Eindhoven University of Technology

MASTER

Throughput time reduction in an e-fulfilment context : design and evaluation of a job sequencing tool

Eras, J.

Award date:
2020

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 17. Jun. 2025

Department of Industrial Engineering and Innovation Sciences

Operations, Planning, Accounting and Control Group

Throughput time reduction in an e-fulfilment context: Design and evaluation of a job sequencing tool

A case study conducted at Ingram Micro Commerce & Lifecycle Services in Waalwijk, the Netherlands

J. (Jordy) Eras

BSc Industrial Engineering

MSc Finance

Student Identity Number 0885747

In partial fulfilment of the requirements for the degree of

Master of Science

in Operations Management and Logistics

Supervisors:

dr. ir. H.P.G. (Henny) van Ooijen	Eindhoven University of Technology, OPAC
dr. ir. R.A.C.M. (Rob) Broekmeulen	Eindhoven University of Technology, OPAC
R. (Ralph) Crombeecke BSc	Operations manager BFC, Ingram Micro CLS

3rd Assessor:

prof. dr. ir. I.J.B.F. Adan	Eindhoven University of Technology, OPAC

Tilburg, 21st of June 2020

TUE. School of Industrial Engineering.

Series Master Thesis Operations Management and Logistics

Subject headings: 3PL, e-fulfilment, job sequencing, job priority management, NUB-rule, throughput time reduction, warehouse, waiting time reduction, Work In Next Queue.

It should be noted that all numbers and costs mentioned in this thesis are fictitious due to confidentiality reasons

Abstract

The importance of optimizing customer response time and reducing throughput time in e-fulfilment is widely recognized among academics. As the picking process is the most labor-intensive process within warehouse operations, most research is focused on improving order picking efficiency. However, integral design solutions for optimizing the picking process and consecutive sorting and packing operations simultaneously are still lacking. In response, this research designs and evaluates a job sequencing tool in a dynamic e-fulfilment context and tests its ability to reduce throughput time by incorporating a variant of the Work In Next Queue logic while taking both next-day and same-day cut-off times into account. Additionally, a FUB-rule is designed which is a modification of the NUB-rule. Results of a case study at Ingram Micro CLS demonstrate that throughput and waiting time reductions between [3.0 – 8.2] and [6.5 – 19.8] respectively can be realized. It is found that these reductions are accompanied by better workload balance among the outbound work centers which has several operational benefits that may result in productivity gains. However, the exact mechanisms through which workload balancing reduces throughput times are not clear as better performance is not always accomplished through higher levels of workload balancing in the case study results. Following from some limitations of this research, several interesting directions for further research are identified that may be leveraged by practitioners as well as academics and graduate students.

Management Summary

Problem statement and assignment

The e-commerce business is rapidly growing; even with the recent COVID-19 pandemic online retailers thrive while traditional brick-and-mortar retailers face many operations-related challenges. Along with fierce market competition, this pressurizes e-retailers to provide customers with short lead times. Thus, minimizing throughput time should be a top-priority objective within warehouses.

This thesis was written in the context of a case study at Ingram Micro Commerce & Lifecycle Services in Waalwijk, the Netherlands. Within the case study company, waiting time and idle time are observed which negatively impacts throughput time. It is believed that this is caused by a misalignment between the order picking process and consecutive sorting and packing operations. More specifically, there are workload imbalances among the work centers.

The literature study suggests that better alignment of picking and consecutive sorting and packing operations will smoothen the workload which would reduce both waiting time and idle time while orders are still fulfilled within the required due dates. Hence, throughput time can be reduced.

The main question that is addressed in this thesis is whether throughput time can be reduced by better aligning operations through workload balancing. A job sequencing tool for pick batch picking that considers workload balancing is designed in this thesis and applied in a case study to test whether throughput time can be reduced. The assignment of this thesis was defined as follows:

"Design a job sequencing tool for pick batch picking that considers workload balancing such that throughput time is reduced"

Conceptual model

In this thesis a job sequencing tool was designed that takes workload balancing into account by applying a variant of the Work in Next Queue logic to a job shop environment in which two main flows and multiple job types are distinguished. The first flow comprises a two-stage tandem process (picking and

packing) where each process consists of multiple independent parallel servers. The second flow is a three-stage tandem process (picking, pool completion and sorting) in which an assembly operation of jobs is required (pool completion) for sorting to commence. This assembly process is considered when sequencing jobs by a modification of the NUB-rule to a FUB-rule.

Case study results

The job sequencing logic was built in a simulation tool to test whether throughput time, waiting time and pool completion time can be reduced through workload balancing. Several (sensitivity) scenarios were developed and compared to base case scenarios where the current sequencing logic is used.

First, throughput time as well as variability of throughput times can be reduced when the proposed logic is used compared to the current logic. This results in the outbound process from picking to sorting and packing to be more controllable and predictable: picking is more aligned with sorting and packing. The sources of throughput time reduction lie in 1) waiting time reduction for mono jobs and 2) in waiting and pool completion time (PCT) reduction for multi pools. For mono jobs, waiting time reductions are observed between 6.5 – 13.4 percent. For multi jobs waiting time reduction is larger: between 7.4-19.8 percent. Additionally, for multi pools, throughput time is reduced by reducing pool completion time. In line with expectation, PCT reduction for the Black Friday week suggests that the magnitude of PCT reduction is smaller in busier periods. Still, performance is better than in the base case. Moreover, in practice the case study company uses troubleshooter operators to manage the prioritization of jobs such that multi pools are completed sooner. Around €80,000 a year can be saved solely by automatically managing this prioritization if the proposed logic were to be applied.

Generalizability of the results is thoroughly discussed in this thesis where it is stressed that one must think through whether the assumptions made in this thesis' case study are valid for the business in consideration. However, it is argued that the proposed job sequencing tool can be applied in other businesses.

Conclusions

It is concluded from the case study at Ingram Micro CLS that throughput time can be reduced through waiting time and pool completion time reduction. This is not only beneficial for customer response time but also in terms of how the conveyor system and the queues are utilized.

Throughput time reductions and waiting time reductions between [3.0 – 8.2] and [6.5 – 19.8] respectively were found in this research. These reductions seem relatively small. However, since workload is more balanced, productivity gains may be expected as less shifting of operators between work centers is required. Additionally, reductions for multi pools were always found to be larger compared to reductions for mono jobs which is explained by the fact that pool completion is sped up; without requiring manual prioritization of jobs. In busier periods as well, gains in terms of throughput and waiting time are obtained, as well as for pool completion time. However, then the added value of the logic in terms of pool completion time is smaller.

Unlike expected, larger reductions were not always achieved through better workload balancing. However, reductions were always accompanied by better workload balance compared to the sequencing logic applied in the case study company. Thus, no solid conclusions are drawn regarding the exact mechanisms through which reductions are achieved through better balancing.

Recommendations and further research

Several interesting directions for further research are identified; some of which follow from some limitations of this research. Additionally, specific recommendations are made relevant for both practitioners and academic.

Ingram Micro CLS is recommended to first make sure that data is available in real-time for the logic to be able to function properly and it is recommended that Ingram Micro CLS invests resources in ensuring the real-time availability of this data as it can also be used for future operations research. A good starting point would be to include this data in the newly developed Warehouse Execution System (WES); a system that would be very useful to the sequencing tool as it contains information regarding the location and characteristics of jobs in the warehouse.

Preface

Tilburg, 18th of June 2020

Writing this preface is one of the most rewarding moments of my life as it concludes my graduation project of the Master Operations Management and Logistics at Eindhoven University of Technology (TU/e) which I conducted at Ingram Micro Commerce & Lifecycle Services (CLS). I have learned many valuable things from a lot of people during this research for which I want to express my gratitude.

I want to thank Henny van Ooijen, my first university supervisor, for our many laughs and valuable discussions. Not only did he guide me through the master's thesis process, he also supervised me during the Bachelor End Project back in 2017. I also want to thank Rob Broekmeulen, my second university supervisor, for his feedback, but foremost importantly I want to thank him for his relentless enthusiasm. I am sure that I will never encounter a person more passionate about the world of retail operations in my life; whenever I see an empty shelve in the supermarket, I can't help it but to think of his lectures on the importance of a solid inventory policy.

Next, I want to thank Ralph Crombeecke, my first company supervisor, who acquainted me with the key players within Ingram Micro CLS and bol.com and allowed me to pitch my research to them while also challenging me to think critically. Also, I want to express my sincere gratitude to Tess Peltenburg who, at her time at Ingram Micro was always willing to support me whenever I needed it. Our valuable discussions lead to many important insights.

Also, I want to thank Thom Houweling, a friend and fellow graduate at Ingram Micro CLS at the Veerweg location. Our discussions took both our theses to a higher level. Additionally, I want to thank Lisa van Lierop, for her willingness to read the concept version of this thesis and most importantly, for our valuable times in Austria and Cuba.

Lastly, I want to thank my beloved parents for always believing in me and supporting me during the most important decisions in life.

Jordy Eras

Table of Contents

List of Figures

List of Tables

Abbreviations

3PL	Third-party logistics (service provider)
BFC	Bol.com Fulfilment Center
EDD	Earliest Due Date
FUB	Fraction of Unscheduled Branches
I/O	Input/Output
NUB	Number of Uncompleted Branches
MSE	Mean Squared Error
OLS	Ordinary Least Squares
PCT	(Picking) Pool Completion Time
VAS	Value-Added Services
WES	Warehouse Execution System
WMS	Warehouse Management System

1. Introduction

This section first describes the case study context in section 1.1. Section 1.2 describes the warehouse operations performed within the case study company. Section 1.3 elaborates on order batching as this process initiates outbound warehouse operations. Section 1.4 concludes this chapter by describing the problem that is addressed by this thesis.

1.1 Case study context

Ingram Micro Commerce & Lifecycle Services, hereafter named Ingram Micro CLS, is a third-party logistics provider (3PL) that provides e-fulfillment solutions as well as advanced omni-channel solutions for customers all over the world. With over 150 distribution centers in 45 countries, Ingram Micro CLS processes hundreds of thousands of items a day for its customers. Most customers are e-retailers (e.g. ASOS, bol.com and Zalando) while other customers are omni-channel retailers that sell products both online as well as in traditional stores (e.g. De Bijenkorf and Zara).

Ingram Micro CLS mainly provides receiving, storage, picking, sorting and packing solutions. In addition, it provides extra services such as (inter)national returns management, transportation management, stock management and business intelligence. For in- and outbound transportation, Ingram Micro CLS relies on external parties.

This research was conducted at the Bol.com Fulfilment Center (BFC), which was put into operation in 2017 and since then has been operated by Ingram Micro CLS. With sales of 2.8 billion euros in 2019, bol.com continues to be the largest online retailer in The Netherlands. The company's rapid growth (about 30% per year) requires continuous improvement of existing processes and considerations of capacity expansions. Additionally, with start of the COVID-19 pandemic growth accelerated even more. In 2019, construction was started to expand the BFC to 100.000 square meters. Bol.com's demand is highly seasonal where peak demand (starting in the week of black Friday in November until Christmas in December) and off-peak demand (January until October) can be distinguished.

1.2 Warehouse operations

Figure 1 provides an overview of the warehouse operations performed in the BFC. Operations start with goods receival and finish in the packing area. When packed, conveyor belts transport the orders to external parties that provide outbound transport (e.g. PostNL).

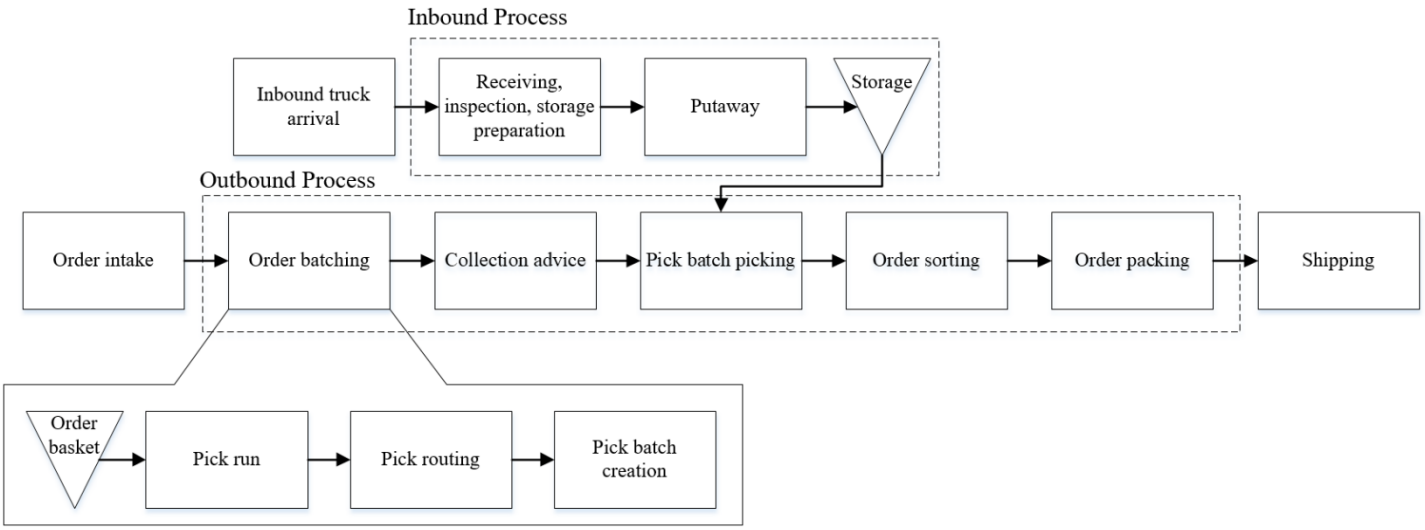


Figure 1: General warehouse operations BFC.

The inbound process comprises receiving and putaway. The first process starts with the arrival of trucks. Upon arrival, trucks are unloaded and pallets with goods are stored near the loading docks. Next, goods are inspected and prepared for storage by unloading the pallets and subsequently placing the items in totes (see Figure 2) which are transported to storage locations by conveyor belts. In the putaway process, operators manually put items from the totes in storage locations.

Figure 2: Totes used for putaway and pick batch picking.

The outbound process comprises picking, sorting and packing. Based on bol.com's customers' preferences, each order is assigned a cut-off time at which the order must be packed. Picking starts with pickers collecting a pick batch suggested by the collection advice that takes cut-off time into account. Picking is done following the zone picking policy which allows pickers to retrieve SKUs from within a single zone (Petersen and Aase, 2004). A pick batch contains multiple items that are directed to the same packing area and fit in one tote. Totes can contain multiple orders that consist of one item each (mono totes). In this case, once a pick batch is completed, the tote is transported to its primary packing destination (either manual or mechanical) via a conveyor belt.

Alternatively, totes can contain parts of orders that consist of multiple items (multi totes). These totes are part of a multi-tote pool and are sent to a tote buffer called the stingray. Once the multi-tote pool is complete (i.e. all pick batches within that pool are picked), all totes within that pool are transported to either manual or automatic sorting stations. As multi orders contain multiple items spread over multiple totes, multi orders are completed by assembling the items that together complete the multi order. This process is performed at the sorting stations. Once sorted, orders are packed manually. Finally, a conveyor belt transports the orders to the outbound transportation provider for shipping. The outbound flow of totes is presented in Figure 3.

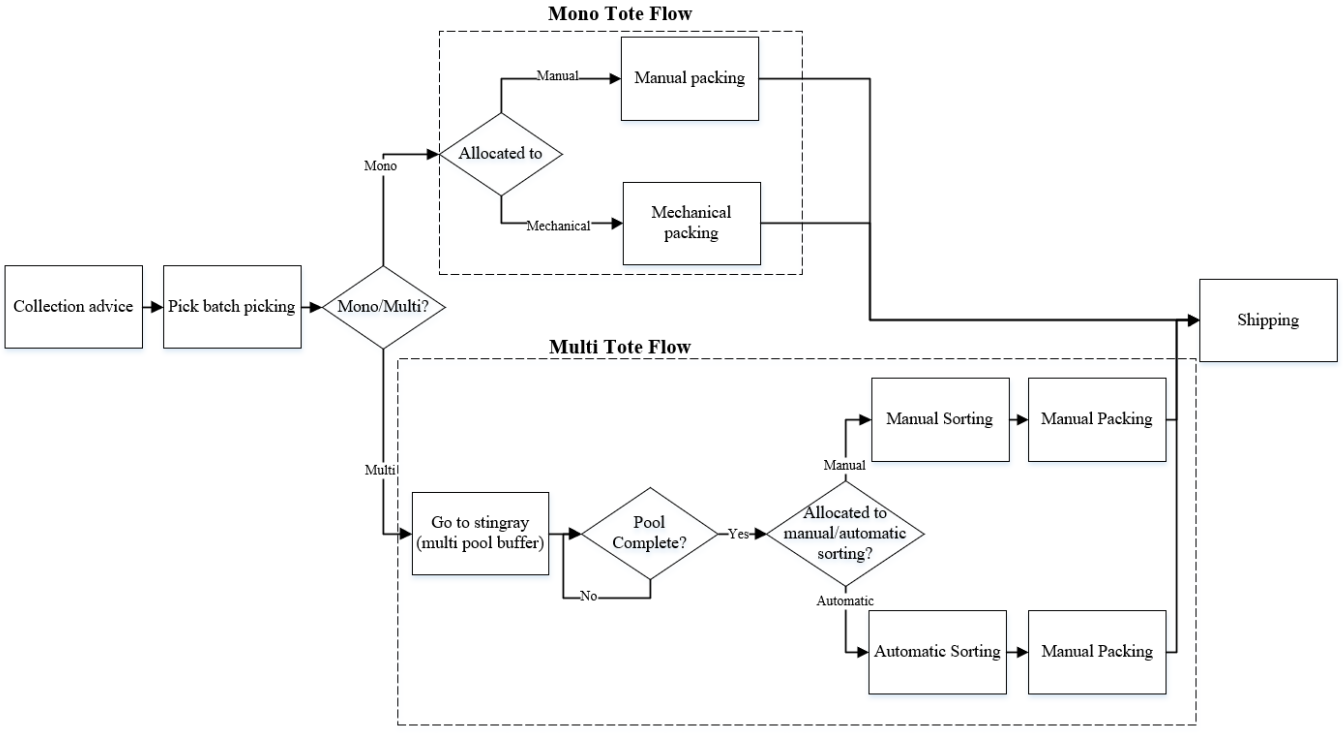


Figure 3: Outbound flow of totes.

1.3 Order batching

When an order is placed at the bol.com website, the order is put in the order basket. Once there are enough orders in the basket, these orders are released for picking. When a batch of orders is released, the Warehouse Management System (WMS) first checks which of these orders are due first. Subsequently, an algorithm called Pacman maximizes the number of items per zone while minimizing the number of zones and totes (pick batches) used. Next, an algorithm called Blinky reshuffles pick batches within these zones such that walking distance is minimized. Then, the optimal route is found by an algorithm called Roadrunner. Items are picked in batches and as a result, one pick batch may contain multiple (or parts of) orders. Combining multiple orders into pick batches is a policy that reduces picking time significantly compared to strict-order picking (Gibson and Sharp, 1992; Petersen, 1997; De Koster, Poort, and Wolters, 1999). Finally, a sequencing logic (the collection advice) provides pickers with batches to pick based on cut-off times. Section 3.2 further elaborates on the sequencing logic.

Jobs are released to picking in the form pick batches. There are multiple job types which are referred to as pick sequences (see Table 1). More on these job types is discussed in section 3.

Table 1: Pick sequences.

Pick sequence code	Name
101	Mono High Risk
102	Mono Manual VAS
103	Mono Manual Regular
104	Mono Smartmailer
105	Mono Cartonwrap
106	Multi High Risk
107	Multi Manual VAS
108	Multi Manual
109	Multi Automatic Sorting

Orders that require same-day delivery are processed on ‘High Risk’ outbound lines and orders that require gift wrapping are processed on Value Added Services (VAS) stations. Smaller, non-fragile items can be processed on the Smartmailer or Cartonwrap packing machines. When batching, one basically aims to balance the workload between picking, sorting and packing operations such that there is enough workload for each of the pick sequences while ensuring that orders are processed within their required due dates (i.e. 100 percent order fulfilment).

1.4 Problem description

Although one aims to achieve a balanced workload by batching orders for various pick sequences (i.e., outbound sorting and packing lines), in practice it is experienced that the order picking process is not aligned with sorting and packing operations.

This misalignment is observed by packing and sorting operators standing still when there is no work for them whereas there could have been work for them. This buffer starvation (or idle time) at packing (or sorting) line i may be a result of 1) order pickers processing pick batches for other outbound lines or 2) a lack of pick batches for outbound line i in the WMS. Also, it occurs that order pickers pick too many pick batches for one particular outbound line at a time; generating a peak in the workload for that outbound line with long waiting times as a result.

Consequently, waiting time and idle time occur which negatively impacts throughput time. Whether an outbound line is subject to waiting time or idle time varies over the day. Most often a disbalance is observed when comparing the mono packing and multi sorting stations which can be illustrated by means of an example. As mentioned in section 1.2, a distinction can be made between mono-totes and multi-totes. Now suppose:

- Multi-pool *ABC* consists of multi pick batch 1 and multi pick batch 2 which must be picked in zone x and y respectively.
- Multi-pool *ABC* must be sorted (i.e. the orders are within the pick batches are assembled) at the Multi Manual Sorting work centers.
- Mono Manual Regular packing does not have available capacity
- Multi Manual Sorting station has ample capacity.

After jobs (pick batches) are released, jobs are queued for picking. Then, these jobs are sequenced to picking operators largely based on the FCFS rule. At some point in time, the queue of jobs for zones x and y may look like Table 2.

Table 2: Example of the order of picking scheme suggested by the current job sequencing logic.

Order of picking:	Zone x	Zone y
1	**Multi pick batch 1**	Mono pick batch
2	Mono pick batch	Mono pick batch
3	Mono pick batch	Mono pick batch
4		**Multi pick batch 2**

The scheme of jobs as presented in Table 2 results in multi pick batch 1 being sequenced (i.e. picked) first while multi pick batch 2 will be sequenced once all mono pick batches in zone y have been picked. As a result, from a workload balancing point of view, multi pool ABC will arrive at multi sorting later than desired; sorting can only be completed once multi pick batch 2 arrives. At mono packing on the other hand, mono pick batches are piling up with waiting time as a result.

Firstly, and most importantly, the main negative consequence of the workload imbalances across the work centers is that throughput time is negatively affected (long waiting times); i.e., plenty of work in the queue at some work centers whereas operators stand idle at other work centers.

Secondly, the completion of multi pools (as in the example of multi pool *ABC*) is experienced as problematic. It occurs multiple times a day that sorting team leaders phone the picking department (or the control room) saying that they have no complete multi pools to process. Then, they request the picking department to prioritize jobs that would complete pools. In an attempt to speed up pool completion, there are operators fully focused at pool completion. The following is quoted from bol.com's internal documentation: *"We use daily around 10 hours of troubleshooting to manage the prioritizing. The control room has full focus on prioritizing during the day"*. This quote refers to situations where certain jobs are manually prioritized over others (e.g. prioritizing multi pick batch 2 from Table 2). However, pool completion is still found to be taking too long.

Thirdly, to cope with the imbalance the control room and work centers' team leaders shift operators from work center A to work center B when idleness is observed at A and plenty of work is in the queue at B. Next, when the imbalance shifts (i.e., B is idle while now A is busy), operators are shifted again. This back and forth movement of personnel occurs several times a day. This experienced as hindering and unpleasant by the control room (and team leaders) as the feeling of being in control is reduced.

It is believed that better alignment of picking and consecutive sorting and packing operations (i.e., smoothening the workloads) will reduce waiting time (hence throughput time). Additionally, it is believed that reducing the above-mentioned pool completion time will positively contribute to this waiting time reduction. Moreover, it is expected that operators are shifted less often.

2. Assignment

Section 2.1 provides a brief literature review that is concluded by the identification of three literature gaps. Section 2.2 states the problem and the assignment of this thesis. Section 2.3 defines the project scope. Section 2.4 elaborates on the research methodology and provides the thesis outline.

2.1 Literature review

Market competition has been pressurizing e-retailers to provide customers with short delivery lead times. Orders are placed online throughout the day and may need to be picked, possibly sorted, packed and shipped on that day. Hence, minimizing order throughput time should be a top-priority objective within warehouses. Most research is purely focused on the order picking process in the context of throughput times. In academic research only little attention is paid to the consecutive processes of sorting and packing. However, capacity management of picking, sorting, and packing operations altogether is a problem that is common in practice (Van Nieuwenhuyse, de Koster, and Colpaert, 2007).

For example, literature on the order batching problem mostly aims to minimize pickers' total travel distance by consolidating orders. However, the impact of batch sizes on the subsequent sorting and packing operations is largely neglected. Smaller batch sizes may result in longer total travel distance at picking and orders piling up waiting to be picked but may also reduce the interarrival time of batches at sorting and packing. Larger batch sizes on the other hand may reduce total travel distance at picking but may result in significant workload fluctuations at sorting and packing operations.

In aligning operations, a question that arises is whether the capacities are used for the right activities at the right time. Van Den Berg (2007) stresses that processes should be completed just in time to prevent the following effects:

- Long waiting time at consecutive operations because of workload imbalances;
- Idle time at consecutive operations because of workload imbalances;
- Late completion of urgent activities because less urgent activities are performed first;
- Poor utilization of space, since goods must wait for other processes to finish; and
- Increased capacity requirements because operators perform activities that could wait until later.

Balancing workload can reduce overtime and idle time (buffer starvation), but may also increase employees' physical fitness and reduce absenteeism (De Leeuw and Wiers, 2015). Ideally, the workload should be constant during the day as it simplifies planning and operation (Kim, 2018). The effect of workload balancing by means of job management as opposed to FCFS is demonstrated in Figure 4.

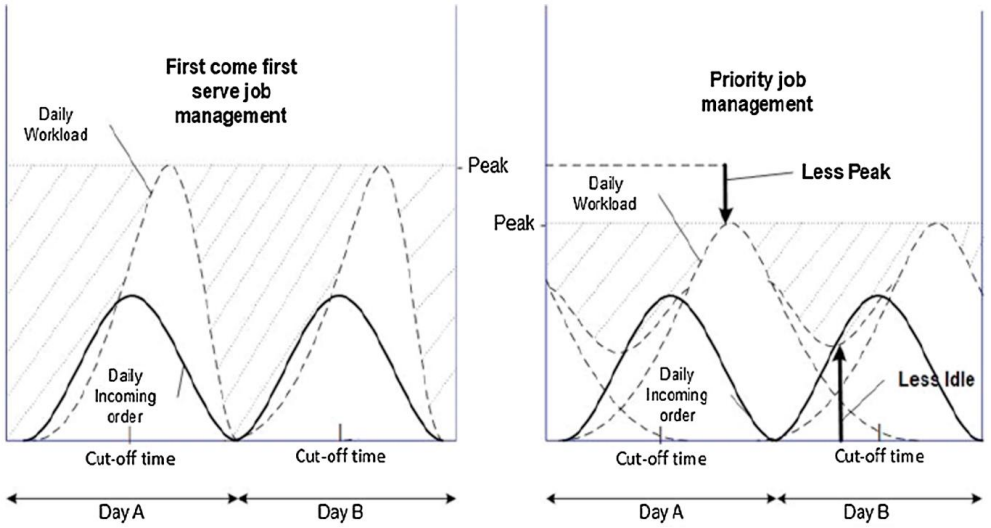


Figure 4: Daily incoming orders and two job management methods by Kim (2018).

Deciding on when to perform which task is addressed in the literature on job scheduling. Job scheduling is a decision-making process with many applications in manufacturing and service industries. A large body of literature is focused on the scheduling problem where it is assumed that all jobs are known at the beginning of the planning period. For an overview of the classification of these static scheduling problems we refer to Pinedo (2012) and for solutions to these static problems we refer to the literature review by Tyagi, Varshney, and Chandramouli (2013).

In e-fulfilment orders arrive throughout the day and thus the set of jobs is not known at the beginning of the planning period. These dynamic problems know no optimal solutions and hence heuristics are often used for sequencing jobs. Dispatching rules (e.g., EDD) are examples of such heuristics that are frequently used in practice given the ease of interpretation, implementation and predictability. However, these rules mostly use a single criterion to decide when to sequence which job.

2.1.1 Literature gaps

The importance of optimizing customer response time and reducing throughput time in e-fulfilment is widely recognized among academics. As the picking process is the most labor-intensive process within warehouse operations, most research is focused on improving order picking efficiency. However, integral design solutions for optimizing the picking process and consecutive operations simultaneously are still lacking.

Job scheduling can smoothen workload which reduces throughput and idle time, but research assumes that the set of jobs is known at the beginning of the planning period (static situation). Research fails to recognize that orders are not known at the beginning of the planning period. Hence, methods designed for static situations are not applicable to dynamic situations (e.g., e-fulfilment) where jobs arrive over time and frequent schedule modification or reaction to changing circumstances may be required.

Also, research that incorporates both same-day and next-day delivery in an e-fulfilment context is lacking. In addition, research makes simplifying assumptions on processing rates (e.g. deterministic) and considers situations with a limited amount of possible order flows.

Clearly, there is a gap in academic literature which is three-fold: (1) Global integral designs for picking and consecutive sorting and packing operations in e-fulfilment are lacking; (2) Research on scheduling and sequencing problems in e-fulfilment is scarce and assumes static situations in which all jobs are known at the beginning of the planning period; and (3) Research does not incorporate the presence of both same-day and next-day cut-off times in sequencing jobs.

In conclusion, further research is needed to develop an integral design for picking and consecutive sorting and packing operations in an e-fulfilment setting while using dynamic methods such that throughput time and idle time is reduced.

2.2 Problem statement and assignment

Within the case study company, waiting time and idle time are observed which negatively impacts throughput time. It is believed that this is caused by a misalignment between the order picking process and consecutive sorting and packing operations. More specifically, there are workload imbalances among the work centers.

Literature study suggests that better alignment of picking and consecutive sorting and packing operations will smoothen the workload which would reduce both waiting time and idle time while orders are still fulfilled within the required due dates. Hence, throughput time can be reduced.

The main question that is addressed in this thesis is whether throughput time can be reduced by better aligning operations through workload balancing. A job sequencing tool for pick batch picking that considers workload balancing will be designed in this thesis and applied in a case study to test whether throughput time can be reduced. Thus, the assignment of this project is defined as follows:

"Design a job sequencing tool for pick batch picking that considers workload balancing such that throughput time is reduced"

Ingram Micro CLS will serve as the case study company for which the tool will be tested to see if a reduction in throughput and waiting time can be achieved; i.e., whether it is able to reduce the negative effects of workload imbalances.

2.3 Project scope

As described in section 1.2, various operations are performed within the BFC. The project scope is graphically presented in Figure 5.

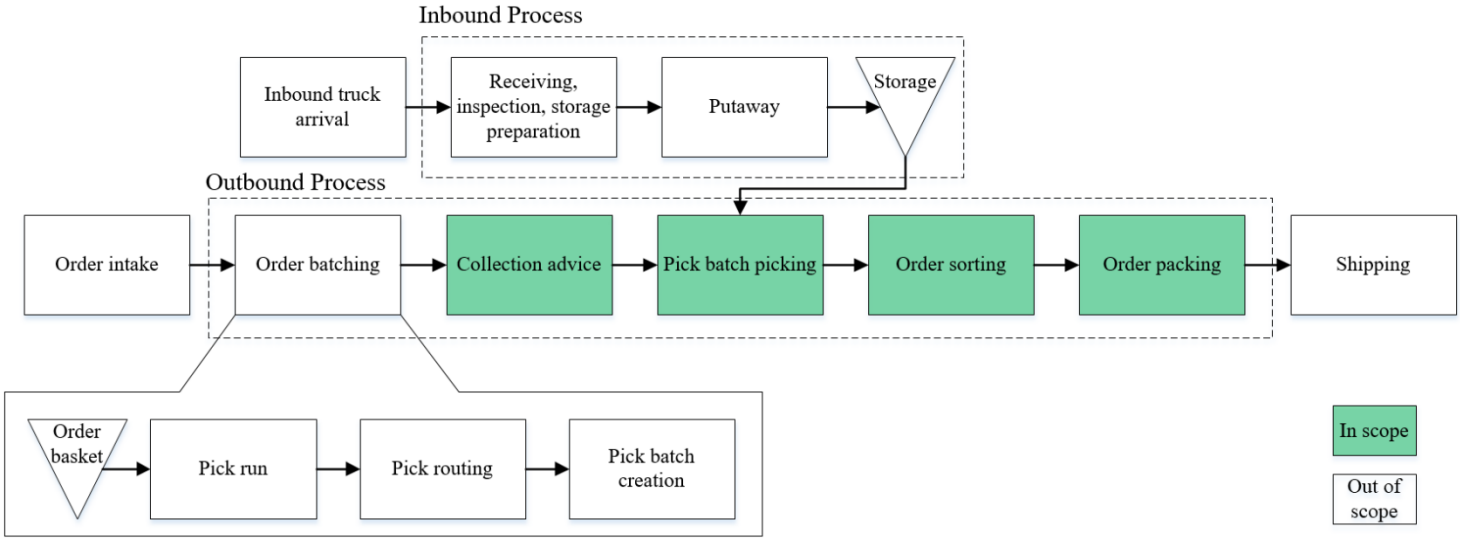


Figure 5: Project scope.

This project focusses on the outbound process; hence the inbound process is out of scope. Also, as shipping is performed by an outbound transportation provider out of Ingram Micro CLS' control, it is not within the scope of this project. Order batching involves various routing considerations (e.g., travel distance and travel time minimization) which is not within the scope of this project. However, the pick batches that are created as a result of the order batching process are within the scope of this project.

The assignment of this thesis is to design a job sequencing tool for pick batches (jobs). The process of allocating jobs to order pickers is referred to as the collection advice. Hence, we focus on the collection advice as the results from that advice determine the workload that will be generated at sorting and packing operations. The current job sequencing logic used for the collection advice is explained in section 3.2.

After a job is sequenced, the job is processed within the picking department (i.e., the pick batch is picked). The processes where one can distinguish pick batches (totes) are within the scope of this project. Hence, mono packing and sorting are within scope. Multi packing is out of scope as it processes customer orders instead of pick batches.

2.4 Research methodology and thesis outline

The research methodology used in this research is based on the reflective design by Van Aken, Van Berends, and Van der Bij (2012). This methodology is graphically presented in Figure 6.

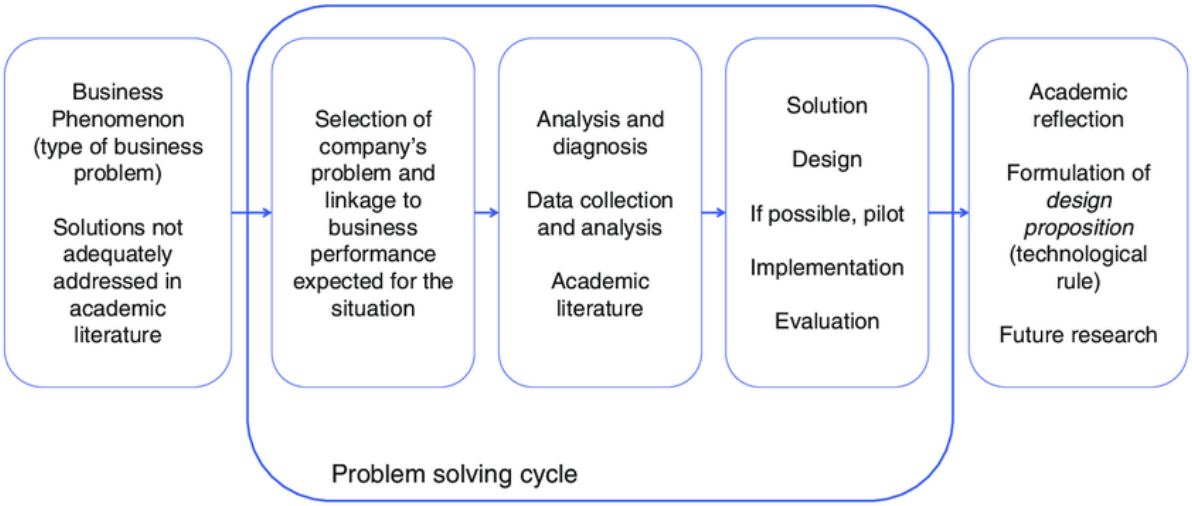


Figure 6: Reflective design (Van Aken et al., 2012).

The business problem is described in section 1.4 and literature gaps were identified in section 2.1.1. Section 3 provides a detailed analysis of the current state within the case study company where focus lies on describing the process environment, the current job sequencing logic, throughput and processing times. Section 4 turns to the solution design by presenting the conceptual model. The results of the case study are presented and discussed in section 5 and section 6 elaborates on the implementation. Section 7 concludes this thesis and provides practical recommendations for practitioners as well as recommendations for further research for academics.

3. Detailed analysis

This section provides a detailed analysis of the current state within the case study company. Section 3.1 classifies the process environment (referred to as machine environment in academic research). Section 3.2 elaborates on the current job sequencing logic. Section 3.3 describes the throughput times of the various mono and multi job types. Section 3.4 describes the processing times.

3.1 Process environment

A simplified representation of the process environment is presented in Figure 7. Mono and multi orders can be distinguished. Mono orders are packed at one of the three work centers after picking while multi orders must first be collected from a number of pick batches (also referred to as a pool) at one of the two sorting work centers. As mentioned in section 2.3, packing for multi orders is out of scope. Therefore, multi packing is not part of Figure 7.

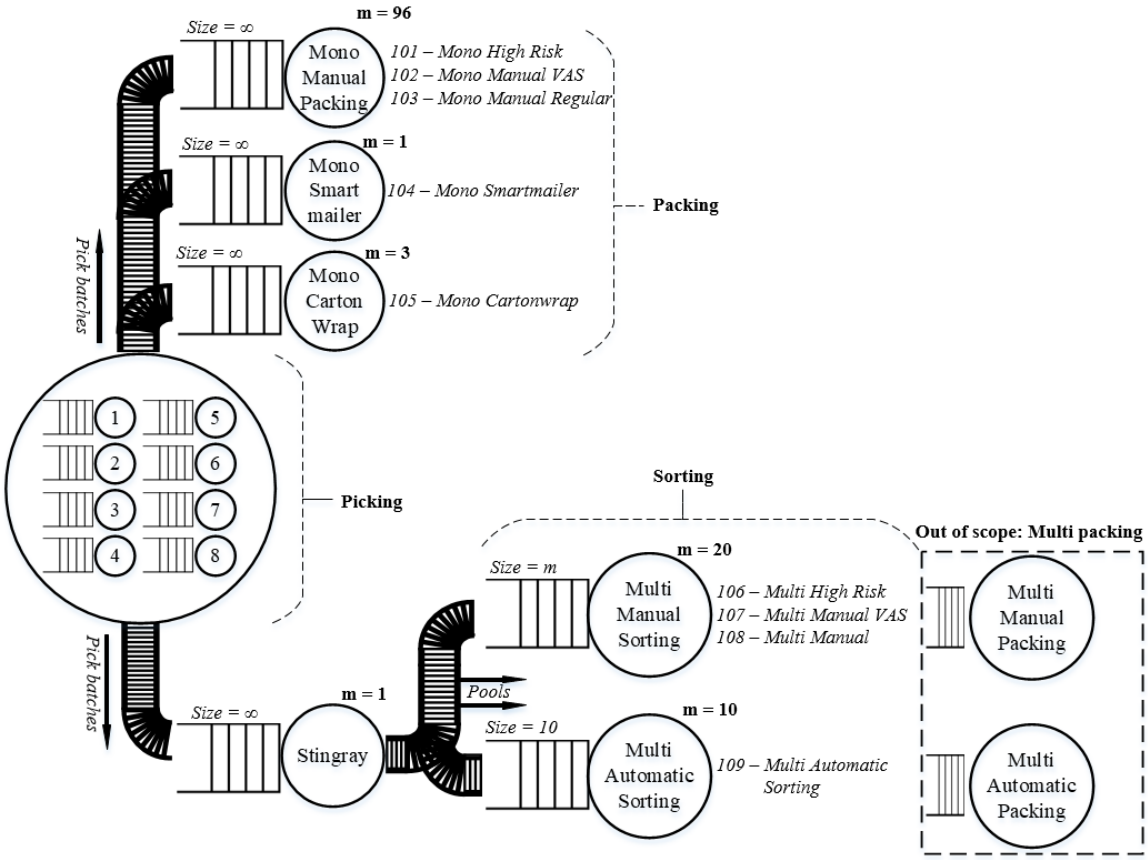


Figure 7: Simplified representation of the process environment.

Picking and the various packing and sorting lines are referred to as ***work centers***. The picking work center comprises eight picking regions, each with a queue of jobs to be picked within that specific

region. Within the packing and sorting work centers, one or more job types are processed by m ***servers*** in parallel. A ***job*** refers to the unit of processing at the various work centers. For example, Mono Manual Packing comprises at most 96 servers and process three job types: 101, 102, and 103. Packing uses pick batches (as picking does). Sorting however uses a number of pick batches (a pool) that together contain multiple complete multi orders. Thus, for picking and packing operations, a job refers to a pick batch. Whereas when sorting is considered, a job refers to a number of pick batches (a pool). Mono jobs requiring same-day delivery (High Risk, 101) or gift wrapping (102) are packed at the mono manual packing work center. For multi jobs that require same-day delivery (106) or gift wrapping (107) sorting is performed at the multi manual sorting work center.

In terms of the various classifications of machine environments the process environment presented in Figure 7 can not be considered a 'pure' job or flow shop. In a pure flow shop, all jobs have the same order of processing (routing). In a pure job shop, the routing is different per job or job type. The process environment does not match the classification of a flow shop as jobs do not all have the same route. It better qualifies for a job shop environment where each job type has a pre-determined route (or flow) and some job types have the same routing. Still, two main flows exist:

1. The ***mono flow***: a two-stage flow where jobs are picked and subsequently packed. Each job type has a specific routing where three routes can be distinguished (to each of the packing work centers).
2. The ***multi flow***: a three-stage flow where jobs are picked, batched into pools in the stingray and subsequently sorted. Each job that is picked is stored in the stingray where it waits until the other pick batches that together form a pool are picked. Once a pool is complete (i.e., all pick batches from the pool arrived in the stingray), the stingray releases the pool to one of the two designated sorting work centers if the work center has a free server or has space available in the queue. Hence, the stingray process is analogous to a batching process.

Throughout the day, workload imbalances are observed: idle time at work center i while work center j has work for the upcoming x hours in the queue. To cope with these imbalances, operators are shifted back and forth from work center to work center several times a day. Also, in an attempt to speed up

pool completion sorting team leaders regularly request the picking department to prioritize multi jobs that would complete these pools.

3.2 Current job sequencing logic

This section describes the logic that decides which job is selected from the queue of jobs at the picking work center when requested by a picking operator. Once created, jobs are queued for picking. The current logic the WMS uses for sequencing jobs (pick batches) to the picking process is the following:

1. An order picker requests for a job.
2. If there are unscheduled jobs due within one hour proceed to *step 4*. If not, proceed to *step 3*.
3. Allocate the oldest job, regardless of cut-off times of other jobs. Go to *step 5*.
4. Allocate the job which is due first (i.e., has the lowest cut-off time) and if there are more jobs with the same (lowest) cut-off, allocate the oldest job.
5. Terminate.

Note: a pick batch (job) generally contains multiple orders or parts of orders each having its own a cut-off time at which it must be packed. The job cut-off time is the minimum of the cut-off times of the orders within the pick batch.

The sequencing logic described above is in fact a dispatching rule largely based on the Earliest Due Date rule and the FIFO discipline. It is simple, easy to implement and it prioritizes jobs that are due within a short timeframe. However, it fails to take workload balancing between the work centers into account. Also, it uses a single criterion (cut-off time) and fails to consider job characteristics such as the expected processing time of the job. Moreover, together with the order of job creation within the WMS, it results in workload fluctuations at packing and sorting work centers.

3.3 Throughput times

It is expected that a more balanced workload will reduce the throughput time of jobs. Thus, this section provides insights in the current throughput times of the different job types. A distinction is made between mono and multi jobs as these require different operations. Mono jobs represent pick batches whereas a multi job represents a pool consisting of multiple pick batches. Data is obtained for jobs created from October 6th, 2019 until December 28th, 2019.

For mono jobs, throughput time is defined as the time elapsed between job allocation to the order picker and the moment at which the last item within the job was packed. This time includes picking, transportation and packing. Figure 8 shows the distribution of throughput times per mono job type. Note that the total number of observations per throughput time is cumulative. For example, for throughput time 40 minutes, there is a total of 4100 observations, which is the sum of the jobs for all job types.

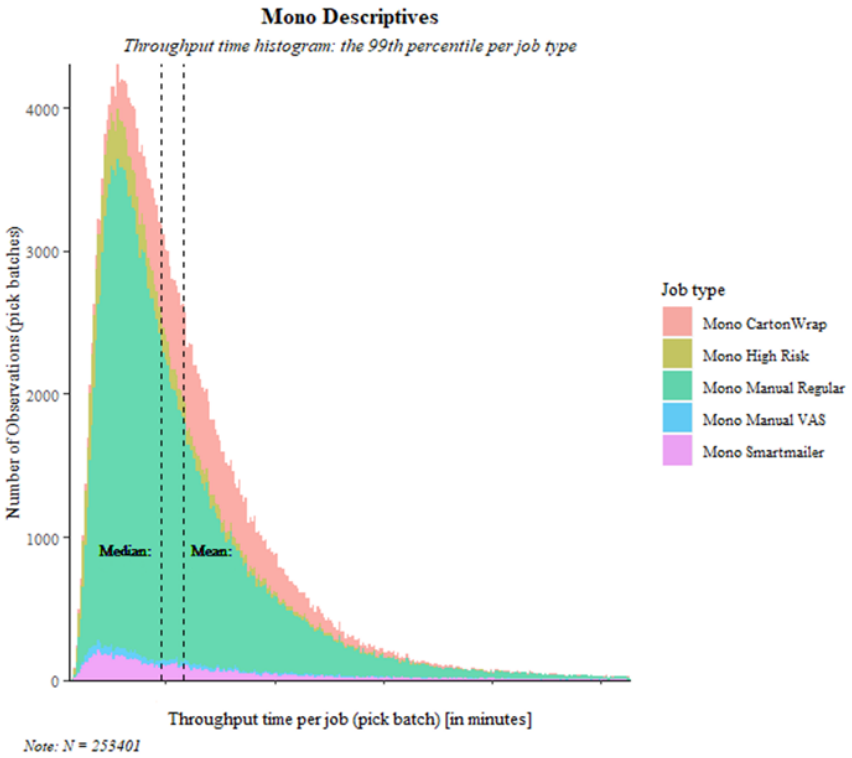


Figure 8: Throughput time histogram: the 99th percentile per mono job type.

It can be observed that there is a right-skewed distribution of throughput times for mono jobs and clear differences exist between the job types. First, the number of observations differs substantially. Mono Manual Regular comprises the lion's share of all mono jobs. This follows from the fact that the number of jobs that requires same-day delivery or gift wrapping is small compared to regular jobs. Also, the

mono manual packing work center has no restriction on item size or item fragileness; hence it can process all items stored in the warehouse.

For multi, multiple pick batches form a pool (on average, a pool contains 5.0 pick batches). Hence, throughput times are analyzed on the pool-level. As mentioned in section 2.3, packing for multi jobs is out of scope. Throughput time is defined as the time elapsed between the allocation of the first job of a pool to an order picker and the moment at which the last item of the pool was sorted. This time includes picking, transportation and sorting. In addition, this time includes the time it takes to pick all jobs within a pool which is referred to as pool completion time. Figure 9 shows the throughput time histogram for multi pools per job type.

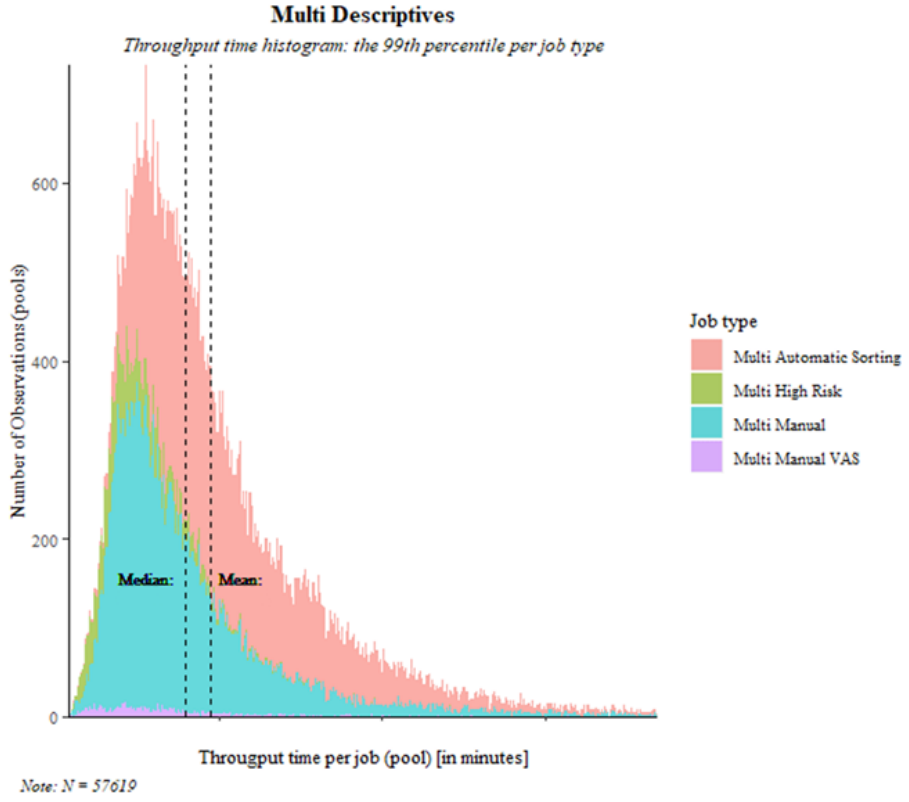


Figure 9: Throughput time histogram: the 99th percentile per multi job type.

Although the shape is similar to the throughput time for mono jobs, the mean and median throughput times for multi pools are substantially larger. This is largely explained by the pool completion time which is included in the throughput time for multi. Most pools are processed on the automatic sorting work center whereas relatively few pools require gift wrapping or same-day delivery.

3.4 Processing times

This section explores the processing times for the various operations performed in the work centers. This is for the purpose of providing the reader with an indication of these times and how these are composed. A more detailed analysis is provided when turning to the case study in section 5.

3.4.1 Picking time

Picking is the process of manually retrieving items from storage locations, collecting these in a tote and subsequently putting the tote on a conveyor belt that transports the tote to the outbound area. The process is initiated by the allocation of a job (pick batch) to an operator as a result of the sequencing logic and terminates when the operator places the tote on the conveyor belt. The picking operation is the same for each job type. Then, picking time of job i is defined in $Equation$ (1).

$$PickingTime_i = ToteOnConveyorTimestamp_i - JobAllocationTimestamp_i \quad (1)$$

This time includes the setup time an operator needs to collect an empty tote, putting it on a cart, walking the cart to the first pick location and traversing the aisles (fixed component). Figure 10 presents a scatterplot of item quantity per job and picking time as well as the density function of both variables.

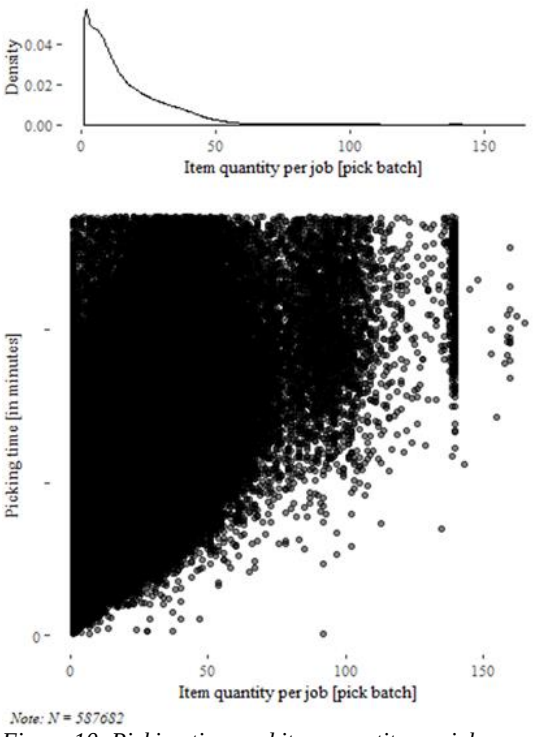


Figure 10: Picking time and item quantity per job.

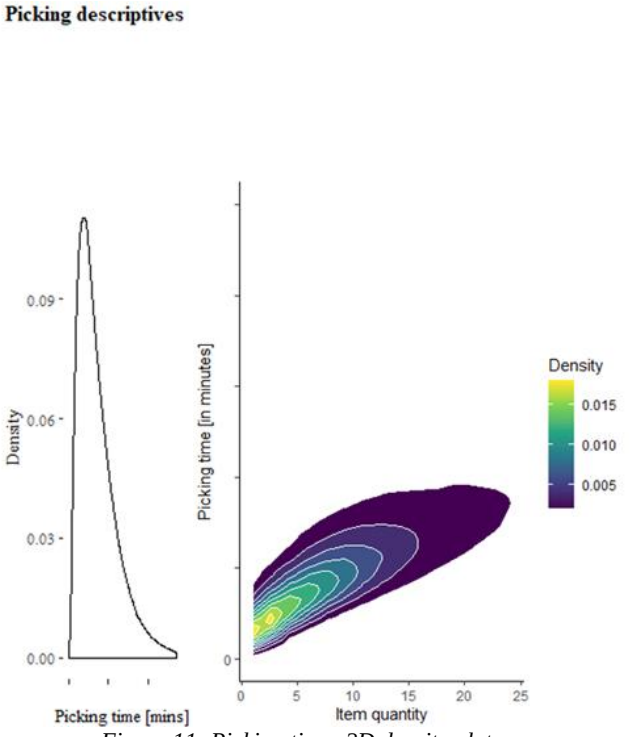


Figure 11: Picking time: 2D density plot.

As a result of over plotting, it seems that there is a substantially large dispersion among picking time for certain item quantities. However, the 2D density plot in Figure 11 better displays the relation between item quantity and picking time (the more yellow, the denser the number of observations).

In general, it can be observed that picking time increases as the number of items in the job increases. Additionally, Figure 10 shows a group of jobs clustered around item quantity 140. This cluster is explained by the jobs directed to the Smartmailer packing work center. The jobs processed on this machine solely contain small items (e.g., books and DVDs). Because of the smaller size, many items fit in one tote. These jobs generally contain many items per item type compared to other jobs. As a result, a picking operator can retrieve many items upon visiting a storage location. For the Smartmailer jobs, size is restricted to 140 items per job which explains the cluster around 140.

Table 3 presents the mean, median, standard deviation and number of observations for picking time both Smartmailer and regular (non-Smartmailer) picking jobs.

Table 3: Processing time descriptives: picking time.

Regular jobs	**Smartmailer jobs**
Mean: 5.9	Mean: 7.3
Median: 4.8	Median: 5.1
SD: 4.5	SD: 6.7
N = 581708	N = 10663

The above descriptives indicate that indeed differences exist in picking time when distinguishing these picking job types. Given this, it is worthwhile to distinguish regular picking jobs and Smartmailer picking jobs in further analyses.

3.4.2 Packing time

Once mono jobs are picked, they are transported to one of the three packing work centers: Mono Manual Packing, Mono Smartmailer, or Mono Cartonwrap. Within the Mono Manual Packing work center, two types of packing can be distinguished: regular packing and VAS packing. Thus, from a packing perspective, four different job types are distinguished: 1) Manual regular Packing; 2) Manual VAS

packing; 3) Smartmailer; and 4) Cartonwrap. Packing time is the time difference between the moment of packing completion of the first and the last item. Then, packing time of job i is defined as:

$$PackingTime_i = LastItemPackedTimestamp_i - \ FirstItemPackedTimestamp_i \quad (2)$$

Table 4 presents the mean, median, standard deviation and number of observations for packing time for the four packing job types.

Table 4: Processing time descriptives: packing time.

Manual regular	**Manual VAS**	**Smartmailer**	**Cartonwrap**
Mean: 8.3	Mean: 10.2	Mean: 3.8	Mean: 3.8
Median: 5.5	Median: 3.0	Median: 1.4	Median: 2.8
SD: 9.3	SD: 16.6	SD: 5.3	SD: 4.4
N = 195170	N = 3680	N = 10663	N = 43296

3.4.3 Sorting time

Once all multi jobs from a pool are picked, the stingray releases the pool to either the manual or the automatic sorting work center. Sorting time of job (pool) i is defined as in $Equation$ (3).

$$SortingTime_i = LastItemOfPoolimestamp_i - \ FirstItemOfPoolimestamp_i \quad (3)$$

Table 5 presents the mean, median, standard deviation and number of observations for sorting time for the two sorting work centers.

Table 5: Processing time descriptives: sorting time.

Manual sorting	**Automatic sorting**
Mean: 12.5	Mean: 14.2
Median: 9.1	Median: 11.8
SD: 12.2	SD: 9.4
N = 28034	N = 30237

4. Conceptual model

This section presents the model for sequencing unscheduled jobs to the various picking regions. It prioritizes the job from the region's queue that is the 'best job' to allocate. Figure 12 provides an example of the queues of jobs at the eight picking regions. Within each region, multiple operators work in parallel each working on one job at a time where each job has a specific size (number of items) and may contain multiple (or parts of) customer orders. A job's picking time is mainly determined by the job size. Once a mono job is picked, it moves to its designated outbound packing work center. Multi jobs are part of a pool which can only move to its outbound sorting work center once the entire pool is picked. Multi jobs are transported to the stingray which batches pools and releases these pools when a pool is completely picked (section 3.1). Below, each letter represents a pool of jobs.

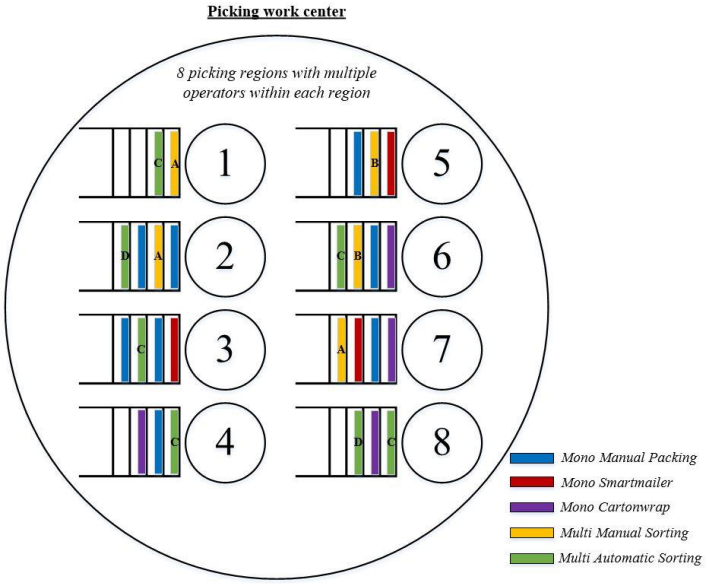


Figure 12:Example of unscheduled jobs at the picking regions.

The process environment is dynamic in which customer orders (hence jobs) arrive throughout the day; the queue of jobs is continuously renewed and never complete. Hence, finding the optimal solution to these problems is not possible. The model is a dispatching rule (an approximate method) that considers three main requirements: 1) cut-off time; 2) steady workflow and 3) pool completion. Section 4.1 states these requirements and defines several metrics and section 4.2 formulates the model.

4.1 Requirements and metrics

The three main requirements of the model are cut-off time, steady workflow and pool completion. Once an operator requests for a new job in a specific picking region, the model evaluates several conditions and performs various calculations for each unscheduled job in the region's queue.

4.1.1 Cut-off time

First, it is assessed whether there are jobs that require immediate allocation. Finishing jobs within the cut-off time is crucial as achieving 100 percent order fulfilment is what Ingram Micro CLS strives for. Certainly, when near cut-off time, jobs that still can be completed in time should be prioritized. The metric that is used to decide when a job must be prioritized based on cut-off time is referred to as the safety margin. The logic behind this is largely analogues to minimum-slack-time (MST) dispatching as discussed by Baker (1984). The safety margin of the job is the time difference between the earliest moment of job completion and the cut-off time of the job:

$$SafetyMargin_i = TTC_i - ECT_i \tag{4}$$

The time to cut-off of a job is defined as the difference between the current time and the cut-off time:

$$TTC_i = CutOffTime_i - t \tag{5}$$

The earliest completion time job i (directed to work center j) is defined as:

$$ECT_i = E_{PickingTime}[Job_i] + TransportationTime_{i,j} + E_{QueuingTime}[Job_i] + E_{PackingTime}[Job_i] \tag{6}$$

This time includes the expected picking time, transportation time, expected queueing time at the outbound work center, and processing time at the outbound work center. Expected queuing time is defined as the sum of the expected packing times for all jobs in the queue of the designated work center divided by the number of active workstations (m). For multi jobs, expected sorting time instead of packing time is used.

Although processing times are estimated, they remain uncertain. A configurable safety parameter is incorporated which can be used to cope with the uncertainty in processing times. When $SafetyMargin$ lies below this safety parameter, the job is prioritized. When multiple jobs lie below the

threshold, the largest job is prioritized such that the likelihood of satisfying as much customers as possible is higher. If multiple jobs are 'largest', the oldest job is prioritized. More on the value of the configurable safety parameter is discussed when turning to the case study. The job cut-off time is the minimum of the cut-off times of the orders within the pick batch. For pools this time is the minimum of the cut-off times of the pick batches that form the pool.

4.1.2 Steady workflow

Second, if none of the jobs require immediate allocation for them to be finished in time, it is evaluated which job could best be allocated to the picking operator to ensure a steady workflow towards the packing and sorting work centers as much as possible. As a steady workflow reduces workload peaks, the likelihood of blocking (i.e., totes looping on the conveyor belt, waiting for a work center to become available) is reduced as well. A steady workflow is enhanced when jobs are sequenced to picking while considering to which outbound work center each job in the queue is routed to. In the simplest form this implies that priority is given to the job that is routed to the work center with the smallest queue. Then, jobs routed to busy work centers are postponed while jobs routed to work centers that will run dry soon are prioritized. This logic is referred to as the 'Work in Next Queue' logic.

The metric used for assessing the queue of outbound work center j is $WINQ_j$. The work in next queue at work center j ($WINQ_j$) is defined as the sum of the expected processing times of the jobs (when processed at work center j) in the queue of work center j divided by the number of active work stations (m) within work center j. As processing times differ per work center, 'Work' is expressed in time units rather than number of jobs. However, defining $WINQ_j$ is not as straightforward as in the single-machine situation since multiple picking regions exist, each with a multiple parallel operator. Ideally, we would prioritize the job that, upon leaving the picking department, goes to work center with the lowest $WINQ_j$. The workloads in the work centers as estimated at time t are most likely different from the workloads in the queues as observed at time $t + PickingTime[Job_i]$ as a result of new allocations made during $(t,\ t + PickingTime[Job_i])$. However, the dynamic nature of the problem

makes it impossible to know in advance what new allocations will be made during $(t,\ t + E_{PickingTime}[Job_i])$. Therefore, we make the following crucial assumption:

Assumption 1: *When calculating $WINQ_j$ at time t, it is assumed that no new allocations are made during $(t,\ t + E_{PickingTime}[Job_i])$.*

In addition, for calculating $WINQ_j$ at time t, an assumption must be made on what part of the jobs that had already been allocated to picking at time t, is finished at time $t + E_{PickingTime}[Job_i]$. One can assume that once a job is picked **none**, **some** or **all** of the previously allocated jobs are picked.

For the purpose of calculating $WINQ_j$ at time t, including **none** of the jobs that were previously dispatched, but are still being picked, is wrong. If these jobs were to be completely ignored, we might prioritize a job directed to a seemingly 'empty' work center whereas there are many jobs currently being picked for that work center.

Alternatively, if we assume that **some** of the jobs that were previously dispatched are picked when the job (that is considered in the dispatching decision) leaves the picking department, we would face another problem. $WINQ_j$ in this case would have to be modeled as a function of $E_{PickingTime}[Job_i]$. That would, together with the no new allocations *assumption 1*, imply that the job with the largest $E_{PickingTime}[Job_i]$ would always be favored over the other jobs (more jobs finished while no new allocations were made).

Including **all** jobs that are being picked in the $WINQ$ calculation at time t allows us to take into account the results of previous dispatching decisions: the jobs being picked at time t are a result of previous decisions aimed at workload balancing. Hence, taking these jobs into account in calculating $WINQ$ would give us a better representation of the workloads in transfer to the work centers. However, a distinction must be made between mono (packing) and multi (sorting) work centers. Mono work centers process jobs (pick batches) one by one; hence for mono work centers we make the following assumption:

Assumption 2a: *For mono work centers, all jobs directed to the work center that are still being picked are included as workload when calculating $WINQ_j$ at time t.*

For multi work centers, the above should not be assumed as sorting work centers do not process jobs one by one. Rather, these work centers process pools: a set of specific jobs that can only processed once all of them have been picked. For example, suppose a situation in which for a pool consisting of five jobs, four jobs are being (or have been) picked. Including these four jobs increases $WINQ_j$ whereas in fact there is no actual workload in the queue of that sorting work center as the pool is not completely picked. In terms of $WINQ_j$, we could falsely decide not to prioritize a job for that sorting work center. Hence, the following assumption is made for multi work centers:

Assumption 2b: *For multi work centers, the jobs that are being (or have been) picked are only included if all jobs from the pool were already dispatched at time t when calculating $WINQ_j$.*

4.1.3 Pool completion

In achieving a steady workflow, the model must recognize that multi jobs are processed in batches (pools) at the sorting work centers. A pool can only move to a sorting work center once all pick batches within that pool are picked. Thus, stimulating timely pool completion is an important requirement of the model. A dispatching rule similar to the NUB-rule (Number of Uncompleted Branches) might suit the purpose of stimulating pool completion. Within the case study context, this rule would prioritize jobs from pools that have a low number of unfinished jobs. Once more jobs are completed (i.e., more pick batches from the pool have been picked), and hence the completion time increases, the remaining jobs are assigned a higher priority. If pool size were constant, the NUB-rule could be applied for speeding up pool completion. However, as pool sizes may differ, this rule would favor small pools over large pools. The downside of this method is that in practice this would result in large pools being processed at the end of the day.

Therefore, we must modify this rule to achieve the desired effect of early pool completion where pool sizes differ. The following modification of this rule is proposed: the Fraction of Unscheduled Branches (FUB) rule. When expressed as a fraction (or percentage), small pools are no longer favored over large pools, but the pools that are almost completely allocated (or picked) are favored over pools with a low fraction of jobs allocated. Then, for job i, FUB_i is defined as in $Equation\ (7)$.

$$FUB_i = \frac{PoolSize_i - JobsFromPoolAllocated_i}{PoolSize_i} \quad (7)$$

where, $PoolSize_i$ is the number of jobs (pick batches) that together form a pool and $JobsFromPoolAllocated_i$ is the number of jobs from that pool that had already been allocated to picking.

4.2 Model formulation

This section formulates the model. When allocating a job to an order picker the model mainly decides on the following three metrics: $SafetyMargin_i$, $WINQ_j$, and FUB_i. The following procedure summarizes how the model prioritizes a job from the region's queue when requested by an operator. The model is graphically presented in Figure 13. Note that at first all unscheduled jobs from the region are considered but that, depending on the metrics, different subsets may be considered.

When requested at time t, evaluate the following for the unscheduled jobs in picking region x:

1. Are there jobs that require immediate allocation ($SafetyMargin < Parameter$). *If yes, go to step 2. Else, go to step 4.*
2. Does only one job require immediate allocation? *If yes, allocate it. Else, go to step 3*
3. Is the largest job size a unique value? (number of items in the pick batch)? *If yes, allocate the largest job. Else, allocate the oldest job from the 'largest' jobs.*
4. Is the lowest $WINQ$ a unique value? *If yes, go to step 6. Else, go to step 5.*
5. Does at least one sorting (multi) work center correspond to this value? *If yes, go to step 7. Else, go to step 8.*
6. Is the outbound work center that has the lowest $WINQ$ a sorting (multi) work center? *If yes, go to step 7. Else, go to step 8.*
7. Is the lowest FUB a unique value? *If yes, allocate the job with the lowest FUB. Else, go to step 8.*
8. Is the lowest $SafetyMargin$ a unique value? *If yes, allocate the job with the lowest $SafetyMargin$. If no, go to step 9.*
9. Is the lowest $E_{PickingTime}[Job_i]$ a unique value? *If yes, allocate the job with the lowest $E_{PickingTime}[Job_i]$. Else, allocate the oldest job.*
10. Terminate.

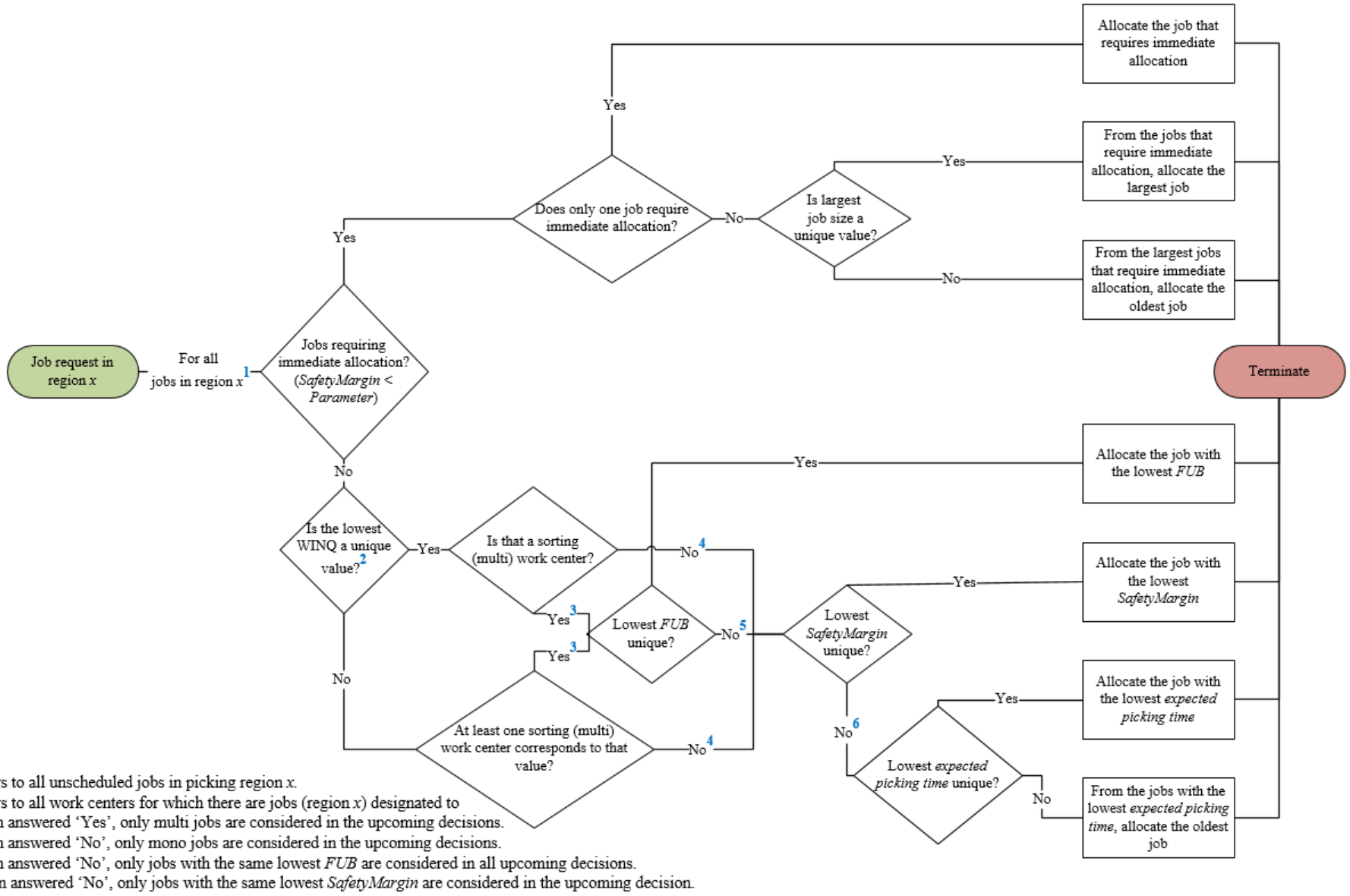
Step 1 until *3* assess whether immediate allocation is required and when multiple jobs require immediate allocation the largest job is allocated such that the likelihood of satisfying the most customers is higher.

If no immediate allocation is required, the model turns to workload balancing by assessing

$WINQ_j$ in *step 4*. If two work centers have the same $WINQ_j$ different decision can be made. When both work centers are mono work centers, one is indifferent for which of these work centers a job is allocated. However, when one of these work centers is a sorting (multi) and the other is a packing (mono) work center, only the multi jobs are considered for allocation as this increases the likelihood of completing a pool as a result of the current allocation. When the lowest $WINQ_j$ is a sorting work center, FUB_i is evaluated to speed up pool completion.

It must be prevented as much as possible that other jobs eventually require immediate scheduling as a result of the current allocation decision. Thus, *Step 8* ensures that the job with lowest $SafetyMargin_i$ is prioritized. When indifferent in $SafetyMargin_i$, the model prioritizes the job with the lowest expected picking time such that a new job requests are made sooner. The latter is analogous to the shortest-processing-time (SPT) rule. When indifferent in picking time, simply the oldest job is allocated.

Summarizing, the model first evaluates whether jobs require immediate allocation to prevent them being late due to unforeseen circumstances and to deal with the uncertainty of processing times. Then, for the jobs that do not require immediate allocation, the model evaluates which job it should allocate to an order picker to ensure that workload is balanced as much as possible.



[1]: Refers to all unscheduled jobs in picking region *x*.
[2]: Refers to all work centers for which there are jobs (region *x*) designated to
[3]: When answered 'Yes', only multi jobs are considered in the upcoming decisions.
[4]: When answered 'No', only mono jobs are considered in the upcoming decisions.
[5]: When answered 'No', only jobs with the same lowest *FUB* are considered in all upcoming decisions.
[6]: When answered 'No', only jobs with the same lowest *SafetyMargin* are considered in the upcoming decision.

Figure 13: Graphical representation of the conceptual model.

5. Case study

Ingram Micro CLS serves as the case study company for which it is analyzed using simulation whether the proposed dynamic job sequencing logic, as opposed to the current job sequencing logic, can reduce throughput times, pool completion time, and imbalances in workload among the work centers.

Section 5.1 addresses the assumptions made regarding the case study. Section 5.2 describes the simulation procedure and validates the simulation. Section 5.3 reports on the input for the case study. Section 5.4 presents the primary case study results of several scenarios. Section 5.5 provides a sensitivity analysis and section 5.6 summarizes the results. Readers only interested in the practical relevance of the results are directly referred to section 5.7 as this is a stand-alone section providing the key insights obtained from the case study results.

5.1 Assumptions

In Appendix A, all assumptions are included and for some it is discussed in more detail how the results may be affected by the assumptions. The most important assumptions are summarized below.

- Once a picking operator finishes a job (that is, she puts the tote on the conveyor belt), a new job is immediately allocated to the operator;
- For work centers that are inactive in the first shift but activated in the second shift of the day (buffer-requiring work centers), it is desired in the company that one hour of workload is present upon activating the work centers;
- Jobs arrive at the pick regions under the round-robin policy;
- Picking operators are evenly distributed over the eight picking regions and do not switch regions;
- Each outbound work center has a single queue that operates under FCFS;
- Mono work center queues and the stingray have infinite capacity; and
- Transportation times (from pick region to work center) are assumed to be deterministic but different for each region-work center combination.

5.2 Simulation procedure and validation

This section describes the simulation procedure and validation. A detailed description (including motivation) is included in Appendix B. Several processes and characteristics of the case study company lead to the decision of using discrete event simulation based on exact data (i.e., true historical processing times, true job arrival data) to test the model. A base case simulation was built using four weeks of data from Monday, October 7, 2019 until Sunday, November 3, 2019. Figure 14 describes the number of jobs processed per day and indicates the proportion of multi and mono jobs.

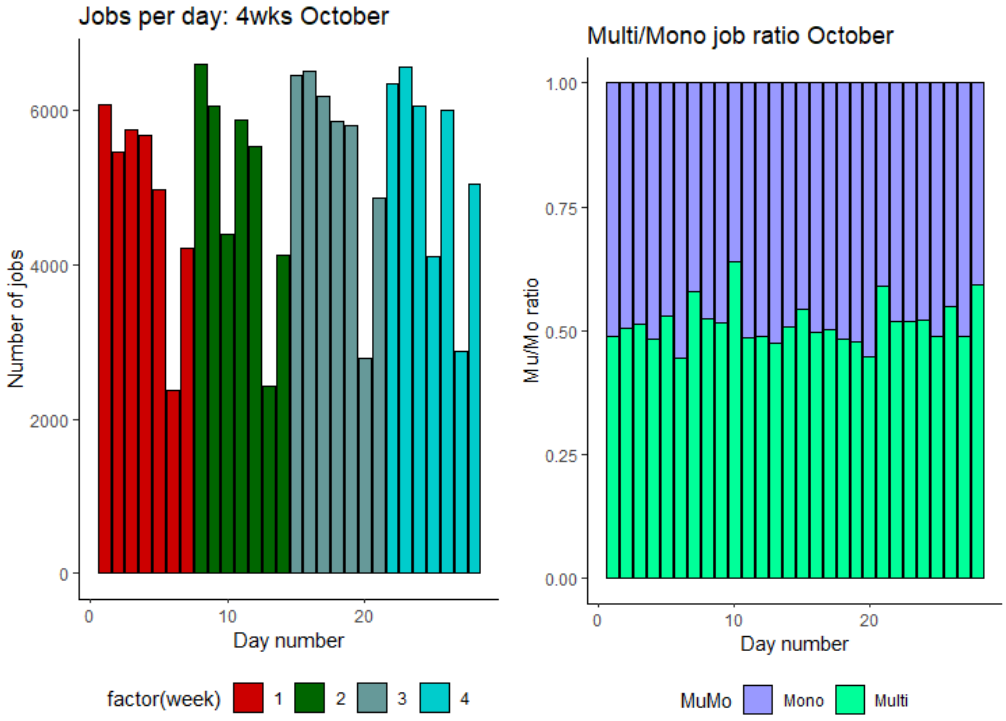


Figure 14: October job descriptives: jobs per day and multi mono ratio.

A clear week pattern can be observed where there is a slight increase in the number of jobs from week 1 to week 4. Additionally, a clear day pattern can be observed within the weeks where Mondays and Tuesday are the busiest and on Saturday and Sunday relatively less jobs are processed. The average multi mono ratio is 0.52; on average there are as many mono jobs as there are multi jobs.

Although case study data on the capacities was used, two adjustments were made. First, no data was available on the exact number of pick operators per region. Thus, in the simulation pick operators are evenly distributed among the pick regions and then the number is adjusted such that the highest utilization of the eight regions does not exceed 90% in the simulation. In practice operators may move

from one region to another, but in principle they remain in one region. In the simulation however, they remain only in one region which may result in unrealistically high simulated utilizations in some regions which is correct by the above. Secondly, the same as the above holds for the outbound work centers; operators may move from work center to work center when imbalances are observed. This data was also not available and hence the number of operators or machines was adjusted such that 99% of the jobs is finished in time as this is achieved in practice.

When validating the simulation, flaws were detected in the calculations of the work centers' utilization of the simulation software (simmer package in R). After reporting these bugs to simmer's authors, they were fixed. Table 6 describes the mean, standard deviation and coefficient of variation for throughput times of the 90th percentile actual data versus the simulation data (base case).

Table 6: Throughput times: actual versus base case reduced sample.

Throughput time (in mins)	**Actual (90th percentile)**			**Base case**		
	Mean	**SD**	C_v	**Mean**	**SD**	C_v
Mono jobs	53.6	26.7	0.50	46.2	30.6	0.66
Multi pools	88.0	43.0	0.49	69.6	40.9	0.59

Lower mean throughput times can be observed which is explained by some of the assumptions. However, Table 6 shows that the standard deviation of the actual throughput times and the simulated throughput times are about the same. This is a promising result as although the simulation on average finds lower throughput times, the variation among these throughput times is similar to the actual throughput times.

Summarizing, the simulation model is found to be valid, but one must inevitably be careful when interpreting the results. Therefore, the results of the simulation are always interpreted relative to the base case simulation.

5.3 Input

The first input parameter for the model is the configurable safety parameter to which the $SafetyMargin$ of a job is compared. Recall that the safety margin of a job is the time difference between its earliest expected moment of completion if it were sequenced now and its cut-off time. The lower the safety margin, the more likely it becomes that a job is not finished before the cut-off time.

Stakeholders in the company stated that it was desired that some adjustable parameter was present in the proposed logic to be accepted by the operators in the control room as it would allow for adjusting the proposed logic such that one would more have the feeling of being in control. Conversations with company operations stakeholders made clear that it is desired that the configurable safety parameter is set to 15 minutes. This builds in some measure of safety to stimulate the timely finishing of jobs. Sensitivity regarding the configurable safety parameter will be assessed further on in this research.

Another important input parameter for the model to be tested in the case study is the expected processing time of jobs. Ideally, historical data is used to estimate these processing times as this captures true historical behavior of the processes. For example, average processing times (obtained from historical data) or regression analysis can be used. Either way, it is important that it is taken into account that different work centers (and job types) should be distinguished.

In Appendix C and D, processing times are analyzed and estimated in detail. First, historical averages per item quantity of the job are used (Appendix C). Then it is argued that fitting a curve using OLS regression to these averages might be interesting to use as input for the model as well (Appendix D). Transportation time is estimated in Appendix E.

5.4 Results

This section first describes which results are of interest and section 5.4.1 presents the primary results.

In comparing the dynamic job sequencing logic presented in this thesis and the current job sequencing logic (the base case), we are interested in three general types of results:

- **The throughput time of jobs:** the main metric of interest. It is expected that throughput time is reduced as a result of waiting time reduction by using the proposed sequencing logic compared to the current sequencing logic.
- **The time it takes to completely pick a multi pool:** the dynamic logic should in general speed up pool completion such that pools can be processed at the sorting work centers.
- **The variation in utilization among the outbound work centers:** the logic is designed with the purpose to reduce throughput time through workload balancing. The variation in

utilization among the work centers provides an indication whether workload is indeed balanced across the work centers.

Throughput times of jobs is measured for mono jobs and multi pools separately (see also section 3.3). For mono jobs, throughput time is defined as the time elapsed between job allocation to the order picker and the moment at which the last item within the job was packed. This time includes picking, transportation and packing. For multi pools, throughput time is defined as the time elapsed between the allocation of the first job of a pool to an order picker and the moment at which the last item of the pool was sorted. This time includes picking, transportation and sorting. In addition, this time includes the time it takes to pick all jobs within a pool which is referred to as pool completion time (PCT). Picking pool completion time (PCT) of pool i is measured as:

$$PCT_i = LastToteOnConveyorTimestamp_i - \ FirstJobAllocationTimestamp_i \quad (8)$$

which is the time difference between the end of picking for the last job of the pool and the start of picking of the first job of the pool.

Variation in utilization among the work centers is measured by squared errors rather than absolute errors such that larger errors are weighted more. Firstly, variation is measured per hour as within the case study company, one is steering the process based on hourly output. Then, the control room acknowledges what hourly output each work center can process and based on that it releases the jobs to the picking process. Secondly, variation is aggregated to variation per shift as the number of operators and machines is planned per shift and the person responsible for job releasement differs per shift. First, variation in utilization among the work centers per hour is measured by the mean squared error:

$$MSE_h = \frac{1}{n}\sum_{j=1}^{n}\left[U_{j,h} - \ \bar{U}_h\right]^2 \quad (9)$$

where MSE_h is the mean squared error for hour h, $U_{j,h}$ is the utilization of work center j for hour h and $\bar{U}_h$ is the mean utilization of all work centers for hour h. Then, the MSE per shift is found by summing the MSE for all hours of the shift:

$$MSE_s = \sum_{h=1}^{\infty} MSE_h \qquad \forall h, h \in S \qquad (10)$$

The MSE as a measure of variability in utilizations indicates how much workload fluctuates among the work centers. The larger the MSE, the larger the workload fluctuations. When lower values would be observed when the dynamic sequencing logic is applied as opposed to the current logic, the dynamic logic would prove its capability in reducing workload imbalances. Lower values of MSE_s indicate lower workload fluctuations among the work centers throughout the shift; that is, utilizations of the work centers are more aligned.

To test whether the proposed dynamic job sequencing logic outperforms the current job sequencing logic, the base case simulation is adjusted by using the proposed sequencing logic instead of the current sequencing logic. This alternative case is then compared to the base case simulation based on the result metrics presented above.

Three scenarios (discussed on the next page) will be tested for which it is expected that regardless of the scenario throughput time and waiting time, pool completion time and variation in utilization among the work centers is reduced. More specifically, three hypotheses are constructed regarding the three-result metrics. First, it is expected that throughput time and waiting time are lower in the alternative case compared to the base case. Hence the first hypothesis is Hypothesis 1:

- ***Hypothesis 1:*** mean throughput and waiting time are lower in the alternative case versus the base case.

Second, it is expected that pool completion is sped up as a result of the proposed sequencing logic:

- ***Hypothesis 2:*** mean pool completion time is lower in the alternative case versus the base case.

Third, it is expected that utilization variation among the work centers per shift is lower in the alternative case simulation compared to the base case simulation. Thus, the third hypothesis is:

- ***Hypothesis 3:*** variation among the utilizations of the outbound work centers is lower in the alternative case compared to the base case.

Fourth, it is expected that larger throughput time reductions are obtained through better workload balancing:

- ***Hypothesis 4:*** larger throughput and waiting time reductions are obtained through better workload balancing.

Several scenarios are explored where the alternative case is compared to the base case when adjusting simulation parameters (see Table 7). To allow for a fair comparison of the results, the base case and alternative case are similar in terms of capacity per scenario.

Table 7: Simulation scenarios.

Scenario	Period	Processing time used in simulation	Processing time used for job sequencing	Base case	Alternative case
1	4 wks. Oct	Historical	Historical	*Current job sequencing logic*	*Proposed dynamic job sequencing logic*
2a	4 wks. Oct	Historical	***Historical averages (Appendix C)***		
2b	4 wks. Oct	Historical	***Estimated using OLS regression (Appendix D)***		

The dynamic job sequencing logic as proposed in this thesis uses the *expected* processing time of jobs in deciding which job to sequence. In scenario 1, expected processing times are set equal to historical processing times and known upon arrival of the job. Scenario 2a and 2b are similar to scenario 1 but instead of using the true historical processing times, either historical averages or estimates of these averages obtained using OLS regression are used to estimate processing times.

Comparing the scenarios, it is expected that when the true historical processing times are known upon making the sequencing decision (scenario 1), the best results are obtained as one can then best decide on what job to sequence given the true workload present at the outbound work stations. Historical averages analyzed in Appendix C were observed to fluctuate for large item quantities; hence it was decided to smooth these estimates by fitting a curve using OLS regression. Thus, it is expected that results of scenario 1 are followed by scenario 2b and subsequently scenario 2a:

- ***Hypothesis 5:*** results for scenario 1 are better than results for scenario 2a and 2b, where 2b is expected to outperform scenario 2a.

The scenarios use four weeks of October 2019 data; a month characterized by a lower demand compared to the high season months (November and December). Sensitivity of the results to the period, estimation

method and the configurable safety parameter is discussed in section 5.5. Section 5.6 summarizes the results and offsets the results against several hypotheses. Practical relevance and generalizability are commented on in section 5.7. Detailed results for the primary scenarios are included in Appendix F. Detailed results for all (sensitivity) scenarios along with a scenario description are in Appendix G. All (sensitivity) hypotheses are also included in Appendix H.

5.4.1 Primary results

Table 8 summarizes the results for scenario 1, 2a and 2b relative to the base case. Results for throughput, waiting, and pool completion time for all scenarios were found by t-tests to be significantly different from the base case results.

Table 8: Results summary: primary.

	Throughput Time [in mins]			**Waiting Time [in mins]**			**Pool Completion Time [in mins]**		**Utilization**	
	Mo	**Mu**	**Mo/Mu**	**Mo**	**Mu**	**Mo/Mu**	**Mu**		**Shift**	
	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\overline{\Delta MSE}$	%Shifts better
Base	46.2 (30.6)	40.1 (15.0)	-	21.7 (27.1)	8.3 (11.3)	-	13.6 (5.2)	-	-	-
1	43.7 (26.7)	36.9 (13.9)	5.4/8.0	19.2 (22.8)	7.1 (9.7)	11.5/14.5	12.6 (5.0)	7.4	186.3	56
2a	44.1 (27.6)	36.9 (13.8)	4.5/8.0	19.6 (23.9)	7.1 (9.9)	9.7/14.5	12.7 (5.3)	6.6	168.7	60
2b	43.7 (27.4)	36.7 (13.8)	5.4/8.5	19.3 (23.6)	6.9 (9.8)	11.1/16.9	12.5 (5.1)	8.1	135.6	63

Note. $\bar{X}$ (SD) refers the mean (standard deviation) values. * refers to the percentual reduction in $\bar{X}$ for the specified scenario compared to the base case. Mono jobs and multi pools are abbreviated to Mo and Mu respectively where Mu refers to the 75th percentile in terms of pool completion (or throughput) time for multi pools (see Appendix F for further details). Mo/Mu specifies that reductions for Mo and Mu are considered. %Shifts better specifies the percentage of shifts in which the specified scenario outperforms the base case in terms of MSE_s.

It can be observed that in all scenarios, waiting times (hence throughput times) are reduced compared to the base case. Reduction is larger for multi pools compared to mono jobs. Thus, supporting evidence is found for Hypothesis 1. Comparing the scenarios, waiting time reduction for mono jobs is largest in scenario 1 whereas for multi pools reduction is largest in scenario 2b. Additionally, scenario 2a in which historical averages are used yields lower waiting time reductions compared to the other scenarios. Thus, it suggests that perfectly knowing the processing times upon the sequencing decision does not necessarily imply larger throughput time reductions compared to when estimates are used. Hence,

Hypothesis 5 must be rejected when it comes to throughput and waiting time reduction.

Pool completion time is sped up in all scenarios compared to the base case which provides supporting evidence for Hypothesis 2. Reduction is largest in scenario 2b which suggests that knowing the true historical processing times does not mean that it outperforms in terms of PCT which again provides evidence that Hypothesis 5 must be rejected. To illustrate how PCT is affected by the proposed job sequencing tool, Figure 15 plots the PCT density graphs for the alternative and base case for scenario 1. The density function is shifted towards lower PCTs and more centered around the mean in the alternative case versus the base case; i.e., pool completion is sped up and more predictable.

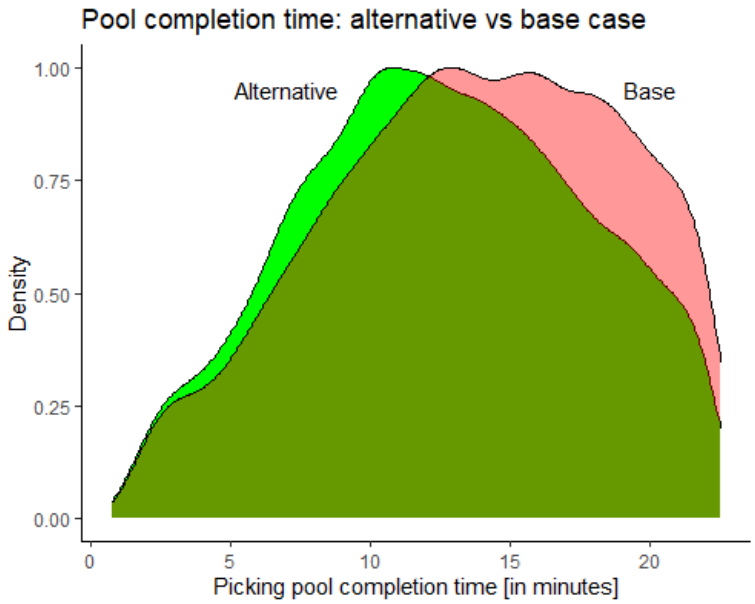


Figure 15: PCT Density graphs 75th percentile: Scenario 1.

Utilization variation among the outbound work centers is lower for all scenarios compared to the base case which indicates that the proposed sequencing logic is able to balance workload more than the current job sequencing logic; which is indicated by positive $\overline{\Delta MSE}$. The %Shifts better indicates that in more than half of the shifts, the proposed logic is better able to balance workload. This provides supporting evidence for Hypothesis 3. However, when comparing the alternative case to the base case per shift, it can be observed. Figure 16 plots the difference in MSE for the base case and the alternative case simulation for scenario 1. Green bars indicate that the MSE_s in that shift is lower when the proposed logic is used compared to the current logic.

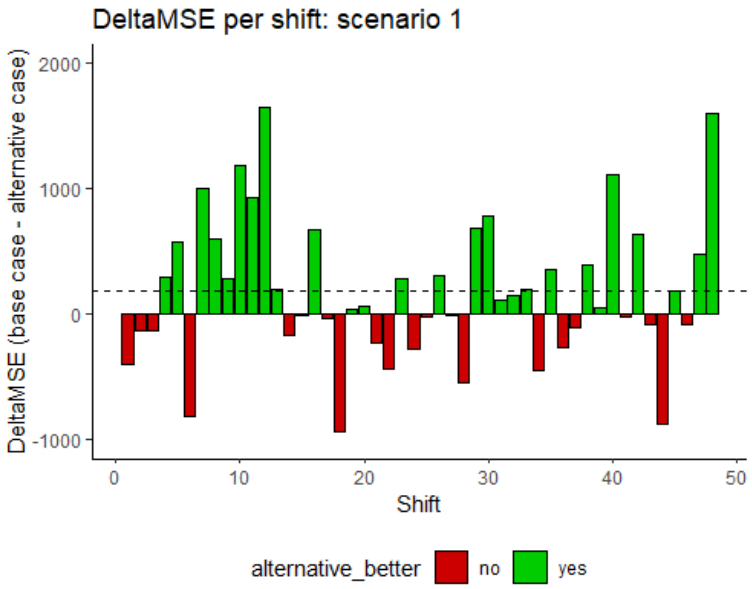


Figure 16: Delta MSE per shift: Scenario 1.

It can be observed that, although on average there is a positive effect, it differs per shift whether the alternative case outperforms the base case. Still, the alternative case outperforms the base case in 56 percent of the shifts, 57 percent of the hours, and 68 percent of the days for scenario 1. Similar results are obtained for scenario 2a and 2b. Thus, although supporting evidence is found for Hypothesis 3, results are not always better than the base case.

Comparing the scenarios in terms of $\overline{\Delta MSE}$ scenario 1 performs best and performance is relatively worst for scenario 2b. In practice, processing times are not perfectly known in advance; hence must be estimated when deciding on job sequencing. When comparing scenario 2a and 2b in terms of throughput time reduction and utilization variation it can be observed that lower $\overline{\Delta MSE}$ does not necessarily imply larger throughput time reductions whereas it was expected that lower variation would imply lower throughput times. Hence, Hypothesis 4 is partially supported by the results as although reductions are obtained through better workload balance, not necessarily *larger* reductions are obtained through better workload balance. The next section elaborates on the sensitivity of the results.

5.5 Sensitivity analysis

This section analyses the sensitivity of the results to 1) the estimation error of the expected processing times; 2) the simulation period; and 3) the configurable safety parameter. Several hypotheses are constructed which will be discussed further on in section 5.6. Detailed sensitivity results for 1) and 2) are presented in Appendix G. Note that all hypotheses are summarized in Appendix H.

5.5.1 Estimation error

The results presented in the previous section indicated that throughput time reduction varied per scenario where performance, although better than in the base case, was worst for scenario 2a in which historical averages were used for estimating processing times. Using regression equations (scenario 2b) on the other hand yielded better results in terms of waiting time reduction. To test whether the results are sensitive to estimation errors, scenario 1 is adjusted to sensitivity scenarios S1a and S1b in which the estimated processing times are set equal to 80% and 120% of the true historical processing times respectively. That is, in scenario S1a (S1b) there is a structural underestimation (overestimation) of the processing times of 20 percent.

It is expected that regardless of over- or underestimating, the proposed logic will outperform the current logic as workload of the outbound work centers is taken into account when sequencing jobs. This is expected for throughput time, waiting time, pool completion time and workload balancing. Hence the following hypothesis is constructed:

- ***Sensitivity hypothesis S1:*** regardless of over- or underestimating processing times, the alternative case outperforms the base case simulation in terms of throughput time, waiting time, pool completion time and workload balancing.

Whether either over- or underestimating yields the best results is less straightforward. For overestimating on the one hand, one could argue that overestimating results in worse performance compared to scenario 1 because of lower values for $SafetyMargin$ which results in the logic leaning more towards prioritizing jobs that are large in terms of item quantity. Then, the logic would more often ignore information regarding the workloads at the outbound work centers. For underestimating on the other hand, one could argue that workload information is taken into account more often when

dispatching jobs. However, the dynamics are complex as either over- or underestimating simultaneously affects $SafetyMargin$ as well as it affects $WINQ$ of the work centers.

Table 9 presents the result for scenario S1a, S1b, 1 and the base case. Results for throughput, waiting, and pool completion time for all scenarios were found by t-tests to be significantly different from the base case results.

Table 9: Sensitivity results: estimation error.

	Throughput Time [in mins]			**Waiting Time [in mins]**			**Pool Completion Time [in mins]**		**Utilization**	
	Mo	**Mu**	**Mo/Mu**	**Mo**	**Mu**	**Mo/Mu**	**Mu**		**Shift**	
	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\overline{\Delta MSE}$	%Shifts better
Base	46.2 (30.6)	40.1 (15.0)	-	21.7 (27.1)	8.3 (11.3)	-	13.6 (5.2)	-	-	-
1	43.7 (26.7)	36.9 (13.9)	5.4/8.0	19.2 (22.8)	7.1 (9.7)	11.5/14.5	12.6 (5.0)	7.4	186.3	56
S1a -20%	44.8 (27.1)	37.8 (14.6)	3.0/5.7	20.3 (23.4)	7.7 (10.6)	6.5/7.2	12.7 (5.2)	6.6	203.5	67
S1b + 20%	43.2 (27.1)	37.0 (13.9)	6.5/7.7	18.8 (23.2)	7.1 (9.8)	13.4/14.5	12.5 (4.9)	8.1	197.3	63

Note. $\bar{X}$ (SD) refers the mean (standard deviation) values. * refers to the percentual reduction in $\bar{X}$ for the specified scenario compared to the base case. Mono jobs and multi pools are abbreviated to Mo and Mu respectively where Mu refers to the 75th percentile in terms of pool completion (or throughput) time for multi pools (see Appendix F for further details). Mo/Mu specifies that reductions for Mo and Mu are considered. %Shifts better specifies the percentage of shifts in which the specified scenario outperforms the base case in terms of MSE_s.

It can be observed that sensitivity of the results to estimation errors varies per metric. Variation in utilization among the work centers is lower regardless of over- or underestimating the processing times which provides supporting evidence for Sensitivity Hypothesis S1. Results for pool completion time are insensitive to estimation error. Additionally, different throughput and waiting time effects are observed for mono jobs and multi pools. Results for mono jobs seem more sensitive to the estimator where overestimating performs best. For multi pools on the other hand, neither over- nor underestimating yields better results: best results are obtained in scenario 1 where the exact processing times are known in advance.

Most importantly, it can be observed that larger throughput time reductions do not necessarily come with better workload balances; which again provides only partial support for Hypothesis 4.

Summarizing, this sensitivity analyses revealed that regardless of over- or underestimating the expected processing times by 20 percent, better results are obtained compared to the base case. Dynamics are complex as either over-or underestimating impacts the result metrics differently.

5.5.2 Simulation period

As October is used as a primary simulation period, it is worthwhile to analyze the sensitivity of the results to a change in simulation period. The week of Black Friday is used (Monday, November 25th until Sunday December 1st); a week in which on each day, a specific product category was promoted. This period is interesting to analyze for mainly two reasons:

Firstly, compared to October, this is a busier period due to promotions (Black Friday) and as a result, more capacity is used to process the larger number jobs. This might impact the results as work centers are overall operating at nearly maximum capacity; hence variation in utilization among these work centers is lower. Secondly, the multi/mono job ratio shifts more towards multi jobs as customers, on average, order more products at once. This might impact the results as the proposed sequencing logic considers the completion of pools. Still, it is expected that the proposed logic will outperform the base case. However, it is expected that benefits from using the proposed logic as opposed to the current logic are smaller as there are both more jobs as well as more operators active in busier periods. Then, under the current logic for multi pools, it is more likely that some job will complete a multi pool; hence the added value of choosing which job completes a multi pool is lower compared to when there is fewer operators or less demand. Combining the above, the following combined hypothesis is constructed:

- ***Sensitivity hypothesis S2:*** regardless of the simulation period, the alternative case outperforms the base case simulation in terms of throughput time, waiting time, pool completion time and workload balancing but effects are smaller in the busier period.

Similar scenarios as for the base results are tested. More specifically the following scenarios are used:

- **Scenario S2:** where the true historical processing times are known (analogous to scenario 1);
- **Scenario S2a:** where processing times are estimated using historical averages (analogous to scenario 2a; and
- **Scenario S2b:** where regression equations as presented in Appendix D are used (analogous to scenario 2b).

Table 10 presents the results. Results for throughput, waiting, and pool completion time for all scenarios were found by t-tests to be significantly different from the base case results.

Table 10: Sensitivity results: simulation period.

	Throughput Time [in mins]			Waiting Time [in mins]			Pool Completion Time [in mins]		Utilization	
	Mo	Mu	Mo/Mu	Mo	Mu	Mo/Mu	Mu		Shift	
	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\overline{\Delta MSE}$	%Shifts better
Base	56.5 (34.9)	44.6 (16.1)	-	27.9 (30.4)	8.6 (11.9)	-	13.8 (4.6)	-	-	-
S2	53.5 (31.7)	42.1 (14.8)	5.3/5.6	24.8 (27.0)	7.3 (10.5)	11.1/15.1	13.3 (4.8)	3.6	137.3	71
S2a	53.1 (31.3)	41.6 (14.6)	6.0/6.7	24.4 (26.6)	7.0 (10.2)	12.5/18.6	13.3 (4.8)	3.6	130.6	50
S2b	52.7 (31.3)	41.4 (14.5)	6.7/7.2	24.6 (26.6)	6.9 (10.2)	11.8/19.8	13.3 (4.9)	3.6	140.1	64

Note. $\bar{X}$ (SD) refers the mean (standard deviation) values. * refers to the percentual reduction in $\bar{X}$ for the specified scenario compared to the base case. Mono jobs and multi pools are abbreviated to Mo and Mu respectively where Mu refers to the 75th percentile in terms of pool completion (or throughput) time for multi pools (see Appendix F for further details). Mo/Mu specifies that reductions for Mo and Mu are considered. %Shifts better specifies the percentage of shifts in which the specified scenario outperforms the base case in terms of MSE_s.

Throughput time reductions can be observed for mono jobs and multi pools regardless the scenario which suggests that the results for throughput time reduction are not very sensitive to the simulation period and it provides supporting evidence for Sensitivity Hypothesis S2.

Pool completion time is reduced in all three scenarios compared to the base case. However, the results suggest that for this busy November week, PCT is unaffected by the estimation method; average PCT is reduced by 3.6 percent in all three scenarios. Compared to the PCT reductions found in October (7-8 percent) this is a smaller reduction which is explained by the increase Multi/Mono ratio: given that there are relatively more multi jobs compared to mono jobs, the likelihood of a pick operator being allocated to a multi job is higher and hence pools are picked sooner. Hence, the added value of the sequencing logic as proposed in this thesis is smaller when it comes to PCT for higher Multi/Mono ratios which also explains the smaller throughput time reductions for multi pools.

Utilization variation is lower compared to the base case for all three scenarios. Thus, although more pressure is put on the work centers, the proposed sequencing logic still outperforms the current sequencing logic in terms of workload balancing. However, a better balance does not imply larger reductions providing once more only partial support for Hypothesis 4.

5.5.3 Safety parameter

This section explores the sensitivity of the results to the value of the configurable safety parameter. The value for the configurable safety parameter was set to 15 minutes as desired by company stakeholders. However, it might be that lower or higher values yield better or worse results. To test whether this is the case, two sensitivity scenarios are used on the October data as in the base results. In both scenarios, expected processing times are estimated by regression equations; that is both scenarios are analogous to scenario 2b besides the value of the configurable safety parameter. Scenario S3a uses a value of 30 and scenario S3b uses a value of 0 for parameter. Scenario 2b is chosen as the scenario to which results are compared because in practice, expected processing times are estimated and scenario 2b yielded better results than scenario 2a (see section 5.4.4).

It is expected that regardless of the value of the safety parameter, the proposed logic outperforms the current logic as information regarding the expected processing times and workloads at the work centers is taken into account. More specifically, the following hypothesis is constructed:

- ***Sensitivity hypothesis S3:*** regardless of the value for the safety parameter, the alternative case outperforms the base case simulation in terms of throughput time, waiting time, pool completion time and workload balancing.

When comparing safety parameter values 0, 15 and 30, it is expected that better results are obtained for a value of 0 as the calculation of $SafetyMargin$ already incorporates expected queuing time. Hence, usage of a safety parameter directs the proposed logic away from using workload information but directs the logic towards unnessiracy immediate allocation with waiting time as a result. The following hypothesis is constructed:

- ***Sensitivity hypothesis S4:*** lower safety parameter values yield better results compared to higher parameter values.

However, no large effects are expected as most jobs' cut-off times lie at the end of the second shift. Table 11 presents the results for the various scenarios. Results for throughput, waiting, and pool completion time for all scenarios were found by t-tests to be significantly different from the base case results.

Table 11: Sensitivity results: safety parameter.

	Value	Throughput Time [in mins]			Waiting Time [in mins]			Pool Completion Time [in mins]		Utilization	
		Mo	Mu	Mo/Mu	Mo	Mu	Mo/Mu	Mu		Shift	
		$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\overline{\Delta MSE}$	%Shifts better
Base	-	46.2 (30.6)	40.1 (15.0)	-	21.7 (27.1)	8.3 (11.3)	-	13.6 (5.2)	-	-	-
2b	15	43.7 (27.4)	36.7 (13.8)	5.4/8.5	19.3 (23.6)	6.9 (9.8)	11.1/16.9	12.5 (5.1)	8.1	135.6	63
S3a	30	43.9 (27.5)	36.8 (13.7)	5.0/8.2	19.4 (23.7)	7.0 (9.8)	10.6/15.7	12.6 (5.1)	7.4	185.5	63
S3b	0	43.7 (27.4)	36.7 (13.7)	5.4/8.5	19.3 (23.6)	7.0 (9.8)	11.1/15.7	12.5 (5.1)	8.1	155.0	63

Note. $\bar{X}$ (SD) refers the mean (standard deviation) values. * refers to the percentual reduction in $\bar{X}$ for the specified scenario compared to the base case. Mono jobs and multi pools are abbreviated to Mo and Mu respectively where Mu refers to the 75th percentile in terms of pool completion (or throughput) time for multi pools (see Appendix F for further details). Mo/Mu specifies that reductions for Mo and Mu are considered. %Shifts better specifies the percentage of shifts in which the specified scenario outperforms the base case in terms of MSE_s. Scenario S3a and S3b are analogous to scenario 2b where scenario S3a and S3b use safety parameter values of 30 and 0 respectively.

It can be observed that regardless of the scenarios, the base case is outperformed in throughput, waiting and pool completion time as well as workload balancing. This provides supporting evidence for Sensitivity Hypothesis S3. Additionally, there is a relatively small difference in results when comparing scenarios 2b, S3a and S3b. Unlike expected, results when safety parameter is set to zero are not better. Rather, results are similar as when 15 is used which provides partial support for Sensitivity Hypothesis S4. For utilization variation however, no such effects can be observed: although variation is reduced, the reduction is not larger for lower, and smaller for higher values of the safety parameter.

5.6 Results summary

This section summarizes and discusses the results based on the hypotheses. Results of previous (sensitivity) scenarios are aggregated in Table 12. All hypotheses developed in the previous sections are included in Appendix H and a version of Table 12 that includes standard deviations along with scenario descriptions is included in Appendix I.

Table 12: Results summary: primary and sensitivity

	Value	Throughput Time [in mins]			Waiting Time [in mins]			Pool Completion Time [in mins]		Utilization	
		Mo	Mu	Mo/Mu	Mo	Mu	Mo/Mu	Mu		Shift	
		$\bar{X}$	$\bar{X}$	$\bar{X}$ %↓*	$\bar{X}$	$\bar{X}$	$\bar{X}$ %↓*	$\bar{X}$	$\bar{X}$ %↓*	$\overline{\Delta MSE}$	%Shifts better
Base Oct	-	46.2	40.1	-	21.7	8.3	-	13.6	-	-	-
1	15	43.7	36.9	5.4/8.0	19.2	7.1	11.5/14.5	12.6	7.4	186.3	56
2a	15	44.1	36.9	4.5/8.0	19.6	7.1	9.7/14.5	12.7	6.6	168.7	60
2b	15	43.7	36.7	5.4/8.5	19.3	6.9	11.1/16.9	12.5	8.1	135.6	63
S1a	15	44.8	37.8	3.0/5.7	20.3	7.7	6.5/7.2	12.7	6.6	203.5	67
S1b	15	43.2	37.0	6.5/7.7	18.8	7.1	13.4/14.5	12.5	8.1	197.3	63
S3a	30	43.9	36.8	5.0/8.2	19.4	7.0	10.6/15.7	12.6	7.4	185.5	63
S3b	0	43.7	36.7	5.4/8.5	19.3	7.0	11.1/15.7	12.5	8.1	155.0	63
Base Nov	15	56.5	44.6	-	27.9	8.6	-	13.8	-	-	-
S2	15	53.5	42.1	5.3/5.6	24.8	7.3	11.1/15.1	13.3	3.6	137.3	71
S2a	15	53.1	41.6	6.0/6.7	24.4	7.0	12.5/18.6	13.3	3.6	130.6	50
S2b	15	52.7	41.4	6.7/7.2	24.6	6.9	11.8/19.8	13.3	3.6	140.1	64

Note. $\bar{X}$ refers the mean values. * refers to the percentual reduction in $\bar{X}$ for the specified scenario compared to the base case. Mono jobs and multi pools are abbreviated to Mo and Mu respectively where Mu refers to the 75th percentile in terms of pool completion (or throughput) time for multi pools. Mo/Mu specifies that reductions for Mo and Mu are considered. %Shifts better specifies the percentage of shifts in which the specified scenario outperforms the base case in terms of MSE_s. The scenarios in the upper part of the table should be interpreted relative to the base October scenario whereas the lower part of the table should be interpreted to the base case for 1-week November data.

Supporting evidence is found for Hypotheses 1, 2, and 3: throughput, waiting and pool completion time as well as variation in utilization are lower for all scenarios relative to the corresponding base case scenarios. Moreover, reductions for multi pools are larger compared to reductions for mono jobs; largely contributed to the fact that pool completion is sped up. Additionally, supporting evidence is found for Sensitivity Hypotheses S1, S2, S3, and partial evidence is found for S2 and S4.

Most interestingly, only partial support was found for Hypothesis 4 as although in the results reductions are obtained through better workload balance, not necessarily *larger* reductions are obtained through better workload balance. Hence, this finding conflicts with the expectations and challenges the fundamentals of the proposed sequencing logic; recall that the $WINQ$ is at the heart of the model.

However, since throughput and waiting time reductions in the results are always accompanied by better mean workload balance per shift ($\overline{\Delta MSE}$ is positive in all scenarios), it is believed that better balance does good rather than it does harm. Still, the exact dynamics seem unclear and not explained by the results obtained in this research.

Interestingly, evidence is found that Hypothesis 5 should be rejected: scenario 1 does not outperform scenario 2a and 2b. Thus, knowing the true historical processing times does not mean that it outperforms in terms of throughput time reduction nor PCT reduction. This is explained that when there are very large historical processing times observed for jobs at a certain work center, the sequencing tool will sequence less jobs towards that work center as it (falsely) observes a large Work In Next Queue (see Appendix F on scenario 1 for further details). Such outliers are also found when simply historical averages (scenario 2a); explaining its underperformance compared to scenario 1 and 2b. These outliers are not found when processing times are estimated by the smooth curves found by OLS regression (see Appendix D for more details).

With respect to Sensitivity Hypothesis 2, it was expected that the proposed tool would still reduce throughput time PCT, and enhance workload balancing, but it was expected that effects would be much lower compared to the primary results for the October month. Thus, although busier, there is still room for improving throughput times in busier periods. However, in line with expectation, the added value of the tool in terms of PCT reduction is lower in busier periods.

Another interesting finding not in line with expectation is regarding Sensitivity Hypothesis S4 which is partially supported by the results. Recall that the configurable safety parameter was incorporated (and set to 15 minutes) upon request of company stakeholders such that there would be an increased feeling of being in control. However, results reveal that no such parameter is necessary; only calculating the $SafetyMargin$ and only requiring immediate allocation when $SafetyMargin$ is lower than zero suffices. Results imply that regardless of the value (0, 15, or 30 minutes) for the configurable safety parameter, throughput, waiting and pool completion time are reduced. Results when using 0 or 15 reductions are lower compared to when 30 is used. It was expected that results were best when the value was set to zero as usage. However, results for 0 and 15 are similar which suggests that no additional safety parameter is needed for the model to outperform the current logic.

5.7 Discussion practical relevance

This section discusses the practical relevance of the results and hence includes some aspects of the previous sections. Further on it reports on the generalizability of the results in section 5.7.1

First, throughput time as well as variability of throughput times can be reduced when the proposed logic is used compared to the current logic. This results in the outbound process from picking to sorting and packing to be more controllable and predictable: picking is more aligned with sorting and packing (i.e., the right activities are more performed at the right time). Whether a throughput time reduction is achieved is found to be insensitive to the method of processing time estimation used in this thesis. However, best results are obtained when the processing times are estimated using the regression equations on the mean historical processing times per item quantity of the job or pool.

The sources of throughput time reduction lie in 1) waiting time reduction for mono jobs and 2) in waiting and pool completion time reduction for multi pools. For mono jobs, waiting time reductions are observed between 6.5 – 13.4 percent. For multi jobs waiting time reduction is larger: between 7.4-19.8 percent. In terms of the process environment in the simulation, this only implies that mono jobs wait shorter in de work centers' queues and multi jobs wait shorter in the queues as well as in the stingray. However, in practice it occurs in the case study company that jobs start looping on the conveyor system when no space is available in the work center queues. Then, more pressure is put on the conveyor system which implies that transportation time increases. In a peak period (e.g., week of Black Friday) this might even imply a so-called dead lock which means that the system runs into a stop as a result of too much jobs in either the conveyor system or the stingray. Hence, next to its role in throughput time reduction and increasing customer response time, waiting time reduction reduces the likelihood of the latter to happen which makes it an even more important phenomenon to reduce.

Additionally, for multi pools, throughput time is reduced by reducing pool completion time. In line with expectation, PCT reduction for the Black Friday week suggests that the magnitude of PCT reduction are smaller in busier periods. Still, performance is better than in the base case. Additionally, variation in PCT as measured by the standard deviation is reduced as well making PCT more predictable. Next to its role in throughput time reduction, in practice this implies that also the number of uncomplete pools that are waiting in the stingray is reduced. This in turn implies that less pressure is

put on the stingray reducing the likelihood of a deadlock. Put differently, a possible expansion decision for the stingray may be delayed as in case of the proposed logic, fewer incomplete pools reside in it as a result of speeding up pool completion. Moreover, in practice the case study company uses troubleshooter operators to manage the prioritization of jobs such that multi pools are completed sooner. Around €80,000 a year can be saved solely by automatically managing this prioritization if the proposed logic were to be applied.

Unlike expected, the results for the Black Friday week suggest that although busier, there is still room for throughput and waiting time reduction that is relatively similar (percentagewise) to the October scenarios. However, benefits from using the proposed logic as opposed to the current logic were expected to be smaller as there are both more jobs as well as more operators active in busier periods; then more pressure would be put on the system reducing waiting time improvement possibilities. Thus, in busier periods, the proposed logic performs beyond expectation based on the Black Friday week data.

Another interesting finding is that it is not necessary to apply an additional configurable safety parameter when deciding on whether a job requires immediate allocation for it to be finished in time; that is, safety parameter set to 0 suffices.

Throughput and waiting time reductions were expected to be achieved once workload is more balanced; it was expected that larger reductions in throughput and waiting time would be accomplished through better balance of the workloads among the outbound work centers. However, the results suggest that this is not the case; better workload balance as measured by utilization variation does not necessarily imply larger throughput time reductions. Hence, this finding conflicts with the expectations and challenges the fundamentals of the proposed sequencing logic; recall that the $WINQ$ is at the heart of the model. However, since throughput and waiting time reductions in the results are always accompanied by better mean workload balance per shift ($\overline{\Delta MSE}$ is positive in all scenarios), it is believed that better balance does good rather than it does harm. Still, the exact dynamics seem unclear and not explained by the results obtained in this research. However, as overall less imbalances are observed when comparing the utilizations of the outbound work centers, in practice this would imply that less back and forth movement of operators from work center to work center would be required. The

back and forth movements of operators are found to be unpleasant obstructing the controllability of the operations. Additionally, productivity loss occurs as operators must walk from one place to another. Thus, as the proposed logic on average better balances workload, productivity gains expected.

Summarizing, the results of the case study at Ingram Micro CLS suggest that throughput time can be reduced through waiting time and pool completion time reduction. Throughput time reductions are relatively small but larger when considering waiting time. Still, usage of the proposed logic may result in additional benefits as not only customer response time is improved but also in terms of how the conveyor system and the queues are utilized. These reductions are always accompanied by lower variations in utilization per shift which is expected lead to productivity gains as less back and forth movement of operators is required. However, the exact mechanisms through which workload balancing reduces throughput times are not clear as better performance is not always accomplished through higher levels of workload balancing. Finally, the tool automatically ensures that pool completion time is sped up which results in cost savings.

5.7.1 Generalizability

The proposed logic was designed to reduce throughput time through workload balancing where multiple outbound work centers and job types are distinguished, an assembly (or batching) process is included (i.e., pool completion in the stingray), and both same-day and next-day cut-off times are recognized. Although tested in the e-fulfillment business at a 3PL in the Netherlands, the logic may also be applied in other types of business. For example, in the manufacturing industry where both single- and multi-item products must be manufactured-to-order or assembled under due date restrictions. Or in the service industry, where different tasks (that arrive throughout the day) must be performed and possibly assembled. If several servers and job types can be recognized, and processing times can be estimated up front, the proposed logic can be applied.

However, one must carefully think through whether the assumptions made in this thesis' case study are valid for the business in consideration. For example, the infinite capacity assumptions for the stingray is valid for the case study company but in other businesses, capacity restrictions on the stingray (or an equivalent process) might be strictly necessary which is not currently incorporated in the

proposed logic; although incorporation of such restriction would be easily made it was not tested in this research.

Additionally, one must be careful when applying the logic to situations in which cut-off times can be observed throughout the day. In the case study, cut-off times are observed at 12:45, 13:45, 14:45 while the larger part of cut-off times is observed in the evening at 23:45 or at night at 01:45. The logic was not tested if the cut-off times were denser; i.e. closer to each other. Additionally, the logic was not tested in the absence of cut-off times (or when there is only one cut-off time at the end of the day); a situation that might occur in other businesses. However, it is expected that, under the assumptions stated in this thesis, throughput time gains can still be obtained. Still, one must be aware that the result presented in this thesis are obtained by exact calculation rather than Monte Carlo simulation. It would be interesting to further explore the dynamics of the logic in such simulation setting.

6. Implementation

This section provides some guidelines for implementing the proposed sequencing tool in practice. In principle, the output of the logic is a job that is sequenced to a picking operator. This can be in the form of a task popping up on a handheld scanner or even as simple as a printed document that is handed over to the operator. Once a job is requested in a particular region the program should perform calculations for each of the jobs in the region's queue according the logic and it should return the job that is found by the logic to the operator. No large investments in computer software or machines are required. Implementing the logic should be simple once the following information is available to the logic once a job is requested:

- The current time (i.e., the time at the moment a job is requested);
- The jobs and job characteristics (e.g., cut-off time, item quantity, outbound work center number) of the jobs in the queue of each region;
- The method for computing expected processing times;
- The number of operators (or machines) for each of the outbound work centers in the current shift and the next shift; and
- The status of each job in the system. That is, it must be known where a job is in the warehouse (e.g., being picked, in transport, or in the queue at an outbound work center) such that workloads present at each of the work centers can be computed. With respect to the case study company, the recently developed Warehouse Execution System (WES) that contains this data could be perfectly used for this.

The logic can be programmed in the WMS itself or it can be programmed in some other computer language (e.g., Python or Java) and work together with the WMS. The latter is preferred to ensure maintainability of the logic and to avoid unnecessary long computation times. Maintainability is important as the method for computing expected processing times may have to be update once processing times change (e.g., through improving the packing process). Moreover, it is very important that real-time data is available on how many operators are working in each outbound work center as the logic heavily relies on this measure in determining for what work center a job should be sequenced.

7. Conclusions and recommendations

This section summarizes the conclusions in section 7.1. Then, based on the findings and limitations of this research, section 7.2 provides recommendations for practitioners and the case study company, and section 7.2 provides recommendations for academics.

7.1 Conclusions

The e-commerce business is rapidly growing; even with the recent COVID-19 pandemic online retailers thrive while traditional brick-and-mortar retailers face many operations-related challenges. Along with fierce market competition, this pressurizes e-retailers to provide customers with short lead times. Thus, minimizing throughput time should be a top-priority objective within warehouses.

From theory it is known that workload balancing may lead to lower throughput times. Therefore, in this thesis a job sequencing tool was designed that takes workload balancing into account by applying a variant of the Work In Next Queue logic to a job shop environment in which two main flows and multiple job types are distinguished. The first flow comprises a two-stage tandem process (picking and packing) where each process consists of multiple independent parallel servers. The second flow is a three-stage tandem process (picking, pool completion and sorting) in which an assembly operation of jobs is required (pool completion) for sorting to commence. This assembly process is considered when sequencing jobs by a modification of the NUB-rule to a FUB-rule.

A case study at a 3PL in the Netherlands was conducted in which the logic proved its ability to reduce throughput time through workload imbalance reduction which is not only beneficial in terms of customer response time but also in terms of how the conveyor system and the queues are utilized. Throughput time reductions and waiting time reductions between [3.0 – 8.2] and [6.5 – 19.8] respectively were found in this research. These reductions seem relatively small. However, since workload is more balanced, productivity gains may be expected as less shifting of operators between work centers is required. Additionally, reductions for multi pools were always found to be larger compared to reductions for mono jobs which is explained by the fact that pool completion is sped up; without requiring manual prioritization of jobs. In busier periods as well, gains in terms of throughput

and waiting time are obtained, as well as for pool completion time However, then the added value of the logic in terms of pool completion time is smaller.

Unlike expected, larger reductions were not always achieved through better workload balancing. However, reductions were always accompanied by better workload balance compared to the sequencing logic applied in the case study company. Thus, no solid conclusions are drawn regarding the exact mechanisms through which reductions are achieved through better balancing.

Findings are relevant to academics as it opens new research possibilities regarding the use of the Work In Next Queue and the proposed modification of the NUB-rule to the FUB-rule. As a point of departure academics are recommended to shed more light on the dynamics between workload balancing and waiting time reduction as this research finds mixed evidence on this relationship.

Findings of this research are relevant for practitioners as throughput time reductions and possibly productivity gains can be obtained with no to little investment costs; only by altering the logic of job sequencing. However, one should carefully evaluate whether the assumptions made in this research can be applied to their business before deciding on implementing the logic.

7.2 Recommendations for practicioners

Foremost importantly, Ingram Micro CLS is recommended to apply the proposed sequencing tool in its operations. However, it must first ensure that the proper data is available (see section 6). In this research some assumptions had to be made due to the unavailability of reliable data. For example, historical data on the precise number of operators per pick region or outbound work center was not available. The logic heavily relies on the number of active operators or machines upon making a sequencing decision. This data should be available in real-time for the logic to be able to function properly and it is recommended that Ingram Micro CLS invests resources in ensuring the real-time availability of this data as it can also be used for future operations research. A good starting point would be to include this data in the newly developed Warehouse Execution System (WES); a system that would be very useful to the sequencing tool as it contains information regarding the location and characteristics of jobs in the warehouse. Moreover, it should be logged and communicated to the logic when operators are shifted from one work center to another; highlighting the importance of real-time reliable data availability.

Second, when all data is in place, all information is available to the logic and it is decided to implement the proposed sequencing tool, the company is recommended to thoroughly test the logic before ‘going live’ with it. For example, on a Saturday morning (when no packing or sorting operations are performed) some dummy jobs should be created, and it should be tested whether the logic indeed behaves as desired. Next, it should be evaluated whether different results are obtained prior to lunch versus after lunch when more jobs are requested as a result of higher customer demand. It is expected that larger reductions are obtained prior to lunch as this research found reductions to be lower for busier periods. However, it is recommended that this is verified.

Third, when the sequencing tool has been applied in practice for a few weeks, Ingram Micro CLS is recommended to investigate the possibility of incorporating the putaway process in the tool; i.e., how the proposed tool can be used to steer both the picking as well as the putaway process. For example, it could be analyzed whether it would be possible to require a pick operator to perform a putaway operation upon finishing the picking job when ‘sufficient’ workload is observed at all outbound work centers while still reducing throughput time through workload balancing.

Most importantly, regardless whether Ingram Micro CLS decides to go with the proposed sequencing tool, it should focus on real-time data availability and stop with the manual prioritization of jobs that would complete multi pools as resources are wasted in doing so and the burden frustrates team leaders of the various work centers as well as the control room. The proposed Fraction of Unscheduled Branches (FUB) rule proves to be very useful in reducing pool completion time.

Lastly, Ingram Micro CLS is recommended to continue to be eager to work with universities and graduate students as various directions for further (academic) research can be identified that might be relevant for practitioners as well (see the next section).

7.3 Recommendations for academics

There are several limitations of this study that can be addressed in further research.

Firstly, the results are obtained through exact calculation of historical case study data which impacts the generalizability of the results. Future research might expand this thesis by developing a Monte Carlo simulation in which fitted distributions are used for estimating the processing times and arrival data. Alternatively, research might address multivariate approaches to estimate processing times; particularly picking time is of the interest as factors such as walking distance, number of aisles and pick locations determine picking time. However, the added value of these estimations in terms of the results is not clear as this research suggests that using simpler model specifications already results in performance gains.

Secondly, this research assumed that a new job is immediately allocated to a picking operator upon finishing the previous job. It could as well be that if the sequencing decision is delayed, a better sequencing decision can be made as more jobs can be considered making the proposed tool more dynamic. Additionally, it would be interesting to investigate whether new jobs can be created upon finishing a previous job (rather than only sequenced). Then, for what work center should a job be created and how should that job be constructed (e.g., number of items or pick locations) given the workloads at the work centers? This provides interesting research opportunities for academics or graduate students.

Additional research opportunities can be directed at investigating the effects when the logic would ignore the safety margin and only focus on workload balancing. It is expected that workload would even be balanced more, and throughput time could be reduced even further as no due date considerations are required. Also, one of the assumptions is that jobs arrive at the pick regions under the round-robin policy. It might be interesting to investigate how the results are affected when different job arrival patterns exist per pick region. Moreover, it would be interesting to analyze how possible variability in transportation time may be included in the sequencing tool.

8. References

Aken, van, J. E., Berends, J. J., and Bij, van der, J. D. (2012). *Problem Solving in Organizations: A Methodological Handbook for Business and Management Students.* Cambridge: Cambridge University Press.

Baker, K. R. (1984). Sequencing Rules and Due-Date Assignments in a Job Shop. *Management Science*, *30*(9), 1093–1104. https://doi.org/10.1287/mnsc.30.9.1093

De Koster, M. B. M., Poort, E. S. V. Der, and Wolters, M. (1999). Efficient orderbatching methods in warehouses. *International Journal of Production Research*, *37*(7), 1479–1504. https://doi.org/10.1080/002075499191094

De Leeuw, S., and Wiers, V. C. S. (2015). Warehouse manpower planning strategies in times of financial crisis: Evidence from logistics service providers and retailers in the Netherlands. *Production Planning and Control*, *26*(4), 328–337. https://doi.org/10.1080/09537287.2014.904531

Gibson, D. R., and Sharp, G. P. (1992). Order batching procedures. *European Journal of Operational Research*, *58*(1), 57–67. https://doi.org/10.1016/0377-2217(92)90235-2

Kim, T. Y. (2018). Improving warehouse responsiveness by job priority management: A European distribution centre field study. *Computers and Industrial Engineering*. https://doi.org/10.1016/j.cie.2018.12.011

Petersen, C. G. (1997). An evaluation of order picking routeing policies. *International Journal of Operations and Production Management*, *17*(11), 1098–1111. https://doi.org/10.1108/01443579710177860

Petersen, C. G., and Aase, G. (2004). A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*, *92*(1), 11–19. https://doi.org/10.1016/j.ijpe.2003.09.006

Pinedo, M. L. (2012). *Scheduling* (5th ed.). New York: Springer.

Tyagi, N., Varshney, R. G., and Chandramouli, A. B. (2013). Six Decades of Flowshop Scheduling Research. *International Journal of Scientific & Engineering Research*, *4*(9), 854–864.

Ucar, I., Smeets, B., and Azcorra, A. (2019). Simmer: Discrete-event simulation for R. *Journal of Statistical Software*, *90*. https://doi.org/10.18637/jss.v090.i02

Van Den Berg, J. P. (2007). *Integral Warehouse Management: The Next Generation in Transparency, Collaboration and Warehouse Management Systems* (1st ed.). Utrecht: Management Outlook.

Van Nieuwenhuyse, I., de Koster, R., and Colpaert, J. (2007). Order batching in multi-server pick-and-sort warehouses. *Dtew - Kbi_0731*. Leuve: Department of Decision Sciences and Information Management (KBI). Retrieved from https://lirias.kuleuven.be/bitstream/123456789/175475/1/KBI_0731_research paper.pdf

Appendix A Assumptions

This appendix states the assumptions made in the case study. First, the most important assumptions mentioned in Section 5.1 are discussed in more detail Second, assumptions regarding buffer-requiring work centers are discussed. Then, the assumptions regarding the process environment itself are discussed.

Firstly, it is assumed that once a picking operator finishes a job (that is, she puts the tote on the conveyor belt), a new job is immediately allocated to the operator. This is a reasonable assumption for the case study company as the digital handheld scanners immediately propose a new job upon finishing the current job.

Secondly, for work centers that are inactive in the first shift but activated in the second shift of the day (buffer-requiring work centers), it is desired in the company that one hour of workload is present upon activating the work centers. In response, the proposed sequencing logic gradually generates workload for these buffer-requiring work centers when there is sufficient workload at the active work centers.

Thirdly, it is assumed that jobs are evenly distributed over the eight picking regions. That is, jobs are created round-robin for the regions. In practice, it occurs that some regions get more jobs than others which results in a disbalance in workloads across the pick regions. However, initiatives are set up to ensure that jobs are more evenly distributed to avoid these imbalances making this assumption a reasonable one.

Another assumption made is that each outbound work center has a single queue that operates under FCFS. Then, jobs go to the first available workstation. In practice, a workstation itself might have a small queue where jobs wait until that specific workstation becomes available. Thus, compared to the actual waiting times, the assumption may result in shorter waiting times as a result of pooling. Still, the conveyor system in general sends jobs to the first available workstation making the single queue assumptions a fair approximation of reality.

Also, it is assumed that the mono work center queues and the stingray have infinite capacity; that is, infinitely many jobs can be stored in them. In practice this is not the case, but it rarely occurs

that the system is overloaded making this a fair assumption unlikely to impact the results.

Additionally, transportation times (from pick region to work center) are assumed to be deterministic but different for each region-work center combination. In practice it may occur that jobs loop several times on the conveyor system or errors may occur making this a strong assumption. However, to prevent results to be affected by these effects, the assumption is made. Still, it will result in lower simulated throughput times compared to reality.

Buffer-requiring work centers

Some work centers such as the Smartmailer and the automatic sorting work center are inactive until the start of the second shift. However, upon activation there must be jobs to process in the queue at these work centers. Otherwise these work centers would be immediately idle upon activation, waiting for jobs to be picked. The conceptual model as described in section 4.2 does not take this into account: it only allocates jobs to order pickers such that the workload among the active work centers is balanced. For the model to be relevant for the case study company, it is desired that this buffering as described above is taken into account.

For the buffer-requiring work centers, stakeholders within the case study company mentioned that it is desired to have a workload present at work center for one hour. In generating workload for these work centers a trade-off exists. One should not prioritize batches for these buffer-requiring work centers when they are inactive, and other work centers are likely to run dry soon. Rather, when there is 'sufficient' workload at the active work centers, one can consider prioritizing a job for the buffer-requiring work center such that upon activation no idle time is incurred. The following will be used to ensure that sufficient workload is available for the buffer-requiring work centers upon activation:

When requested at time t, evaluate the following for the unscheduled jobs in picking region x that do not require immediate allocation:

1. Is the lowest $WINQ$ for the active work centers larger than one hour? *If yes, go to step 2. Else, go to step 3.*
2. Evaluate whether the lowest $WINQ$ for buffer-requiring work centers is a unique value and follow the conceptual model while only considering jobs directed to buffer-requiring work centers.

3. Follow the conceptual model as described before where the job is allocated directed to the active work center with the lowest $WINQ$ while only considering active work centers.

Process environment

This section states the assumptions made regarding *the process environment* in the case study company. Four different assumption categories are distinguished: 1) general assumptions regarding the process environment or applicable to multiple work centers; 2) assumptions regarding the (processes within) picking work center; 3) assumptions regarding the (processes within) three packing work centers; 4) assumptions regarding the (processes within) two sorting work centers.

Table A1: Assumptions: General.

General Assumptions

(G1)	Within the picking work center, a job is defined as a pick batch that may contain multiple items.
(G2)	Within the packing work centers, a job is defined as a pick batch that may contain multiple items and each item corresponds to a single customer order and each customer only contains one item.
(G3)	Within the sorting work centers, a job is defined as a pool which is a set of specific pick batches that together contain multiple items. In general, a pool contains multiple customer orders (at least one) consisting of at least two items.
(G4)	Apart from the picking work center, all work centers have queues that operate under FCFS.
(G5)	The stingray is a batching process that collects multi pick batches into specific, pre-determined pools and only releases a pool when a sorting work center has space available.
(G6)	The stingray has infinite capacity and operates under FCFS.
(G7)	The processing time of job i at work center j is equal to its true historical processing time. Note that the expected processing time of job i at work center j is not necessarily equal to its true historical processing time; rather this time is estimated as explained in section 5.3.
(G8)	A job is always processed on its designated outbound work center.
(G9)	$TransportationTime_{x,i,j}$ of job i, picked in region x to work center j is deterministic (jobs will never loop on the conveyor system).
(G10)	If work center j has space available for a pool: once the last job from a pool is picked in region x, it takes $TransportationTime_{x,i,j}$ for the entire pool of jobs to arrive at sorting work center j. If work center j does not have space available for a pool: the time it takes for the pool to travel from the stingray to work center j.
(G11)	If work center j does not have space available for a pool: pool travel time from the stingray to work center j is equal to the shortest travel time of all picking regions to work center j.
(G12)	The conveyor system has infinite capacity (no congestion or blocking).
(G13)	There are no preemptive outages (no machine failures) and no non-preemptive outages (planned machine maintenance).
(G14)	Operators are identical and do not get tired.
(G15)	Jobs that are not finished on the current day are processed on the next day.

Table A2: Assumptions: Picking.

Picking Assumptions

(Pi1)	The picking work center consists of eight identical picking regions (there are two towers, each with four floors) that operate independently.
(Pi2)	Items are equally distributed across (and within) the picking regions. Hence, jobs (pick batches) are equally distributed among the regions.
(Pi3)	Operators are equally distributed across the regions.
(Pi4)	Operators operate independently, in parallel and there is no congestion in the picking aisles.
(Pi5)	Operators operate within one region only and do no switch region, even if there is no work for them in their current region.
(Pi6)	There can be infinitely many jobs in the queues of the picking regions.
(Pi7)	Which job is dispatched depends on the rule that is in place (either the current job sequencing logic as in section 3.2 or the proposed alternative procedure).
(Pi8)	A pick batch always fits in one tote and will be put in one tote only.
(Pi9)	A pick batch (job) is always processed within one region.
(Pi10)	Walking distances are zero.
(Pi11)	The Picking work center is active from 08.00 – 0.30. The day shift is from 8.00 – 16.30 and the evening shift is from 16.30 – 0.30.
(pi12)	The number of active operators per shift is based on historical data.

Table A3: Assumptions: Packing.

Packing Assumptions

(Pa1)	One job (pick batch) is processed by one machine (or operator) only.
(Pa2)	Operators do not switch between work centers and workstations.
(Pa3)	There can be infinitely many jobs in the queues of the packing work centers.
(Pa4)	For the Mono Manual Packing work center (m = 96, at most): - Stations within the work center are identical; no distinction is made between VAS stations or High Risk stations. - Each station is operated by one operator and there are as much stations active as there are operators within the work center with a maximum of 96.
(Pa5)	The Mono Smartmailer work center (m = 1) is operated by two operators.
(Pa6)	For the Mono Cartonwrap work center (m = 3, at most): - The machines at the work center are identical. - Each machine is operated by two operators.
(Pa7)*	The Packing work centers can be active from 09.00 – 1.30. The day shift is from 9.00 – 17.30 and the evening shift is from 17.30 – 1.30.
(Pa8)	The number of active operators (stations) per shift is based on historical data.

Table A4: Assumptions: Sorting.

Sorting Assumptions

(S1)	Multi packing has enough capacity the process whatever is processed by the sorting work centers: multi packing is aligned with multi sorting for both work centers.
(S2)	One job (a pool) is processed by one operator only.
(S3)	Operators do not switch between work centers and workstations.
(S4)	The Multi Manual work center (m = 20, at most) has a queue length equal to m.
(S5)	The Multi Automatic Sorter can process at most 15 jobs (pools) at a time based on the number of operators that is active within the work center.
(S6)	The Multi Automatic Sorter has space in the queue for 10 pools regardless of the size of the pool. If the queue is full, pools are stored in the stingray.
(S7)	The Multi Manual Sorting work center is active from 09.00 – 1.30. The day shift is from 9.00 – 17.30 and the evening shift is from 17.30 – 1.30.
(S8)*	The Multi Automatic Sorting work center can be active from 09.00 – 1.30. The day shift is from 9.00 – 17.30 and the evening shift is from 17.30 – 1.30.
(S10)	The number of active operators (stations) per shift is based on historical data.

Note that the packing and sorting work centers are activated an hour after the picking work center is activated. This is done to ensure that upon activation, sorting and packing have jobs to process. If packing and sorting were activated together with picking, idle time would be incurred at packing and sorting until picking had processed jobs.

Pa7* and S8*: Unlike the other outbound work centers, buffer-requiring work centers such as Smartmailer and Multi Automatic Sorting work center are not necessarily active the entire day (see ***Assumption S8***). In busier periods these work centers may be active throughout the day (e.g., on Mondays, the entire peak season, or during promotion periods). In general however, this work center is inactive during the day shift and activated at the start of the evening shift.

Appendix B Detailed simulation procedure and validation

Discrete event simulation was used based on exact data to test the model. Usage of exact case study data (i.e., true historical processing times, true job arrival data) allows one to capture the true process behavior within the case study company. For example, it might occur that on some day, for some reason (e.g., promotions) customers bought more products for one particular outbound line compared to the others or some jobs may have had relatively long or short processing times. Additionally, it allows for capturing the batching behavior (i.e., releasement of jobs) of the control room. However, a disadvantage of using exact data is that generalizability of the results may be at risk. An alternative approach could be to use a Monte Carlo simulation in which elements of uncertainty are incorporated (e.g., by using fitted distributions to arrival or processing data) and the simulation is replicated multiple times to obtain (perhaps more robust) results. However, this would result in information loss regarding company specific process behavior. Therefore, it was decided to use exact company data to test the model. The reader should be aware of this when interpreting the results.

The simmer (Ucar, Smeets, and Azcorra, 2019) package for R was used as the simulation tool. First, a base case simulation (that uses the current job sequencing logic) was set by the following steps:

- Set up the process environment (see section 3.1) in R using the simmer package and program the current job sequencing logic.
- Load historical job data for which the following variables are required: *arrival time*, *job size* (number of items in the job), *cut off time*, *historical processing time*s, *outbound work center* (to which the job is directed) and the *job type* (pick sequence code). In addition, for multi jobs, pool data should be included: *pool ID*, *pool size* (number of items in the pool), *number of jobs in the pool*.
- Load historical data on the capacities, i.e. the number operators or machines per shift, and adjust it for breaks.
- Run the simulation and obtain the utilizations per shift of each of the pick regions.
- Adjust the number of picking operators per shift such that the highest utilization of the eight regions does not exceed 90% while each region has the same number of pickers.

- Rerun the simulation and adjust the number of operators or machines of the outbound work centers such that 1) all jobs are processed on the day that they were created and 2) 99% of the jobs is finished in time.
- Per simulated day obtain 1) the hourly utilizations of the outbound work centers; 2) the time it took for each multi pool to be completely picked and 3) the throughput time for each job (from the start of picking until finished).

To validate whether the base case simulation resembles the process environment in the case study company, job arrival data for the month October was used. Four weeks were simulated starting from Monday, October 7, 2019 until Sunday, November 3, 2019.

It is impossible to model all aspects of the processes within the case study company. Therefore, simplifying assumptions were made regarding the process environment. To check whether the simulation resembles the process environment under these assumptions, several crucial aspects of the simulation were evaluated. The validation process revealed that some bugs existed within the simmer package. For example, some jobs were unintentionally dropped from the simulation. Also, flaws were detected in the calculations of the work centers' utilization. After reporting these bugs to simmer's authors, they were fixed. It was verified that capacities of the work centers are correctly adjusted upon changing shifts. Picking operators are assigned jobs according to the current job sequencing logic (section 3.2). Jobs' processing times are equal to their historical processing time. Jobs go to the work centers corresponding to their job type. All multi jobs are send to the stingray where multi pool formation takes place. Pools wait in the stingray until there is available capacity at their designated sorting work centers. Hourly utilization is calculated by the fraction of time a resource is in use during the hour. To test whether the simulation correctly handles extremes, it was verified that utilization was 100 percent when the capacity (number of operators) was too low compared to the number of jobs. Consequently, and as suggested by queuing theory, when utilization approached 100 percent queue length explodes, resulting in jobs not being finished in time and processed on the next day. This behavior would also be observed in practice when the number of customer orders is too high compared to capacity.

It can be concluded that the most crucial aspects of the process environment are correctly incorporated in the simulation. Still, the simulation must not be considered a true representation of the real process behavior in the case study company. Table B1 compares the actual throughput times and the throughput times as simulated in the base case simulation.

Table B1: Throughput times: actual versus base case full sample

Throughput time (in mins)	***N***	**Actual**			**Base case**		
		Mean	**SD**	C_v	**Mean**	**SD**	C_v
Mono jobs	70053	67.2	60.6	0.90	46.2	30.6	0.66
Multi pools	13034	105.3	73.8	0.68	69.6	40.9	0.59

It can be observed that both the mean as well as the variability of the throughput times are lower in the base case simulation compared to the actual times. When looking at the actual data, outliers can be observed (throughput times of +10hrs); outliers of such magnitude are not observed in the base case results.

These outliers are explained by external factors causing delays that cannot be modeled. Examples are defect products detected before processing the jobs and machine (or transportation) failures that result in long waiting times. Considering these outliers, it is worthwhile to compare the actual throughput times for the 90th percentile to the base case results (Table B2).

Table B2: Throughput times: actual versus base case reduced sample

Throughput time (in mins)	**Actual (90th percentile)**			**Base case**		
	Mean	**SD**	C_v	**Mean**	**SD**	C_v
Mono jobs	53.6	26.7	0.50	46.2	30.6	0.66
Multi pools	88.0	43.0	0.49	69.6	40.9	0.59

Still, lower mean throughput times can be observed which may be explained by some of the assumptions (see section 5.1). For example, regarding the work centers' queues; in the simulation each work center has a single FCFS queue from which jobs go to the first available workstation and in practice jobs might wait for particular workstations to become available.

However, compared to the full sample, the reduced sample shows that the standard deviation of the actual throughput times and the simulated throughput times are about the same. This is a

promising result as although the simulation on average finds lower throughput times, the variation among these throughput times is similar to the actual throughput times.

Summarizing, the simulation model is found to be valid, but one must inevitably be careful when interpreting the results. Therefore, the results of the simulation are always interpreted relative to the base case simulation.

Appendix C Processing time estimation: historical averages

Simply averaging the historical processing times might be tempting as it is very straightforward. For example, the average picking time of a job (N = 592,371) is 5.9 minutes. However, the standard deviation of 4.5 minutes indicates that there is quite some variation among these picking times. By simply averaging the historical picking time, one neglects an important aspect that largely determines picking time: item quantity. It is possible to determine bins for which the processing times are averaged. An example of a bin could be all jobs with item quantity between 1-10. This might be useful when there are few observations per item quantity. However, the data used in the case study is quite large (October, November and December) which allows us to use small bins; as small as one.

A distinction is made between regular picking jobs and Smartmailer picking jobs (see section 3.4.1). Figure C1 plots the mean picking time and standard deviation per item quantity for regular picking jobs.

Figure C1: Picking time: regular jobs (N = 581708).

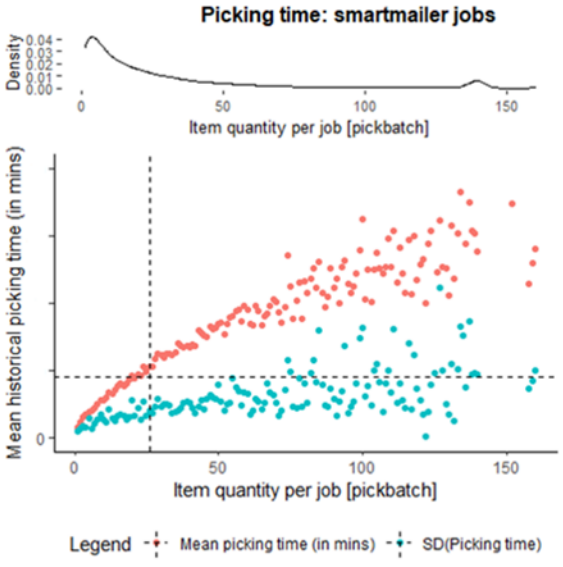


Figure C2 Picking time: smartmailer jobs (N = 10663).

The horizontal and vertical dashed line indicate the mean picking time and item quantity of 5.9 (7.4) minutes and 16 (26) respectively for regular (Smartmailer) jobs. Clearly, size of the job is an important variable for picking processing times. Similar to picking time, packing and sorting time also show

processing time to be related to item quantity. It can also be observed in Figure C3 and C4 that clearly distinct quantity-processing time relations exist across the various work centers and job types.

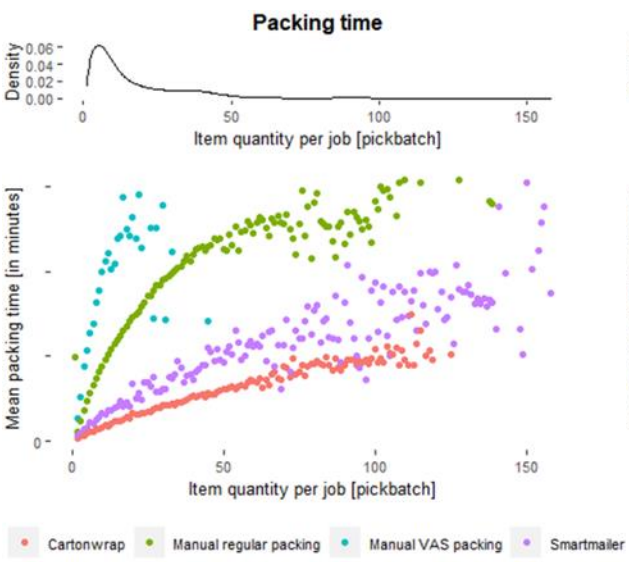


Figure C3: Historical averages: packing time (N = 229605).

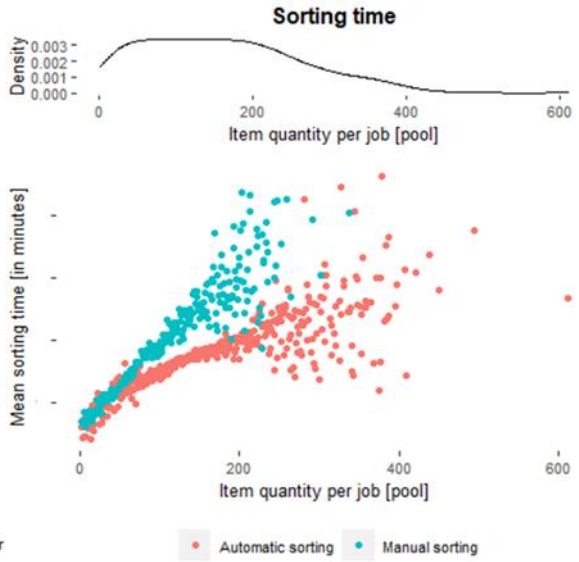


Figure C4: Historical averages: sorting time (N = 52271).

It can also be observed that variation in processing time increases as item quantity increases which is a result of a smaller number of observations. For example, for regular jobs picking time, there are 27688 observations for item size 4, 1021 observations for item size 52 and 4 observations for item quantity 120). Then, the historical average picking time of jobs with size 120 might not be a good estimate for the picking time. However, for smaller sizes, a clear pattern can be observed. Therefore, fitting a curve to the average processing times might better predict processing times which is done in the Appendix.

Appendix D Processing time estimation: OLS regression

The average processing times per item quantity presented in the previous section suggest that clear patterns exist where processing time is a function of item quantity. This appendix is aimed at modeling processing times as a function of item quantity. More specifically, OLS regression is used to fit a curve to the average processing times. The historical averages suggest that processing time is increasing in item quantity at a decreasing rate. 'Increasing in item quantity' makes sense as adding more items to a job generally will require the operator to visit more storage locations (walking time). 'Decreasing rate' is explained by the following: when the number of items increases by 'a lot', the likelihood of picking multiple items of the same type from the same storage location increases; hence picking time decreases. A model specification that allows modeling such non-linear behavior is the power regression model specification in $Equation\ (D1)$:

$$ProcessingTime_i = a \cdot ItemQuantity_i^{\beta_1} + \varepsilon_i \quad \text{(D1)}$$

where $ProcessingTime_i$ is the processing time of job i, alpha and beta are parameters to be estimated and ε_i is a mean zero, finite variance error term. It is expected that alphas and betas are positive (where beta is expected to be smaller than one) which would describe the above behavior. Note that the closer to 1 the value for beta the more linear the function.

Picking time estimation

Table D1 indicates that picking time for both regular and Smartmailer jobs is strongly correlated with item quantity where a stronger relationship exits for Smartmailer jobs.

Table D1: Pearson's correlations of item quantity and picking time.

	Regular jobs	Smartmailer jobs
	$PickingTime_i$	$PickingTime_i$
$ItemQuantity_i$	.750***	.830***
Observations	581708	10663

Note. Significance levels are denoted by ***, ** and * for 1%, 5%, and 10% respectively.

Figure D1 presents the average picking times per item quantity in which a curve is fitted that resulted from OLS regression using the power model specification.

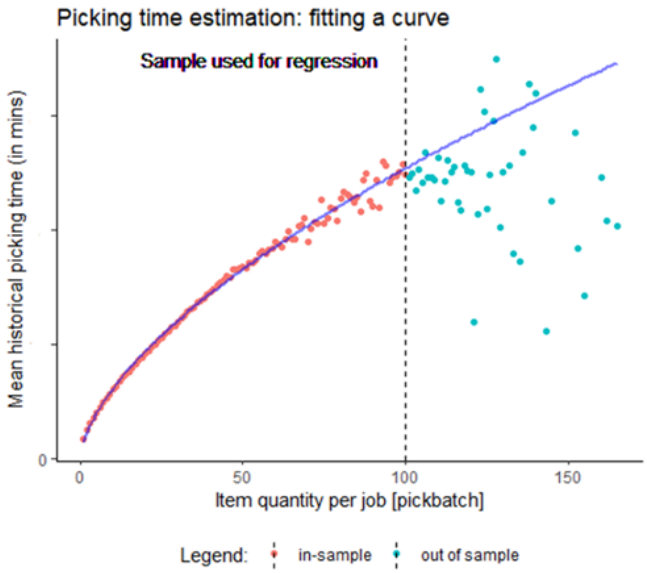


Figure D1: Picking time (regular) estimation: fitting a curve.

From item quantity 100 onwards, variation in mean picking time increases. Hence, the red points were used in the sample to obtain the model specification and fitting the curve. Table D2 summarizes the model results.

Table D2: Results of power model specifications: average picking time regular jobs.

Dep. var: $mean(PickingTime_i)$ **Indep. Var:** $ItemQuantity_i$	***Fitted curve*** Mean: 5.9 min Median: 4.8 min SD: 4.5 min N = 100 *Power*	***Historical averages***
$\widehat{\beta_1}$	0.493	
$\hat{a}$	1.191	
Prob > F	0.00	
Goodness of fit: N = 581708		
R^2_{adj}	0.58	0.58
MAE	1.74	1.74
$MDAE$	1.14	1.12
$MAPE$	0.27	0.27
$RMSE$	2.87	2.87

Note. Parameter estimates are statistically significant at the 1% level. Note that the curve is fitted on the historical average data points whereas goodness of fit is assessed for the individual observations.

In terms of predictive power, the fitted curve and historical averages perform equally well. The adjusted R-squared indicates that 58 percent of the variation in picking time is explained by the model. The mean absolute error (MAE) indicates that the average absolute error is 2.18 minutes whereas the median

absolute error (MDAE) is 1.42 minutes. The mean absolute percentage error (MAPE) indicates that on average, the predictor is wrong by 34 percent and the RMSE represents the mean of squared errors which is 3.59 minutes.

Table D3 presents the results for Smartmailer jobs.

Table D3: Results of power model specifications: average picking time Smartmailer jobs.

Dep. var: $mean(PickingTime_i)$ **Indep. Var:** $ItemQuantity_i$	***Fitted curve*** Mean: 7.3 min Median: 5.1 min SD: 6.7 min N = 50 *Power*	***Historical averages***
$\widehat{\beta_1}$	0.520	
$\hat{a}$	1.038	
Prob > F	0.00	
Goodness of fit: N = 10663		
R^2_{adj}	0.73	0.74
MAE	2.00	1.93
$MDAE$	1.06	1.05
$MAPE$	0.23	0.25
$RMSE$	3.54	3.40

Note. Parameter estimates are statistically significant at the 1% level. Note that the curve is fitted on the historical average data points whereas goodness of fit is assessed for the individual observations.

It can be observed that the fitted curve underperforms historical averages in terms of goodness of fit in four out of the five measures.

Picking time analyses were also done when distinguishing three types of picking jobs: regular, Smartmailer, and Cartonwrap. The predictive power of all model specifications was lower for regular picking jobs and Cartonwrap jobs (adjusted R-Squared below .40). In addition, it was also analyzed if distinguishing small jobs and large jobs (where various values for small and large were used) would increase the predictive power of the model specifications. However, this was not the case. Note that no distinction is made between mono and multi jobs as, from a picking perspective, these are similar.

Packing time estimation

In this section, average packing time per item quantity is estimated by fitting a curve to the historical average packing time per job type. The power regression specification is used to fit this curve.

In table D4, it can be observed that packing time is significantly related to item quantity.

Table D4: Pearson's correlations of item quantity and packing time.

	Manual regular	Manual VAS	Smartmailer	Cartonwrap
	$PackingTime_i$	$PackingTime_i$	$PackingTime_i$	$PackingTime_i$
$ItemQuantity_i$	.700***	.817***	.622***	.332***
Observations	195170	3680	10663	43296

Note. Significance levels are denoted by ***, ** and * for 1%, 5%, and 10% respectively.

Figure D2 presents the mean and standard deviation for historical packing time for mono regular jobs on the left and the fitted curve on the right.

Packing time estimation: Manual regular jobs

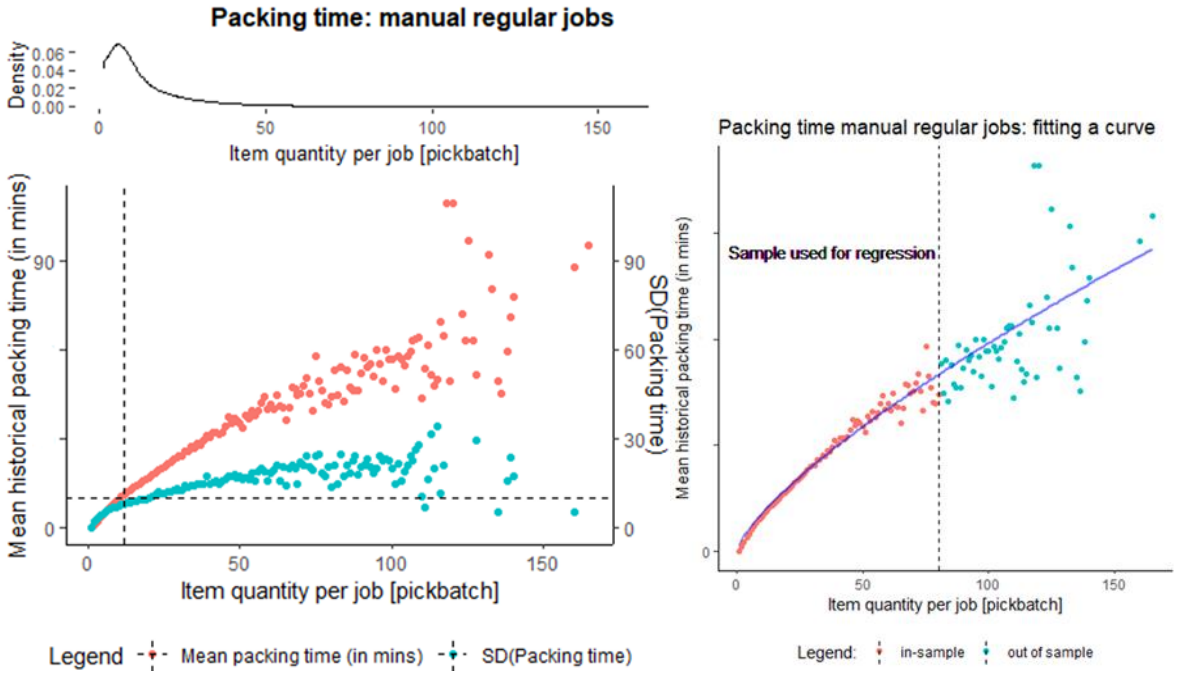


Figure D2: Fitted curve mean packing time: manual regular jobs.

Results of the power model specification are presented in Table D5.

Table D5: Results of power model specifications: average packing time manual regular jobs.

Dep. var: $mean(PackingTime_i)$ **Indep. Var:** $ItemQuantity_i$	***Fitted curve*** Mean: 8.3 min Median: 5.5 min SD: 9.3 min N = 80 *Power*	***Historical averages***
$\widehat{\beta_1}$	0.593	
$\hat{a}$	1.555	
Prob > F	0.00	
Goodness of fit: N = 195170		
R^2_{adj}	0.67	0.67
MAE	2.90	2.00
$MDAE$	2.02	0.87
$RMSE$	4.53	4.30

Note. Parameter estimates are statistically significant at the 1% level. Note that the curve is fitted on the historical average data points whereas goodness of fit is assessed for the individual observations. MAPE is not calculated as packing times of zero are in the data (for jobs with item quantity is 1).

Packing time estimation: Manual VAS jobs

Figure D3 presents the mean and standard deviation for historical packing time per item quantity for mono VAS jobs on the left and the fitted curve on the right.

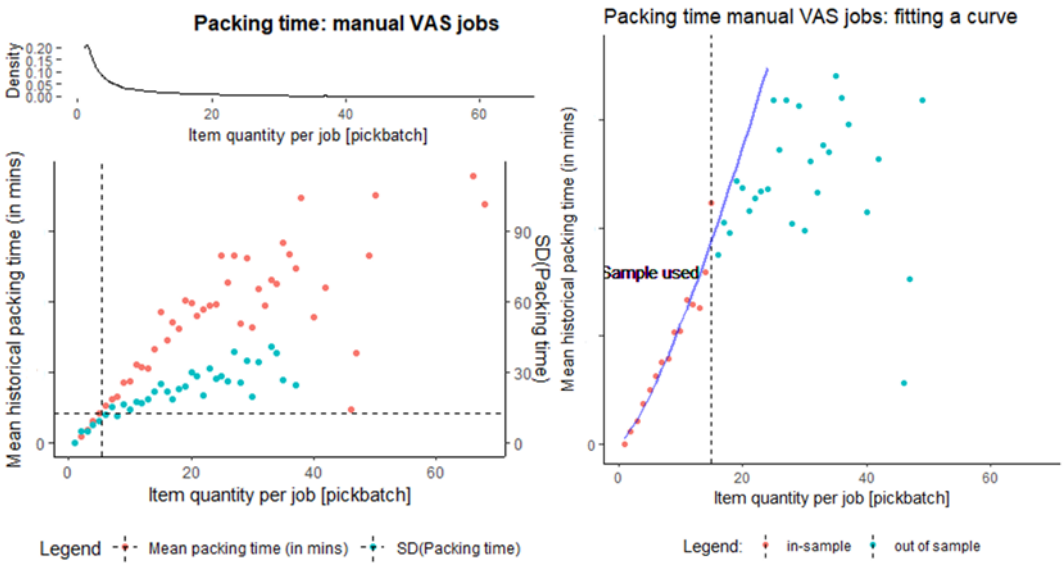


Figure D3: Fitted curve mean packing time: manual VAS jobs.

The number of observations per item quantity is low compared to the other job types. For example, there are just 29 jobs with item quantity 20 whereas there are 1197 jobs with item quantity 1. The fitted curve on the right suggests a convex relationship which cannot be reasonably argued. Additionally, if the more data points are used (i.e., more item quantities) the curve becomes concave, but this results in enormous errors for the lower item quantities. Hence, we prefer the use of historical average packing times over curve fitting for manual VAS jobs.

Packing time estimation: Smartmailer jobs

Figure D4 presents the mean and standard deviation for historical packing time per item quantity for Smartmailer jobs on the left and the fitted curve on the right.

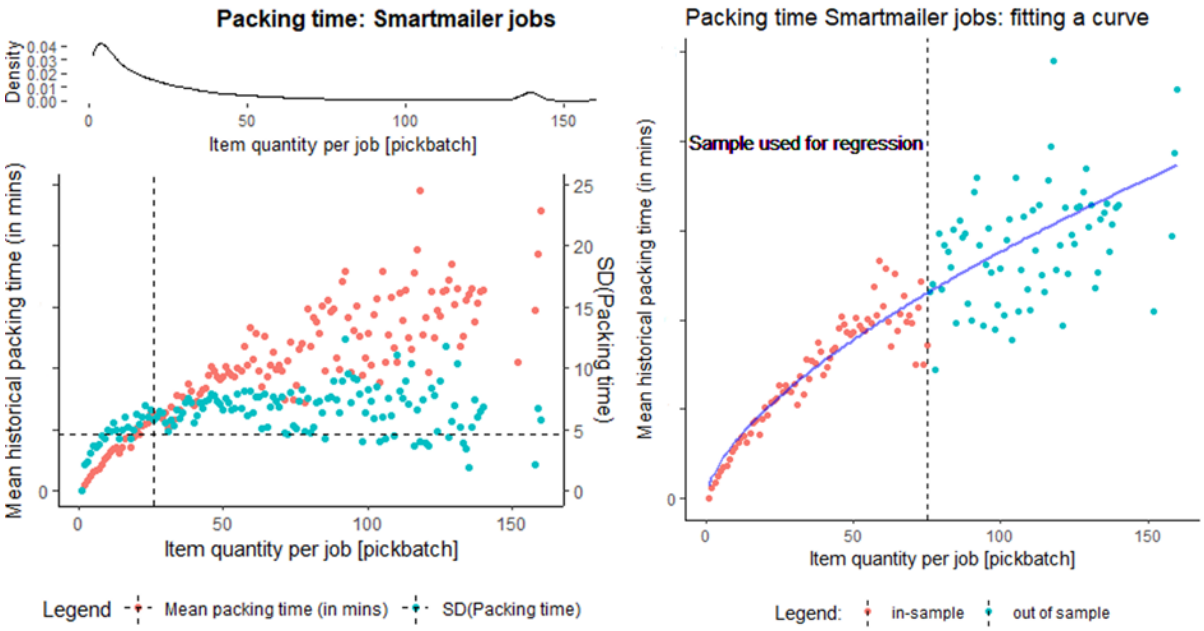


Figure D4: Fitted curve mean packing time: Smartmailer jobs.

Results of the power model specification are presented in Table B3.

Table D6: Results of power model specifications: average packing time Smartmailer jobs.

Dep. var: $mean(PackingTime_i)$ **Indep. Var:** $ItemQuantity_i$	***Fitted curve*** Mean: 3.8 min Median: 1.4 min SD: 5.3 min N = 75 *Power*	***Historical averages***
$\widehat{\beta_1}$	0.512	
$\hat{a}$	0.582	
Prob > F	0.00	
Goodness of fit: N = 10663		
R^2_{adj}	0.41	0.43
MAE	2.73	2.52
$MDAE$	1.78	1.51
$RMSE$	4.06	4.00

Note. Parameter estimates are statistically significant at the 1% level. Note that the curve is fitted on the historical average data points whereas goodness of fit is assessed for the individual observations. MAPE is not calculated as packing times of zero are in the data (for jobs with item quantity is 1).

Packing time estimation: Cartonwrap jobs

Figure D5 presents the mean and standard deviation for historical packing time per item quantity for Cartonwrap jobs on the left and the fitted curve on the right.

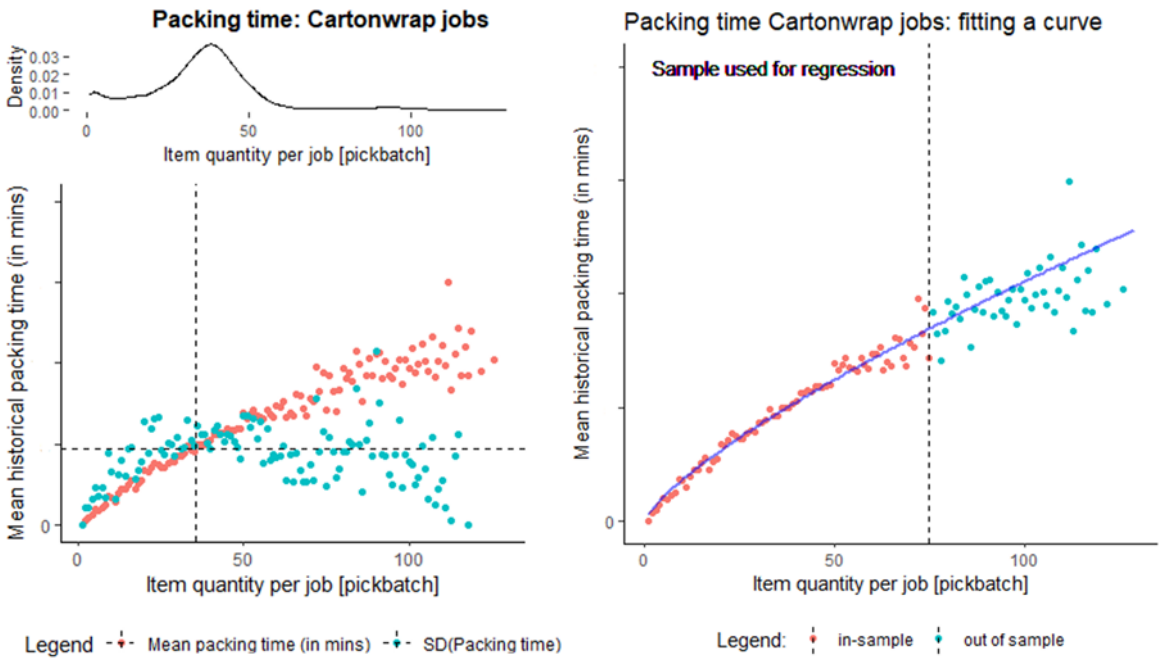


Figure D5: Fitted curve mean packing time: Cartonwrap jobs.

Results of the power model specification are presented in Table D7.

Table D7: Results of power model specifications: average packing time Cartonwrap jobs.

Dep. var: $mean(PackingTime_i)$ **Indep. Var:** $ItemQuantity_i$	***Fitted curve*** Mean: 3.8 min Median: 2.8 min SD: 4.4 min N = 75 <u>*Power*</u>	***Historical averages***
$\widehat{\beta_1}$	0.610	
$\hat{a}$	0.252	
Prob > F	0.00	
Goodness of fit: N = 43296		
R^2_{adj}	0.11	0.12
MAE	1.93	1.94
$MDAE$	1.38	1.42
$RMSE$	4.10	4.10

Note. Parameter estimates are statistically significant at the 1% level. Note that the curve is fitted on the historical average data points whereas goodness of fit is assessed for the individual observations. MAPE is not calculated as packing times of zero are in the data (for jobs with item quantity is 1).

Sorting time estimation

In this section, average sorting time per item quantity is estimated by fitting a curve to the historical average sorting time sorting work center. Note that for sorting, item quantity refers to the number of items in the pool (a collection of pick batches). The power regression specification is used to fit this curve.

In Table D8, it can be observed item quantity is significantly related to sorting time.

Table D8: Pearson's correlations of item quantity and sorting time.

	Manual sorting	Automatic sorting
	$SortingTime_i$	$SortingTime_i$
$ItemQuantity_i$	$.426^{***}$	$.312^{***}$
Observations	28034	30237

Note. Significance levels are denoted by ***, ** and * for 1%, 5%, and 10% respectively.

Sorting time estimation: Manual sorting

Figure D6 presents the mean and standard deviations for historical sorting time per item quantity for pools processed at Manual sorting on the left and the fitted curve on the right.

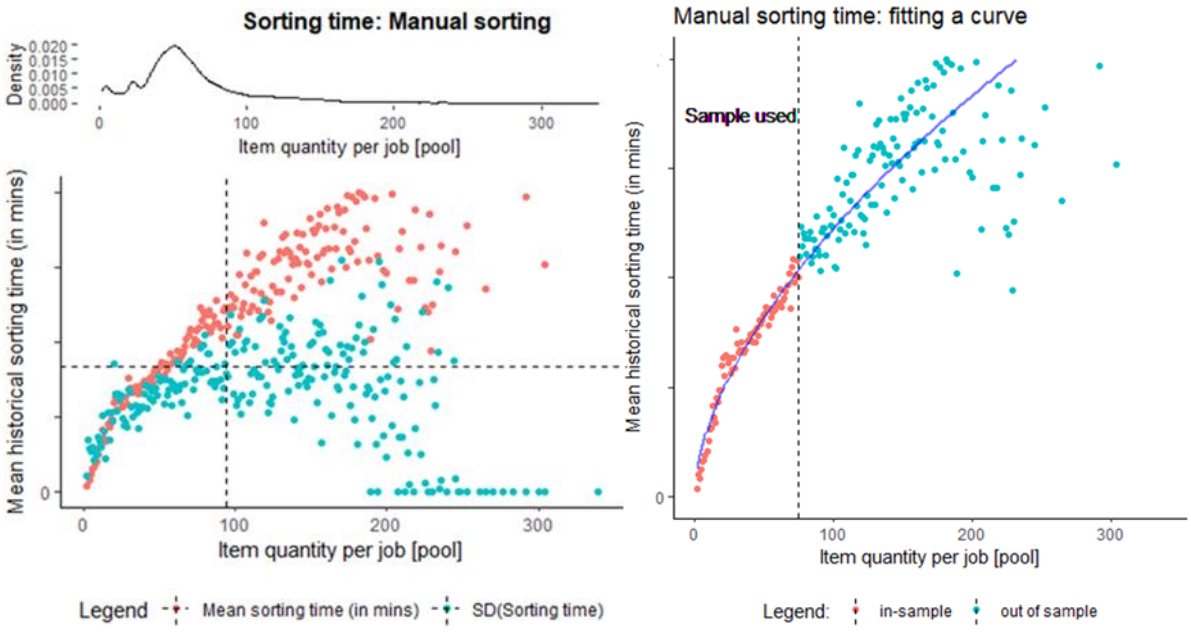


Figure D6: Fitted curve mean sorting time: Manual Sorting

Results of the power model specification are presented in Table D9.

Table D9: Results of power model specifications: average sorting time Manual.

Dep. var: $mean(PackingTime_{i)}$ **Indep. Var:** $ItemQuantity_i$	***Fitted curve*** Mean: 12.5 min Median: 9.1 min SD: 12.2 min N = 110 <u>*Power*</u>	***Historical averages***
$\widehat{\beta_1}$	0.480	
$\hat{a}$	1.263	
Prob > F	0.00	
Goodness of fit: N = 28034		
R^2_{adj}	0.19	0.20
MAE	7.19	7.07
$MDAE$	5.22	5.11
$MAPE$	0.73	0.60
$RMSE$	11.34	11.21

Note. Parameter estimates are statistically significant at the 1% level. Note that the curve is fitted on the historical average data points whereas goodness of fit is assessed for the individual observations.

Sorting time estimation: Automatic sorting

Figure D7 presents the mean and standard deviations for historical sorting time per item quantity for pools processed at Automatic sorting on the left and the fitted curve on the right.

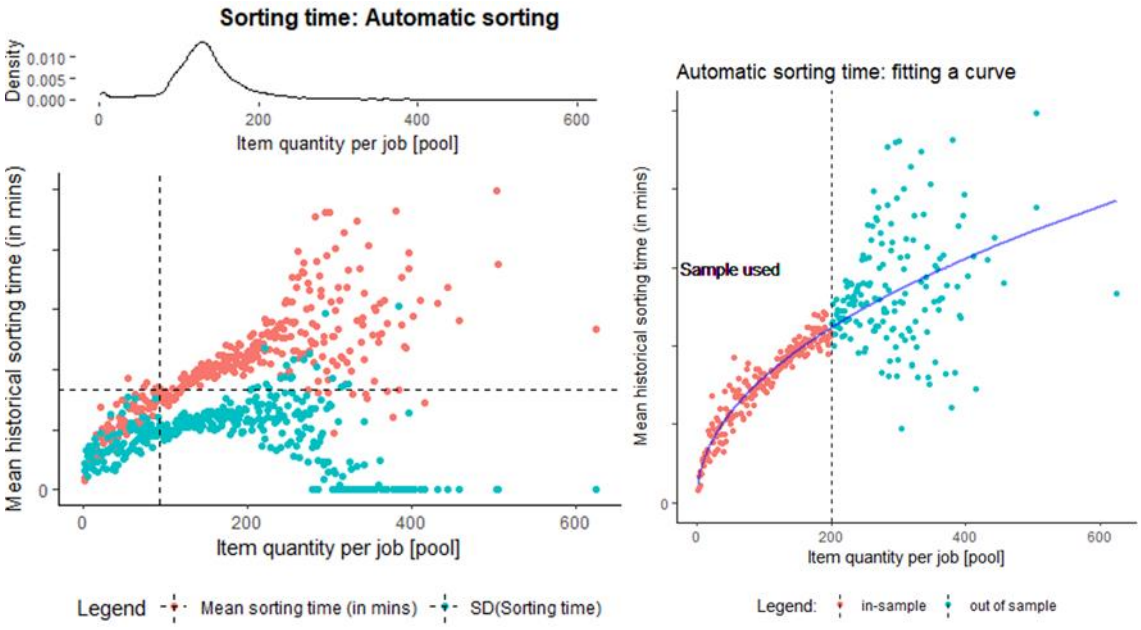


Figure D7: Fitted curve mean sorting time: Automatic Sorting

Results of the power model specification are presented in Table D10.

Table D10: Results of power model specifications: average sorting time Automatic.

Dep. var: $mean(PackingTime_{i)}$ **Indep. Var:** $ItemQuantity_i$	***Fitted curve*** Mean: 14.2 min Median: 11.8 min SD: 9.4 min N = 200 <u>*Power*</u>	***Historical averages***
$\widehat{\beta_1}$	0.385	
$\hat{a}$	1.394	
Prob > F	0.00	
Goodness of fit: N = 30237		
R^2_{adj}	0.10	0.13
MAE	6.32	6.23
$MDAE$	4.84	4.73
$MAPE$	0.51	0.49
$RMSE$	8.99	8.86

Note. Parameter estimates are statistically significant at the 1% level. Note that the curve is fitted on the historical average data points whereas goodness of fit is assessed for the individual observations

Appendix E Transportation time estimation

As mentioned in the assumptions, transportation time from the picking regions to specific work centers is assumed to be deterministic. It is based on historical averages. The time it takes for the first tote of a pool to arrive at a sorting work center is equal to the average travel time from the stingray to that particular work center.

Table E1: Transportation time from picking regions to outbound work centers.

	Outbound work center					
Picking region		*Mono Manual Packing*	*Mono Smartmailer*	*Mono Cartonwrap*	*Multi Manual Sorting*	*Multi Automatic Sorting*
	1	9.0	8.2	7.5	5.0	4.6
	2	6.8	6.0	5.4	5.0	4.6
	3	9.0	8.2	7.4	5.0	4.6
	4	6.8	5.9	5.3	5.0	4.6
	5	10.1	9.4	8.7	5.0	4.6
	6	7.4	6.6	6.0	5.0	4.6
	7	10.0	9.4	8.64	5.0	4.6
	8	7.4	6.5	6	5.0	4.6

Note. Transportation time is expressed in minutes.

Appendix F Detailed results: primary

This appendix presents the detailed results for the primary scenarios 1, 2a and 2b. Recall that we are interested in the throughput and waiting time, pool completion time and utilization variation.

Scenario 1

This section presents the results for scenario 1 where the expected processing times used for job sequencing are set equal to the true historical processing times for each of the jobs.

Throughput time – scenario 1

Table F1 presents the throughput time and waiting time results for scenario 1.

Table F1: Throughput and waiting time: Scenario 1.

		Mean (median) throughput time	SD (C_v)	Mean (median) waiting time	SD (C_v)
Mono jobs *N = 70053*	**Base**	46.2 (38.3)	30.6 (0.66)	21.7 (12.0)	27.1 (1.25)
	Alternative	43.7 (37.2)	26.7 (0.61)	19.2 (11.7)	22.8 (1.19)
Multi pools *N = 13033*	**Base**	69.6 (64.4)	40.7 (0.58)	29.3 (20.6)	33.2 (1.13)
	Alternative	68.2 (62.1)	41.9 (0.61)	27.1 (17.9)	32.6 (1.20)

It can be observed that throughput times are lower for both mono jobs (-5.4%) as well as multi pools (-2%) in the alternative case compared to the base case. Additionally, there is lower variability in throughput time for mono jobs (lower C_v). Lower throughput times however can only be a result of waiting time reduction as processing and transportation times are fixed. For mono jobs a mean waiting time reduction of 11.5 percent can be observed. Additionally, relative variability of waiting times is reduced as well.

For multi pools, lower throughput times can be observed as well but the variability in throughput times is larger compared to the base case. For this to understand, we first turn to analysis on pool completion time.

Pool completion time – scenario 1

The mean, median and standard deviation for pool completion time (PCT) are presented in Table F2.

Table F2: Pool completion time: Scenario 1.

N = 13034	Mean (in min)	Median (in min)	SD (in min)	C_v
Base	17.6	16.5	8.9	0.51
Alternative	18.4	15.0	13.7	0.75

It can be observed that on average, PCT is longer for the alternative case. However, the lower median indicates lower PCT for the lower half of the pools; i.e., skweness is reduced. Variability in PCT is larger for the alternative case. Then, there is a long tail with high PCTs. This makes sense as the proposed sequencing logic gradually creates workload for buffer-requiring work centers throughout the day (Appendix A). Then, depending on the $WINQ_j$ of the active work centers, jobs might be chosen to gradually generate workload without focussing on quick pool completion; hence some pool completion times are larger compared to the base case. Thus, it is worthwile to consider a reduced sample where longer times are excluded. Table F3 presents the results for the 75th percentile in terms of PCT.

Table F3: Pool completion time (75th percentile): Scenario 1.

N = 9775	Mean (in min)	Median (in min)	SD (in min)	C_v
Base	13.6	13.9	5.2	0.38
Alternative	12.6	12.5	5.0	0.40

Now it can be observed that PCT is indeed reduced for the larger part of the pools; mean PCT is 7.4 percent lower compared to the base case. Figure F1 plots the PCT density graphs for the alternative and base case. The density function is shifted towards lower PCTs and more centered around the mean in the alternative case versus the base case; i.e., pool completion is sped up and more predictable.

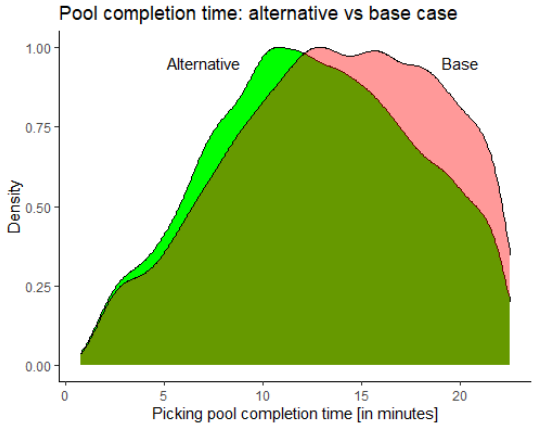


Figure F1: PCT Density graphs 75th percentile: Scenario 1.

Now if we consider the throughput times for multi pools in the previous section again, throughput times were found to be lower, but variability was found to be larger. This is in line with the findings on PCT

above where longer PCTs are observed for some of the pools. Table F4 presents the multi pool results for the 75th percentile in terms of throughput time.

Table F4: Throughput and waiting time (multi pool 75th percentile): Scenario 1

		Mean (median) throughput time	SD (C_v)	Mean (median) waiting time	SD (C_v)
Multi pools *N = 6851*	**Base**	40.1 (39.4)	15.0 (0.37)	8.3 (2.8)	11.3 (1.36)
	Alternative	36.9 (36.1)	13.9 (0.38)	7.1 (2.0)	9.7 (1.37)

Now, a mean throughput time (waiting time) reduction of about 8.0 (14.5) percent can be observed compared to the base case while variability is lower as well.

Utilization – scenario 1

Whether the proposed logic outperforms the current logic in terms of variation in utilization among the work centers is measured by the difference in MSE_s for the base case compared to the MSE_s for the alternative case scenario (ΔMSE_s).

On average, there is a positive ΔMSE_s ($\overline{\Delta MSE} = 186.3$) which indicates that the variation in utilization among the work centers per shift, on average, is smaller when the dynamic sequencing logic (alternative case) is used compared to the current sequencing logic. To see whether variation differs among the shifts, Figure F2 plots the difference in MSE for the base case and the alternative case simulation. Green bars indicate that the MSE_s in that shift is lower when the proposed logic is used compared to the current logic.

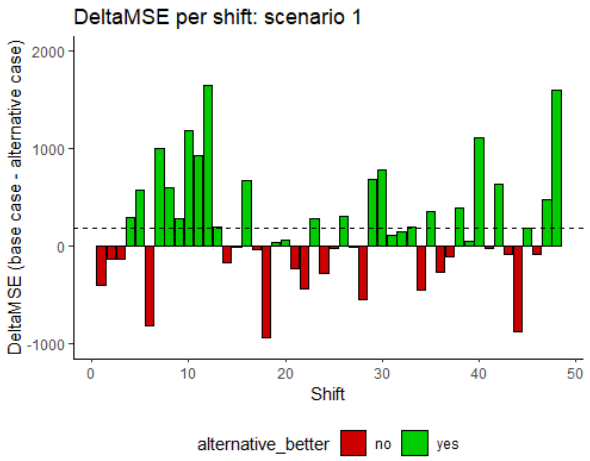


Figure F2: Delta MSE per shift: Scenario 1.

It can be observed that, although on average there is a positive effect, it differs per shift whether the alternative case outperforms the base case. Still, the alternative case outperforms the base case in 56

percent of the shifts, 57 percent of the hours, and 68 percent of the days.

Additionally, it can be observed that the average ΔMSE_s for the shifts in which the alternative case outperforms is larger compared to the absolute average ΔMSE_s in which it underperforms.

However, large negative ΔMSE_s are observed for shift 6, 18 and 44; i.e., the second shift of day 3, 10 and 25 respectively. For shift 18 and 44, the data shows that for one hour, the utilization among the work centers varies considerably for the alternative case. More specifically for that hour, the utilization of work center 1 is about five times lower compared to the other four work centers. This is not observed in the base case. Clearly, compared to the base case, less workload was generated for work center 1 in the alternative case; i.e., fewer jobs for that work center were sequenced.

Recall that the proposed sequencing logic generally sequences jobs directed to the work center with the lowest work in next queue ($WINQ_j$); a number that is calculated by summing the *expected* processing times of jobs sequenced for a specific work center. When looking at the processing times, extremely large, unrealistic processing times (packing time > 1.5 hour) are observed for jobs directed to work center 1 on the above-mentioned days. As in scenario 1 the expected processing times are set equal to the true historical processing times, $WINQ_1$ was increased by unrealistically large amounts once these jobs were sequenced for picking. As a result, no new jobs for work center 1 were sequenced to picking as $WINQ_1$ was larger compared to the other work centers; explaining the low utilization for that work center (hence the large variation among the work centers).

Figure F3 plots the difference in MSE for the base case and the alternative case simulation in which the above-mentioned hours are excluded.

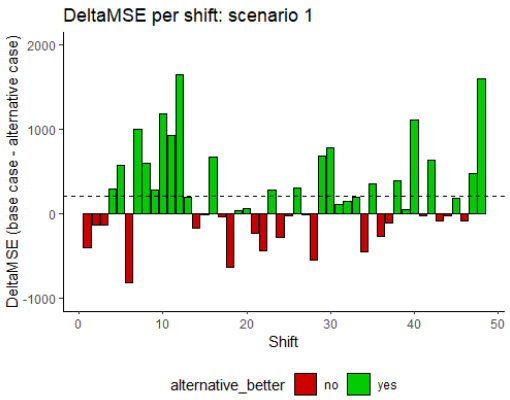


Figure F3: Delta MSE per shift: Scenario 1 excluding two hours.

Excluding these two hours considerably reduced MSE_s for the alternative case for shift 18 and 44. This might indicate that the results are sensitive to how expected processing times are estimated which is analyzed when turning to scenario 2a and 2b.

Scenario 2a

This section presents the results for scenario 2a where the expected processing times used for job sequencing are estimated by the mean historical processing times per item quantity (see Appendix C).

Throughput time – scenario 2a

The results on mono jobs in Table F5 indicate that both mean and median throughput time as well as variability in throughput times is lower compared to the base case. That is, mean throughput (waiting) time is 5 (10) percent lower compared to the base case.

Table F5: Throughput and waiting time: Mono jobs scenario 2a.

		Mean (median) throughput time	SD (C_v)	Mean (median) waiting time	SD (C_v)
Mono jobs *N = 70053*	**Base**	46.2 (38.3)	30.6 (0.66)	21.7 (12.0)	27.1 (1.25)
	Alternative scenario 2a	44.1 (37.8)	27.6 (0.64)	19.6 (11.5)	23.9 (1.21)

Table F6 indicates that the throughput (waiting) time for multi pools is about 8 (15) percent lower compared to the base case while variability in throughput and waiting times is reduced. The variability of waiting times is slightly larger in relative terms but lower in absolute terms compared to the base case.

Table F6: Throughput and waiting time (multi pool 75th percentile): Scenario 2a.

		Mean (median) throughput time	SD (C_v)	Mean (median) waiting time	SD (C_v)
Multi pools *N = 6851*	**Base**	40.1 (39.4)	15.0 (0.37)	8.3 (2.8)	11.3 (1.36)
	Alternative scenario 2a	36.9 (36.3)	13.8 (0.37)	7.1 (1.8)	9.9 (1.39)

Thus, although historical averages are used to estimate the processing times, throughput time is still reduced compared to the base case.

Pool completion time – scenario 2a

Table F7 indicates that both mean and median PCT are lower compared to the base case. Standard deviation and the coefficient of variation on the other hand are slightly higher which is explained by pools picked for buffer-requiring work centers.

Table F7: Pool completion time (75th percentile): Scenario 2a.

N = 9775	Mean (in min)	Median (in min)	SD (in min)	C_v
Base	13.6	13.9	5.2	0.38
Alternative scenario 2a	12.7	12.5	5.3	0.42

Thus, despite estimates (historical averages) are used for the expected processing times, PCT is still sped up. More specifically, the mean (median) PCT is reduced by 7 (10) percent compared to the base case.

Utilization – scenario 2a

There is a positive $\overline{\Delta MSE}$ (168.7), which indicates that variation on average is smaller than in the base case. Figure F4 indicates variation to differ among the shifts; most of the shifts the alternative case outperforms the base case and it is relatively better on outperforming days than it is worse on underperforming days. Still, the alternative case outperforms the base case in 60 percent of the shifts, 59 percent of the hours, and 61 percent of the days.

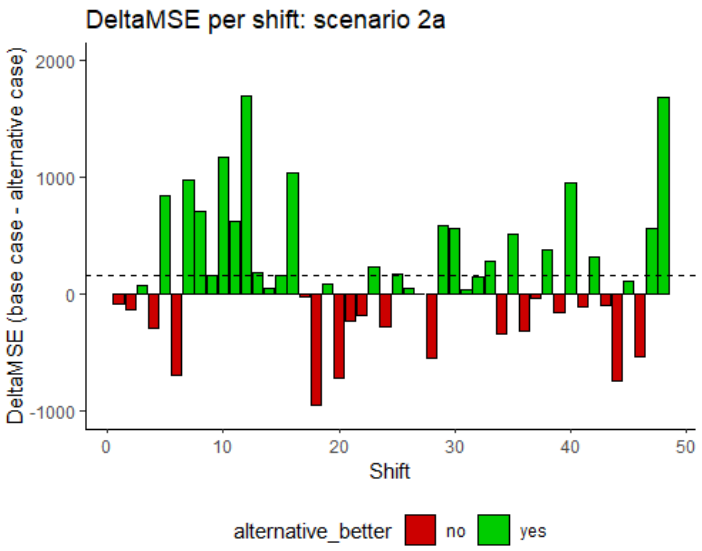


Figure F4: Delta MSE per shift: Scenario 2a.

Scenario 2b

This section presents the results for scenario 2b where the expected processing times used for job sequencing are estimated by the equations fitted to the average historical processing times (see Appendix B).

Throughput time – scenario 2b

The results for mono jobs in Table F8 indicate that both mean and median throughput time as well as variability in throughput times is lower compared to the base case. That is, mean throughput (waiting) time is 5 (11) percent lower compared to the base case.

Table F8: Throughput and waiting time: Mono jobs scenario 2b.

		Mean (median) throughput time	SD (C_v)	Mean (median) waiting time	SD (C_v)
Mono jobs *N = 70053*	**Base**	46.2 (38.3)	30.6 (0.66)	21.7 (12.0)	27.1 (1.25)
	Alternative scenario 2b	43.7 (37.4)	27.4 (0.63)	19.3 (11.3)	23.6 (1.22)

Similarly, for multi pools a throughput time and waiting time reduction can be observed of 9 and 17 percent respectively compared to the base case (Table F9).

Table F9: Throughput and waiting time (multi pool 75th percentile): Scenario 2b.

		Mean (median) throughput time	SD (C_v)	Mean (median) waiting time	SD (C_v)
Multi pools *N = 6851*	**Base**	40.1 (39.4)	15.0 (0.37)	8.3 (2.8)	11.3 (1.36)
	Alternative scenario 2b	36.7 (35.9)	13.8 (0.38)	6.9 (1.4)	9.8 (1.41)

Pool completion time – scenario 2b

The results in Table F10 demonstrate that both mean and median PCT are reduced. More specifically, PCT is sped up by 8 percent. Additionally, the slightly lower standard deviation indicates that variability in PCT is reduced as well.

Table F10: Pool completion time (75th percentile): Scenario 2b.

N = 9775	Mean (in min)	Median (in min)	SD (in min)	C_v
Base	13.6	13.9	5.2	0.38
Alternative scenario 2b	12.5	12.4	5.1	0.41

Utilization – scenario 2b

There is a positive $\overline{\Delta MSE}$ (135.6), which indicates that variation on average is smaller than in the base case. Similar to previous scenarios, performance differs among the shifts (see Figure F5). Still, the alternative case outperforms the base case in 63 percent of the shifts, 59 percent of the hours, and 61 percent of the days.

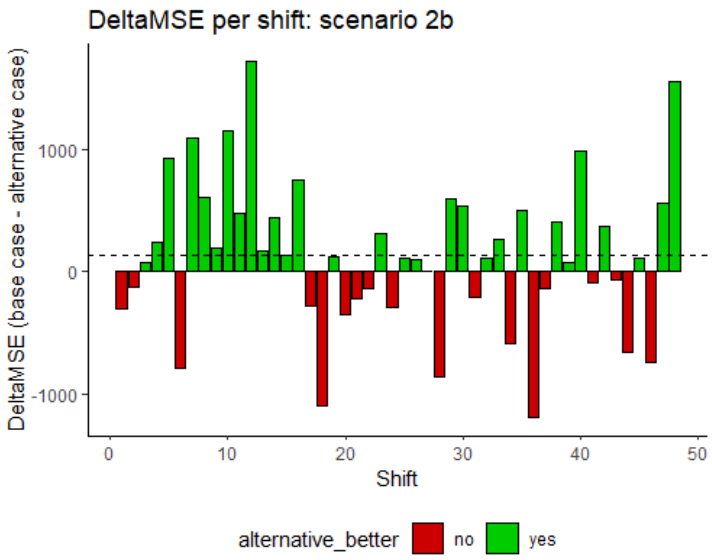


Figure F5: Delta MSE per shift: Scenario 2b.

Appendix G Detailed results: sensitivity

This appendix presents the detailed results for the three sensitivity analyses: 1) estimation error; 2) simulation period; and 3) safety parameter.

Estimation Error

Table G1 presents the result for scenario S1a, S1b, 1 and the base case. Results for throughput, waiting, and pool completion time for all scenarios were found by t-tests to be significantly different from the base case results.

Table G1: Sensitivity results: estimation error.

	Throughput Time [in mins]			**Waiting Time [in mins]**			**Pool Completion Time [in mins]**		**Utilization**	
	Mo	**Mu**	**Mo/Mu**	**Mo**	**Mu**	**Mo/Mu**	**Mu**		**Shift**	
	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\overline{\Delta MSE}$	%Shifts better
Base	46.2 (30.6)	40.1 (15.0)	-	21.7 (27.1)	8.3 (11.3)	-	13.6 (5.2)	-	-	-
1	43.7 (26.7)	36.9 (13.9)	5.4/8.0	19.2 (22.8)	7.1 (9.7)	11.5/14.5	12.6 (5.0)	7.4	186.3	56
S1a -20%	44.8 (27.1)	37.8 (14.6)	3.0/5.7	20.3 (23.4)	7.7 (10.6)	6.5/7.2	12.7 (5.2)	6.6	203.5	67
S1b + 20%	43.2 (27.1)	37.0 (13.9)	6.5/7.7	18.8 (23.2)	7.1 (9.8)	13.4/14.5	12.5 (4.9)	8.1	197.3	63

Note. $\bar{X}$ (SD) refers the mean (standard deviation) values. * refers to the percentual reduction in $\bar{X}$ for the specified scenario compared to the base case. Mono jobs and multi pools are abbreviated to Mo and Mu respectively where Mu refers to the 75th percentile in terms of pool completion (or throughput) time for multi pools (see Appendix F for further details). Mo/Mu specifies that reductions for Mo and Mu are considered. %Shifts better specifies the percentage of shifts in which the specified scenario outperforms the base case in terms of MSE_s.

In terms of variation in utilization among the work centers, both over- and underestimating yield lower variation compared to the base case (positive $\overline{\Delta MSE}$). Hence, when the proposed job sequencing logic is used, workload is balanced more compared to the current sequencing logic regardless of the estimation error. However, both S1a and S1b outperform scenario 1, where underestimating yields the best results in terms of $\overline{\Delta MSE}$ and shift-performance. Still, these effects are relatively small.

When considering pool completion time, throughput and waiting times. Pool completion time is reduced by nearly the same amounts as in scenario 1, compared to the base case. Hence, pool completion time seems insensitive to the estimation error. This makes sense as the fraction of unscheduled branches (see section 4.1.3) is independent of the expected processing times of jobs.

Compared to the base case, throughput, waiting times and variability of both is reduced

regardless of over- or underestimating. For mono jobs, throughput time reduction is higher when overestimating. For multi pools on the other hand, it can be observed that scenario 1 performs best in throughput and waiting time reduction closely followed by scenario S1b. When underestimating in S1a, throughput time reduction is lower. This suggests that multi pool throughput times are relatively more sensitive to underestimating rather than overestimating where overestimating hurts less than underestimating.

Simulation period

As October is used as a primary simulation period, it is worthwhile to analyze the sensitivity of the results to a change in simulation period. One week of November data serves as the input for this analysis. More specifically, the week of Black Friday is used (Monday, November 25th until Sunday December 1st); a week in which on each day, a specific product category was promoted. This period is interesting to analyze for mainly two reasons.

Firstly, compared to October, this is a busier period due to promotions (Black Friday) and as a result, more capacity is used to process the larger number jobs. This might impact the results as work centers are overall operating at nearly maximum capacity; hence variation in utilization among these work centers is lower. Secondly, the multi/mono job ratio shifts more towards multi jobs as customers, on average, order more products at once. This might impact the results as the proposed dynamic sequencing logic considers the completion of pools. Still, it is expected that the proposed logic will outperform the base case. However, it is expected that benefits from using the proposed logic as opposed to the current logic are smaller as there are both more jobs as well as more operators active in busier periods. Then, for example under the current logic more for multi pools, it is more likely that some job will complete a multi pool and the added value of choosing which job completes a multi pool is lower compared to when there is fewer operators or less demand. Combining the above, the following combined hypothesis is constructed:

- ***Sensitivity hypothesis S2:*** regardless of the simulation period, the alternative case outperforms the base case simulation in terms of throughput time, waiting time, pool completion time and workload balancing but effects are smaller in the busier period.

Figure G1 plots the number of jobs per day for the Black Friday week. The average number of jobs per day is 11506 as opposed to 5175 for the October data. The average daily Multi/Mono ratio is 0.64 as opposed to 0.51 for October.

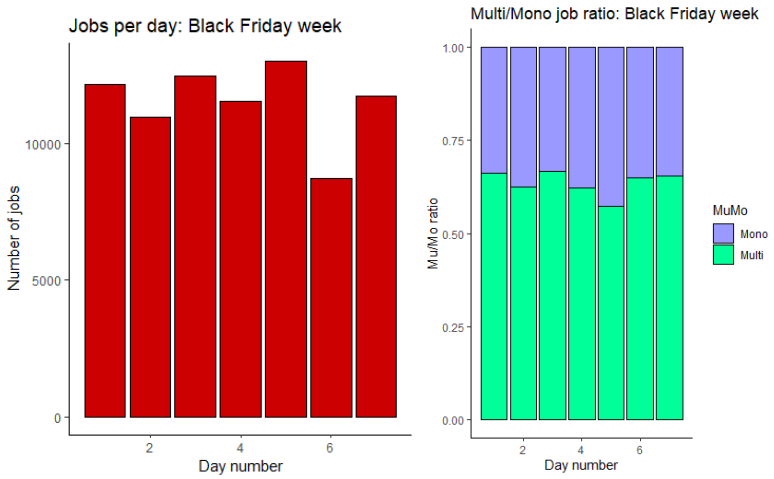


Figure G1: November (1 wk.) job descriptives: jobs per day and multi mono ratio.

Similar scenarios as for the base results are tested. More specifically the following scenarios are used:

- **Scenario S2:** where the true historical processing times are known (analogous to scenario 1);
- **Scenario S2a:** where processing times are estimated using historical averages (analogous to scenario 2a; and
- **Scenario S2b:** where regression equations as presented in Appendix D are used (analogous to scenario 2b).

Table G1 presents the results for throughput time, PCT and utilization variation for the above-mentioned scenarios for 1 week of November as the simulation period. Results for throughput, waiting, and pool completion time for all scenarios were found by t-tests to be significantly different from the base case results.

Table G1: Sensitivity results: simulation period.

	Throughput Time [in mins]			Waiting Time [in mins]			Pool Completion Time [in mins]		Utilization	
	Mo	Mu	Mo/Mu	Mo	Mu	Mo/Mu	Mu		Shift	
	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\overline{\Delta MSE}$	%Shifts better
Base	56.5 (34.9)	44.6 (16.1)	-	27.9 (30.4)	8.6 (11.9)	-	13.8 (4.6)	-	-	-
S2	53.5 (31.7)	42.1 (14.8)	5.3/5.6	24.8 (27.0)	7.3 (10.5)	11.1/15.1	13.3 (4.8)	3.6	137.3	71
S2a	53.1 (31.3)	41.6 (14.6)	6.0/6.7	24.4 (26.6)	7.0 (10.2)	12.5/18.6	13.3 (4.8)	3.6	130.6	50
S2b	52.7 (31.3)	41.4 (14.5)	6.7/7.2	24.6 (26.6)	6.9 (10.2)	11.8/19.8	13.3 (4.9)	3.6	140.1	64

Note. $\bar{X}$ (SD) refers the mean (standard deviation) values. * refers to the percentual reduction in $\bar{X}$ for the specified scenario compared to the base case. Mono jobs and multi pools are abbreviated to Mo and Mu respectively where Mu refers to the 75th percentile in terms of pool completion (or throughput) time for multi pools (see Appendix F for further details). Mo/Mu specifies that reductions for Mo and Mu are considered. %Shifts better specifies the percentage of shifts in which the specified scenario outperforms the base case in terms of MSE_s.

Firstly, it can be observed from the base case that both mean, and variation of throughput times are higher compared to the October analyses for both mono jobs and multi pools. This follows from the fact that more pressure is put on the work centers due to higher demand in this period (average daily number of jobs is more than doubled) with waiting times as a result. Picking pool completion time on the other hand, on average, is similar to the October base case.

Throughput time reductions can be observed for mono jobs and multi pools regardless for all three scenarios (i.e., regardless of the estimation method for expected processing times).This suggests that the results for throughput time reduction are not very sensitive to the simulation period. However, compared to the October data, throughput time reductions for multi pools are slightly smaller in this Black Friday week. Mono throughput time reductions are slightly higher.

Pool completion time is reduced in all three scenarios compared as well to the base case. However, the results suggest that for this busy November week, PCT is unaffected by the estimation method; average pool completion time is reduced by 3.6 percent in all three scenarios. Compared to the PCT reductions found in October (7-8 percent) this is a smaller reduction. The smaller reduction is explained by the increase Multi/Mono ratio: given that there are relatively more multi jobs compared to mono jobs, the likelihood of a pick operator being allocated to a multi job is higher and hence pools are picked sooner. As a result, the added value of the dynamic sequencing logic as proposed in this thesis seems smaller when it comes to pool completion time for higher Multi/Mono ratios. This also

explains the smaller throughput time reductions for multi pools.

Utilization variation is lower compared to the base case for all three scenarios. Thus, although more pressure is put on the work centers, the proposed sequencing logic still outperforms the current sequencing logic in terms of workload balancing.

Summarizing, when it comes to sensitivity of the results to the simulation period: throughput time reductions are robust to the simulation period; the added value pool completion time reduction is smaller compared to the October data; and a better balance exists in terms of utilization variation among the work centers per shift.

Appendix H Hypotheses

In this appendix, the hypotheses that were developed in section 5.4 and 5.5 are summarized. Per hypothesis it is stated whether it is supported, partially supported, or not supported by the results.

- ***Hypothesis 1:*** mean throughput and waiting time are lower in the alternative case versus the base case. Supported.
- ***Hypothesis 2:*** mean pool completion time is lower in the alternative case versus the base case. Supported.
- ***Hypothesis 3:*** variation among the utilizations of the outbound work centers is lower in the alternative case compared to the base case. Supported.
- ***Hypothesis 4:*** larger throughput and waiting time reductions are obtained through better workload balancing. Partially supported.
- ***Hypothesis 5:*** results for scenario 1 are better than results for scenario 2a and 2b, where 2b is expected to outperform scenario 2a. Not supported.
- ***Sensitivity hypothesis S1:*** regardless of over- or underestimating processing times, the alternative case outperforms the base case simulation in terms of throughput time, waiting time, pool completion time and workload balancing. Supported.
- ***Sensitivity hypothesis S2:*** regardless of the simulation period, the alternative case outperforms the base case simulation in terms of throughput time, waiting time, pool completion time and workload balancing but effects are smaller in the busier period. Partially supported.
- ***Sensitivity hypothesis S3:*** regardless of the value for the safety parameter, the alternative case outperforms the base case simulation in terms of throughput time, waiting time, pool completion time and workload balancing. Supported.
- ***Sensitivity hypothesis S4:*** lower safety parameter values yield better results compared to higher parameter values. Partially supported.

Appendix I Results summary

Table I1 describes the various (sensitivity) scenarios. Table I2 summarizes the results of all (sensitivity) scenarios while also including standard deviations.

Table I1: Description of scenarios

Scenario	Period	Description
1	4 wks. Oct	Expected processing times are equal to the true historical processing times
2a	4 wks. Oct	Expected processing times estimated by historical averages (Appendix C)
2b	4 wks. Oct	Expected processing times estimated by regression equations (Appendix D)
S1a	4 wks. Oct	Underestimation of processing times (-20%)
S1b	4 wks. Oct	Overestimation of processing times (+20%)
S3a	4 wks. Oct	Analogous to scenario 2b, but configurable safety parameter set to 30
S3b	4 wks. Oct	Analogous to scenario 2b, but configurable safety parameter set to 0
S2	1 wk. Nov	Apart from period, analogous to scenario 1
S2a	1 wk. Nov	Apart from period, analogous to scenario 2a
S2b	1 wk. Nov	Apart from period, analogous to scenario 2b

Table I2: Results summary: primary and sensitivity

	Value	Throughput Time [in mins]			Waiting Time [in mins]			Pool Completion Time [in mins]		Utilization	
		Mo	Mu	Mo/Mu	Mo	Mu	Mo/Mu	Mu		Shift	
		$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\bar{X}$ (SD)	$\bar{X}$ %↓*	$\overline{\Delta MSE}$	%Shifts better
Base Oct	-	46.2 (30.6)	40.1 (15.0)	-	21.7 (27.1)	8.3 (11.3)	-	13.6 (5.2)	-	-	-
1	15	43.7 (26.7)	36.9 (13.9)	5.4/8.0	19.2 (22.8)	7.1 (9.7)	11.5/14.5	12.6 (5.0)	7.4	186.3	56
2a	15	44.1 (27.6)	36.9 (13.8)	4.5/8.0	19.6 (23.9)	7.1 (9.9)	9.7/14.5	12.7 (5.3)	6.6	168.7	60
2b	15	43.7 (27.4)	36.7 (13.8)	5.4/8.5	19.3 (23.6)	6.9 (9.8)	11.1/16.9	12.5 (5.1)	8.1	135.6	63
S1a	15	44.8 (27.1)	37.8 (14.6)	3.0/5.7	20.3 (23.4)	7.7 (10.6)	6.5/7.2	12.7 (5.2)	6.6	203.5	67
S1b	15	43.2 (27.1)	37.0 (13.9)	6.5/7.7	18.8 (23.2)	7.1 (9.8)	13.4/14.5	12.5 (4.9)	8.1	197.3	63
S3a	30	43.9 (27.5)	36.8 (13.7)	5.0/8.2	19.4 (23.7)	7.0 (9.8)	10.6/15.7	12.6 (5.1)	7.4	185.5	63
S3b	0	43.7 (27.4)	36.7 (13.7)	5.4/8.5	19.3 (23.6)	7.0 (9.8)	11.1/15.7	12.5 (5.1)	8.1	155.0	63
Base Nov	15	56.5 (34.9)	44.6 (16.1)	-	27.9 (30.4)	8.6 (11.9)	-	13.8 (4.6)	-	-	-
S2	15	53.5 (31.7)	42.1 (14.8)	5.3/5.6	24.8 (27.0)	7.3 (10.5)	11.1/15.1	13.3 (4.8)	3.6	137.3	71
S2a	15	53.1 (31.3)	41.6 (14.6)	6.0/6.7	24.4 (26.6)	7.0 (10.2)	12.5/18.6	13.3 (4.8)	3.6	130.6	50
S2b	15	52.7 (31.3)	41.4 (14.5)	6.7/7.2	24.6 (26.6)	6.9 (10.2)	11.8/19.8	13.3 (4.9)	3.6	140.1	64

Note. $\bar{X}$ (SD) refers the mean (standard deviation) values. * refers to the percentual reduction in $\bar{X}$ for the specified scenario compared to the base case. Mono jobs and multi pools are abbreviated to Mo and Mu respectively where Mu refers to the 75th percentile in terms of pool completion (or throughput) time for multi pools. Mo/Mu specifies that reductions for Mo and Mu are considered. %Shifts better specifies the percentage of shifts in which the specified scenario outperforms the base case in terms of MSE_s. The scenarios in the upper part of the table should be interpreted relative to the base October scenario whereas the lower part of the table should be interpreted to the base case for 1-week November data.