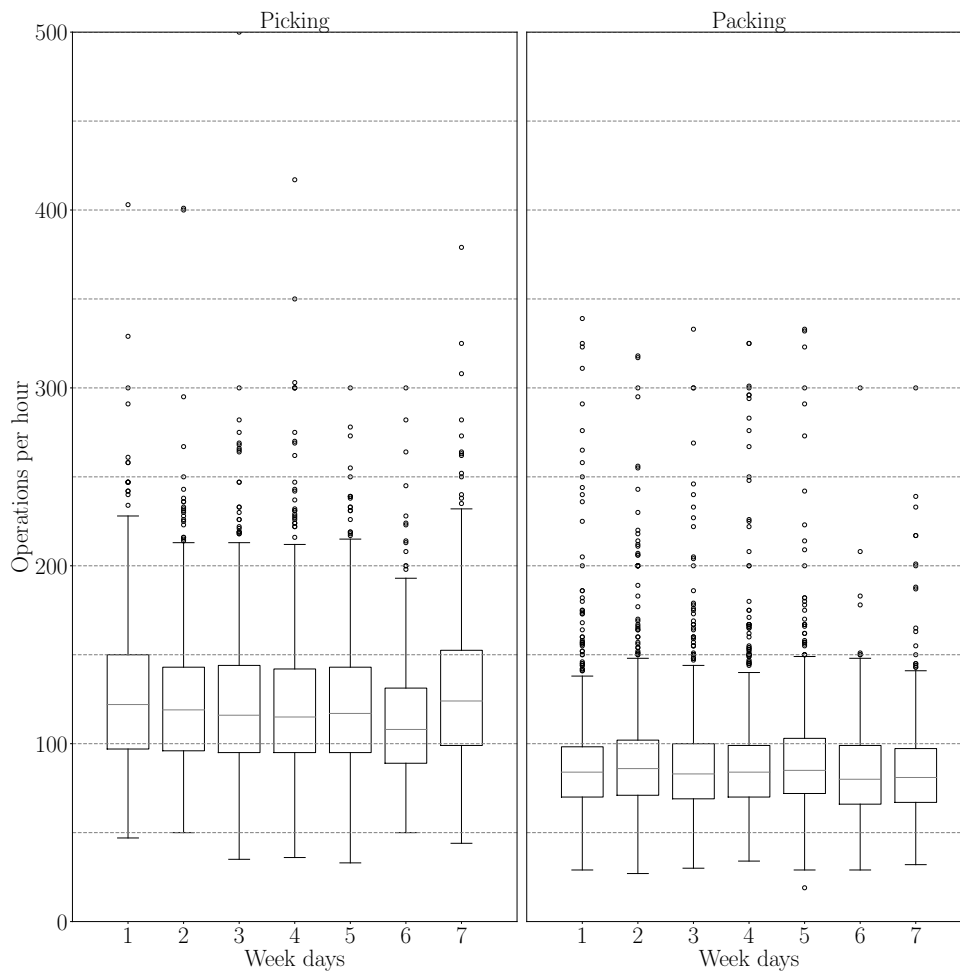


Figure 3.3: Productivity per work day

Motivators offering short-term incentives (Figure 3.4), such as extra payments, appear to have little effect. No significant difference can be observed for overtime, while for double-time, the interquartile range increases only for the picking function. Switching tasks to avoid monotonous work produces a positive productivity trend with increasing alternations.

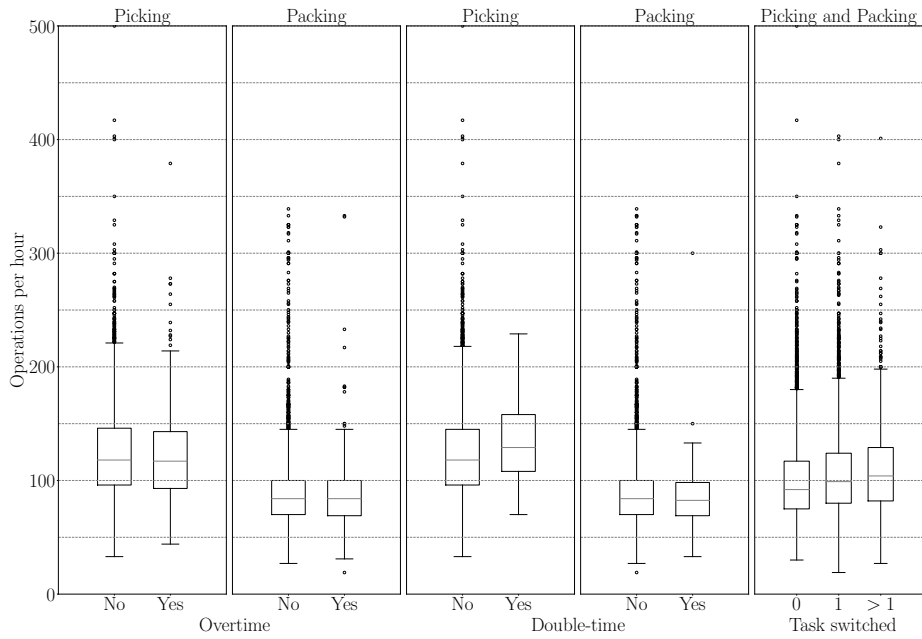


Figure 3.4: Productivity with overtime, double-time and task switching

The development of productivity in the first 40 working days (eight weeks) reveals different learning curves for picking and packing (Figure 3.5). Picking shows a steady learning curve over the first six weeks until a stable productivity level is established. In contrast, packing reaches a stable productivity level in week 3, with two preceding weeks of increasing productivity. New employees usually start in picking and may change to packing with more experience, which may explain the earlier maturity in the packing function. Outliers decrease in both functions within the first two weeks (14 working days).

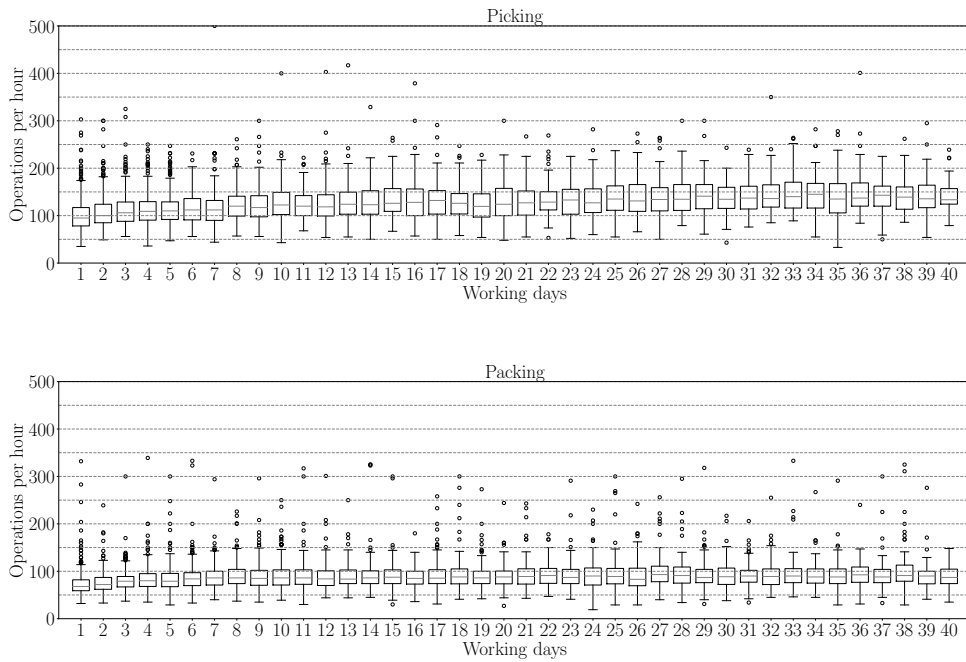


Figure 3.5: Picking and packing productivity in the first 40 days

To gain insights into what drives the productivity learning rate in the first eight weeks, we examine individual long-term stimulators (Figure 3.6). Job

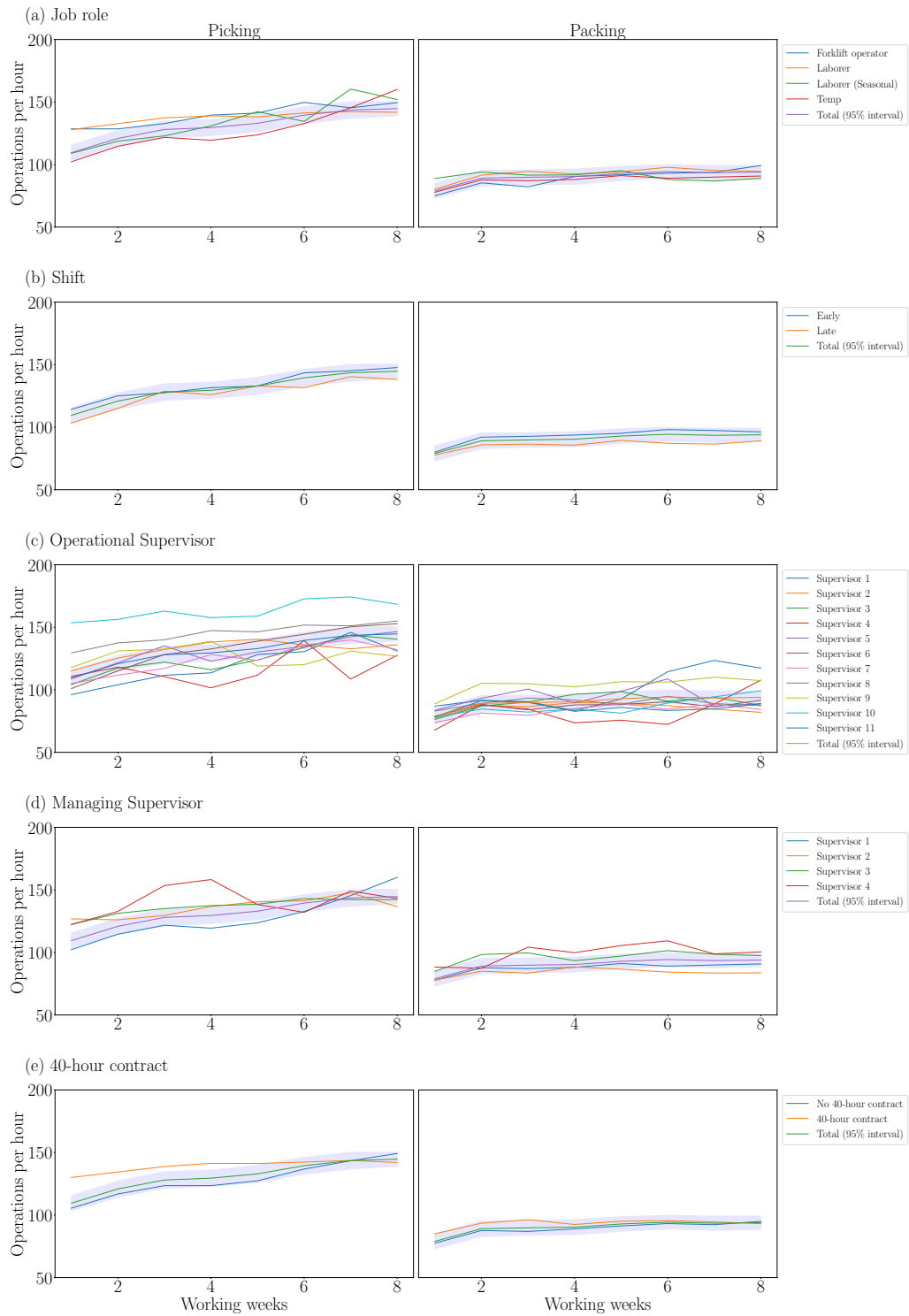


Figure 3.6: Productivity by long-term stimulators

role (Figure 3.6a) does not appear to impact on productivity development. In the picking function, no role shows consistently better performance than another. For packing, only one role (laborer) constantly performs above average, while temporary labor is below average for all eight weeks. Other packing roles

exhibit no distinct pattern. A different picture can be seen in the shift assignment (Figure 3.6b). Apart from week 3 in the picking function, all other weeks in both functions indicate higher productivity for the early shift compared with the late shift. On the other hand, different supervisors have no strong effect on productivity development at either operational or managerial level (Figures 3.6c and 3.6d). Although two operational supervisors correlate with better productivity in one function (supervisor 10 in picking, supervisor 9 in packing), these individuals' higher productivity does not apply to the other function. A possible explanation is their expertise in one of the two functions. Similarly, no managing supervisor is consistently better than another in both functions. Supervisor 1 is superior to supervisor 2 in packing, but below average in picking. Overall, the performance of supervisors at the management level shows less variation than that of operational supervisors. Lastly, employees with 40-hour contracts start with above-average productivity in both functions (Figure 3.6e). This advantage flattens out in weeks 3 and 4 for picking and packing, respectively. Those without 40-hour contracts start below average but show a steep learning curve over the eight weeks, resulting in a turning point in the last two weeks.

In summary, the data show few clear distinguishing variables, such as assigned shifts or tasks switched. Most variables offer limited insights (40-hour contract, job role, function, months) or are inconclusive (supervisors, payments, weekdays).

3.3.3 *Baseline and simple predictor*

For workforce planning, the LSP assumes that after the initial training it takes warehouse associates four full working weeks to achieve the full productivity of a tenured employee. A working week is defined as five working days for a warehouse associate, regardless of whether or not these are consecutive. Starting with a productivity level of 40% in the first working week, regressive increases to 60%, 75% and 80% are assumed in weeks 2, 3 and 4 respectively (Table 3.2). These productivity assumptions are based on historical data and represent workforce averages. As can be inferred from sub-section 3.3.2, the

Table 3.2: Assumed ramp-up of new hires' productivity

Working week	Cumulative work days	Assumed productivity
1	1-5	40%
2	5-10	60%
3	11-15	75%
4	16-20	80%
≥5	≥21	100%

LSP's assumptions are too simplistic for use as a baseline in our study. For example, they are neither employee-specific, nor do they consider different learning curves between functions. Hence, we introduce a more robust baseline for com-

parison. Our model makes predictions at an individual employee level, and therefore we introduce three simple, employee-based predictors. All three simple predictors leverage past productivity per employee to forecast the future. They rely on building Cumulative Moving Averages (CMA) or Simple Moving Averages (SMA), and differ only in the length of historical data stored.

The first simple predictor, CMA (Equation 3.1), is based on an employee's overall productivity level, and calculated by averaging all available n data points of previous productivity observations p_t :

$$\text{CMA}_{n+1} = \frac{p_1 + p_2 + \dots + p_n}{n} \quad 1 \leq n. \quad (3.1)$$

The second predictor, SMA5 (Equation 3.2), builds a moving productivity average over the last five working days:

$$\text{SMA5}_{n+1,k=5} = \frac{1}{k} \sum_{t=n-k+1}^n p_t \quad 5 \leq n. \quad (3.2)$$

The rationale for a short historical horizon is to capture productivity changes more quickly to account for the initial learning curve. The downside is that event-driven variations have a stronger impact on productivity predictions. The third simple predictor, SMA10 (Equation 3.3), takes account of the past 10 working days:

$$\text{SMA10}_{n+1,k=10} = \frac{1}{k} \sum_{t=n-k+1}^n p_t \quad 10 \leq n. \quad (3.3)$$

With a mid-range horizon, this predictor balances the short-term learning curve and the employee's overall productivity level. We replace SMA5 and SMA10 with CMA in cases where $n < k$.

3.4 DATA PREPARATION AND METHOD

This section clarifies the methods we used to prepare the data and build the predictions model. We introduce a structured approach to identify potential factors influencing productivity. We use this approach to enrich the primary data by extracting meaningful features. We then present a short overview of ensemble tree methods and our tuning of relevant parameters in this process.

3.4.1 Feature generation

Operational warehouse employees' productivity is driven by many factors. In order to better understand what variables come into play, we provide a structured framework that categorizes these factors into four impact categories: (1) Warehouse, (2) Operator, (3) Shift and (4) Product (Table 3.3). Within these categories, we list potential factors influencing productivity, with no claim to being fully exhaustive. For each factor in a category, we show which primary variables v_i are available in our dataset. We also list the features we extracted from the primary variables.

Table 3.3: Framework of productivity impact factors

Impact category	Impact factor	Data variable	Extracted feature
Warehouse	Location	*	-
	Design	*	-
	Size	*	-
	Process maturity	*	-
	Degree of automation	*	-
Operator	Job role	v ₄	Job role
	Salary	v ₆	40-hour contract
	Age	*	-
	Experience	v ₁	Total days worked
		v ₁₁	Total hours worked
		v ₁₃	Total quantity processed
	Training	*	-
	Days off	v ₁	Days off since last shift
	Sick days	*	-
	Past performance	v ₁₆	SMA _{P,5} , SMA _{P,10} , CMA _{P,n}
Shift	Date	v ₁	Month of the year
			Day of the week
	Function	v ₃	Function
	Shift type	v ₅	Shift type
	Work monotony	v ₃	Functions switched per shift
	Supervisors	v ₇	Operational supervisor
		v ₈	Managing supervisor
	Extra payments	v ₉	Overtime
		v ₁₀	Double-time
	Workforce size	*	-
Product	Quantity	v ₁₂	Daily orders picked
		v ₁₃	Daily quantity picked
	Volume	v ₁₄	Daily volume picked
	Weight	v ₁₅	Daily weight picked
	Special Handling requirements	*	-

Note: * unavailable data, relevant for future research

The first impact category is the warehouse in which the operation takes place. Impact factors affecting productivity performance include location (e.g., places with high or low humidity), design (e.g., easy or difficult to access aisles), size (e.g., short or long travel distances), process maturity (e.g., standard operating procedures) and extent of automation (e.g., picking robots).

The second impact category is the actual operator who conducts the work. The corresponding impact factors are the job role (e.g., full-time employee or temporary labor), salary (e.g., different hourly wages), age (e.g., physical fitness), work experience (e.g., accumulated work hours), training (e.g., onboarding workshops), days off (e.g., recreation time), sick days (e.g., overall health) and past performance (e.g., historical performance indicators).

The third impact category is the shift environment in which the operator is placed. Potential impact factors are the date (e.g., extreme temperatures in some months, or proximity of weekdays to the weekend), function (e.g., differ-

ing operations per hour per function), shift type (e.g., reduced attention during night shifts), work monotony (e.g., assembly line versus job shop), supervisors (e.g., ability to motivate), extra payments (e.g., increased wage for overtime work) and workforce size (e.g., congestion caused in aisles).

The fourth impact category is the product that is being processed. The efficiency with which a product can be picked and packed is affected by the quantity (e.g., economies of scale for higher quantities), volume (e.g., large products are more unwieldy), weight (e.g., faster exhaustion when handling heavy products) and special requirements (e.g., lower operability of breakable, hazardous or cooled products).

The extent of available data variables enables us to study a high proportion of impact factors.

3.4.2 *Ensemble tree methods*

We use an ensemble tree method to build our prediction model. While single trees are the simplest form of decision tree models, the most popular ensemble tree methods are Random Forest (RF) and Gradient Boosting (GB) (Hastie et al. (2015), Friedman (2002), Breiman (2001)). Ensemble tree methods offer several advantages over established and easily comprehensible methods such as linear regression. They do not require extensive pre-processing of data, such as normalization of variables, they allow processing of categorical and numerical variables simultaneously, and they can cope with unbalanced and non-linear data. Compared with single decision trees, ensemble methods are more robust because the overfitting associated with single trees is reduced, which may increase accuracy in relation to unseen data. In this sub-section, we present a concise introduction to tree methods for predictions.

A single tree model is intuitive, as it follows a tree structure that divides the dataset at decision nodes into two, or sometimes more, subsets (branches). At the end of each branch may be another decision node or a final prediction value (leaf). A decision node has a decision parameter and a corresponding decision threshold. The parameter is one of the variables in the dataset. The decision threshold is a value in the range of all values of the chosen variable. It is selected to minimize the variance of the target variable in the resulting subset. In a fully grown tree, decision nodes are added until only one value exists for each branch. This is cumbersome and leads to over-fitting of the resulting model; hence, it is likely to be a bad predictor for unseen data. As a result, a stop criterion is used which limits, for example, the number of decision nodes (depth of the tree) or the maximum variance to be achieved (purity of a branch).

Ensemble tree methods make use of single trees by combining them. This can be done simultaneously (bagging) or sequentially (boosting).

RF is a bagging algorithm as it builds multiple single trees simultaneously. Each tree uses a random subset based on a selection of randomized observations and variables. Individual weak learners (single trees) are then combined to make a collective prediction (strong learner). For regression purposes, consensus is achieved by averaging all weak learners. Owing to the parallelization

of individual tree building, RFs can be built in a similar time to single trees. They also perform better on unseen data, as over-fitting is reduced compared with single trees. A drawback is reduced interpretability.

GB, on the other hand, is a boosting algorithm that builds single trees one after another. During this process, the current tree focuses on improving the part of the previous tree that had the highest prediction error. The error is defined as any suitable loss function. Owing to its sequential and self-learning nature, GB often outperforms RFs, but is slower. Friedman (2000) proposes EGB which uses a greedy algorithm to decrease the processing time. In our research, we focus on EGB for its prediction performance advantage, but also report the results of RF as a reference for our analysis.

3.4.3 *Hyper-parameter tuning*

Hyperparameters define the conditions and boundaries of the algorithm's learning process. Prediction performance may be adversely affected by poorly chosen hyperparameters. Before hyperparameters are tuned, the data are pre-processed. Most importantly, we transform variables that show heteroscedasticity, and create dummy variables for categorical values. Hyperparameter optimization in EGB requires a range of test values to be defined for relevant parameters (Appendix B.1).

Max_depth determines the depth of a tree by defining the maximum number of sequential nodes to be used. The default value is six, which we increase in increments of two until a maximum of 10 concatenated variables-nodes is used in a single tree. We also examine the case when only four variables are used. The Min_child_weight hyperparameter has a default value of 1, meaning that nodes can split data until only one observation is left. This allows the algorithm to reflect specific cases, but may also lead to overfitting. Hence we test all single-digit values in increments of two to allow the model to be more conservative. Eta is the learning rate of the algorithm and lower values allow the algorithm to make smaller steps for better optimization, but simultaneously increase computational time. Given the efficiency of EGB, we allow values smaller than the default of 0.3, first in decimal increments, then with two low rates of 0.05 and 0.01. The Gamma hyperparameter defines by how much a node needs to reduce the error when splitting the dataset. Higher values reduce overfitting at the expense of lower granularity. While the default is 0, we start with 0.01 and increase test values by a factor of 10 up to a maximum value of 10. Subsample and Colsample_bytree reflect the proportion of randomly selected observations (rows) and features (columns) respectively. Given that we use training and test datasets as well as cross-validation which already partitions the observations, we limit Subsample to high values of 0.9 and 1 (default). In Colsample_bytree we test all ratios in increments of 0.25.

Each combination of hyperparameter test values is evaluated against our loss function which is equal to the Root Mean Squared Error (RMSE) in Equation 3.4:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (\hat{p}_t - p_t)^2}, \quad (3.4)$$

where p_t represents the time spent per operation, a common performance metric for warehouse productivity (Staudt et al. (2015b)).

To find the best hyperparameters while making them generalizable to unseen data, we use 10-fold cross-validation. The dataset is randomly split 80/20 into training and test subsets respectively. This is repeated ten times, and hyperparameter tuning is conducted on each of the ten training subsets. We chose the set of hyperparameters with the lowest average RMSE over all ten training subsets. The prediction performance reported later is based on the RMSE for the test subset only.

The calculations are performed on a Windows server with a six core Intel(R) Xeon(R) CPU E5-2630, with 2.30GHz and 64 gigabyte ram. All calculations are implemented in Python v3.9.

3.5 PREDICTING NEW HIRES' PRODUCTIVITY

3.5.1 *Prediction results*

Our model predicts the productivity of warehouse employees in their first eight weeks of employment. For all eight weeks, our model outperforms the prediction accuracy³ of the LSP, all three simple predictors and the RF (Table 3.4). In comparison with the LSP's predictions, EGB shows an RMSE improvement of 41.6% in picking and 52.5% in packing. In relation to the three simple predictors, EGB shows further RMSE improvements in the range of 20.2-20.9% and 25.3-26.1%, respectively. Even compared with RF, EGB improves the prediction accuracy by 5.5% and 9.1%, respectively. Segmenting the prediction results into individual weeks reveals that EGB outperforms the LSP in all weeks by at least 17.7% for picking and 22.5% for packing. The highest improvements are observed in week 1 (60.7% and 65.7%). This is favorable, given that least information is available and variance in new hires' productivity is highest in the first week. EGB always outperforms the simple predictors in the picking function, while in packing it does so in all but two weeks. In those two weeks, only SMA5 is marginally better, by 3.4% in week 4 and 1.5% in week 7. In the more volatile first four of the eight weeks, EGB prevails over RF with only one exception, in week 2 in the picking function. However, it should be noted that even the simple predictors successfully improve the LSP's current model by at least 26.2% for picking and 35.7% for packing. The variation between SMA5 and SMA10 is low, with a maximum difference of 0.6 and 0.8 percentage points, respectively. The prediction results of the EGB model are achieved by tuning the hyperparameters for each function individually. A disadvantage is that the cross-validation required is computationally expensive. Table 3.5 shows that the trained hyperparameters are sufficiently insensitive when interchanged: using hyperparameters tuned in the training dataset of one function and applying them to the model for the other function yields similar results. Hence, it is sufficient to tune hyperparameters once for each dataset.

³Significance of $p < 0.01$ for cross-validated RMSE versus all other models

Table 3.4: Prediction results

Week	Model	Picking			Packing		
		RMSE	$\Delta_{\text{Model, LSP}}$	$\Delta_{\text{EGB, Model}}$	RMSE	$\Delta_{\text{Model, LSP}}$	$\Delta_{\text{EGB, Model}}$
All	LSP	29,60	0,0%	41,6%	67,18	0,0%	52,5%
	SMA ₅	21,67	26,8%	20,2%	43,23	35,7%	26,1%
	SMA ₁₀	21,76	26,5%	20,5%	42,66	36,5%	25,1%
	CMA	21,86	26,1%	20,9%	42,76	36,4%	25,3%
	RF	18,29	38,2%	5,5%	35,15	47,7%	9,1%
	EGB	17,29	41,6%	0,0%	31,94	52,5%	0,0%
1	LSP	34,58	0,0%	60,7%	81,83	0,0%	65,7%
	SMA ₅	21,42	38,1%	36,6%	55,36	32,3%	49,3%
	SMA ₁₀	21,42	38,1%	36,6%	55,36	32,3%	49,3%
	CMA	21,42	38,1%	36,6%	55,36	32,3%	49,3%
	RF	16,46	52,4%	17,5%	36,92	54,9%	24,0%
	EGB	13,58	60,7%	0,0%	28,07	65,7%	0,0%
2	LSP	29,79	0,0%	38,6%	71,90	0,0%	58,3%
	SMA ₅	20,58	30,9%	11,1%	33,96	52,8%	11,7%
	SMA ₁₀	21,45	28,0%	14,7%	33,95	52,8%	11,7%
	CMA	21,45	28,0%	14,7%	33,95	52,8%	11,7%
	RF	18,01	39,5%	-1,6%	31,75	55,8%	5,5%
	EGB	18,30	38,6%	0,0%	29,99	58,3%	0,0%
3	LSP	25,38	0,0%	17,7%	64,05	0,0%	50,0%
	SMA ₅	22,46	11,5%	6,9%	38,73	39,5%	17,3%
	SMA ₁₀	23,00	9,4%	9,1%	37,81	41,0%	15,3%
	CMA	23,37	7,9%	10,6%	37,99	40,7%	15,7%
	RF	21,29	16,1%	1,8%	35,14	45,1%	8,9%
	EGB	20,90	17,7%	0,0%	32,02	50,0%	0,0%
4	LSP	21,44	0,0%	30,1%	55,91	0,0%	38,8%
	SMA ₅	20,38	4,9%	26,4%	34,79	37,8%	1,6%
	SMA ₁₀	19,90	7,2%	24,7%	33,12	40,8%	-3,4%
	CMA	19,89	7,2%	24,6%	34,24	38,8%	0,0%
	RF	16,65	22,3%	10,0%	34,57	38,2%	1,0%
	EGB	14,99	30,1%	0,0%	34,23	38,8%	0,0%
5	LSP	28,33	0,0%	41,2%	48,45	0,0%	40,7%
	SMA ₅	20,14	28,9%	17,3%	32,42	33,1%	11,4%
	SMA ₁₀	19,36	31,7%	13,9%	32,73	32,4%	12,2%
	CMA	19,69	30,5%	15,4%	33,15	31,6%	13,3%
	RF	17,03	39,9%	2,2%	31,45	35,1%	8,6%
	EGB	16,66	41,2%	0,0%	28,74	40,7%	0,0%
6	LSP	27,90	0,0%	18,9%	55,45	0,0%	22,5%
	SMA ₅	27,80	0,4%	18,6%	47,61	14,1%	9,8%
	SMA ₁₀	27,45	1,6%	17,5%	45,92	17,2%	6,5%
	CMA	27,80	0,4%	18,6%	44,36	20,0%	3,2%
	RF	22,13	20,7%	-2,3%	42,37	23,6%	-1,4%
	EGB	22,64	18,9%	0,0%	42,95	22,5%	0,0%
7	LSP	27,19	0,0%	23,7%	39,48	0,0%	27,5%
	SMA ₅	22,40	17,6%	7,4%	32,31	18,2%	11,4%
	SMA ₁₀	21,47	21,0%	3,4%	28,20	28,6%	-1,5%
	CMA	23,57	13,3%	12,0%	28,87	26,9%	0,8%
	RF	22,40	17,6%	7,4%	26,65	32,5%	-7,4%
	EGB	20,75	23,7%	0,0%	28,63	27,5%	0,0%
8	LSP	26,74	0,0%	34,4%	59,63	0,0%	30,3%
	SMA ₅	20,91	21,8%	16,2%	42,62	28,5%	2,5%
	SMA ₁₀	21,47	19,7%	18,4%	42,16	29,3%	1,4%
	CMA	19,17	28,3%	8,6%	42,99	27,9%	3,3%
	RF	17,30	35,3%	-1,3%	39,71	33,4%	-4,7%
	EGB	17,53	34,4%	0,0%	41,57	30,3%	0,0%

Table 3.5: Recycling of hyperparameters

		Function used for hyperparameter tuning	
		Picking	Packing
RMSE at applied function	Picking	17.29	31.64
	Packing	17.02	31.94

3.5.2 Insights from variable analysis

To understand what drives prediction accuracy, we examine the permutation importance of EGB. Permutation importance provides information on the importance of each variable used in the model. It indicates the percentage by which the model's metric would worsen if that variable were omitted (Breiman (2001)). We observe that the variable order differs depending on the function (Figure 3.7). For picking, the *Product* impact category (see Table 3.3) with the two variables *Quantity processed* and *Orders processed* shows the highest permutation importance. On the other hand, *Weight processed* and *Volume processed* contribute little to prediction accuracy. The *Operator* impact category also improves prediction performance, with three variables in the top five. These three variables, *Total hours worked*, *Quantity processed* and *SMA5*, represent employees' experience and past performance. Other variables in these categories and all variables in the *Shift* impact category have a subsidiary effect.

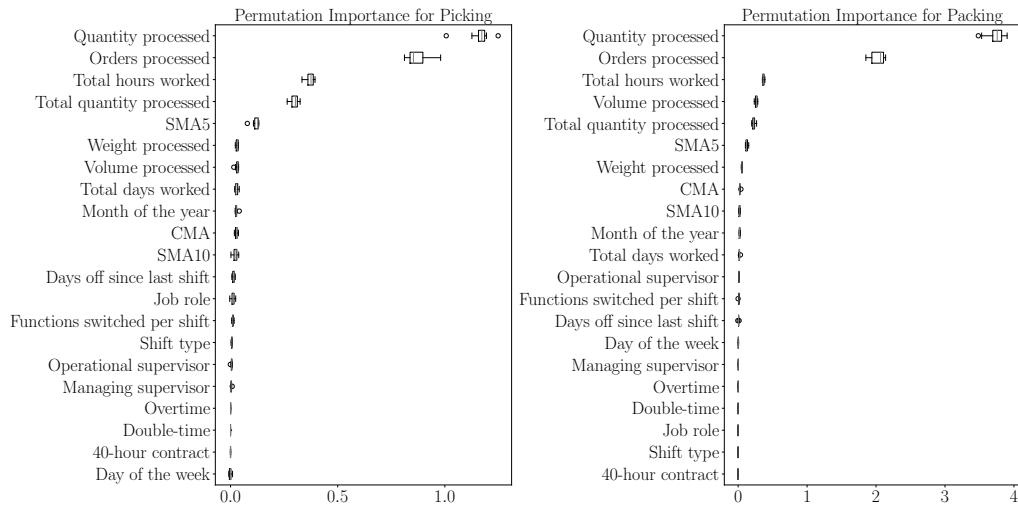


Figure 3.7: EGB permutation importance

In the packing function, three of the top four variables are in the *Product* impact category. Similarly to the picking function, *Quantity processed* and *Orders processed* are the most important variables. Variables in the *Operator* category do not have the same permutation importance as in the picking function, and the importance of the *Shift* impact category is as low as in picking. Overall, the key difference is that the *Product* category is most important in the pack-

ing function, while a combination of workload and experience is relevant in the picking function. A potential explanation is that picking is more complex due to longer travel paths and interactions with picking robots, which require experience to master the process. In contrast, packing has standardized work procedures that are easier to internalize.

The permutation importance ranking provides an indication of relationships between the variables and prediction performance. To confirm this, we repeat the prediction using a full factorial design for impact categories (Table 3.6), thereby testing all combinations of the three categories.

Table 3.6: Effect of impact category variables on EGB performance

Impact Categories	Picking			Packing		
	RMSE	MAE (SD)	$\Delta_{MAE,All}$ (95%KI)	RMSE	MAE (SD)	$\Delta_{MAE,All}$ (95%KI)
<i>Operator</i> × <i>Shift</i> × <i>Product</i>	17.29	11.2 (13.2)		31.94	19.5 (25.3)	
<i>Operator</i> × <i>Shift</i>	19.52	13.1 (14.5)*	-1.91 (-3.01, -0.82)	35.61	21.7 (28.2)**	-2.25 (-4.16, -0.34)
<i>Operator</i> × <i>Product</i>	17.29	11.5 (13.0)	-0.28 (-1.32, 0.75)	31.36	19.8 (24.4)	-0.27 (-2.04, 1.50)
<i>Shift</i> × <i>Product</i>	18.69	13.0 (13.5)*	-1.81 (-2.86, -0.76)	35.23	24.1 (25.7)*	-4.56 (-6.38, -2.74)
<i>Operator</i>	19.93	13.5 (14.6)*	-2.37 (-3.48, -1.27)	35.36	21.4 (25.7)**	-1.88 (-3.79, -3.90)
<i>Product</i>	21.57	15.3 (15.2)*	-4.13 (-5.25, -3.00)	35.21	25.2 (24.6)*	-5.68 (-7.46, -3.90)
<i>Shift</i>	22.01	15.5 (15.7)*	-4.31 (-5.45, -3.16)	40.03	26.7 (29.8)*	-7.21 (-9.18, -5.23)

Notes: *p < 0.01, **p < 0.05 (significance of MAE increase versus *Operator* × *Shift* × *Product*)

We observe that compared with using all variables, using only those from the *Product* and *Operator* impact categories achieves similar prediction performance. The Mean Absolute Error (MAE) differences of -0.28 (picking) and -0.27 (packing) are not significant. The *Shift* × *Product* and *Operator* × *Shift* category combinations have similar RMSE prediction performance but perform significantly worse than using all variables, confirming that the *Shift* variables are less important. This is further supported by the fact that prediction performance is lowest when variables from only the *Shift* category are used. Of the three instances of the experiment design that use only one variable category, *Operator* outperforms the others.

3.5.3 Impact on staff planning

The previous section has shown that our model outperforms the LSP's forecast, as well as all other baselines. To quantify the impact, we consider two cases: under and overestimating the actual productivity (U and O). Both cases may incur costs (c_U and c_O).

If productivity is underestimated, employees will work faster than assumed. In the best scenario, employees will finish earlier than planned, clock out, and be paid the corresponding salary, or use their remaining time to help others. However, in many other scenarios employees incur additional costs. Employees may finish as planned but sit idle before clocking out. They may also intentionally work slower during the day. Repeated or accumulated over-productivity may also lead to congestion in subsequent processes.

If productivity is overestimated, employees will need more time than planned to complete the given work. Additional costs will be incurred, as employees may be paid extra for the overtime. Employees in downstream processes may

need to wait, which will also result in additional wages. This may ultimately lead to reduced service levels as shipping cut-off times may be missed resulting in penalty payments or loss of reputation.

In any given shift, productivity may be underestimated for some employees and overestimated for others. In such cases, they may potentially support each other (Cantor and Jin (2019)), which will reduce U and O . We denote this support probability as s . As a result, Equation 3.5 determines the cost $C_{U,O}$ of under and overestimating productivity:

$$C_{U,O} = c_U U + c_O O - \min(U, O)s(c_U + c_O). \tag{3.5}$$

We compare $C_{U,O}$ in our EGB model with the LSP's current model and the best simple predictor (SMA5). To gain a comprehensive overview, we consider a total of 231 scenarios. For s we test probabilities between 0.0 and 1.0 in decimal steps. The lower bound of 0.0 describes the case where no support is given by over-productive employees to under-productive employees. The upper bound of 1.0 reflects a situation in which the entire remaining capacity of over-productive employees is used to support others. We analyze relative ratios of c_U and c_O . Starting in balance, where c_U equals c_O , we look in both directions until one assumes twice the value of the other. The cost reduction impact I_{EGB} is calculated using Equation 3.6:

$$I_{EGB} = \frac{C_{U,O} - C'_{U,O}}{C_{U,O}}, \tag{3.6}$$

and illustrated in Figure 3.8.

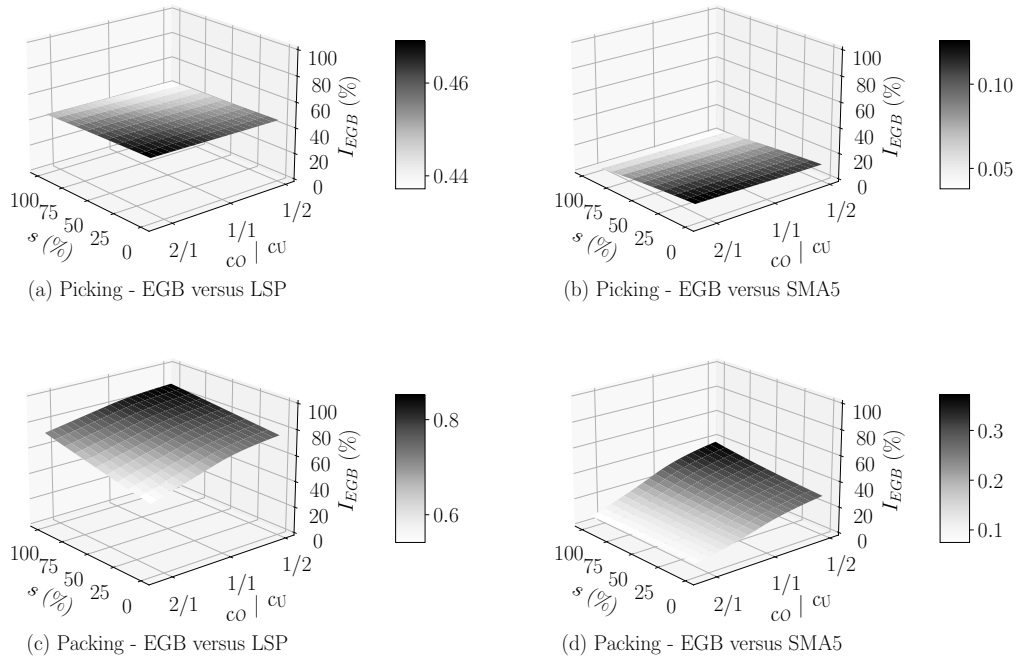


Figure 3.8: Cost reduction potential of EGB

For the picking function, the cost reduction impact of EGB versus the LSP's model and SMA5 is shown in Sub-figure 3.8a and 3.8b, respectively. EGB achieves a cost reduction of 44-47% compared with the LSP model, and 3-13% compared with the SMA5. Both show their highest potential where employees do not support each other and where c_O is twice as high as c_U . The lowest impact is achieved in situations where full support would be reasonable. With an increasing c_O/c_U ratio, the impact increases. As EGB produces more underestimations of actual productivity, it profits in situations where c_O exceeds c_U . On the support probability axis, the potential of EGB decreases with higher support values. This is because LSP and SMA5 are slightly better balanced between under- and overestimating productivity. As a result, they benefit more if mutual support is assumed.

Sub-figures 3.8c and 3.8d show the cost reduction potential of EGB versus the LSP's model (53-86%) and SMA5 (7-38%) in the packing function. Contrary to picking, the highest potential is achieved for low c_O/c_U ratios and high support probabilities. On the opposite side of the axes, the cost reduction is lowest. The LSP's model and SMA5 are highly imbalanced towards underestimating productivity; hence, higher c_U values result in increased cost reduction potential for EGB. Given EGB's balanced ratio between U and O, it also benefits from high support probabilities.

Overall, EGB shows a positive cost reduction potential in all scenarios across both functions. The magnitude in our case ranges from 3 to 86%. Owing to EGB's lower tendency for imbalance, it shows higher profits than losses in scenarios where support between employees is assumed. EGB shows even greater potential when the baselines are highly biased towards either under- or overestimating productivity.

3.6 DISCUSSION AND CONCLUSION

3.6.1 *Synthesis*

In this study, we build an EGB model to predict the productivity of newly hired warehouse associates in two functions, picking and packing. To do this, we devise a framework to identify the four main categories that impact on productivity: the warehouse in which the operation takes place, the operator, the shift in which the operator is placed and the product. We partner with a world-leading LSP to collect an extensive dataset with over 10 million data records across 16 variables. The dataset covers all but the warehouse category. We extract additional features from the 16 source variables in accordance with our framework of productivity impact factors (Table 3.3). For evaluation purposes, we introduce three baselines and RF in addition to the LSP's assumption of a four-week static ramp-up phase. First, we find that our EGB model in combination with the extracted features outperforms the LSP's model, all three baselines and RF. Compared with the LSP's model, we reduce the RMSE by 42% for the picking and 53% for the packing function. RMSE reductions of at least 20% and 25% are achieved versus the baselines, and the RF results are improved by 6% and 9%, respectively. Second, we show that hyperparameters tuned for one function

can be recycled for use in the EGB model of another function. This allows significant reductions in computation time, which is usually more expensive for EGB than for RFs. Third, we test a full factorial design of all three variable categories, and prove that variables in the Operator and Product categories are sufficient to achieve the same prediction accuracy as when all three categories are used. Hence, practitioners are recommended to focus their data collection on these two categories. Fourth, we calculate that the improved productivity prediction reduces the cost of over- and underestimating productivity by up to 47% in the picking and 86% in the packing function. This calculation uses a cost-based evaluation that considers under- and overestimations of productivity, and the potential equalization effect if both are apparent at the same time.

Given the extent of our collected data, the size and maturity of the analyzed warehouse, and the magnitude of improvements to predictions, we expect our approach using EGB to be generalizable to other warehouses. With regard to data beyond the scope of our research, we presume that impact factors such as age and training received will further improve prediction accuracy. However, warehouse factors have a more severe impact on generalizability. In facilities with substantially diverging setups in terms of location, design, maturity and automation, the one-to-one applicability of our approach may be reduced.

3.6.2 *Future research*

We propose two areas for future research. The first covers variables derived from our framework (Table 3.3) that could not be obtained during our data collection. At the employee level, this concerns the employee's age and training received. In particular, we hypothesize that the amount of training received might be used as a variable to further improve prediction accuracy. Furthermore, different warehouses should be considered in order to understand the effect of different locations, designs, sizes, process maturity and degree of automation, and particularly how well our findings can be generalized to other warehouse set-ups. The second potential area for further research covers other environments to which our model might be transferred. Within the scope of warehouses, this might be achieved by broadening the productivity prediction to other functions, such as inbound or put-away. The scope of future research might also be broadened to consider other manual work areas, such as the production or service industries.

This chapter has shown that the daily productivity of warehouse employees can be predicted using EGB, outperforming the LSP's assumption and other baselines. Together with Chapter 2, we investigated the two executing warehouse resources, i.e., equipment and labor. The following Chapter 4 considers the fundamental idea of physical warehouses. In a more disruptive approach, we investigate whether the warehouse concept of cross-docking can be applied virtually to allow dynamic transfers between delivery vehicles.

A MULTI-ALGORITHM FOR DYNAMIC TRANSSHIPMENTS WITH PERFECT SYNCHRONIZATION

The following chapter is based on Falkenberg et al. (2022).¹

4.1 INTRODUCTION

Direct deliveries within hours or even minutes have gained popularity, driven by the growth of e-commerce and changing consumer habits toward instant gratification (Dablanc et al. (2017)). Although direct deliveries have existed in the courier service industry for some time, these service offerings have been extended more recently to food outlets, online retailers and grocery stores. Examples include UberRush for VIP food delivery and other courier services, Amazon Prime Now for some of Amazon's retail business, and Gorillas and Food Rocket, which deliver groceries within minutes in Germany and the United States, respectively. Increased demand for direct deliveries, in combination with already rising parcel volumes in cities with constrained road networks, may intensify congestion and impair the environment (Zhen et al. (2021), Liu et al. (2019a)). Although carried out mainly by small vehicles, sometimes even with single unit-sized capacity, direct deliveries are often inefficient due to empty rides when returning to their origin (Dablanc et al. (2017), Lamngård and Andersson (2014)).

As a result, novel ways to reduce transport time are welcomed to bring down costs and emissions (Rincon-Garcia et al. (2020)). Concepts such as ride-sharing (Enzi et al. (2020)) and crowd-sourced logistics using ad hoc drivers (Arslan et al. (2018)) leverage existing traffic to limit the need for additional vehicles, although it is not always clear whether traffic congestion is actually mitigated (Beojone and Geroliminis (2021)). In addition to using existing traffic for new volumes, the footprint of current volumes can be mitigated by reducing the proportion of empty rides, such as backhauls (Lamngård and Andersson (2014)).

This chapter addresses the latter issue, and in particular the case of direct shipments where transport vehicles return to their origin after delivery, such as in food deliveries by drivers employed directly by restaurants. These direct deliveries are often sub-optimal because no routing is utilized to increase efficiency by allowing multi-stops, as in the Vehicle Routing Problem (VRP).

One solution to these inefficiencies is the concept of cross-docking, where transshipment locations with little or no storage capacity are used to transfer goods from one vehicle to another in order to reduce distribution costs (Grangier et al. (2017)). This may be beneficial in situations where multiple suppliers deliver to the same set of recipients. Rather than each supplier delivering to the drop-off locations individually, they meet at cross-docks where their loads are

¹*This unpublished working paper with the title "A multi-algorithm for dynamic transshipments with perfect synchronization" was written by Sven Falkenberg, Stefan Spinler and Arne Strauss, and is currently under review at the journal "European Journal of Operational Research".*

redistributed. Loads composed of various sub-loads from the different delivery locations are then delivered to the final recipients. For instance, supermarkets do not usually receive deliveries directly from, for example, a pasta factory delivering to multiple other supermarkets afterward. Rather, they receive trucks with the right mix of products from multiple suppliers, which have been consolidated at cross-docks. The drawbacks of cross-docking are the costs of the facility and handling processes (Masson et al. (2014b)), the dependence on suitable locations, and lack of flexibility to amend the location if circumstances change. In cities, locations for cross-docking opportunities are rare, and shorter delivery distances often do not justify the additional handling time. This may explain why cross-docking tends to be leveraged for long-haul transport rather than in urban areas.

To circumvent this problem, we propose the concept of dynamic transfers with perfect synchronization for direct deliveries. In this context, dynamic means that transfer points are flexible, and thus permit transshipment activities at any location. In order to allow this without the need for physical storage facilities, perfect synchronization between vehicles is required. The total distance can thereby be reduced if the increased mileage of the delivery trip can be offset by shorter return trips. Although transfers at any location are difficult to realize in practice, we consider them for two reasons. First, formulation of a continuous optimization problem offers advantages in terms of mathematical methods and computational speed. The continuous location coordinates can later be approximated to the nearest feasible ones if required. Second, in future scenarios with drone-based deliveries, it is reasonable to assume that transfers in controlled airspace will be viable at any location. The technical prerequisites for transfers between drones are described in a patent by IBM (Szondy (2017)).

Consider two simple examples illustrated in Figure 4.1, each with two delivery requests, r and r' . We assume that both are pizza deliveries, carried out by restaurant-owned vehicles, such as bicycles. Pizza r is picked up at pizza place p_r and delivered to the customer who ordered the pizza at d_r . After the delivery, bicycle v_r returns to the pizzeria p_r . The same holds for $v_{r'}$ analogously. Using dynamic transfers rather than individual deliveries, both bicycles could meet at transfer location $t_{r,r'}$ and exchange pizzas. After the exchange, v_r delivers pizza r' to the respective customer $d_{r'}$, and returns to its native pizza place p_r . Similarly, pizza r is delivered to d_r by $v_{r'}$, which then returns to $p_{r'}$. Assuming Euclidean distances, the total distance traveled is reduced from 16 to 12.9 (-19%) in Example 1 and from 17.9 to 16.9 (-6%) in Example 2.

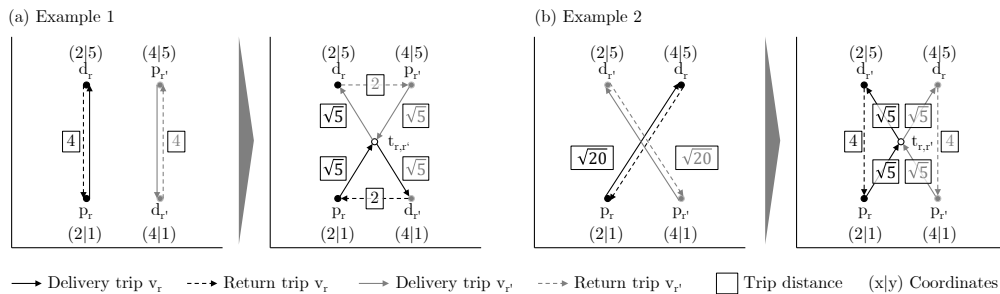


Figure 4.1: Total distance reduction through transfers

In this research, our comprehensive literature review shows that dynamic transfer locations with perfect synchronization have not yet been studied. We propose a multi-algorithm that solves large instances of the problem in under three minutes. For this purpose, we combine a problem-specific geometric algorithm to reduce the solution space, Sequential Least Squares Programming (SLSQP) to find the optimal transfer location between requests, and Edmonds' Blossom Algorithm (EBA) to determine which requests should be paired. Compared with direct deliveries, we show in a numerical simulation that dynamic transfers reduce costs by 15% on average, without violating time-window constraints. This is achieved mainly by cutting the non-value-adding return part of the total trip distance, thus still allowing short delivery times. We confirm the simulation results by applying our algorithm to real food delivery data in the city of Bordeaux, where we find an average cost reduction of 11%. We also show that the effort required for data collection and the complexity of the algorithm are reasonable, and that a simple central coordinator suffices to realize the reported potential.

The remainder of this chapter is structured as follows. In Section 4.2 we present a literature overview, and in Section 4.3 we define the problem and its complexity. We then present our multi-algorithm in Section 4.4, and provide numerical simulation and case study results in Section 4.5. In Section 4.6, we synthesize our findings, and highlight opportunities for future research.

4.2 RELATED WORK AND LITERATURE CONTRIBUTION

A broader consideration of transshipments has emerged in the last decade and particularly in recent years. Transshipments are presented mainly as a sub-problem of VRPs, specifically in relation to two variants, the Pickup and Delivery Problem (PDP) and the Dial-a-Ride Problem (DARP). The PDP is concerned with the pickup and delivery of goods and often focuses on minimizing cost or distance, whereas the DARP deals with the transport of people and also considers convenience factors. If transshipments are allowed, the two problem types are usually referred to as the Pickup and Delivery Problem with Transfers (PDPT) and the Dial-a-Ride Problem with Transfers (DARPT). An overview of research on these topics is shown in Table 4.1.

Given the complexity of the problem, most authors focus on static problem setups in which all necessary data are available upfront. Only Zhou and Lin (2019), Deleplanque and Quilliot (2013a), Thangiah et al. (2007) and Reinhardt et al. (2013) propose dynamic concepts. As the PDPT is NP-hard (Rais et al. (2014)), the proposed optimal solutions only consider problems of small size with six to nine requests (Rais et al. (2014), Pierotti and Van Essen (2021) and Cortés et al. (2010)). Speranza et al. (2016)'s survey of intermediate facilities highlights synchronization of vehicles' arrival and departure as an interesting variant for future research. However, all but four authors avoid the need for synchronization by providing storage opportunities. For the PDPT, some kind of physical storage place for goods is provided, whereas in the DARPT it is assumed that waiting times at transfer locations will be acceptable. Rais et al. (2014) develop a mixed-integer programming model for the PDPT with a syn-

Table 4.1: Research dealing with transfers of requests

Authors	Problem type	Transfer points		Approach		Method	Optimal
		Location	Storage	Design	Requests		
Takoudjou et al. (2012)	PDPT	Fixed	Yes	Static	50	Variable Neighbourhood Descent	No
Thangiah et al. (2007)	PDPT	Fixed	Yes	Dynamic	159	Insertion-based algorithm	No
Danloup et al. (2018)	PDPT	Fixed	Yes	Static	25	Genetic algorithm	No
Qu and Bard (2012)	PDPT	Fixed	Yes	Static	25	Greedy randomized adaptive search procedure	No
Wolfinger (2021)	PDPT	Fixed	Yes	Static	55	Large neighborhood search	No
Peng et al. (2019)	PDPT	Fixed	Yes	Static	5	Hybrid partial swarm optimization	No
Kerivin et al. (2008)	PDPT	Fixed	Yes	Static	15	Branch & Cut	No
Masson et al. (2014b)	PDPT	Fixed	Yes	Static	85	Branch & Cut	No
Wolfinger and Salazar González (2021)	PDPT	Fixed	Yes	Static	10	Branch & Cut	No
Minic and Laporte (2006)	PDPT	Fixed	Yes	Static	100	Insertion-based algorithm	No
Voigt and Kuhn (2021)	PDPT	Fixed	Yes	Static	100	Adaptive large neighborhood search	No
Sampaio et al. (2020)	PDPT	Fixed	Yes	Static	100	Adaptive large neighborhood search	No
Zhou and Lin (2019)	PDPT	Fixed	Yes	Dynamic	45	Adaptive boundary relaxation heuristic	No
Oertel (2001)	PDPT	Fixed	Yes	Static	40	Column generation heuristic	No
Mues and Pickl (2006)	PDPT	Fixed	Yes	Static	70	Column generation heuristic	No
Masson et al. (2013)	PDPT	Fixed	Yes	Static	193	Adaptive large neighborhood search	No
Rais et al. (2014)	PDPT	Fixed	No	Static	7	Mixed integer programming	Yes
Shang and Cuff (1996)	DARPT	Fixed	Yes	Static	9	Multi-objective heuristic	No
Masson et al. (2014a)	DARPT	Fixed	Yes	Static	193	Adaptive large neighborhood search	No
Pierotti and Van Essen (2021)	DARPT	Fixed	Yes	Static	9	Mixed integer linear programming	Yes
Cortés et al. (2010)	DARPT	Fixed	Yes	Static	6	Branch & Cut	Yes
Hammouda et al. (2020)	DARPT	Fixed	Yes	Static	10	Simulated annealing	No
Masson et al. (2012)	DARPT	Fixed	Yes	Static	193	Adaptive large neighborhood search	No
Deleplanque and Quilliot (2013b)	DARPT	Fixed	Yes	Static	144	Adaptive large neighborhood search	No
Bögl et al. (2015)	DARPT	Fixed	Yes	Static	40	Branch & Cut	No
Schönberger (2017)	DARPT	Fixed	Yes	Static	100	Genetic algorithm	No
Twist et al. (2021)	DARPT	Fixed	No	Static	500	Simulated annealing	No
Reinhardt et al. (2013)	DARPT	Fixed	No	Dynamic	555	Simulated annealing	No
Deleplanque and Quilliot (2013a)	DARPT	Dynamic	Yes	Dynamic	96	Insertion-based algorithm	No
Our research	PDPT	Dynamic	No	Static	100	Geometric heuristic with SLSQP and EBA	No

chronization constraint for transshipments. They optimize the problem solution with seven requests, but with fixed transfer points at pickup and delivery nodes. Reinhardt et al. (2013) investigate passengers with reduced mobility in an airport environment, who cannot be left alone and thus require synchronization during transfers. The authors present a simulated annealing-based greedy heuristic in combination with local search using advanced neighborhoods. They optimize the problem with up to 555 passengers in a dynamic environment by maximizing the number of passengers taken care of while minimizing excess time. However, they consider a maximum of four transfer points with fixed locations. Their solution is improved by Twist et al. (2021), at the expense of abandoning the dynamic environment. To our knowledge, only Deleplanque and Quilliot (2013a)’s study discusses the use of dynamic transfer points, but they do not impose synchronization. Given the dynamic environment in which they analyze up to 96 requests, it is unclear how many are considered simultaneously.

In summary, given the problem’s complexity, most research focuses on non-optimal solutions, considers few transfer points with fixed locations, and offers storage opportunities to avoid synchronization constraints.

However, in situations where direct deliveries are made in urban areas, for example by restaurant-managed drivers, the number of requests may be high, and physical storage facilities to allow transfers may be costly or even infeasible, and sometimes unsuitable for perishable products. Nevertheless, urban areas with high road density allow drivers to meet almost anywhere for immediate product transfers. Our research addresses this opportunity by providing a multi-algorithm to solve large problem instances in which the number of trans-

shipment points is unlimited, their locations dynamic, and transfers synchronized. Our research thus contributes to the presented literature in considering a unique combination of transfer point characteristics that have not previously been considered.

4.3 PROBLEM DEFINITION

4.3.1 Notation and cost evaluation

We consider a Cartesian coordinate system in a two-dimensional Euclidean space. We define a set of delivery requests R and a set of homogeneous vehicles V with unit-sized capacity Q . The load of request $r \in R$ is transported from pickup point $p_r \in P$ to delivery point $d_r \in D$. All requests are known upfront, and their loads are dispatched at once and must be delivered within a time-window constraint tw_r . Each load consumes the capacity Q , so a vehicle can carry only one load at any given time. A vehicle $v_r \in V$ conducts one trip, starting and ending at p_r , resulting in a fleet size of $|V| = |R|$. We consider two options. The first represents the classic case of direct shipments, where vehicle v_r picks up r at p_r , delivers the load to d_r and returns to p_r . In the second option, a central decision-maker exists that coordinates a network of businesses that allow transshipments between each other. In this option, we relax the pairing constraint between v_r and r and allow v_r to meet one, and only one, other $v_{r'}$ at any transfer point $t_{r,r'}$. The constraint of a maximum of one transfer, also used by Deleplanque and Quilliot (2013a), reflects realistic setups, as further transfers would reduce the perceived convenience (Masson et al. (2014b)). At the transfer point, v_r hands over r to $v_{r'}$ and in turn receives r' , after which v_r then delivers r' to $d_{r'}$ and returns to p_r . Similarly, $v_{r'}$ delivers r to d_r and returns to $p_{r'}$. Loads cannot be stored at $t_{r,r'}$, so synchronized transfers are required and transfer times are ignored. Hence, waiting time $wt_{v_r, v_{r'}}$ applies if v_r or $v_{r'}$ arrive at $t_{r,r'}$ before the other. The maximum number of transfer points is $\frac{|V|}{2}$ if each vehicle meets another.

In our problem formulation, one time unit equals one distance unit. The Euclidean distance between two coordinates c and c' is given by Equation 4.1, and thus holds the triangle inequality:

$$d(c, c') = \sqrt{(c'_x - c_x)^2 + (c'_y - c_y)^2}. \quad (4.1)$$

If two requests, r and r' , are under investigation for a potential transfer, we denote them as a request pair $rp_{r,r'}$. For each request pair, we differentiate between the total cost of either not using a transfer location ($c(rp_{r,r'})$) or using one ($c(rp_{r,r'}, t_{r,r'})$).

In Equation 4.2, we determine the cost of a request pair without using a transfer location by summing the delivery and return distances for both requests, r and r' :

$$c(rp_{r,r'}) = d(p_r, d_r) + d(d_r, p_r) + d(p_{r'}, d_{r'}) + d(d_{r'}, p_{r'}). \quad (4.2)$$

In Equation 4.3, if a transfer location $t_{r,r'}$ is used, the cost of $rp_{r,r'}$ is calculated by combining the distance traveled by v_r and $v_{r'}$. The cost of v_r is calculated as the distance from pickup location p_r to transfer location $t_{r,r'}$, v_r 's potential waiting time for $v_{r'}$ at $t_{r,r'}$ if the distance from $p_{r'}$ to $t_{r,r'}$ is longer, the distance traveled by v_r to deliver r' to $p_{r'}$ after the transfer, and v_r 's return to p_r . The cost of $v_{r'}$ is calculated analogously:

$$\begin{aligned} c(rp_{r,r'}, t_{r,r'}) = & d(p_r, t_{r,r'}) + \max(d(p_{r'}, t_{r,r'}) - d(p_r, t_{r,r'}), 0) \\ & + d(t_{r,r'}, d_{r'}) + d(d_{r'}, p_r) \\ & + d(p_{r'}, t_{r,r'}) + \max(d(p_r, t_{r,r'}) - d(p_{r'}, t_{r,r'}), 0) \\ & + d(t_{r,r'}, d_r) + d(d_r, p_{r'}). \end{aligned} \quad (4.3)$$

A feasible solution with cost $c(rp_{r,r'}, t_{r,r'})$ is given if the delivery times of r and r' do not violate time constraints tw_r and $tw_{r'}$ respectively. Equation 4.4 shows the constraint formulation for r :

$$d(p_r, t_{r,r'}) + \max(d(p_{r'}, t_{r,r'}) - d(p_r, t_{r,r'}), 0) + d(t_{r,r'}, d_r) \leq tw_r. \quad (4.4)$$

4.3.2 Problem complexity

The complexity of the formulated problem increases with the number of requests. A simple problem with four requests is shown in Figure 4.2.

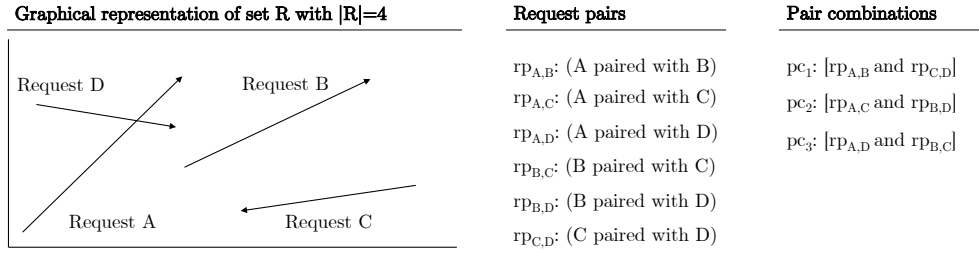


Figure 4.2: Example of transfer complexity with four requests

Each request may involve a transshipment with one of the other requests, resulting in a total of six request pairs to be evaluated. As the order within pairs does not matter, the total number of potential request pairs $|RP|$ is calculated using the binomial coefficient (Equation 4.5):

$$|RP| = \binom{n}{k} = \binom{|R|}{2} = \frac{|R|!}{2!(|R|-2)!} \quad |R| \in \mathbb{N} + 1. \quad (4.5)$$

It must then be decided which combination of request pairs minimizes the total distance. Choosing one pair affects the solution space, as other pairings become unavailable. If, for example, request A in Figure 4.2 is chosen to meet with B, then neither A nor B can be involved in a transfer with C or D. The only remaining request pair which can be added to the current solution is

the transfer between C and D. To calculate the total number of possible pair combinations PC, we use Equation 4.6:

$$|PC| = \begin{cases} \prod_{m=0}^{M=\frac{|R|-2}{2}} (|R| - 2k - 1) & |R| \in 2\mathbb{N} \\ \prod_{m=0}^{M=\frac{|R|-1}{2}} (|R| - 2k - 1) & |R| \in 2\mathbb{N} + 1. \end{cases} \quad (4.6)$$

While the number of request pairs grows moderately with r , the pair combinations increase drastically owing to the property of geometric series. Appendix C.1 shows the number of request pairs and pair combinations for various values of $|R|$. A realistic magnitude of 50 requests requires consideration of 1,225 request pairs and investigation of 5.8×10^{31} pair combinations, hence becoming infeasible.

4.3.3 Brute-force solution

A brute-force algorithm to solve the problem is shown in Algorithm 1.

Algorithm 1 Brute-force algorithm for direct deliveries with dynamic transfers

Require: CG, R, P, D, TW, T

Ensure: $|R| \in 2\mathbb{N}$, $d(c, c') \leftarrow \sqrt{(c'_x - c_x)^2 + (c'_y - c_y)^2}$

```

1: procedure BRUTEFORCE(R)
2:   RP = All request pairs  $rp_{r,r'}$  of  $r \cup r' \in R$  with  $r \neq r'$ 
3:   for  $rp_{r,r'}$  in RP do
4:     CostWithoutTransfer =  $c(rp_{r,r'})$ 
5:     BestCost = CostWithoutTransfer
6:     for t in T do
7:       CostWithTransfer =  $c(rp_{r,r'}, t_{r,r'})$ 
8:       if CostWithTransfer < BestCost then
9:         if  $d(p_r, t_{r,r'}) + \max(d(p_{r'}, t_{r,r'}) - d(p_r, t_{r,r'}), 0)$ 
            $+ d(t_{r,r'}, d_{r'}) \leq tw_r$ , then
10:        if  $d(p_{r'}, t_{r,r'}) + \max(d(p_r, t_{r,r'}) - d(p_{r'}, t_{r,r'}), 0)$ 
            $+ d(t_{r,r'}, d_r) \leq tw_{r'}$ , then
11:          BestCost = CostWithTransfer
12:        Improvement $_{r,r'}$  = CostWithoutTransfer - BestCost
13:   PC = All combinations of  $rp \in RP$  such that no element of a  $rp$ 
         appears in another pair within a pair combination  $pc$ 
14:   ImprovementOfBestPC = 0
15:   for  $pc$  in PC do
16:     CostOfPC =  $\sum_{n=1}^{RP} Improvement_{r,r'}$ 
17:     if CostOfPC > ImprovementOfBestPC then
18:       ImprovementOfBestPC = CostOfPC

```

The input is given in the form of a coordinate grid CG with fixed side length and sets R with P and D, time-window constraints TW and potential transfer locations T. The latter are all, theoretically infinite, coordinates in CG. First, all potential request pairs are created by pairing each r with all $r' \in R \setminus r$. For each request pair, the baseline cost $c(rp_{r,r'})$ of direct delivery without transfer

is calculated. Next, the lowest cost of using a transfer location $c(rp_{r,r'}, t_{r,r'})$ is calculated for each request pair. Therefore, each coordinate in the defined coordinate grid is tested as a potential transfer location. If the delivery time using the transfer location does not violate the time-window constraint, and $c(rp_{r,r'}, t_{r,r'})$ improves $c(rp_{r,r'})$, then a new best cost is found and the improvement is stored. Ultimately, the total improvement for each possible pair combination is calculated by adding the improvements for each request pair. The pair combination with the highest total improvement provides the final solution. For example, suppose we have 50 delivery requests in a 100×100 grid and, for simplicity, the pickup, delivery and transfer points are located only at integer coordinates. Neglecting the minuscule calculations, one would need to calculate the baseline cost for 1,225 transfer scenarios. Then, for each of the 1,225 scenarios, the cost of 10,000 varying transfer locations would be computed and eventually fed into 5.8×10^{31} request pair combinations.

Finding the optimum by conducting a full enumeration, as in this algorithm, is inefficient and computationally expensive. Three steps in Algorithm 1 are especially problematic in that regard. First, considering all request pairs (line 2) is excessive and increases the complexity of the following steps in finding the optimal transfer location and pair combinations. Second, testing all locations (lines 6-12) to find the optimal transfer location is cumbersome because, theoretically, an infinite number of locations in a two-dimensional Euclidean space must be checked. Third, for a problem of practical size, the number of pair combinations to be checked (lines 13-18) becomes infeasible given the nature of geometric series. In the next section, we present a problem-specific algorithm consisting of three sub-algorithms to solve each of these problems in reasonable computation time.

4.4 ALGORITHM FORMULATION

4.4.1 Sub-algorithm 1: Reducing the solution space

We show that a full enumeration is computationally expensive. One difficulty is that all request pairs are initially considered, although, intuitively, some pairs should not be examined. Two examples of such cases are shown in Figure 4.3.

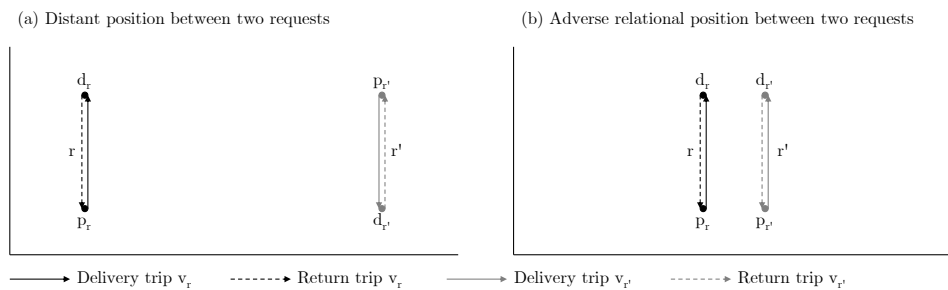


Figure 4.3: Exemplary request pairs with no improvement through transfers

Some deliveries (Figure 4.3a) are so far apart that either the combined distance to a common transfer location exceeds the combined distance of the two

delivery trips, or the distance to the new delivery location using a transfer location exceeds the time-window constraint. To reduce the solution space based on unfavorable distances, we can simply remove request pair $rp_{r,r'}$ from set RP according to Equation 4.7:

$$RP \setminus rp_{r,r'} \text{ if } = \begin{cases} d(p_r, p_{r'}) \geq d(p_r, d_r) + d(p_{r'}, d_{r'}) \text{ or} \\ d(p_r, d_{r'}) \geq tw_{r'} \text{ or} \\ d(p_{r'}, d_r) \geq tw_r. \end{cases} \quad (4.7)$$

Other deliveries (Figure 4.3b) are in close proximity, but their relative position makes a transfer obsolete. In the example, the two deliveries run in parallel and in the same direction. Hence, meeting at a transfer location would only increase the total distance.

The conditions under which the relative position of two requests leads to a situation that offers no improvement potential are not intuitive. We propose a structured approach to evaluate those positions to steer the decision process. First, we define how the relational position of two requests, r and r' , is determined. To do so, we rotate the coordinate plane, such that the request r under consideration has a vertical position and is directed upwards (Figure 4.4).

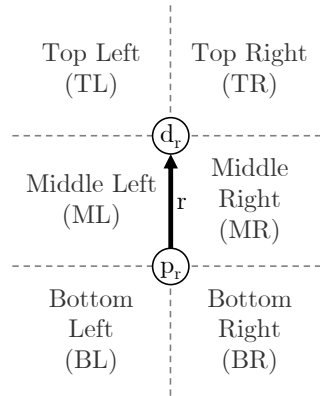
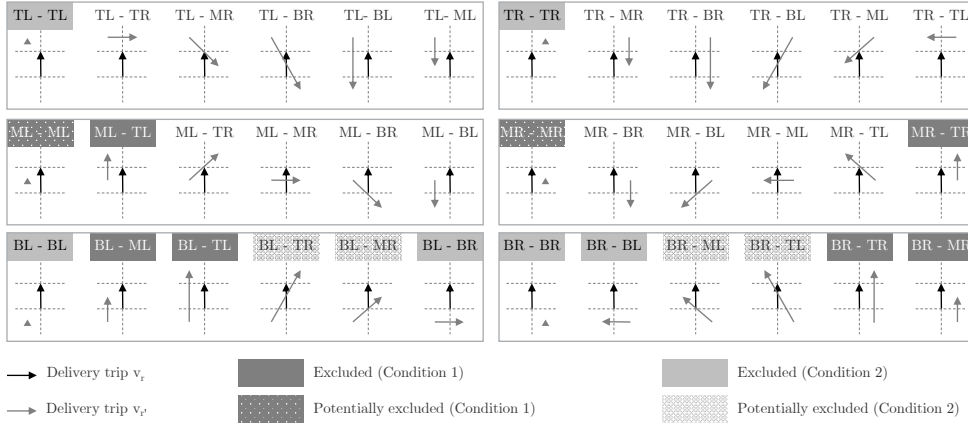


Figure 4.4: Surrounding areas from the perspective of a request

From this perspective, we can define whether a coordinate is located above d_r (top), below p_r (bottom) or between p_r and d_r (middle), and whether it is placed to the left or right of request r . This yields six surrounding areas: Top Left (TL), Top Right (TR), Middle Left (ML), Middle Right (MR), Bottom Left (BL) and Bottom Right (BR). Any other request r' with $p_{r'}$ and $d_{r'}$ can be allocated to one of these six areas. Consequently, we obtain 36 potential positions of request r' in relation to r (Figure 4.5). For example, the first case (TL-TL) describes a request r' that starts somewhere in the TL area and ends in the same area.

To determine which cases offer no improvement potential, one must understand that two independent conditions favor a transfer of goods.

Condition 1: At least part of r' is next to r (areas ML or MR) and the direction of r' is opposite to that of r (that is any direction facing downwards). For an example, see Figure 4.1a.

Figure 4.5: All positions of request r' in relation to r

Condition 2: Requests r and r' have a direct or indirect crossing point. An indirect crossing point is apparent if a crossing would exist assuming the vector of request r is extended beyond the delivery location d_r . For an example, see Figure 4.1b.

Based on *Condition 1*, we exclude six cases (ML-TL, MR-TR, BL-ML, BL-TL, BR-TR, BR-MR) in which r' is (in some part) next to r but in a similar direction. As both, r and r' are directed upwards with no direct crossing, there is no transfer location that could improve the base scenario. Two cases, ML-ML and MR-MR, can only be excluded if r' is not directed downwards. We also exclude six cases (TL-TL, TR-TR, BL-BL, BL-BR, BR-BR, BR-BL) based on *Condition 2*. In these scenarios, no direct or indirect crossing exists between r and r' . Another four cases (BL-TR, BL-MR, BR-ML, BR-TL) are excluded if r' undercuts r , resulting in a no-cross situation. To determine which of the 36 cases holds for a given request pair, we propose the transformation shown in Figure 4.6.

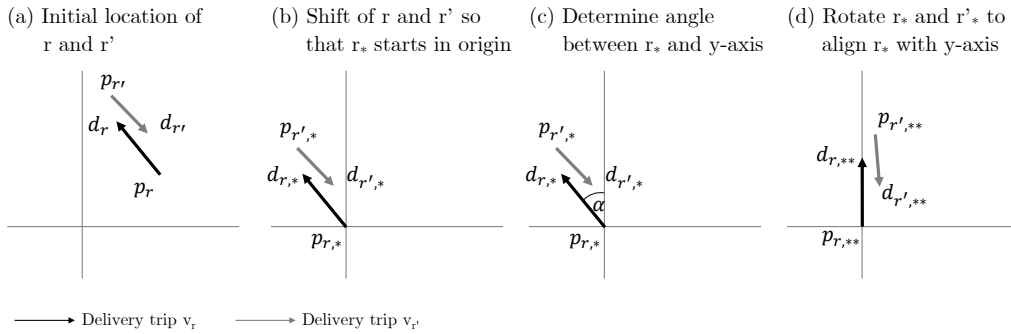


Figure 4.6: Transformation of transfer scenarios to determine relational positions

First, we take the initial request pair (Figure 4.6a) consisting of request pair r and r' , and shift it so that $p_{r,*}$ starts at the origin (Figure 4.6b). Equations 4.8

show how to transform the initial x and y -coordinates (c_x, c_y) of the two requests into shifted coordinates (c_{x^*}, c_{y^*}) :

$$\begin{aligned} c_{x^*} &= c_x - p_{r,x} \\ c_{y^*} &= c_y - p_{r,y}. \end{aligned} \quad (4.8)$$

Second, we calculate the angle α between the vector of r_* and the y -axis (Figure 4.6c) using Equation 4.9:

$$\alpha = \cos^{-1} \left(\frac{d_{r,y^*}}{\sqrt{d_{r,x^*}^2 + d_{r,y^*}^2}} \right). \quad (4.9)$$

Finally, we use the obtained angle α to rotate the entire request pair (Figure 4.6d) using Equation 4.10 so that $p_{r,**}$ starts at the origin and $d_{r,**}$ lies above it on the y -axis:

$$\begin{bmatrix} c_{x^{**}} \\ c_{y^{**}} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \times \begin{bmatrix} c_{x^*} \\ c_{y^*} \end{bmatrix}. \quad (4.10)$$

$\text{RelationalPosition}(rp_{r,r'})$ is the function that conducts the steps in Equations 4.8 to 4.10 to determine which of the 36 relational positions between r and r' holds, and to decide whether to exclude $rp_{r,r'}$ from RP. Algorithm 2 then reduces the initial solution space in which all request pairs are considered. It first generates all request pairs, then excludes them if the distance between two requests is too large to find meaningful transfer locations and finally applies the $\text{RelationalPosition}(rp_{r,r'})$ function to exclude further request pairs if their relational position is unfavorable.

Algorithm 2 Solution Space Reduction

Require: R, P, D, TW

Ensure: $|R| \in 2\mathbb{N}$, $d(c, c') \leftarrow \sqrt{(c'_x - c_x)^2 + (c'_y - c_y)^2}$

- 1: **procedure** SOLUTIONSPACEREDUCTION(RP)
 - 2: RP = All request pairs $rp_{r,r'}$ of $r \cup r' \in R$ with $r \neq r'$
 - 3: **for** $rp_{r,r'}$ in RP **do**
 - 4: **if** $d(p_r, p_{r'}) \geq d(p_r, d_r) + d(p_{r'}, d_{r'})$ **then** $RP \setminus rp_{r,r'}$
 - 5: **else if** $d(p_r, d_{r'}) \geq tw_{r'}$ **then** $RP \setminus rp_{r,r'}$
 - 6: **else if** $d(p_{r'}, d_r) \geq tw_r$ **then** $RP \setminus rp_{r,r'}$
 - 7: **else** $\text{RelationalPosition}(rp_{r,r'})$
-

4.4.2 Sub-algorithm 2: Finding the optimal transshipment location

Using the previous algorithm, we have reduced the solution space by excluding request pairs whose cost cannot be improved through transfer locations. For each of the remaining request pairs, we must now determine the best feasible transfer location $t_{r,r',\text{Opt}}$. Only then can we compare the cost without transfers $c(rp_{r,r'})$ to the cost with transfers $c(rp_{r,r'}, t_{r,r'})$ and decide whether a transfer

location improves the initial situation of direct deliveries. To find $t_{r,r',\text{Opt}}$, we formulate the following minimization problem (Equations 4.11):

$$\begin{aligned}
& \min_{t_{r,r',\text{Opt}}} d(p_r, t_{r,r'}) + d(p_{r'}, t_{r,r'}) \\
& \quad + \max(d(p_{r'}, t_{r,r'}) - d(p_r, t_{r,r'}), d(p_r, t_{r,r'}) - d(p_{r'}, t_{r,r'})) \\
& \quad + d(t_{r,r'}, d_r) + d(t_{r,r'}, d_{r'}) \\
& \text{s.t.} \quad d(p_r, t_{r,r'}) + \max(d(p_{r'}, t_{r,r'} \\
& \quad - d(p_r, t_{r,r'}), 0) + d(t_{r,r'}, d_{r'}) \leq tw_r \\
& \quad d(p_{r'}, t_{r,r'}) + \max(d(p_r, t_{r,r'} \\
& \quad - d(p_{r'}, t_{r,r'}), 0) + d(t_{r,r'}, d_r) \leq tw_{r'}.
\end{aligned} \tag{4.11}$$

The objective function minimizes the combined delivery cost of r and r' when using a transfer location. The cost of delivery is based on Equations 4.2 and 4.3, but ignores the return distances $d(d_r, p_r)$ and $d(d_{r'}, p_{r'})$ as they do not depend on $t_{r,r'}$. The two constraints ensure that the delivery time when using a transfer location does not exceed time windows tw_r and $tw_{r'}$.

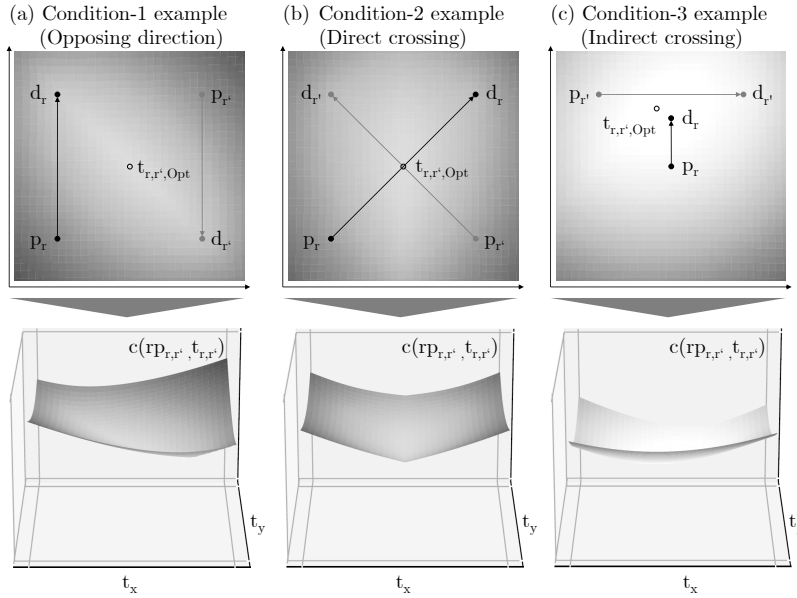


Figure 4.7: Costs of various transfer locations

Ignoring time window constraints, Figure 4.7 shows the location of $t_{r,r',\text{Opt}}$ for three example request pairs. The upper illustration (Figure 4.7a) shows two requests falling under *Condition 1*, as introduced in subsection 4.4.1, where requests are next to each other in opposing directions. The diagonal lighter shaded area from top left to bottom right indicates that $t_{r,r'}$ generally yields lower cost if the distance to the two pickup locations is equal, leading to no waiting time at the transfer location. The bottom illustration shows a three-dimensional representation of the total cost $c(rp_{r,r'}, t_{r,r'})$ for the upper example, depending on x and y -coordinates for $t_{r,r'}$. Figures 4.7b and 4.7c show examples for *Condition 2* with a direct and indirect crossing, respectively. All three examples show convex cost functions for $t_{r,r'}$, which holds for all combi-

nations of two requests and one transfer location. This is owing to the fact that the squared Euclidean distance function between two coordinates is convex. As the sum of convex functions is again convex, our cost function, comprising of four distance functions, is also convex.

To further simplify the optimization problem, we state that it never makes sense for a request r to wait for another request r' , as it is always better to be on the move than to waste time waiting. We conclude that the optimal transshipment point will be located at a coordinate equally distant from p_r and $p_{r'}$. Using the x and y -coordinates of both requests $(p_{r,x}, p_{r,y}, p_{r',x}, p_{r',y})$, we derive the equation on which the optimal transfer location must lie. Equation 4.12 describes the midpoint between p_r and $p_{r'}$ and is thus a potential transshipment point with no waiting time:

$$t_{x,y,\text{mid}} = \left(\frac{p_{r,x} + p_{r',x}}{2}, \frac{p_{r,y} + p_{r',y}}{2} \right). \quad (4.12)$$

The equation sought for the optimal transfer location is perpendicular to the equation connecting p_r and $p_{r'}$, hence its slope a_t is described by Equation 4.13:

$$a_t = -\frac{p_{r',x} - p_{r,x}}{p_{r',y} - p_{r,y}}. \quad (4.13)$$

The resulting Equation 4.14 describes the function on which $t_{r,r',\text{Opt}}$ will lie:

$$t_y = -\frac{p_{r',x} - p_{r,x}}{p_{r',y} - p_{r,y}} t_x + \frac{p_{r,y} + p_{r',y}}{2} + \frac{p_{r',x} - p_{r,x}}{p_{r',y} - p_{r,y}} \frac{p_{r,x} + p_{r',x}}{2}. \quad (4.14)$$

This is used in the optimization problem to circumvent the $\max()$ function and to make the objective function dependent on only one variable (t_x instead of t_x and t_y).

To find the optimal transfer location per request pair efficiently, we use the SLSQP algorithm. SLSQP takes a sequential approach to nonlinear optimization with inequality constraints (Kraft (1988)). It is a generalization of the Karush-Kuhn-Tucker (KKT) conditions which consider problems with equality constraints, while KKT is itself a generalization of the Newton method for unconstrained problems (Nocedal and Wright (2006)). For the mathematical optimization, the function to be optimized must be twice continuously differentiable. Both inequality constraints and differentiability are true for our problem. As the starting point for our optimization, we use Equation 4.12.

We define $\text{OptimalTransferLocation}(rp_{r,r'})$ as the function that implements the SLSQP algorithm to find the optimal transfer location for two requests, r and r' .

4.4.3 Sub-algorithm 3: Optimizing the combination of request pairs

Having determined the lowest cost possible for each request pair $rp_{r,r'}$, we must determine which pairs to combine in order to minimize the total cost. Rather than testing all pair combinations $|PC|$, we transfer our problem to the area of graph theory and use existing algorithms to solve the known assignment problem.

The generalized assignment problem deals with matching two groups of the same size with each other (bipartite graph), such as a group of drivers with a group of vehicles. In cases where not all drivers are qualified to drive every car, the Hopcroft-Karp algorithm (Hopcroft and Karp (1971)) aims to find the maximum number of matches. Furthermore, if drivers have preferences for particular vehicles, the Hungarian algorithm (Kuhn (1955), Munkres (1957)) produces solutions for maximum weighted matchings. However, there are no two distinct groups to be matched in our problem, as each request r could be matched with any other request $r' \neq r$. Hence, we use Edmonds' Blossom Algorithm (EBA) to create maximum matchings, as it does not require the general graph to be bipartite (Edmonds (1965a), Edmonds (1965b)).

The EBA considers vertices $v \in V$ and edges $e \in E$, defining an undirected graph $G = (V, E)$. It finds the maximum matchings M , with a matching m being defined as an edge connecting two vertices while each vertex is connected to a maximum of one edge. The EBA builds an augmenting path within G , which is a path starting and ending at a free vertex while alternating matched and unmatched edges. After finding such an augmented path, the algorithm inverts it to increase the number of matchings by one. A crucial part of the EBA is that when a circular path of odd length is discovered, the algorithm contracts it (the blossom). This postpones the decision to create an augmented path and supports the linkage of further augmenting paths to the existing ones. For a detailed explanation, see Feil (2016). For our purpose, we use Kolmogorov's (2009) extension of the EBA to solve the maximum weight-matching problem.

To apply the extended EBA to our problem, we must define three elements: vertices, edges and weights. For the vertices, we define the requests in R . The edges are the connections between two requests r and r' and represent our request pairs $rp_{r,r'}$ (found via sub-algorithm 1). The weights are the potential distance improvements per request pair; that is, either zero for the independent fulfillment cost of both requests or, if better and feasible, the improvement of using a transfer location (found via sub-algorithm 2). We obtain an undirected graph in which each request-node is connected to all other request-nodes. We define `MaximumWeightedMatchings(RP)` as the function that implements the weighted EBA to find the best combination of request pairs minimizing the total cost (sub-algorithm 3).

We can now formulate our multi-algorithm by combining all three sub-algorithms (Algorithm 3) to minimize the cost of direct deliveries by using dynamic transshipments with perfect synchronization.

Algorithm 3 Algorithm for direct deliveries with dynamic transshipments

Require: R, P, D, TW

Ensure: $|R| \in 2\mathbb{N}$, $d(c, c') \leftarrow \sqrt{(c'_x - c_x)^2 + (c'_y - c_y)^2}$

- 1: **procedure** `DYNAMICTRANSFERALGORITHM(R)`
 - 2: `SolutionSpaceReduction(RP)`
 - 3: `OptimalTransferLocation(rpr,r')`
 - 4: `MaximumWeightedMatchings(RP)`
-

4.5 ALGORITHM RESULTS

4.5.1 Numerical simulation

Our algorithm is tested on 30 randomly generated scenarios, each of which is simulated 100 times. For each scenario, we use a squared coordinate grid with a side length of 100. The scenarios differ in the number of requests considered and the time-window constraints imposed. The number of randomly generated requests assumes values of 10, 20, 30, 40, 50, and 100. For each request scenario, we consider maximum delivery times of 10, 20, 30, 40 and 50. The pickup locations are randomly selected, with the implication that no two pickup locations can be located at the same coordinate. The delivery locations within each request are chosen such that they can be directly delivered to without violating the time constraints. We also assume that each delivery location is unique. As a naming convention for each scenario, we use $R|R| - TW|TW|$. For example, $R_{10}-TW_{10}$ describes a scenario with 10 requests and a time-window constraint of 10. For each scenario, we average the results across all 100 simulations. We report the total initial cost, i.e., the delivery and return trip of all requests in the given scenario and the standard deviation of the initial cost. We show separately the cost of delivery and the cost of returns, which are equal in the case of direct deliveries. We also report the total and percentage cost improvement achieved by our algorithm, and the total and percentage change of the delivery and return trip. The results are shown in Table 4.2.

With regard to total cost, we see two intuitive trends. The total cost grows with increasing numbers of requests and larger time windows in a given request scenario. A similar trend is observed in the total cost improvement. The more requests are available in a fixed grid, the more likely it becomes that two requests find a transfer location that reduces their combined cost. Increasing the time-window constraint allows the search radius of each request r to be extended to partner with another request r' , which also increases the cost reduction potential. Scenario $R_{10}-TW_{10}$ achieves almost no cost improvement with an average of only 0.1%. Given the low number of requests and narrow time windows, only one transshipment opportunity is found in every ten simulations. With the most relaxed time-window constraint of 50, our algorithm improves the R_{10} scenario by 8.4%. Comparing $R_{10}-TW_{10}$ with scenarios in which either the requests or the time windows are increased by 100% reveals that time windows have a greater impact on the solution. $R_{10}-TW_{20}$ shows a 1.5% improvement, while $R_{20}-TW_{10}$ produces a 0.8% improvement. A similar relationship is observed if, for example, $R_{20}-TW_{10}$ is compared with $R_{20}-TW_{20}$ and $R_{40}-TW_{10}$, or $R_{50}-TW_{10}$ with $R_{50}-TW_{20}$ and $R_{100}-TW_{10}$. The general performance improvement in relation to the number of requests and time-window constraints is shown in Figure 4.8. Starting with TW_{10} , it can be seen that doubling the time window (TW_{20}) increases the improvement approximately threefold. Another duplication to TW_{40} shows an improvement increase by a factor of two. We also notice that, especially for a lower number of requests between 10 and 30, larger time windows enable steeper improvements.

While TW_{10} and TW_{20} have a rather constant increase across all requests, TW_{30} and TW_{40} almost triple and TW_{50} doubles the impact when requests

Table 4.2: Numerical simulation results

Scenario	R	tw	Cost at initial setup		Cost improvement after algorithm			Transfers
			Total (Std. Dev.)	Delivery/Return	Total (%)	Delivery (%)	Return (%)	
R10-TW10	10	10	131 (15)	65	0.2 (0.1)	-0.1 (-0.2)	0.3 (0.5)	0.1
R10-TW20	10	20	257 (31)	128	4.2 (1.5)	-0.5 (-0.4)	4.7 (3.7)	0.4
R10-TW30	10	30	380 (49)	190	14.6 (3.8)	-2.4 (-1.3)	17.0 (8.9)	1.1
R10-TW40	10	40	499 (64)	249	25.3 (4.9)	-3.7 (-1.5)	29.0 (11.6)	1.4
R10-TW50	10	50	605 (75)	303	52.0 (8.4)	-7.2 (-2.4)	59.3 (19.6)	2.2
R20-TW10	20	10	268 (20)	134	2.4 (0.8)	-0.6 (-0.4)	2.9 (2.2)	0.6
R20-TW20	20	20	521 (43)	260	16.2 (3.1)	-2.7 (-1.0)	19.0 (7.3)	1.9
R20-TW30	20	30	776 (60)	388	57.7 (7.4)	-7.7 (-2.0)	65.4 (16.9)	3.8
R20-TW40	20	40	1021 (85)	510	103.7 (10.1)	-13.4 (-2.6)	117.0 (22.9)	5.0
R20-TW50	20	50	1251 (112)	626	169.9 (13.4)	-16.7 (-2.7)	186.6 (29.8)	6.0
R30-TW10	30	10	395 (27)	198	5.3 (1.3)	-1.2 (-0.6)	6.5 (3.3)	1.4
R30-TW20	30	20	782 (56)	391	36.8 (4.6)	-6.4 (-1.6)	43.2 (11.0)	4.1
R30-TW30	30	30	1153 (77)	576	106.5 (9.)	-13.7 (-2.4)	120.2 (20.9)	7.0
R30-TW40	30	40	1501 (112)	750	200.0 (13.)	-21.4 (-2.9)	221.4 (29.5)	9.0
R30-TW50	30	50	1862 (121)	931	307.2 (16.4)	-29.0 (-3.1)	336.2 (36.1)	10.2
R40-TW10	40	10	529 (29)	264	8.2 (1.5)	-1.8 (-0.7)	10.0 (3.8)	2.1
R40-TW20	40	20	1036 (55)	518	63.6 (6.1)	-10.1 (-1.9)	73.7 (14.2)	6.9
R40-TW30	40	30	1534 (92)	767	169.5 (11.0)	-21.3 (-2.8)	190.8 (24.9)	10.7
R40-TW40	40	40	2013 (118)	1007	302.6 (14.9)	-29.7 (-2.9)	332.3 (33.0)	13.1
R40-TW50	40	50	2479 (177)	1240	459.9 (18.4)	-33.4 (-2.7)	493.3 (39.8)	14.4
R50-TW10	50	10	665 (32)	332	14.1 (2.1)	-2.8 (-0.8)	16.9 (5.1)	3.4
R50-TW20	50	20	1291 (55)	646	96.8 (7.4)	-13.4 (-2.1)	110.2 (17.1)	9.9
R50-TW30	50	30	1894 (102)	947	222.8 (11.7)	-24.6 (-2.6)	247.4 (26.1)	14.1
R50-TW40	50	40	2497 (136)	1248	414.1 (16.5)	-37.3 (-3.0)	451.4 (36.2)	17.1
R50-TW50	50	50	3073 (163)	1536	613.2 (19.9)	-45.3 (-2.9)	658.5 (42.9)	18.9
R100-TW10	100	10	1326 (44)	663	55.5 (4.1)	-9.0 (-1.4)	64.6 (9.7)	12.1
R100-TW20	100	20	2602 (101)	1301	309.4 (11.8)	-35.9 (-2.8)	345.3 (26.5)	28.3
R100-TW30	100	30	3828 (139)	1914	668.9 (17.4)	-56.5 (-3.0)	725.4 (37.9)	36.1
R100-TW40	100	40	4992 (184)	2496	1085.2 (21.7)	-67.5 (-2.7)	1152.7 (46.2)	39.5
R100-TW50	100	50	6166 (194)	3083	1549.9 (25.1)	-71.6 (-2.3)	1621.5 (52.6)	41.8
Average					(15.1)	(-2.0)	(21.3)	10.8

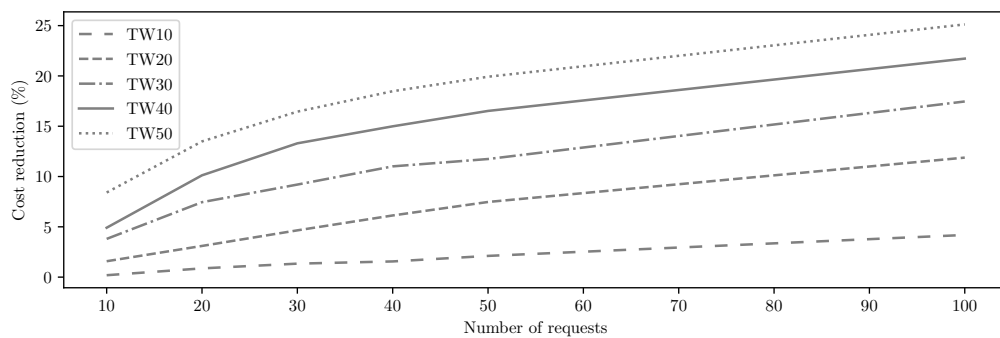


Figure 4.8: Impact of multi-algorithm for each scenario

increase from 10 to 30. The Delivery(%) and Return(%) columns in Table 4.2 reveal the origin of the overall cost reduction. While it is obvious that trip times for direct deliveries must increase if transshipment points are used, we

see that the delivery cost rises little, with a maximum observation of 3.1%. In contrast, the return cost is reduced substantially. In 21 out of 30 scenarios, the cost reduction for the return trip assumes double-digit percentages, with up to 52.6% in scenario R100-TW50. In all scenarios except R10-TW10 and R10-TW20, at least one transfer location is found on average. In R100-TW50, roughly 42 transfer locations are used, resulting in 84 requests utilizing transfers. Just 16 out of 100 requests remain in their original setup with direct deliveries. Even the largest scenario is computed in less than three minutes. We conclude that our multi-algorithm for dynamic transshipments with perfect synchronization during direct deliveries potentially yields substantial cost reduction, especially for setups with many requests and large time windows.

4.5.2 Food delivery case study

In addition to the numerical simulation, we test our algorithm on the case of food deliveries in the city of Bordeaux. For each consignment, the available data contain the latitude, longitude, date and timestamp of pickup and delivery. Based on the timestamp of order placement, we aggregate all orders into increments of five minutes. We assume that all orders placed within these five-minute time horizons could be considered simultaneously for optimization by a central decision-maker. From a total observation period of three months, we select 10 days that have more than 20 requests within a five-minute time bracket (Table 4.3).

Table 4.3: Algorithm results for food delivery data in Bordeaux

Case ID	Date	Hour	Minute	R	tw	Cost at initial setup		Cost improvement after algorithm			Transfers
						Total	Delivery/Return	Total (%)	Delivery (%)	Return (%)	
Bordeaux_1	2016-10-23	8pm	05-09	24	3	59	29	6.8 (11.5%)	7.0 (23.9%)	-0.3 (-0.9%)	7
Bordeaux_2	2016-10-23	8pm	40-44	22	3	64	32	7.2 (11.3%)	8.3 (26.0%)	-1.1 (-3.3%)	9
Bordeaux_3	2016-11-04	8pm	05-09	25	3	79	40	8.4 (10.6%)	10.4 (26.2%)	-2.0 (-5.0%)	9
Bordeaux_4	2016-11-10	8pm	10-14	24	3	54	27	3.2 (6.0%)	4.2 (15.4%)	-0.9 (-3.5%)	9
Bordeaux_5	2016-11-11	7pm	35-39	27	3	66	33	4.9 (7.5%)	5.9 (18.1%)	-1.0 (-3.2%)	9
Bordeaux_6	2016-12-09	8pm	20-24	23	3	59	30	5.8 (9.8%)	7.1 (24.1%)	-1.3 (-4.5%)	8
Bordeaux_7	2017-01-12	8pm	25-39	24	3	57	29	6.3 (11.0%)	6.8 (23.7%)	-0.5 (-1.6%)	6
Bordeaux_8	2017-01-13	8pm	35-39	25	3	52	26	5.4 (10.4%)	5.7 (22.1%)	-0.4 (-1.4%)	7
Bordeaux_9	2017-01-15	7pm	25-39	24	3	51	26	9.7 (18.8%)	10.4 (40.4%)	-0.7 (-2.8%)	9
Bordeaux_10	2017-01-22	9pm	00-04	31	3	64	32	6.3 (9.8%)	7.0 (21.6%)	-0.6 (-2.0%)	9
Average								(10.6%)	(-2.9%)	(24.0%)	8.2

For example, the case labeled "Bordeaux_1" contains 24 requests placed on October 23, 2016 between 08:05:00pm and 08:09:59pm. Given that the longest deliveries are required to cover approximately 2.7 kilometers, we set the delivery constraint to 3.0 kilometers. Using our algorithm, the distance covered for these 10 observations is reduced by 11% on average. This is achieved by using eight transshipments that reduce the return distance by 24%, while the delivery distance is increased by only 3%. These results are in line with the results of comparable scenarios in our numerical simulation (R20-TW10 - R30-TW50), and thus confirm our findings.

4.6 DISCUSSION AND CONCLUSION

4.6.1 *Synthesis*

In this chapter, we propose consideration of dynamic transshipments with perfect synchronization for direct deliveries. We show that the problem's complexity makes a full enumeration infeasible for instances of practicable size. A problem size of 100 requests yields nearly 5,000 potential request pairs and requires 2.7×10^{78} request pair combinations to be tested. We present a problem-specific combination of three algorithms to solve this problem. The first evaluates the relational position of two requests r and r' and, based on 36 structured cases, decides which pairs to exclude from the solution space. The second formulates the optimization problem to find the best transfer location while taking account of time-window constraints, which we simplify and solve using SLSQP. Finally, we transfer the request pair combination problem to graph theory, and leverage EBA to determine the lowest overall distance. In an extensive scenario analysis, we show that our algorithm substantially reduces the initial distance of direct deliveries by 15% on average, and by up to 25% in the largest scenario, while adhering to the time-window constraints. We demonstrate that the total distance is best reduced by relaxing time windows rather than adding requests to increase the chance of transshipments. We confirm our findings in a case study using food delivery data from the city of Bordeaux and show that the algorithm achieves an average distance reduction of 11%.

The main limitations of our research are as follows. First, we assume hub-specific delivery vehicles that return to their origin after direct deliveries, such as private delivery vehicles operated by a pizzeria. This assumption is specific to our problem setup and would differ for external logistics services that conduct deliveries for multiple clients. For that purpose, our algorithm would need to be implemented as a sub-problem of routing optimization. A further limitation is our assumption of absolute flexibility to place transfer locations. Although this may be approximately true for city logistics with a dense network of roads, it is less realistic for rural areas with, for example, large and inaccessible fields or forests. In our setup, assuming that transfers do not consume time is not a limiting factor, as this is analogous to reducing the delivery time window, which is implicitly considered in our scenarios. However, in an optimization in which multiple deliveries are subsequently served over a fixed period of time, it is necessary to include transfer times. Positive transfer times might result in being able to fulfill fewer orders overall. Lastly, we do not consider quality-related factors. For example, temperature-controlled deliveries might be adversely affected by transshipments, especially if more than one transfer were allowed. Transshipments also increase the probability of errors, such as dropping or mixing up products during the transfer process.

Our findings have the following managerial implications. From a cost perspective, allowing sufficiently large time windows to interact with customers is more important than the volume game of increasing sales numbers. Nevertheless, synergy effects and increased profits must be considered in addition to the cost-only perspective. Furthermore, allowing transfers within direct shipments substantially reduces the non-value-adding return part of the total trip distance.

This will enable businesses not only to reduce costs, but also to improve sustainability. For example, reducing the distances driven correlates with reductions of carbon and other emissions. Reductions in the total required distance will also translate into fewer vehicles needed for a given number of requests, which will reduce both congestion and the emissions realized during vehicle production. Another effect of lower mileage and fewer cars is reduced demand for drivers. Given the severe driver shortage in recent years, transshipments might mitigate this problem.

In addition, given that networks of businesses may participate in such a system of transshipment, the complexity of implementation is low for two reasons. First, the physical effort is moderate, as only one transfer location is added to the original trip, while the effort to return is reduced. Second, the algorithm is easily implemented, as few input parameters are required. The algorithm is also efficient and solves our largest instance in under three minutes.

4.6.2 *Future research*

Future research might focus on two areas: extending our model, and testing it in different contexts. One extension might be to increase the number of transshipments allowed per delivery. In a scenario with unlimited transfers, requests would be taken care of by all vehicles operating on the same linear route, similar to airplanes being covered by different air traffic controllers. Furthermore, a dynamic representation should be considered to allow the optimization to consider newly added requests in real-time. Another extension might be to increase the capacity constraints to allow more than one unit load to be carried per vehicle, which would ultimately result in a routing problem. With regard to another capacity issue, a non-homogeneous fleet might be of interest, as different food places might have varying storage boxes that limit exchanges of loads. Lastly, including quality parameters might be an option to reflect the quality of transport, such as the number of transfers or extended delivery times.

Our method might also be applied to other contexts. During the picking process in warehouses, employees often use picking boxes to collect items for an order, and keep these boxes until delivery to the packing area. Rather than having a fixed assignment between box and employee, our algorithm might help to determine when it would make sense for containers to be exchanged between employees. Another example might be long-haul transport of various kinds. Whenever an underutilized back-haul exists, for example due to imbalanced market demand or customer networks, transshipments might be useful. For example, trucks might meet at roadhouses and exchange goods using truck-mounted forklifts. Ultimately, large industrial business complexes that use intra-logistic deliveries might benefit from transfers during direct deliveries.

SUMMARY AND OUTLOOK

5.1 SYNTHESIS, MANAGERIAL IMPLICATIONS AND RESEARCH CONTRIBUTIONS

The research presented in this dissertation was motivated by the critical intermediate role played by warehouses in supply chains, and the opportunities they offer for competitive cost and service advantages. Novel data sources and optimization algorithms can be used to improve the 150,000 warehouses worldwide that account for roughly half of supply chain costs. In our three research projects, we look at predictions of warehouse equipment breakdowns, predictions of warehouse employees' productivity, and the potential for cross-docking-like transshipments in direct deliveries.

In Chapter 2, we build a predictive maintenance model for Material Handling Equipment (MHE) by forecasting breakdowns. First, we develop a structured framework to define meaningful prediction features by looking at how the MHE interacts with the environment. We then combine novel data from shock sensors with master and usage data on that equipment to generate the defined features. In a comprehensive study, we analyze eight machine-learning algorithms and their ability to incorporate these novel data sources. Given the difficulty of highly imbalanced data (breakdown events occurred on only 125 out of 10,722), the five algorithms capturing non-linear relationships achieve high prediction performance, with an F1 score of > 0.85 . An evaluation of the costs of breakdowns and maintenance services reveals that, from a cost reduction perspective, the Random Forest Classifier (RFC) achieves the best results. With an accuracy of 99%, precision of 93%, recall of 86% and F1 score of 88%, it detects 108 out of 125 breakdowns, with only eight false positive classifications. Using a second dataset, we show that the features defined through our framework can be applied analogously. We also achieve similar prediction results, with an accuracy of 99%, precision of 95%, recall of 82% and F1 score of 88%, thus showing that our findings are transferable.

From a managerial perspective, we propose that data collection efforts should focus first on the condition-based, then on time-based factors, and lastly on master data. We know that OEMs of MHE already install login terminals and shock sensors as default configurations; hence, data can be obtained at low cost. Given that master data are usually available at no cost, they should also be included, even though they contribute only marginally.

The main contribution of this chapter is to contradict the assumption of most literature that either time- or condition-based factors are responsible for breakdowns, as well as the view that considering both variables would be too complex. First, we show through a feature importance analysis that, in fact, both contribute to the occurrence of breakdowns. Second, we include master data as a third data type in addition to time and condition variables, and show

how such data also contribute to prediction accuracy. Third, we show that machine-learning algorithms are capable of incorporating all three data types simultaneously.

In Chapter 3, we predict the productivity of newly hired warehouse associates within their first eight weeks of employment using Extreme Gradient Boosting (EGB). As in Chapter 2, we first develop a framework to capture factors most likely to impact on productivity, namely the warehouse in which the operation takes place, the operator, the shift in which the operator is placed and the product. We collect data from a Logistics Service Provider (LSP) for two functions on all but the warehouse dimension, and we build an EGB model to predict productivity on a daily basis for each employee individually. For comparison, we use the LSP's actual productivity assumptions, three simple predictors, and a Random Forest (RF) as additional baselines. Considering all eight weeks, our model outperforms the prediction accuracy of the LSP and all other baselines. We show that the EGB improves the Root Mean Squared Error (RMSE) by 41% in the picking and 52% in the packing function. All three simple baselines are improved by around 20% in both functions, and the RF is improved by 5% and 9% in the picking and packing functions, respectively. Our model performs especially well in the critical first weeks when least information is available and productivity variance highest. Compared with the LSP's current cost structure, we show that the prediction improvement translates into a cost reduction of up to 47% in picking and 86% in packing.

With regards to generalizing the method and results, we have two indicators. First, although the two functions differ considerably from an operational perspective, we show that using the tuned hyperparameters of one function and applying them to the prediction model of the other yields very similar prediction results. Second, the substantial prediction improvement is based on an extensive dataset collected in a highly mature warehouse operation. Hence, we believe our method will produce more robust results - not only for other functions, but also to some extent for other warehouses.

Our full factorial design analysis tests all combinations of variable categories, namely Operator, Product and Shift, revealing various managerial implications. We show that when only one of the three categories is used, Shift variables have the lowest prediction accuracy. In addition, using the Operator and Product variables yields the same results as when the Shift category is added. From this perspective, one should not invest time in collecting Shift variables. However, if collection of personal information (Operator category) is problematic owing to concerns by unions and work councils or for other reasons, Shift data should be collected as predictions are improved compared with using Product variables alone.

To the best of our knowledge, productivity predictions have only been considered at an aggregated level, not an employee or daily level. However, disaggregated data are vital if productivity forecasts are to be used for processes such as workforce planning. Hence, our work contributes to a topic relevant to most industrial companies, and is the first of its kind in the academic research body. Our structured framework of relevant productivity factors can be used

by other researchers investigating similar topics.

In Chapter 4, we analyze the emerging demand for instant deliveries, and examine how dynamic and synchronized transfers, similar to cross-dock-like transshipments, reduce the total distance driven. Given the size and complexity of the problem, we propose a combination of three successive algorithms. First, a geometric algorithm evaluates the relational position of two delivery requests, allocates it to one of 36 predefined cases and, based on a set of predefined rules, decides whether to omit it from the solution space. The second algorithm (SLSQP) finds the best transfer location by minimizing the time-window constrained total distance function. Lastly, the EBA finds the best combination of transfer locations to be used. Our multi-algorithm is tested on 30 numerical scenarios with different time constraints and up to 100 requests. On average, the algorithm reduces the initial distance of direct deliveries by 15%. We show that this distance reduction is achieved mainly by shortening the empty return distance (21% on average), while the delivery trip increases moderately (3% on average). These findings are then confirmed in a case study using food delivery data from Bordeaux, with an average distance reduction of 11%.

Our multi-algorithm is problem-specific. As a result, making our methodological approach transferable to other environments, and especially other constraints, will require adjustments. If the boundary conditions remain similar, as in the case study we investigated, the methodology and its results will be generalizable.

Our research has several managerial implications. If the concept of dynamic and synchronized transshipments is used to reduce the total distance, then managers and organizations should focus on interactions with existing customers rather than increased sales activity. This is because relaxing the time-window constraints on current delivery requests will increase the probability of successful transshipments more than adding new requests. In addition to considering dynamic transfers when reducing the total distance is important, organizations struggling with driver shortages, seeking to reduce their ecological footprint or downsizing their delivery fleet should also consider transshipments. However, to realize the identified benefits, organizations must participate in collaborative networks with a central decision-maker steering transshipment decisions. This will require a certain level of trust between participants to allow deliveries through other entities. Although the algorithm is easy to implement and computationally efficient, some initial investment must be made to set up such a system.

The research in this chapter contributes to an emerging literature stream on transshipments. All previous studies assume either fixed transfer locations or storage opportunities to avoid synchronization constraints. Our research appears to be the first to consider both aspects, and thus makes an important contribution.

In conclusion, all three research papers solve relevant problems and are tested with actual data from organizations facing these challenges. They contribute to literature streams on warehouse management, applications of machine-

learning algorithms, use of novel data sources, and transshipments during deliveries.

5.2 FUTURE RESEARCH OPPORTUNITIES

Our findings make several contributions to the literature and have various managerial implications. The limitations of our research highlight potential directions for future research.

In the first two studies, presented in Chapter 2 and 3, we integrate novel data into prediction models. For this purpose, we develop individual frameworks to determine the most important variables. Although we cover a wide range of these variables, we were unable to examine all of them. Based on Chapter 2, future research should investigate the environment in which MHE is located (e.g., warehouse location and design), the impact of operators (e.g., their tenure or qualifications) and additional sensor data (e.g., vibrations, temperature, humidity). In Chapter 3, factors of interest not yet covered include employee characteristics (e.g., age, training received) and various warehouse parameters (e.g., design, size, process maturity, degree of automation). We believe that the latter will be of greater interest, as warehouse parameters differ structurally, which may impact more severely on prediction accuracy. The methods of both chapters might be transferred to other contexts to validate the generalizability of our findings. On the one hand, our methodologies should be tested in similar setups, i.e., additional warehouses. Beyond the scope of warehouses, predictions of moving equipment breakdowns should be tested on vehicles with similar complexity (e.g., ride-on mowers, e-scooters, cranes, farm vehicles, bikes, cars, trucks), whereas productivity predictions might be extended to other areas of manual work (e.g., production and service industries).

The research opportunities arising from Chapter 4 result mainly from relaxation of our model assumptions. Although we show that transshipments reduce the total distance driven compared with vehicles conducting direct deliveries, we assume static scenarios with information known upfront, vehicles with unit-sized capacity, and a maximum of one transfer per delivery. Each assumption offers interesting research potential. Allowing higher than unit-sized capacity might be most interesting, as this would result in more complex routing problems with multiple pickups and deliveries during a tour. Given that routing problems are highly relevant but often complex, we believe that this research path would be the most challenging but would yield the most relevant results. Although our static perspective on the problem can be achieved by aggregating time intervals, considering dynamic problem types is a second area for future research. This is particularly relevant from a practice-oriented perspective, as introducing dynamic information into the problem, such as newly placed delivery requests, may change the current solution. Lastly, and similarly to Chapters 2 and 3, the use of dynamic and synchronized transfers in other setups might be of high practical use. One example might be to analyze whether two warehouse employees' walking distance in the picking process can be reduced if they are permitted to exchange picking containers.

Summarizing, our research not only enriches the current literature, but also offers several new, interesting and relevant research opportunities.

APPENDIX TO CHAPTER 2

A.1 DATA STRUCTURE

MHE	Master Data (MD)	Timestamp (t)	Usage Data (UD)	Extracted Feature (EF)	Breakdown Indicator (BI)
MHE_i	MD_i	$t_{i, Start}$	$UD_{t_i, Start}$	$EF_{t_i, Start}$	$BI_{t_i, Start}$
MHE_i	MD_i	$t_{i, Start+1}$	$UD_{t_i, Start+1}$	$EF_{t_i, Start+1}$	$BI_{t_i, Start+1}$
MHE_i	MD_i	$t_{i, Start+2}$	$UD_{t_i, Start+2}$	$EF_{t_i, Start+2}$	$BI_{t_i, Start+2}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
MHE_i	MD_i	$t_{i, End}$	$UD_{t_i, End}$	$EF_{t_i, End}$	$BI_{t_i, End}$
MHE_{i+1}	MD_{i+1}	$t_{i+1, Start}$	$UD_{t_{i+1}, Start}$	$EF_{t_{i+1}, Start}$	$BI_{t_{i+1}, Start}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

A.2 DESCRIPTION OF STATISTICAL LEARNING METHODS

SLM	Description
Gaussian Naïve Bayes (GNB)	The GNB is known to be a simple algorithm that is computationally inexpensive, even for large data sets and often outperforms sophisticated algorithms. It is based on Bayes Theorem, thus calculating probabilities for an outcome (classification) based on the prior knowledge of information (features), which are possibly related. A strong assumption is that features are independent of each other (Chan et al. (1979)).
K-Nearest-Neighbors (KNN)	The KNN uses k pre-labeled observations that are in closest proximity (nearest neighbors) to the data to be classified. Each neighbor casts a vote based on its own class. The sum of all, sometimes weighted votes, determines the class of the unlabeled observation.
Logistic Regression (LR)	Unlike linear regression, LR predicts classes instead of continuous values. It uses a logistic sigmoid function with a value range from 0 to 1, representing the probability of an observation belonging to a class or not. Maximum likelihood is used to fit the sigmoid function. Compared to many other algorithms, Logistic Regression (LR) offers better interpretability through coefficients which provide information on feature importance.
Random Forest Classifier (RFC)	In RFCs, individual weak learners (uncorrelated decision trees) are combined to form one strong learner (random forest). Each decision tree is trained on a random sample with replacement from the original data. The independently trained decision trees are combined to take a majority vote (bagging) on unlabeled data. RFCs are known to be efficient for a high number of observations and features (Breiman (2001)).
Stochastic Gradient Boosting (SGB)	Similar to RFCs, SGB is an ensemble method that combines multiple weak learners. It does not generate weak learners independently in parallel but uses an iterative approach. By this, a weak learner builds on information gained by the previously built one (Friedman (2002)).
Support Vector Classifier (SVC)	An SVC constructs a hyperplane that optimally separates two classes. The hyperplane thereby maximizes the corridor width between the two classes. The observations lying on the edges of that corridor define its width and are called the support vectors. Support Vector Classifier (SVC)s are able to capture non-linear relationship using kernels to divide classes in higher dimensional data.
Multilayer Perceptron Neural Network (MLP)	Neural networks consist of three layers: input, hidden and output layer. During the feedforward process they feed independent variables (input layer) into linear functions (neurons) and combine them into a non-linear function (hidden layer) to yield the target value (output layer). During backpropagation, the error between output layer and real values are used to optimize the neural net. MLPs use one or more hidden layers and pass information during feedforward only to the next layer, not between neurons of the same layer, within the same neuron or to a neuron of a previous layer.
Relevance Vector Machine (RVM)	With an identical functional form to that of SVC, RVM differs by exploiting a Bayesian learning framework to make probabilistic estimates instead of point estimates. By omitting SVC's complexity parameter, it does not require cross-validation (Tipping (2001)).

A.3 SLM HYPERPARAMETERS

SLM	Python hyperparameter	Tested values
KNN	n_neighbors	1 ^{a,c} , 2 ^b , 3, 4, 5, 6, 7, 8, 9, 10
	weights	distance ^{a,b,c} , uniform
	algorithm	ball_tree ^{a,b,c} , kd_tree
MLP	solver	lbfgs, sgd, adam ^{a,b,c}
	hidden_layer_sizes	(10), (50), (100), (10, 10), (50, 50), (100, 100), (10, 10, 10), (50, 50, 50), (100, 100, 100), (10, 10, 10, 10), (50, 50, 50, 50), (100, 100, 100, 100) ^{a,b,c} , (10, 10, 10, 10, 10), (50, 50, 50, 50, 50), (100, 100, 100, 100, 100)
	max_iter	500 ^{a,b,c} , 1000
	activation	logistic, tanh, relu ^{a,b,c}
	alpha	0.0001 ^c , 0.001 ^{a,b} , 0.01, 0.1, 1
	learning_rate	constant ^{a,b,c} , invscaling
	SGD	objective
SGD	learning_rate	0.001, 0.01, 0.1 ^b , 1 ^{a,c}
	colsample_bytree	0.3 ^c , 0.7 ^{a,b}
	n_estimators	500 ^{a,b,c} , 1000
	max_depth	5 ^c , 10 ^{a,b} , None
	gamma	0 ^{a,b} , 0.1 ^c , 0.2
	reg_alpha	0 ^{a,b,c} , 0.5, 1
	reg_lambda	0 ^{b,c} , 0.5, 1 ^a
	GNB	var_smoothing
LR	solver	liblinear, newton-cg, sag, saga ^{a,b,c} , lbfgs
	penalty	l1 ^{a,b} , l2 ^c , elasticnet, None
	C	0.001, 0.01 ^b , 0.02, 0.03, 0.04, 0.05, 0.06 ^a , 0.07 ^c , 0.08, 0.09, 0.1, 1, 5, 10, 15
	max_iter	500, 1000 ^{a,b,c}
RFC	class_weight	balanced ^{a,b,c} , none
	n_estimators	100, 200, 300, 400 ^{a,c} , 500, 1000, 1500, 2000 ^b
	max_depth	10, 20, 30 ^{a,b,c} , 40, 50, 75, 100, None
	max_features	auto, sqrt ^{a,b,c}
	min_samples_leaf	1 ^{a,b,c} , 2, 4
	min_samples_split	2 ^b , 5 ^{a,c} , 10
SVC	bootstrap	True, False ^{a,b,c}
	kernel	poly ^c , rbf ^{a,b} , sigmoid
	C	0.01, 0.1, 1, 10, 100 ^b , 1000 ^{a,c}
	gamma	0.00001, 0.0001, 0.001, 0.01, 0.1, 1 ^a , 10 ^b , 100 ^c
SVC	degree (only poly kernel)	0, 1, 2, 3, 4 ^c , 5
	RVM	alpha
RVM	beta	1e-06 ^{a,b,c}
	coef0	0 ^{a,b,c}
	coef1	None ^{a,b,c}
	degree	3 ^{a,b,c}
	kernel	rbf ^{a,b,c}
	n_iter	100 ^{a,b,c}
	n_iter_posterior	50 ^{a,b,c}
	threshold_alpha	1e09 ^{a,b,c}
	tol	0.001 ^{a,b,c}

Notes: Best hyperparameter for 7-day forecast (a), 3-day forecast (b) and 1-day forecast (c)

A.4 SLM PREDICTION RESULTS FOR OEM2

Forecast	Metric	SVC	KNN	SGB	RFC	MLP	LR	GNB	NP
7 day	Accuracy	0.99	0.99	0.99	0.99	0.98	0.73	0.88	0.95
	Precision	0.94	0.90	0.95	0.95	0.86	0.13	0.11	0.00
	Recall	0.87	0.88	0.86	0.82	0.73	0.79	0.21	0.00
	F1	0.90	0.89	0.91	0.88	0.79	0.22	0.14	0.00
3 day	Accuracy	0.99	0.99	0.99	0.99	0.98	0.73	0.93	0.98
	Precision	0.84	0.80	0.83	0.90	0.75	0.06	0.08	0.00
	Recall	0.74	0.74	0.55	0.54	0.41	0.78	0.19	0.00
	F1	0.75	0.77	0.66	0.68	0.53	0.11	0.11	0.00
1 day	Accuracy	0.98	0.99	0.99	0.99	0.99	0.73	0.96	0.99
	Precision	0.03	0.07	0.00	0.00	0.00	0.02	0.03	0.00
	Recall	0.05	0.03	0.00	0.00	0.00	0.70	0.13	0.00
	F1	0.04	0.04	0.00	0.00	0.00	0.04	0.05	0.00

Note: RVM not included as it shows non-competitive prediction results for OEM₁, but consumes large amount of computational resources

A.5 IMPACT OF IMPLEMENTING SLM

$$\begin{aligned}
I_{SLM} &= \frac{C_{Base} - C_{BM}}{C_{Base}} \\
&\equiv \frac{c_M r_{BM}(TP + FN) - c_M(TP + FP) - c_M r_{BM}(FN + u_{TP}TP)}{c_M r_{BM}(TP + FN)} \\
&\equiv \frac{TP}{TP + FN} - \frac{TP + FP}{r_{BM}(TP + FN)} - \frac{u_{TP}TP}{TP + FN} \\
&\equiv \frac{TP}{TP + FN} - \frac{1}{\frac{TP+FN}{r_{BM}} - \frac{TP+FP}{TP+FP}} - \frac{u_{TP}TP}{TP + FN} \\
&\equiv \frac{TP}{TP + FN} - \frac{\frac{TP}{TP+FN}}{r_{BM} \frac{TP}{TP+FP}} - \frac{u_{TP}TP}{TP + FN} \\
&\equiv Recall - \frac{Recall}{r_{BM} Precision} - u_{TP} Recall \\
&\equiv Recall \left(1 - \frac{1}{r_{BM} Precision} - u_{TP} \right)
\end{aligned}$$

APPENDIX TO CHAPTER 3

B.1 EGB HYPERPARAMETER TEST VALUES

Hyperparameter	Description	Tested values
Booster	Defines the algorithm type	gbtree
Max_depth	Limits the number of decision node levels	4,6*,8**,10
Min_child_weight	Ensures that the number of observations in the leaves does not fall below a specified threshold	1,3**,5,7,9
Eta	Acts as learning rate to increase robustness by reducing the impact of each new tree	0.01, 0.05, 0.1**, 0.2*, 0.3
Gamma	Sets the minimum error reduction required in a decision node	0.01, 0.1, 1*, 10**
Subsample	Determines what proportion is randomly sampled from observations	0.9**, 1
Colsample_bytree	Determines what proportion is randomly sampled from features	0.25, 0.5, 0.75*, 1**

Notes: * best parameters for the picking function, ** best parameters for the packing function

APPENDIX TO CHAPTER 4

C.1 COMPLEXITY OF REQUEST PAIRS AND PAIR COMBINATIONS

Requests*	Request pairs	Pair combinations
2	1	1
4	6	3
6	15	15
8	28	105
10	45	945
...
50	1,225	5.8×10^{31}
...
100	4,950	2.7×10^{78}

Note: *dummy request is added if total request number is uneven

BIBLIOGRAPHY

- Ahmad, R., Kamaruddin, S., 2012. An overview of time-based and condition-based maintenance in industrial application. *Computers & Industrial Engineering* 63, 135–149. doi:10.1016/j.cie.2012.02.002.
- Amari, S.V., McLaughlin, L., Pham, H., 2006. Cost-effective condition-based maintenance using markov decision processes, in: RAMS '06. Annual Reliability and Maintainability Symposium, 2006., pp. 464–469. doi:10.1109/RAMS.2006.1677417.
- Arslan, A., Agatz, N., Kroon, L., Zuidwijk, R., 2018. Crowdsourced delivery - A dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science* 53, 222–235. doi:10.1287/trsc.2017.0803.
- Baker, R.D., Christer, A.H., 1994. Review of delay-time OR modelling of engineering aspects of maintenance. *European Journal of Operational Research* 73, 407–422. doi:10.1016/0377-2217(94)90234-8.
- Barlow, R., Hunter, L., 1960. Optimum preventive maintenance policies. *Operations Research* 8, 90–100. doi:10.1287/opre.8.1.90.
- Beojone, C., Geroliminis, N., 2021. On the inefficiency of ride-sourcing services towards urban congestion. *Transportation Research Part C: Emerging Technologies* 124, 102890. doi:10.1016/j.trc.2020.102890.
- Bloch, H.P., Geitner, F.K., 2012. Machinery failure analysis and troubleshooting. 4 ed., Butterworth-Heinemann. doi:10.1016/C2010-0-67104-5.
- Bögl, M., Doerner, K., Parragh, S., 2015. The school bus routing and scheduling problem with transfers. *Networks* 65, 180–203. doi:10.1002/net.21589.
- Boland, P.J., Proschan, F., 1982. Periodic replacement with increasing minimal repair costs at failure. *Operations Research* 30, 1183–1189. doi:10.1287/opre.30.6.1183.
- Bond, J., 2017. Top 20 industrial lift truck suppliers. https://www.mmh.com/article/top_20_lift_truck_suppliers_2017. Accessed: 30-04-2020.
- Breiman, L., 2001. Random forests. *Machine Learning* 45, 5–32. doi:10.1023/A:1010933404324.
- Büsch, S., Nissen, V., Wünscher, A., 2017. Automatic classification of data-warehouse-data for information lifecycle management using machine learning techniques. *Information Systems Frontiers* 19. doi:10.1007/s10796-016-9680-8.
- Caesarendra, W., Widodo, A., Yang, B.S., 2010. Application of relevance vector machine and logistic regression for machine degradation assessment. *Mechanical Systems and Signal Processing* 24, 1161–1171. doi:10.1016/j.ymsp.2009.10.011.
- Cantor, D.E., Jin, Y., 2019. Theoretical and empirical evidence of behavioral and production line factors that influence helping behavior. *Journal of Operations Management* 65, 312–332. doi:10.1002/joom.1019.

- Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., da P. Francisco, R., Basto, J.P., Alcalá, S.G.S., 2019. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering* 137, 106024.
- Carvalho, V.M., Nirei, M., Saito, Y.U., Tahbaz-Salehi, A., 2021. Supply chain disruptions: Evidence from the great east japan earthquake. *The Quarterly Journal of Economics* 136, 1255–1321. doi:10.1093/qje/qjaa044.
- Cavalcante, C.A.V., Lopes, R.S., Scarf, P.A., 2018. A general inspection and opportunistic replacement policy for one-component systems of variable quality. *European Journal of Operational Research* 266, 911–919. doi:10.1016/j.ejor.2017.10.032.
- Chan, F., Chan, H., 2011. Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. *Expert Syst. Appl.* 38, 2686–2700. doi:10.1016/j.eswa.2010.08.058.
- Chan, T., Golub, G., LeVeque, R., 1979. Updating formulae and a pairwise algorithm for computing sample variances, in: *COMPSTAT 1982 5th Symposium held at Toulouse 1982*, pp. 30–41. doi:10.1007/978-3-642-51461-6_3.
- Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 321–357. doi:10.1613/jair.953.
- Chen, M., Feldman, R.M., 1997. Optimal replacement policies with minimal repair and age-dependent costs. *European Journal of Operational Research* 98, 75–84. doi:10.1016/0377-2217(95)00320-7.
- Cléroux, R., Dubuc, S., Tilquin, C., 1979. The age replacement problem with minimal repair and random repair costs. *Operations Research* 27, 1158–1167. doi:10.1287/opre.27.6.1158.
- Cortés, C., Matamala, M., Contardo, C., 2010. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research* 200, 711–724. doi:10.1016/j.ejor.2009.01.022.
- Cui, H., Rajagopalan, S., Ward, A.R., 2019. Predicting product return volume using machine learning methods. *European Journal of Operational Research* 281, 612–627. doi:10.1016/j.ejor.2019.05.046.
- Cui, R., Gallino, S., Moreno, A., Zhang, D.J., 2017. The operational value of social media information. *Production and Operations Management* 27, 1749–1769. doi:10.1111/poms.12707.
- Dablanc, L., Morganti, E., Arvidsson, N., Woxenius, J., Browne, M., Saidi, N., 2017. The rise of on-demand ‘instant deliveries’ in european cities. *Supply Chain Forum* 18. doi:10.1080/16258312.2017.1375375.
- Danloup, N., Allaoui, H., Goncalves, G., 2018. A comparison of two meta-heuristics for the pickup and delivery problem with transshipment. *Computers & Operations Research* 100, 155–171. doi:10.1016/j.cor.2018.07.013.

- De Gooijer, J.G., Hyndman, R., 2006. 25 years of time series forecasting. *International Journal of Forecasting* 22, 443–473. doi:10.1016/j.ijforecast.2006.01.001.
- De Koster, R., Le Duc, T., Roodbergen, K.J., 2007. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research* , 481–501doi:10.1016/j.ejor.2006.07.009.
- Deleplanque, S., Quilliot, A., 2013a. Dial-a-ride problem with time windows, transshipments, and dynamic transfer points. doi:10.3182/20130619-3-RU-3018.00435.
- Deleplanque, S., Quilliot, A., 2013b. Transfers in the on-demand transportation: The darpt dial-a-ride problem with transfers allowed, in: *Multidisciplinary international scheduling conference: Theory and applications (MISTA)*, pp. 185–205.
- Deng, Q., Santos, B.F., Curran, R., 2019. A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. *European Journal of Operational Research* 281, 256–273. doi:10.1016/j.ejor.2019.08.025.
- Derman, C., 1963. Optimal replacement and maintenance under markovian deterioration with probability bounds on failure. *Management Science* 9, 478–481. doi:10.1287/mnsc.9.3.478.
- Derman, C., Solomon, H., 1958. Development and evaluation of surveillance sampling plans. *Management Science* 5, 72–88. doi:10.1287/mnsc.5.1.72.
- Doerr, K., Arreola-Risa, A., 2000. A worker-based approach for modeling variability in task completion times. *IIE Transactions* 32, 625–636. doi:10.1023/A:1007659032655.
- Edmonds, J., 1965a. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics* , 125–130.
- Edmonds, J., 1965b. Paths, trees, and flowers. *Canadian Journal of Mathematics* 17, 449–467. doi:10.4153/CJM-1965-045-4.
- Enzi, M., Parragh, S., Pisinger, D., Prandtstetter, M., 2020. Modeling and solving the multimodal car- and ride-sharing problem. *European Journal of Operational Research* 293. doi:10.1016/j.ejor.2020.11.046.
- Falkenberg, S., Spinler, S., 2022a. Integrating operational and human factors to predict daily productivity of warehouse employees using extreme gradient boosting. *International Journal of Production Research* doi:10.1080/00207543.2022.2159563.
- Falkenberg, S., Spinler, S., 2022b. The role of novel data in maintenance planning: Breakdown predictions for material handling equipment. *Computers Industrial Engineering* 169, 108230. doi:10.1016/j.cie.2022.108230.
- Falkenberg, S.F., Spinler, S., Strauss, A., 2022. A multi-algorithm for dynamic transshipments with perfect synchronization. Unpublished working paper.

- Feil, J., 2016. Edmonds's blossom algorithm.
https://algorithms.discrete.ma.tum.de/graph-algorithms/matchings-blossom-algorithm/index_en.html. Accessed: 2021-12-14.
- Fletcher, S., Baines, T., Harrison, D., 2006. An investigation of production workers' performance variations and the potential impact of attitudes. *The International Journal of Advanced Manufacturing Technology* 35, 1113–1123. doi:10.1007/s00170-006-0793-y.
- Fouladirad, M., Paroissin, C., Grall, A., 2018. Sensitivity of optimal replacement policies to lifetime parameter estimates. *European Journal of Operational Research* 266, 963–975. doi:10.1016/j.ejor.2017.10.055.
- Friedman, J., 2000. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29. doi:10.1214/aos/1013203451.
- Friedman, J.H., 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38, 367–378. doi:10.1016/S0167-9473(01)00065-2.
- Gandy, B., Coberley, C., Pope, J., Wells, A., Rula, E., 2014. Comparing the contributions of well-being and disease status to employee productivity. *Journal of occupational and environmental medicine / American College of Occupational and Environmental Medicine* 56, 252–7. doi:10.1097/JOM.000000000000109.
- Gartner, 2021. Future of supply chain: 5 trends to act on now.
<https://emtemp.gcom.cloud/ngw/globalassets/en/supply-chain/documents/trends/future-of-supply-chain-ebook.pdf>. Accessed: 2022-01-24.
- Gartner, 2022. The top 8 supply chain technology trends.
<https://emtemp.gcom.cloud/ngw/globalassets/en/supply-chain/documents/trends/the-top-8-supply-chain-technology-trends.pdf>. Accessed: 2022-01-24.
- van Gils, T., Ramaekers, K., Braekers, K., Caris, A., 2015. Improving operational workforce scheduling in a warehouse using time series forecasting.
- van Gils, T., Ramaekers, K., Caris, A., Cools, M., 2017. The use of time series forecasting in zone order picking systems to predict order pickers' workload. *International Journal of Production Research* 55, 6380–6393. doi:10.1080/00207543.2016.1216659.
- Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.M., 2017. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers & Operations Research* 84, 116–126. doi:10.1016/j.cor.2017.03.004.
- Gu, J., Goetschalckx, M., McGinnis, F., L., 2007. Research on warehouse operation: A comprehensive review. *European Journal of Operational Research* 177, 1–21. doi:10.1016/j.ejor.2006.02.025.
- Hammouda, S., Taffar, M., Lemouari, A., 2020. A simulated annealing for the resolution of "dial-a-ride-problem with transfer" using hybrid neighborhood methods. 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS) , 1–6.

- Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning: Data mining, inference, and prediction. 2 ed., Springer, New York. doi:10.1007/978-0-387-84858-7.
- Hastie, T., Tibshirani, R., Wainwright, M., 2015. Statistical Learning with Sparsity - The Lasso and Generalizations. 1 ed., Chapman & Hall/CRC, New York. doi:10.1201/b18401.
- Heng, A., Tan, A.C.C., Mathew, J., Montgomery, N., Banjevic, D., Jardine, A.K.S., 2009. Intelligent condition-based prediction of machinery reliability. *Mechanical Systems and Signal Processing* 23, 1600–1614. doi:10.1016/j.ymsp.2008.12.006.
- Herzog, M.A., Marwala, T., Heyns, P.S., 2009. Machine and component residual life estimation through the application of neural networks. *Reliability Engineering & System Safety* 94, 479–489. doi:10.1016/j.res.s.2008.05.008.
- Hjartar, K., Krishnakanthan, K., Prieto-Muñoz, P., Shenai, G., Kuiken, S.V., 2019. The CEO's new technology agenda. <https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-ceos-new-technology-agenda>. Accessed: 30-04-2020.
- Hopcroft, J.E., Karp, R.M., 1971. A $n^{5/2}$ algorithm for maximum matchings in bipartite, in: 12th Annual Symposium on Switching and Automata Theory (swat 1971), pp. 122–125. doi:10.1109/SWAT.1971.1.
- Höppner, S., Stripling, E., Baesens, B., vanden Broucke, S., Verdonck, T., 2018. Profit driven decision trees for churn prediction. *European Journal of Operational Research* 284, 920–933. doi:10.1016/j.ejor.2018.11.072.
- Islam, M.R., Ali, S.M., Fathollahi-Fard, A., Kabir, G., 2021. A novel particle swarm optimization-based grey model for the prediction of warehouse performance. *Journal of Computational Design and Engineering* 8, 1–23. doi:10.1093/jcde/qwab009.
- de Jonge, B., Scarf, P.A., 2019. A review on maintenance optimization. *European Journal of Operational Research* 285, 805–824. doi:10.1016/j.ejor.2019.09.047.
- Jorgenson, D.W., McCall, J.J., 1963. Optimal replacement policies for a ballistic missile. *Management Science* 9, 358–379. doi:10.1287/mnsc.9.3.358.
- Juran, D., Schruben, L., 2004. Using worker personality and demographic information to improve system performance prediction. *Journal of Operations Management* 22, 355–367. doi:10.1016/j.jom.2004.05.003.
- Kerivin, H., Lacroix, M., Mahjoub, A., Quilliot, A., 2008. The splittable pickup and delivery problem with reloads. *European Journal of Industrial Engineering* 2, 112–133. doi:10.1504/EJIE.2008.017347.
- Kiefer, A.W., Novack, R.A., 1999. An empirical analysis of warehouse measurement systems in the context of supply chain implementation. *Transportation Journal* 38, 18–27. doi:10.2307/20713387.
- Kim, A., Yang, Y., Lessmann, S., Ma, T., Sung, M.C., Johnson, J.E.V., 2019. Can deep learning predict risky retail investors? A case study in financial risk behavior forecasting. *European Journal of Operational Research* 283, 217–234. doi:10.1016/j.ejor.2019.11.007.

- Klein, M., 1962. Inspection-maintenance-replacement schedules under markovian deterioration. *Management Science* 9, 25–32. doi:10.1287/mnsc.9.1.25.
- Kolmogorov, V., 2009. Blossom V: A new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation* 1, 43–67. doi:10.1007/s12532-009-0002-8.
- Konovsky, M., Cropanzano, R., 1991. Perceived fairness of employee drug testing as a predictor of employee attitudes and job performance. *The Journal of applied psychology* 76, 698–707. doi:10.1037//0021-9010.76.5.698.
- Kraft, D., 1988. A software package for sequential quadratic programming. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht, Wiss. Berichtswesen d. DFVLR.
- Kraus, M., Feuerriegel, S., Oztekin, A., 2020. Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research* 281, 628–641. doi:10.1016/j.ejor.2019.09.018.
- Kuhn, H.W., 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 83–97. doi:https://doi.org/10.1002/nav.3800020109.
- Kumar, K., 2001. Technology for supporting supply chain management. *Communication of the ACM* 44, 58–61. doi:10.1145/376134.376165.
- Lammgård, C., Andersson, D., 2014. Environmental considerations and trade-offs in purchasing of transportation services. *Research in Transportation Business & Management* 10, 45–52. doi:10.1016/j.rtbm.2014.04.003.
- Lavalle, S., Lesser, E., Shockley, R., Hopkins, M., Kruschwitz, N., 2011. Big data, analytics and the path from insights to value. *MIT Sloan Management Review* 52, 21–32.
- Linnéusson, G., Ng, A.H.C., Aslam, T., 2019. A hybrid simulation-based optimization framework supporting strategic maintenance development to improve production performance. *European Journal of Operational Research* 281, 402–414. doi:10.1016/j.ejor.2019.08.036.
- Liu, C., Kou, G., Zhou, X., Peng, Y., Sheng, H., Alsaadi, F., 2019a. Time-dependent vehicle routing problem with time windows of city logistics with a congestion avoidance approach. *Knowledge-Based Systems* 188. doi:10.1016/j.knsys.2019.06.021.
- Liu, Y., Chen, Y., Jiang, T., 2019b. Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach. *European Journal of Operational Research* 283, 166–181. doi:10.1016/j.ejor.2019.10.049.
- Liziwe, N., Beer, M., Delene, V., 2008. Predicting work performance through selection interview ratings and psychological assessment. *South African Journal of Industrial Psychology* 34. doi:10.4102/sajip.v34i3.750.
- Loi, R., Ngo, H.Y., Zhang, L., Lau, V., 2011. The interaction between leader-member exchange and perceived job security in predicting

- employee altruism and work performance. *Journal of Occupational and Organizational Psychology* 84, 669–685. doi:10.1348/096317910X510468.
- Manne, A.S., 1960. Linear programming and sequential decisions. *Management Science* 6, 259–267. doi:10.1287/mnsc.6.3.259.
- Martínez, A., Schmuck, C., Pereverzyev, S., Pirker, C., Haltmeier, M., 2018. A machine learning framework for customer purchase prediction in the non-contractual setting. *European Journal of Operational Research* 281, 588–596. doi:10.1016/j.ejor.2018.04.034.
- Masson, R., Lehuédé, F., Péton, O., 2012. Simple temporal problems in route scheduling for the dial-a-ride problem with transfers, in: *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, Springer. pp. 275–291.
- Masson, R., Lehuédé, F., Péton, O., 2013. An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science* 47, 344–355. doi:10.1287/trsc.1120.0432.
- Masson, R., Lehuédé, F., Péton, O., 2014a. The dial-a-ride problem with transfers. *Computers & Operations Research* 41, 12–23. doi:10.1016/j.cor.2013.07.020.
- Masson, R., Ropke, S., Lehuédé, F., Péton, O., 2014b. A branch-and-cut-and-price approach for the pickup and delivery problem with shuttle routes. *European Journal of Operational Research* 236, 849–862. doi:10.1016/j.ejor.2013.08.042.
- Masud, A., 1985. Forecasting plant labor productivity with a time series model. *Computers & Industrial Engineering* 9, 73–81. doi:10.1016/0360-8352(85)90038-5.
- Matusiak, M., De Koster, R., Saarinen, J., 2017. Utilizing individual picker skills to improve order batching in a warehouse. *European Journal of Operational Research* doi:10.1016/j.ejor.2017.05.002.
- McCall, J.J., 1963. Operating characteristics of opportunistic replacement and inspection policies. *Management Science* 10, 1–191. doi:10.1287/mnsc.10.1.85.
- McCall, J.J., 1965. Maintenance policies for stochastically failing equipment: A survey. *Management Science* 11, 493–615. doi:10.1287/mnsc.11.5.493.
- McKinsey, 2019. Getting a handle on warehousing costs. <https://www.mckinsey.com/business-functions/operations/our-insights/getting-a-handle-on-warehousing-costs#>. Accessed: 2022-01-24.
- Mentzer, J., Dewitt, W., Keebler, J., Min, S., Nix, N., Smith, C., Zacharia, Z., 2001. Defining supply chain management. *Journal of Business Logistics* 22, 1–25. doi:10.1002/j.2158-1592.2001.tb00001.x.
- Mercier, S., Castro, I.T., 2018. Stochastic comparisons of imperfect maintenance models for a gamma deteriorating system. *European Journal of Operational Research* 273, 237–248. doi:10.1016/j.ejor.2018.06.020.

- Minic, S., Laporte, G., 2006. The pickup and delivery problem with time windows and transshipment. *INFOR* 44, 217–227. doi:10.1080/03155986.2006.11732749.
- Mor, R., Bhardwaj, A., Singh, S., Sachdeva, A., 2019. Productivity gains through standardization-of-work in a manufacturing company. *Journal of Manufacturing Technology Management* 30, 899–919. doi:10.1108/JMTM-07-2017-0151.
- Mues, C., Pickl, S., 2006. Transshipment and time windows in vehicle routing, p. 6 pp. doi:10.1109/ISPAN.2005.88.
- Munkres, J., 1957. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* 5, 32–38.
- Nakagawa, T., Kowada, M., 1983. Analysis of a system with minimal repair and its application to replacement policy. *European Journal of Operational Research* 12, 176–182. doi:10.1016/0377-2217(83)90221-7.
- Nikolopoulos, K., Punia, S., Schäfers, A., Tsinopoulos, C., Vasilakis, C., 2021. Forecasting and planning during a pandemic: COVID-19 growth rates, supply chain disruptions, and governmental decisions. *European Journal of Operational Research* 290, 99–115. doi:10.1016/j.ejor.2020.08.001.
- Nocedal, J., Wright, S., 2006. *Sequential Quadratic Programming*. Springer New York, New York, NY. pp. 529–562. doi:10.1007/978-0-387-40065-5_18.
- Oertel, P., 2001. Routing with reloads. Ph.D. thesis. Universität zu Köln.
- Omshi, E.M., Grall, A., Shemehsavar, S., 2019. A dynamic auto-adaptive predictive maintenance policy for degradation with unknown parameters. *European Journal of Operational Research* 282, 81–92. doi:10.1016/j.ejor.2019.08.050.
- Peng, Z., Al Chami, Z., Manier, H., Manier, M.A., 2019. A hybrid particle swarm optimization for the selective pickup and delivery problem with transfers. *Engineering Applications of Artificial Intelligence* 85, 99–111. doi:10.1016/j.engappai.2019.06.006.
- Pierotti, J., Van Essen, T., 2021. MILP models for the dial-a-ride problem with transfers. *EURO Journal on Transportation and Logistics* 10, 100037. doi:10.1016/j.ejtl.2021.100037.
- Powell, S., Schultz, K., 2004. Throughput in serial lines with state-dependent behavior. *Management Science* 50, 1095–1105. doi:10.1287/mnsc.1040.0233.
- Qin, X., Shao, L., Jiang, Z.Z., 2019. Contract design for equipment after-sales service with business interruption insurance. *European Journal of Operational Research* 284, 176–187. doi:10.1016/j.ejor.2019.12.027.
- Qu, Y., Bard, J., 2012. A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research* 39, 2439–2456. doi:10.1016/j.cor.2011.11.016.
- Radner, R., Jorgenson, D.W., 1963. Opportunistic replacement of a single part in the presence of several monitored parts. *Management Science* 10, 70–84. doi:10.1287/mnsc.10.1.70.

- Rais, A., Alvelos, F., Carvalho, M., 2014. New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research* 235, 530–539. doi:10.1016/j.ejor.2013.10.038.
- Reinhardt, L., Clausen, T., Pisinger, D., 2013. Synchronized dial-a-ride transportation of disabled passengers at airports. *European Journal of Operational Research* 225, 106–117. doi:10.1016/j.ejor.2012.09.008.
- Ribeiro, J.P., Barbosa-Povoa, A., 2018. Supply chain resilience: Definitions and quantitative modelling approaches - A literature review. *Computers & Industrial Engineering* 115, 109–122. doi:10.1016/j.cie.2017.11.006.
- Rincon-Garcia, N., Waterson, B., Cherrett, T., Salazar Arrieta, F., 2020. A metaheuristic for the time-dependent vehicle routing problem considering driving hours regulations – an application in city logistics. *Transportation Research Part A Policy and Practice* 137, 429–446. doi:10.1016/j.tra.2018.10.033.
- Russom, P., et al., 2011. Big data analytics. TDWI best practices report, fourth quarter 19, 1–34.
- Sampaio, A., Savelsbergh, M., Veelenturf, L., Van Woensel, T., 2020. Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers. *Networks* 76, 232–255. doi:10.1002/net.21963.
- Sanders, N., Ritzman, L., 2004. Using warehouse workforce flexibility to offset forecast errors. *Journal of Business Logistics* 25, 251–269. doi:10.1002/j.2158-1592.2004.tb00189.x.
- Sasieni, M.W., 1956. A markov chain process in industrial replacement. *OR* 7, 148–155. doi:10.2307/3007561.
- Schönberger, J., 2017. Scheduling constraints in dial-a-ride problems with transfers: A metaheuristic approach incorporating a cross-route scheduling procedure with postponement opportunities. *Public Transport* 9, 243–272. doi:10.1007/s12469-016-0139-6.
- Schwarzkopf, A., Tersine, R., Morris, J., 1988. Top-down versus bottom-up forecasting strategies. *The International Journal Of Production Research* 26, 1833–1843. doi:10.1080/00207548808947995.
- Shang, J., Cuff, C., 1996. Multicriteria pickup and delivery problem with transfer opportunity. *Computers & Industrial Engineering* 30, 631–645. doi:10.1016/0360-8352(95)00181-6.
- Shaw, A.W., 1915. *Some Problems in Market Distribution: Illustrating the Application of a Basic Philosophy of Business*. Harvard University Press.
- Shen, J., Cui, L., Ma, Y., 2018. Availability and optimal maintenance policy for systems degrading in dynamic environments. *European Journal of Operational Research* 276, 133–143. doi:10.1016/j.ejor.2018.12.029.
- Shih, W.C., 2020. Global supply chains in a post-pandemic world. <https://hbr.org/2020/09/global-supply-chains-in-a-post-pandemic-world>. Accessed: 2022-01-24.
- Sivasankaran, P., Shahabudeen, P., 2014. Literature review of assembly line balancing problems. *The International Journal of Advanced*

- Manufacturing Technology 73, 1665–1694.
doi:10.1007/s00170-014-5944-y.
- Sonmez, R., Rowings, J., 1998. Construction labor productivity modeling with neural networks. *Journal of Construction Engineering and Management-asce* 124. doi:10.1061/(ASCE)0733-9364(1998)124:6(498).
- Speranza, M., Guastaroba, G., Vigo, D., 2016. Intermediate facilities in freight transportation planning: A survey. *Transportation Science* 50, 763–789.
doi:10.1287/trsc.2015.0631.
- Srinivasan, M.M., Bozer, Y.A., Cho, M., 1994. Trip-based material handling system: Throughput capacity analysis. *IIE Transactions* 26, 70–89.
doi:10.1080/07408179408966586.
- Statista, 2021a. Number of warehouses worldwide from 2020 to 2025. <https://www.statista.com/statistics/1271245/warehouses-worldwide/>. Accessed: 2022-01-24.
- Statista, 2021b. Share of total supply costs worldwide in 2018 by type of cost. <https://www.statista.com/statistics/1043253/share-of-total-supply-chain-costs-by-type-worldwide/>. Accessed: 2022-01-24.
- Staudt, F.H., Alpan, G., Mascolo, M.D., Rodriguez, C.M.T., 2015a. Warehouse performance measurement: A literature review. *International Journal of Production Research* 53, 5524–5544.
- Staudt, F.H., Alpan, G., Mascolo, M.D., Rodriguez, C.M.T., 2015b. Warehouse performance measurement: A literature review. *International Journal of Production Research* 53, 5524–5544.
- Steinker, S., Hoberg, K., Thonemann, U.W., 2017. The value of weather information for e-commerce operations. *Production and Operations Management* 26, 1854–1874. doi:10.1111/poms.12721.
- Szondy, D., 2017. IBM patent outlines mid-air drone-to-drone cargo transfers. <https://newatlas.com/ibm-patent-delivery-drones-mid-air-transfer/49301/>. Accessed: 2022-01-20.
- Takoudjou, R.T., Deschamps, J.C., Dupas, R., 2012. A hybrid multistart heuristic for the pickup and delivery problem with and without transshipment, in: 9th International Conference on Modeling, Optimization & SIMulation.
- Thangiah, S., Fergany, A., Awan, S., 2007. Real-time split-delivery pickup and delivery time window problems with transfers. *Central European Journal of Operations Research* 15, 329–349. doi:10.1007/s10100-007-0035-x.
- Thomas, H.R., Sakarcin, A.S., 1994. Forecasting labor productivity using factor model. *Journal of Construction Engineering and Management* 120, 228–239. doi:10.1061/(ASCE)0733-9364(1994)120:1(228).
- Thompson, G., Goodale, J., 2006. Variable employee productivity in workforce scheduling. *European Journal of Operational Research* 170, 376–390.
doi:10.1016/j.ejor.2004.03.048.
- Tipping, M., 2001. Sparse bayesian learning and relevance vector machine. *J. Mach. Learn. Res.* 1, 211–244. doi:10.1162/15324430152748236.

- Tiwari, S., Wee, H., Daryanto, Y., 2017. Big data analytics in supply chain management between 2010 and 2016: Insights to industries. *Computers & Industrial Engineering* 115. doi:10.1016/j.cie.2017.11.017.
- Tran, V.T., Yang, B.S., Tan, A.C.C., 2009. Multi-step ahead direct prediction for the machine condition prognosis using regression trees and neuro-fuzzy systems. *Expert Systems with Applications* 36, 9378–9387. doi:10.1016/j.eswa.2009.01.007.
- Tufano, A., Accorsi, R., Manzini, R., 2021. A machine learning approach for predictive warehouse design. *The International Journal of Advanced Manufacturing Technology* doi:10.1007/s00170-021-08035-w.
- Twist, R., Akker, J., Hoogeveen, H., 2021. Synchronizing transportation of people with reduced mobility through airport terminals. *Computers & Operations Research* 125, 105103. doi:10.1016/j.cor.2020.105103.
- Usher, J.S., Kamal, A.H., Syed, W.H., 1998. Cost optimal preventive maintenance and replacement scheduling. *IIE Transactions* 30, 1121–1128. doi:10.1080/07408179808966568.
- Valdez-Flores, C., Feldman, R.M., 1989. A survey of preventive maintenance models for stochastically deteriorating single-unit systems. *Naval Research Logistics* 36, 419–446. doi:10.1002/1520-6750(198908)36:4<419::aid-nav3220360407>3.0.co;2-5.
- Voigt, S., Kuhn, H., 2021. Crowdsourced logistics: The pickup and delivery problem with transshipments and occasional drivers. *Networks* doi:10.1002/net.22045.
- Wang, H., 2002. A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research* 139, 469–489. doi:10.1016/S0377-2217(01)00197-7.
- Wang, X., Li, L., Xie, M., 2019. An unpunctual preventive maintenance policy under two-dimensional warranty. *European Journal of Operational Research* 282, 304–318. doi:10.1016/j.ejor.2019.09.025.
- Weiss, G.H., 1962. A problem in equipment maintenance. *Management Science* 8, 266–277. doi:10.1287/mnsc.8.3.266.
- Wolfinger, D., 2021. A large neighborhood search for the pickup and delivery problem with time windows, split loads and transshipments. *Computers & Operations Research* 126, 105110. doi:10.1016/j.cor.2020.105110.
- Wolfinger, D., Salazar González, J.J., 2021. The pickup and delivery problem with split loads and transshipments: A branch-and-cut solution approach. *European Journal of Operational Research* 289, 470–484. doi:10.1016/j.ejor.2020.07.032.
- Zhen, L., Wu, Y., Wang, S., Yi, W., 2021. Crowdsourcing mode evaluation for parcel delivery service platforms. *International Journal of Production Economics* 235, 108067. doi:10.1016/j.ijpe.2021.108067.
- Zhou, W., Lin, J., 2019. An on-demand same-day delivery service using direct peer-to-peer transshipment strategies. *Networks and Spatial Economics* 19, 409–443. doi:10.1007/s11067-018-9385-2.

Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S., Li, G.P., 2020. Predictive maintenance in the industry 4.0: A systematic literature review. *Computers & Industrial Engineering* 150, 106889.