Axel Vislie Mikkelsen
August Heiervang Dahl

Cloud Material Handling System

Leveraging dynamic dispatching and reinforcement learning in a cloud-enabled shop floor material handling system

Master's thesis in Engineering and ICT
Supervisor: Fabio Sgarbossa
Co-supervisor: Mirco Peron
June 2021

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Axel Vislie Mikkelsen
August Heiervang Dahl

Cloud Material Handling System

Leveraging dynamic dispatching and reinforcement learning in a cloud-enabled shop floor material handling system

Master's thesis in Engineering and ICT
Supervisor: Fabio Sgarbossa
Co-supervisor: Mirco Peron
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Preface

This thesis was carried out during the spring of 2021 at the Department of Mechanical and Industrial Engineering. It concludes our Master of Science in Engineering and ICT at the Norwegian University of Science and Technology.

We want to express our gratitude towards our supervisors, Fabio Sgarbossa and Mirco Peron, for valuable guidance on manufacturing concepts and continuous motivation for improvement during the work with this thesis. We would also like to thank Giuseppe Fragapane for his honest feedback and input regarding the thesis structure and content. At last, a special thanks to the Pathmind support team for helping us understand the underlying dynamics of the platform and the construction of reinforcement learning policies.

Trondheim, June 2021

Axel V. Mikkelsen

Axel Vislie Mikkelsen

August H. Dahl

August Heiervang Dahl

Abstract

Efficient handling of materials and products on manufacturing shop floors is essential to reduce production costs and improve productivity. Although the scientific community has embraced automated material-handling equipment in the wake of Industry 4.0, human-operated vehicles like forklifts and pallet trucks are still the most commonly used equipment for material handling. This thesis investigates how a cloud-enabled shop floor can facilitate dynamic dispatching by automating human and autonomously operated material-handling equipment through a centralized system, coined as the Cloud Material Handling System (CMHS).

The main objective of this study is to determine how a CMHS may improve material handling activities in manufacturing. Specifically, the study evaluates a CMHS in different scenarios to support when it is particularly beneficial in material handling operations. Multiple dispatching methods like heuristic dispatching rules and reinforcement learning policies are evaluated to support how a CMHS can be implemented. A literature study was conducted to disclose research gaps addressed by a CMHS, while a simulation model based on a case study was developed to demonstrate its use in practice.

The results have shown the CMHS's ability to achieve higher productivity in product throughput and equipment utilization than the conventional non-automated benchmark. Performance increases were observed in all scenarios, while the number of required material-handling equipment was reduced by 40%.

The simulation results revealed that the CMHS with reinforcement learning is particularly beneficial for uncertain product arrival rates and workstation failures when product loads were kept in line relative to production capacity. Most prominent were moderate product loads, resulting in a 197% gain in total product throughput. The lower-complexity heuristic methods were on a par, or superior, to the reinforcement learning policy for predictable material flows with high arrival rates.

Further evaluation of the CMHS should be done in collaboration with a practical business case to extract key operation parameters, reducing the number of assumptions, and develop a rigorous economic model for the CMHS.

Sammendrag

Effektiv håndtering av materialer og produkter på produksjonsgulv er viktig for å redusere produksjonskostnadene og forbedre produktiviteten. Selv om det vitenskapelige samfunnet har omfavnet automatisert materialhåndteringsutstyr i kjølvannet av Industry 4.0, er menneskedrevne kjøretøy som gaffeltrucker og palletrucker fortsatt det mest brukte utstyret for materialhåndtering. Denne masteroppgaven undersøker hvordan et skyaktivert produksjonsgulv kan utnytte dynamisk utsendelse ved å automatisere både menneskelige og autonomt betjente materialhåndteringsutstyr, kalt Cloud Material Handling System (CMHS).

Hovedmålet med denne studien er å bestemme hvordan en CMHS kan forbedre materialhåndteringsaktivitetene i produksjonen. Spesielt evaluerer studien en CMHS i forskjellige scenarier for å undersøke når den er spesielt gunstig i materialhåndteringsoperasjoner. Flere utsendelsesmetoder som etablerte heuristikker og læringsmetoder med forsterkningslæring blir evaluert for å undersøke hvordan en CMHS kan implementeres. En litteraturstudie ble utført for å avsløre forskningshull adressert av en CMHS, mens en simuleringsmodell basert på et casestudie ble utviklet for å demonstrere bruken i praksis.

Resultatene har vist CMHSs evne til å oppnå høyere produktivitet når det gjelder gjennomstrømning av produkter og utstyrsutnyttelse enn den konvensjonelle ikke-automatiserte referansen. Ytelsesøkninger ble observert i alle scenarier, mens antall nødvendige materialhåndteringsutstyr ble redusert med 40 %.

Simuleringsresultatene avslørte at CMHS med forsterkningslæring er spesielt gunstig for usikre produktankomster og arbeidsstasjonsfeil når produktbelastninger ble holdt på linje med arbeidsstasjonenes produksjonskapasitet. Mest fremtredende var under normale produktbelastninger, noe som resulterte i en 197 % forbedring i total produktgjennomstrømning. De heuristiske metodene med lavere kompleksitet var på nivå eller bedre enn metodene med forsterkningslæring for forutsigbare materialstrømmer med høy ankomstrate.

Videre evaluering av CMHS bør gjøres i samarbeid med en praktisk business case for å trekke ut viktige driftsparametere, redusere antall antagelser og utvikle en nøyaktig kostnadsmodell for CMHS.

Abbreviations

CMHS	Cloud Material Handling System
CM	Cloud Manufacturing
CPS	Cyber-Physical System
AGV	Automated Guided Vehicle
AMR	Autonomous Mobile Robot
IoT	Internet of Things
IPS	Indoor Positioning System
IPT	Indoor Positioning Technology
MH	Material handling
MHE	Material Handling Equipment
MS	Machine Scheduling
RFID	Radio Frequency Identification
KPI	Key Performance Indicator
ML	Machine Learning
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
PBT	Population-Based Training
PPO	Proximal Policy Optimization
HPO	Hyperparameter Optimization
STD	Shortest Travel Distance
LWT	Longest Waiting Time
CP	Centralized Positioning
NIL	Nearest Idle Location

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With an increasing demand for high-variety and low-volume products (Telgen et al. 2014), leveraging Industry 4.0 (I4.0) to increase flexibility is essential to maintain competitive advantages for manufacturing companies (Centobelli et al. 2016). The emerging and rapidly evolving field of I4.0 has given rise to several new technologies in the world of manufacturing, such as the Internet of Things (IoT), Cyber-Physical Systems (CPS), cloud computing, and big data analytics (Liu and X. Xu 2017). Since the term was coined in 2010, the rapid growth and availability of cloud technologies have spawned a new concept called Cloud Manufacturing (CM; Li et al. 2010).

CM offers rapidly provisioned on-demand network access to a centralized pool of manufacturing resources with minimal management effort (X. Xu 2012). As a result, it can be used together with other I4.0 technologies to reduce establishment costs, scale resources quickly as data volumes grow, and increase manufacturing flexibility (Liu and X. Xu 2017). This recent technological development has encouraged the manufacturing sector to seek out new areas of improvement, emphasizing dynamic decision-making, increased productivity, and reduced costs.

The cloud provides a reliable real-time service over the Internet that facilitates dynamic decision-making and control in the manufacturing shop floor (Yue et al. 2015). Its performance is closely linked to how sensors, chips, and other IoT devices track and trace the physical environment.

However, the implementation of tracking components in today's manufacturing companies is still in its infancy. The report "Digital Factories 2020 – Shaping the future of manufacturing" (PWC-Deutchland 2017) carried out a quantitative survey on over 200 manufacturing companies in Germany, and they found that only 29% of respondents said they had implemented tracking components within their production process. Concurrently, more than twice as many (60%) said they expected to do so by 2022.

The report also demonstrated successful implementations. For instance, Bosch Rexroth uses RFID to track components within a manufacturing cell and machine sensors to perform scheduling and predictive maintenance. Fujitsu's Augsburg factory utilizes cloud-based services, sensors, and RFID tags to achieve fast production to exact customer specifications. Continental Automotive developed a production line with automated scheduling connected to a warehousing system operated by Automated Guided Vehicles (AGVs) and can respond dynamically to changes in product volumes or variances. In short, tracking of manufacturing resources and equipment in the age of I4.0 and CM has had the most significant impact on production scheduling operations as of 2021 (see Section 4.1).

Although manufacturing companies do not use tracking to a great extent today, the CM concept has received much attention in the scientific community. The papers discussing cloud technology and tracking in the manufacturing shop floor define CM as a cloud or centralized connectivity system that interconnects a network of manufacturing resources and equipment to some extent. In the context of CM, production scheduling has been particularly recognized by researchers (Mourtzis et al. 2018; Wan et al. 2017; Kumar et al. 2019; Pakpahan et al. 2018). However, due to the novelty of CM, most papers examine the problem from an architectural perspective with device shop floor configurations, disregarding material handling.

Material handling can be defined as *"(...) the movement, storage, protection, and control of materials throughout the manufacturing and distribution process (including their consumption and disposal)"* (Sgarbossa et al. 2020, p. 88). Material handling activities generally account for 30 to 40% of production costs (Onut et al. 2009). Thus, efficient handling of materials and products on the shop floor is vital to operate a responsive and adaptable production environment (Zangaro et al. 2019).

Literature targeting material handling has almost exclusively focused on automated material-handling equipment (MHE) like Automated Guided Vehicles (AGVs; De Ryck et al. 2020) and Autonomous Mobile Robots (AMRs; Fragapane et al. 2020). As demonstrated in figure 1.1, the AGV is restricted to a fixed path, whereas the

AMR has built-in obstacle avoidance allowing for more flexible movement.

Implementations of automated and decentralized solutions in material handling—particularly with AGVs—have become increasingly popular in the CM context for material handling purposes (De Ryck et al. 2020). Although an abundance of algorithms and control methods have been researched, AGVs are not necessarily applicable for all manufacturing companies (Fragapane et al. 2020). Choosing the appropriate MHE type is a complex decision-making problem as both quantitative (e.g., load capacity, cost, and energy consumption) and qualitative (e.g., flexibility, reliability, and safety) measures need to be taken into consideration (Hellmann et al. 2019).

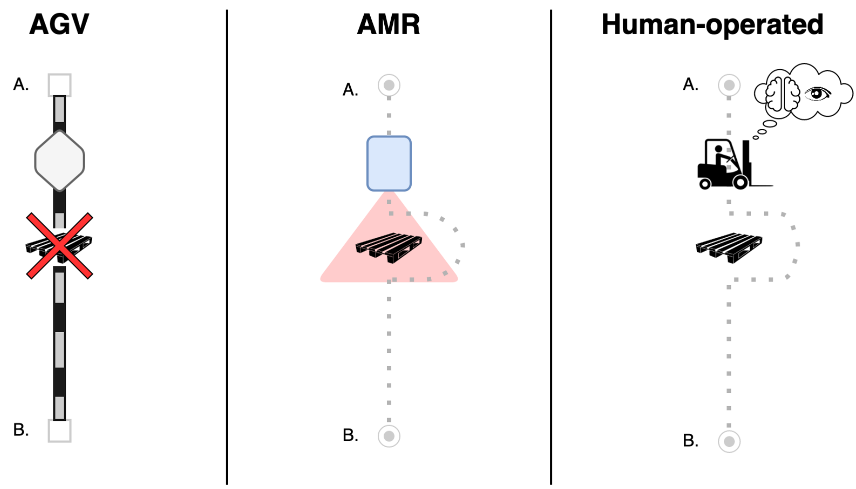

Figure 1.1: Obstacle avoidance comparison between AGVs, AMRs and human-operated MHE

Current literature on material handling seems to reflect that most manufacturing companies use AGVs, AMRs, or other automated (non-human operated) material handling equipment to fulfill their material handling needs. However, the positive outlook on automated MHE in literature has not yet been transferred into practice. The report "Industrial mobility – How autonomous vehicles can change manufacturing" (PWC-US 2018) states that, from the 128 large and mid-size US manufacturers surveyed, only 9% have adopted some type of semi-autonomous or autonomous mobility within their operation. Additionally, warehousing and inventory management is seeing the most prominent growth in automation, whereas the shop floor has seen limited development.

In the shop floor environment, human-operated vehicles like forklifts and pallet trucks are still the most common MHE and are often deemed sufficient for the factory's material handling needs (Dukic et al. 2018). Human-operated MHE is equally capable of avoiding obstacles as AMRs (Figure 1.1), but because such MHE is challenging to automate, research on this topic is underrepresented in literature. This split between the research community's view on automation in theory and its true manifestation in material handling practices reveals a significant research gap.

While non-automated material handling systems can be adequate in some cases, they usually struggle to adapt to dynamic and unpredictable demands in real-time, often leading to low MHE utilization and poor manufacturing performance (Tompkins et al. 2010). Material handling in practice is directly linked to the processes it needs to support, and the permutations of the system's configuration are different for each shop floor. Thus, the conventional non-automated material handling system is difficult to define rigorously. However, as a general rule, it can be classified as human-operated solutions with ad-hoc methods based on trial-and-error (Desrosiers et al. 1995) where human physical interference is commonplace (Sgarbossa et al. 2020). By addressing this issue, there is an evident potential to increase the utilization of MHE, resulting in reduced production costs and increased profits. For manufacturing companies looking to optimize their shop floor operation, elaborate production optimizations might not be value-adding if their material handling system performs under par.

The lack of attention towards human-operated solutions reveals a significant gap between scientific research and the technology utilized in actual manufacturing shop floors. Additionally, manufacturing companies often use AGVs and human-operated MHE in combination (Saputro et al. 2015), demonstrating the need for control methods incorporating several MHE simultaneously. To bridge this gap, the Logistics 4.0 lab at the Norwegian University of Science and Technology proposes a new paradigm to apply cloud services on material-handling equipment, namely a Cloud Material Handling System (CMHS; Sgarbossa et al. 2020).

1.2 CMHS concept

A CMHS implementation can automate human-operated MHE like forklifts and pallet trucks, offering companies an automation alternative without switching out their existing MHE fleet for expensive AGVs or AMRs. Shop floors that require multiple MHE types will also automate their material handling as the cloud engine—the CMHS component responsible for MHE dispatching and job allocation—will differentiate and allocate tasks according to the MHE type required for the job. Thus, from a cost and practical perspective, a company with automation ambitions will benefit from the flexibility of a CMHS.

A CMHS aims to *"satisfy consumers' requests through the available resources in a cloud environment, reducing the complexity of a multilevel hierarchical control system and increasing the manufacturing system's overall flexibility and productivity"* (Sgarbossa et al. 2020, p. 89). The concept combines the CM paradigm with an Indoor Positioning System (IPS), enabling real-time positional MHE data capture from the shop floor and making it available in the cloud to facilitate dynamic dispatching at a reasonable cost. The same way GPS revolutionized travel in the outdoor environment and enabled companies like Uber to find ways to optimize their driver fleet, a CMHS aims to bridge the same gap for the stochastic manufacturing shop floor environment. Compared to established tracking technologies like RFID, where physical scanning is necessary to log position data of MHE, real-time tracking with an IPS offers the high data frequency and precision level needed to provide an Uber-like service on the shop floor.

Figure 1.2 outlines how the MHE position data is collected and processed to allocate jobs according to the active dispatching policy or rule. It is based on the Logistics 4.0 Lab's work at NTNU and the paper Sgarbossa et al. 2020 where the IPS is also used to track materials, pallets, and boxes (called smart objects) at a conceptual level. However, as the location and relevant data points from these objects can be sufficiently tracked and relayed to the cloud engine without an IPS (but with workstation sensors and RFID), this thesis will only focus on MHE tracking to avoid unnecessary complexity.

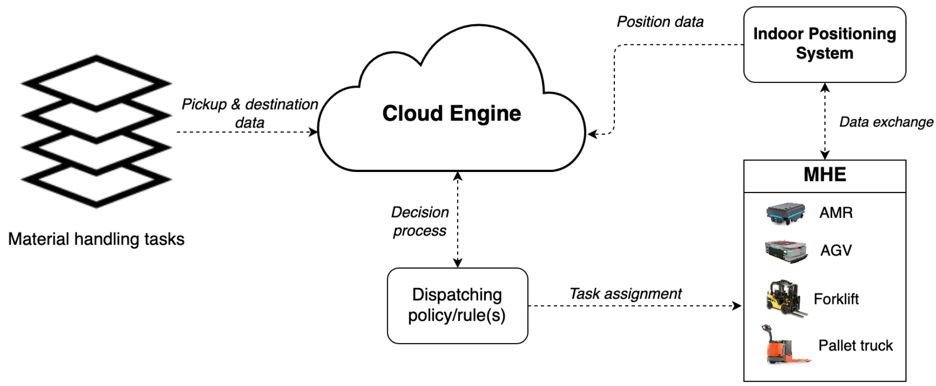

Figure 1.2: Flow of data in and out of the cloud engine

The shop floor is inherently stochastic, and the ability to react quickly to unexpected events and revise plans in a cost-efficient way is essential for manufacturing efficiency (Herrmann 2004). As the number of daily material-handling tasks grows, the importance of a flexible material-handling system increases accordingly. In order to demonstrate the potential of a CMHS's capabilities, the system is compared with conventional non-automated material handling methods. Figure 1.3 is meant to depict such an environment, and although it is not graphically representative for all shop floor environments, it demonstrates the material handling logic. In the conventional material handling case, MHE operators are assigned to predetermined areas/workstation clusters, relying on visual contact to identify and perform new tasks.

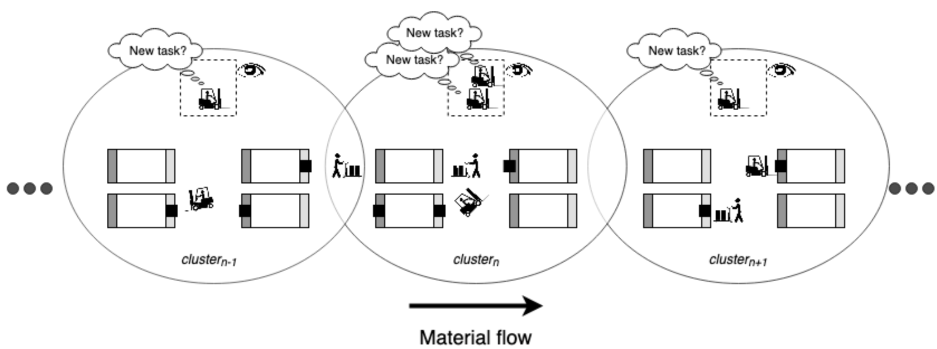

Figure 1.3: Conventional material handling case

The Manufacturing-Uber concept proposed in Greis et al. 2019 allowed machine operators to move unrestricted in workstation cells by scheduling jobs dynamically, reducing machine idle time by increasing their freedom of movement. The same logic applies to MHE controlled by a CMHS as shown in Figure 1.4. Consequently, the visual contact requirement is not present in this case.

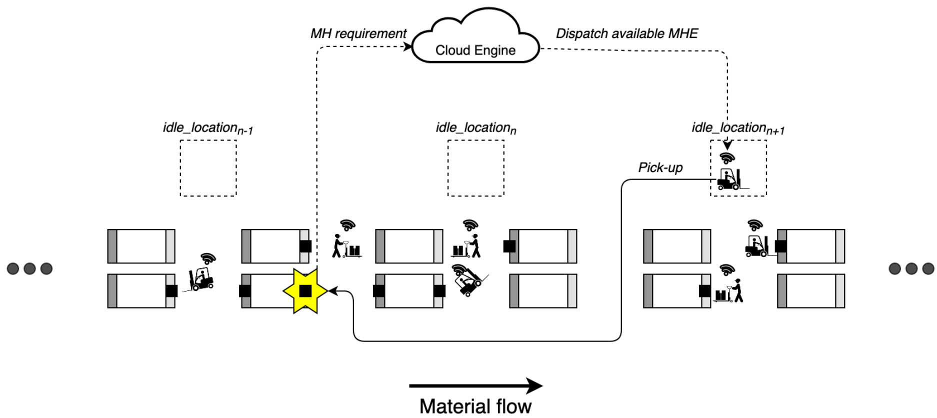

Figure 1.4: Material handling with a CMHS

A vital feature of a CMHS is the ability to enable dynamic dispatching of MHE. Upon completing a job, the MHE will be recognized as unassigned by the cloud engine and immediately be considered for a new one. Dynamic dispatching can boost MHE utilization and manufacturing efficiency, but the policies and rules governing this process are the main drivers behind whether a CMHS implementation is advantageous. Additionally, where the MHE idles may significantly impact the distance traveled to execute a material handling task. Thus, investigating optimal dispatching and idle policies/rules with a CMHS implementation will be one of the main focus areas of this thesis.

1.3 Problem description

A CMHS leverages the interconnection of cloud services, IPS, and dispatching to coordinate material handling activities. These topics have been researched individually, but there has not been any research carried out exploring them in a combined system. In order to determine a CMHS's value-adding properties, there is a need to create a roadmap that accurately demonstrates its potential capabilities in different scenarios for shop floor material handling.

This roadmap will give companies with different needs, objectives, and degrees of expertise the ability to determine if a CMHS is applicable in their operation and, if so, how advanced such a setup should be to improve their overall efficiency. The problem statement can be divided into two parts:

- Should the company implement a Cloud Material Handling System?
- If so, how sophisticated does the logic need to be to obtain satisfactory results?

The first item is concerned with whether or not a CMHS and its inherent automation capabilities of MHE can improve a material handling operation compared to conventional non-automated methods. Upon this foundation, the natural progression is to investigate what degree and in what situations a more sophisticated dispatching logic is beneficial compared to more straightforward approaches. By comparing different complexity levels of dispatching logic to conventional non-automated shop floor scenarios, the goal is to give potential users of a CMHS the tools required to make an educated decision on whether to adopt it or not.

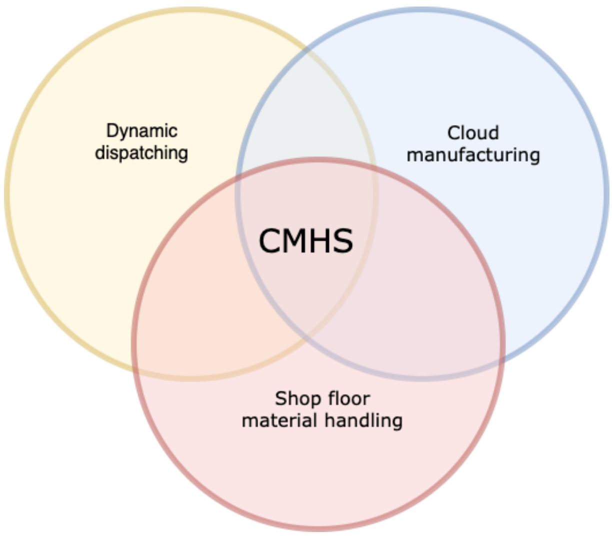

Figure 1.5: Thesis scope illustrated in a venn diagram

As depicted in figure 1.5, this thesis' scope is centered towards three pillars included in a CMHS: dynamic dispatching, cloud manufacturing, and shop floor material handling. Dynamic dispatching is the focal point of the theoretical background, while the literature study will focus on cloud manufacturing and shop floor material handling.

A key aspect to consider when optimizing material handling is to take into account the number of trips taken by the MHE on average each day. If the number is too low, optimization with a CMHS will not make any intuitive sense. As a result, a prerequisite for potential candidates for a CMHS is that the shop floor environment has a considerable level of material flow, where planning each material handling job in advance is impractical due to uncertainties like arrival rates, processing times, and workstation failures.

The warehousing research field is outside the scope of this thesis. Although the warehouse is acting as a supplier and storage unit for the manufacturing process, the shop floor environment often contains more complex material flows due to interdependent processes with a high degree of unpredictability. Hence, an analysis concerning shop floor manufacturing is more applicable for initial research of a CMHS's capabilities.

As this thesis investigates manufacturing from a material handling perspective, the scope is not restricted to a particular production strategy like Make-to-Order, Assemble-to-Order, Engineer-to-Order, or Make-to-Stock. Material handling has been regarded as a process to minimize in traditional manufacturing, so applying a specific production strategy is irrelevant in this context. Rather than applying the CMHS to one particular production environment, this thesis aims to provide guidelines from a purely material handling perspective, allowing each manufacturing company to consider a CMHS for their operation.

1.4 Research objectives

There is a need to develop methods to support when and how to apply a CMHS in manufacturing operations. A simulation model with different scenarios concerning material flow variability and workstation failure is evaluated to decide *when* CMHS is particularly beneficial. The scenarios aim to depict stochastic occurrences on the shop floor to test the flexibility of different material handling methods. In order to determine *how* a CMHS can be applied, methods like traditional dispatching rules and policy generation through reinforcement learning are further explored in the model to demonstrate the performance of a CMHS. The basis of comparison will measure product throughput and MHE utilization across the shop floor.

The analysis will serve as a foundation to develop general guidelines for businesses to decide whether to implement a CMHS. Finally, the thesis proposes a generalized economic model as a first step to assess CMHS's profitability. The research questions are:

RQ1 *How can a CMHS improve the material handling activities in manufacturing in terms of flexibility and productivity performance?*

RQ2 *In what scenarios should a CMHS be applied in manufacturing shop floors compared to traditional dispatching approaches?*

RQ3 *When should reinforcement learning support the decision-making process in a CMHS for dispatching the material handling activities?*

A literature study on the use of CM in shop floor material handling is conducted to disclose research gaps addressed by a CMHS to answer RQ1. Furthermore, a simulation model portraying a stochastic shop floor environment will be used to answer RQ2 and RQ3. Notably, the model focuses its attention on MHE job dispatching methods. The exploration steps can be grouped into seven categories to outline the objectives. Table 1.1 and Figure 1.6 show what, why, and which method is used to answer each research question.

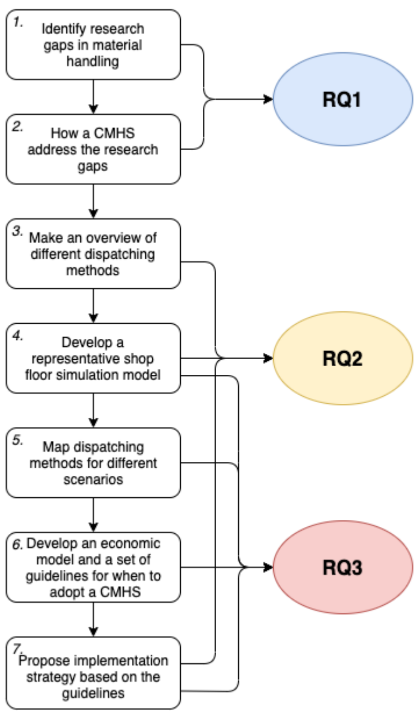

Figure 1.6: A graphical representation of the thesis objectives listed in Table 1.1

What	Why	Method	Related RQ
1. Map CM influence on material handling in manufacturing shop floors	Identify gaps between current literature and real facilities	Literature study	RQ1
2. Develop an understanding of a CMHS and its capabilities	Explain why a CMHS address research gaps on a conceptual level	Literature study	RQ1
3. Explain different dispatching methods and argue their use in a CMHS	Construct an overview of different ways to dispatch MHE in a CMHS	Theoretical background	RQ2/RQ3
4. Develop a shop floor layout and implement it in a simulation model	Obtain a CMHS implementation that demonstrates its potential	Simulation	RQ2/RQ3
5. Compare dispatching rules and DRL in a CMHS with conventional methods for different levels of system variability	Manifest the impact of a CMHS from a practical perspective	Simulation	RQ3
6. Develop general guidelines and an economic model for businesses to decide whether or not to apply a CMHS	Provide advice for CMHS's role in a material handling operation	Discussion / Conclusion	RQ2
7. Propose an implementation strategy	Describe when sophisticated logic like reinforcement learning should be applied, and when easier methods are sufficient	Discussion / Conclusion	RQ2/RQ3

Table 1.1: A systematic representation of how the research questions will be answered

1.5 Thesis structure

The thesis is structured as follows. Chapter 2 presents the theoretical background on dispatching in material handling. Chapter 3 explains why the scientific methods are chosen and how they are used. Moreover, the chapter gives an overview of the model implementation. Chapter 4 outlines the literature study findings and relates the current gaps to the capabilities of a CMHS (RQ1). Chapter 5 presents and analyzes the experimental results from the simulation model (RQ2 and RQ3). Chapter 6 discuss the results concerning the research questions before Chapter 7 provides a conclusion.

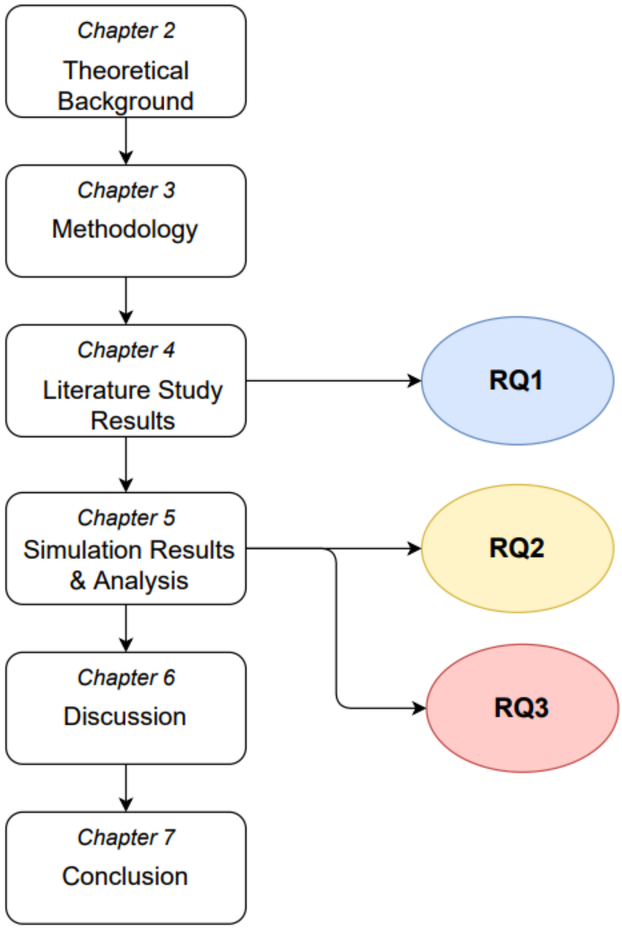

Figure 1.7: Thesis structure overview

Chapter 2

Theoretical Background

The objective of this chapter is to establish a theoretical background for material handling dispatching. First, traditional dispatching methods are outlined. Then, the chapter centers its attention on dynamic dispatching and machine learning, with a particular emphasis on reinforcement learning. Finally, critical components of deep reinforcement learning are presented and used to argue why deep reinforcement learning is appropriate in a CMHS.

2.1 Dispatching in material handling

Several factors influence the performance of a material handling system, most notably the shop floor configuration/layout, the MHE fleet size, and how they are dispatched (Le-Anh et al. 2005). However, a well-developed shop floor layout with an incompatible control method might lead to decreased system performance (Vis 2006). The most common way of controlling the MHE dispatching operation is by using a centralized controller that manages the fleet simultaneously (De Ryck et al. 2020). The shop floor environment is inherently stochastic, and exact information on what should go where and when is seldom known in advance. Planning in this environment becomes more difficult as the complexity and number of material handling jobs increases, facilitating the need for flexible real-time control to optimize shop floor efficiency (Mařık et al. 2007).

A well-established control method in the shop floor environment is the use of dispatching rules. They have been used extensively in academic research on machine scheduling and AGV dispatching (Sabuncuoglu 1998), but as material handling is the main focus of this thesis, machine scheduling is disregarded. Dispatching rules can be split into two categories: Workstation-initiated and MHE-initiated dispatching rules (Vis 2006).

1. Workstation-initiated: A job claims an available MHE according to the dispatch rule
2. MHE-initiated: An available MHE claims a job according to the dispatch rule

The static dispatching rule controls the MHE movements based on intuitive reasoning to achieve good performance (Le-Anh et al. 2005). No best rule applies to all cases, but an appropriate rule can be found for the specific shop floor layout and material handling requirement. Table 2.1 is a compounded list of the most common rules found in academic literature (Vis 2006; Le-Anh et al. 2005; Ho et al. 2006; Sabuncuoglu 1998).

Name	Abbreviation	Rule category
Shortest travel distance	STD	Both
Longest Idle Vehicle	LIV	Workstation-initiated
Least Utilized Vehicle	LUV	Workstation-initiated
Longest Waiting Time	LWT	Both
Greatest Queue Length	GQL	Both

Table 2.1: Static dispatching rules

- *STD* – Dispatching MHE with shortest travel distance to job
- *LIV* – Longest idling vehicle/MHE is dispatched
- *LUV* – Least utilized vehicle/MHE is dispatched
- *LWT* - Dispatching MHE to job that has waited the longest
- *GQL* – Dispatching MHE to job with the biggest queue

Another important aspect of dispatching is locating idle MHE to react as efficiently as possible to a new assignment. The return policy or rule can have a significant effect on response time and is sub-optimal if the MHE fleet has to travel unnecessarily large distances without a load (Vis 2006). Minimizing average fleet response time can lead to a more even distribution of idle MHE on the shop floor. Static rules like *centralized positioning* (CP) and *nearest idle location* (NIL) were proposed in Egbelu 1993 as the most common idle rules in shop floors. Thus, no other rules will be explored going forward.

Although static dispatching rules are simple, offer low computation costs, and are easy to implement in practice, the primary issue for the rules outlined above is that the performance depends heavily on the system state (Le-Anh et al. 2005). No definitive rule supports all possible states of a manufacturing environment, as manufacturing facilities often operate in dynamic environments with unavoidable, unpredictable real-time events (Priore, Gómez, et al. 2014). Unforeseen variations in arrival rates, processing delays, machine failure, and maintenance may cause deviations from the original plan and lead to delays.

2.1.1 Dynamic dispatching

Successful implementation of real-world dispatching systems relies on dispatching in the presence of real-time events, known as dynamic scheduling (Ouelhadj et al. 2009). In the context of shop floor manufacturing, where MHE is responsible for transporting materials across the facility, it is more convenient to address scheduling as dispatching of MHE. Scheduling is an ambiguous term as it may be confused with predetermined planning. Hence, dynamic scheduling is referred to as *dynamic dispatching* going forward.

Due to the NP-hard nature of dispatching problems in complex manufacturing shop floors, exact solutions become unfeasible within a reasonable time (Qin et al. 2021). Hence, optimization-based algorithms are utilized to identify acceptable solutions in dynamic environments.

Optimization-based algorithms

In recent years, most optimization-based algorithms introduced in dynamic dispatching descend from evolutionary algorithms (Liu, L. Wang, et al. 2019). Genetic algorithms (GA), inspired by the concept of natural selection and evolution, are the most prominent in this category due to their easy implementation and conceptual simplicity (Shukla et al. 2017). These algorithms are so-called *metaheuristics*, i.e., a higher-level procedure to find satisfactory solutions to optimization problems.

Using metaheuristics for dispatching problems transforms NP-hard problems into problems with polynomial complexity. This feature is essential for a computationally feasible dynamic dispatching system. However, the main drawback of these algorithms is that they require extensive expert knowledge and human intervention (Y. Wang et al. 2019).

Priore, Gómez, et al. 2014 defines two contradictory characteristics of dynamic dispatching that needs to be addressed:

1. The rule selection must contemplate different information about the manufacturing system in real-time.
2. The rule selection must be completed in such a short amount of time that real operations are not delayed.

Knowledge about the relationship between the environment's state and the dispatching rule applied is vital to achieving these characteristics (Priore, Gómez, et al. 2014). However, the procedure of choosing a dispatching rule from a pre-established set of rules can make real-time dispatching difficult, as the examination of all candidates (e.g., through simulation) may require a significant amount of time.

Nonetheless, the emergence of Industry 4.0 has made it possible to optimize material handling by acquiring knowledge of the environment and determine which rule is the best for each possible system state (Priore, Gómez, et al. 2014). The most prominent way of achieving this goal is through the use of machine learning.

2.1.2 Machine Learning

Machine learning (ML) can be defined as a collection of computational methods using experience to improve performance or to make accurate predictions (Mohri et al. 2018). Machine learning can detect patterns in data with little human intervention and use these patterns to understand new, unseen data (Murphy 2012).

Two of the most common approaches in machine learning are supervised and unsupervised learning. In supervised learning, the goal is to learn a mapping from inputs x to outputs y, given a labeled set of input-output pairs (Murphy 2012). The approach can compare its outputs with the correct outputs because it is labeled and finds errors to modify the model accordingly. Typical applications used with supervised learning are classification (categorical response variable) and regression (continuous response variable).

Unsupervised learning is applicable when the information used for training is not labeled, meaning the target value is unknown. In this scenario, we lack a response variable that can supervise our analysis. The goal is to discover structure or patterns in data, and we seek to understand the relationships between observations (James et al. 2013). Clustering and pattern detection is the most common areas within unsupervised learning.

Although supervised and unsupervised learning approaches are widely used within the ML field, they have some limitations regarding interactive problems. The material handling environment is an example of such a problem. Here, it is often difficult to obtain desired behavior that is both correct and representative of all situations where an agent needs to act (Sutton et al. 2018). Additionally, data from the real world is hard to capture and even harder to structure. Hence, many dynamic dispatching approaches utilizing machine learning are based on interactive learning between the agents and the environment.

2.1.3 Dynamic dispatching with machine learning

Figure 2.1 gives an architectural presentation of machine learning in interaction with a dynamic dispatching system. Here, training and test examples are generated from the simulation model. The machine learning algorithm uses the data acquired to gain dispatching knowledge iteratively.

The knowledge acquired from the algorithm, combined with the manufacturing system's performance and state, is utilized by the real-time control system to determine the best dispatching rule. In a CMHS context, the manufacturing system can be viewed as a facility equipped with an IPS that monitors the manufacturing environment continuously. The real-time control system plays the role of the cloud engine. Finally, the state and performance of the system are analyzed, and the knowledge is refined by generating more training examples from the simulation model until the performance is satisfactory.

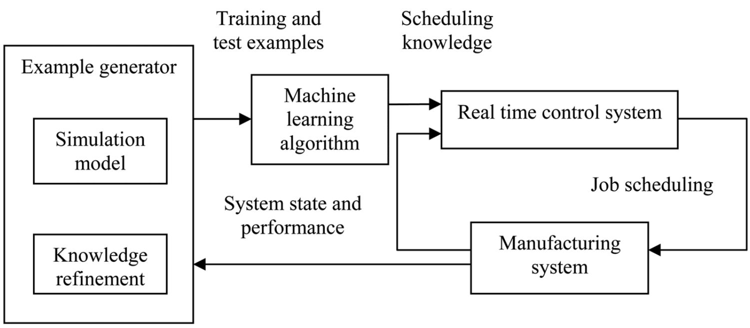

Figure 2.1: Overview of machine learning for dynamic dispatching (from Priore, Gómez, et al. 2014)

Table 2.2 excerpts some literary contributions for dynamic dispatching with machine learning:

Paper	Algorithm	Manufacturing system	Objectives
Kim et al. 2020	Deep neural network	Automated material handling system	Improve machine utilization and throughput
Choi et al. 2011	Inductive learning	Hybrid flow shop	Improve throughput, reduce mean flow time and mean tardiness
Priore, Parreño, et al. 2010	Support-vector machine	Flexible manufacturing system	Reduce mean tardiness and flow time
Y. Wang et al. 2019	Deep reinforcement learning	Multi-workflow scheduling	Minimize makespan, reduce costs
Hwang et al. 2020	Reinforcement learning	Job shop	Reduce total vehicle travel time
Zhou et al. 2021	Reinforcement learning	Cloud-enabled smart factory	Reduce makespan and energy consumption, improve machine utilization and balance machine workloads
Hu et al. 2020	Deep reinforcement learning	Flexible shop floor	Minimze makespan and delay ratio

Table 2.2: Research papers from the literature study

As Table 2.2 shows, current literature in the last years has viewed much attention to solutions with reinforcement learning, deep reinforcement learning in particular (DRL; François-Lavet et al. 2018). The field of DRL has received much attention, mainly because of remarkable results within game environments like Chess, Go, and Atari (Silver et al. 2018; Kaiser et al. 2019). However, in a manufacturing setting, the simultaneous scheduling of machines and material handling systems is rarely considered as it is a very complex problem (Tabatabaei et al. 2018). As a result, applications in real-world environments with simultaneous scheduling are still underexplored.

Today, as computational power is improving rapidly and emerging IPS technologies are under development, the use cases of DRL in a cloud-enabled shop floor are yet to explore. In order to understand how a DRL approach can be formulated in a shop floor context, it is necessary to give an in-depth explanation of one of the main areas of machine learning called reinforcement learning.

2.2 Reinforcement Learning

Reinforcement learning (RL) enables an agent to learn by trial-and-error in an interactive environment from its actions and experience with no predefined data required. Sutton et al. 2018 describes reinforcement learning as *"learning what to do - how to map situations to actions - so as to maximize a numerical signal"* (Sutton et al. 2018, p. 1). This definition assumes that the agents have one or more explicit goals they strive to reach. They continuously work towards these goals by maximizing rewards based on actions in specific states.

The two most crucial base features distinguishing RL from other machine learning approaches are trial-and-error search and delayed rewards (Sutton et al. 2018). An action executed by an agent may not affect only the immediate reward but also the subsequent rewards. As a result, RL needs to be structured in a sequential decision-making manner. This structure is formalized as Markov Decision Processes.

2.2.1 Markov Decision Processes

A Markov decision process (MDP) is a formalization of sequential decision-making (Sutton et al. 2018). The process involves an *agent* that interacts with the *environment* it is placed in, and the agent receives representation of the environment's *states*. In the context of this thesis, agents are viewed as material-handling equipment. The agent performs an *action* based on the representation, and gets a *reward* based on this action. As actions are performed, the agent's goal is to maximize the *cumulative* rewards throughout the process - not only the immediate reward.

Return of rewards

Given a set of states $\mathbf{S}$, a set of actions $\mathbf{A}$, and a set of rewards $\mathbf{R}$. At each time-step t the environment's state and the agent's action is formed as a state-action pair (S_t, A_t). At the next time-step $t+1$, the agent receives a reward based on the action taken from the state, $f(S_t, A_t) = R_{t+1}$.

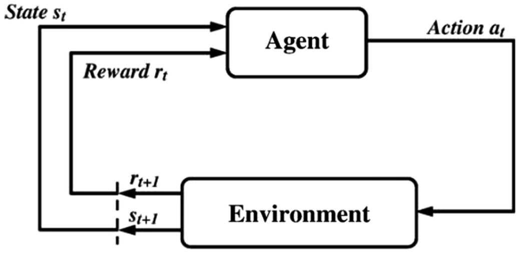

Figure 2.2: Agent and environment interaction in reinforcement learning (from Sutton et al. 2018)

For episodic tasks, the agent's goal would be to maximize these rewards simply by adding the rewards to a final time-step, T. On the other hand, in continuous environments—like in a manufacturing environment—the agent interaction with the environment continues without limits. A discounting factor, $\gamma \in (0, 1)$, is introduced to avoid infinite returns and apply the concept of delayed returns. Therefore, the agent's goal is to maximize the *discounted* return of rewards given by:

$$G_t = \sum_{k=1}^{\infty} \gamma^k R_{t+k+1} \tag{2.1}$$

Policies and value functions

With a newfound way to maximize the discounted return of rewards, it is also essential to consider the probability of an agent choosing an action a from a state s, and how good the action or state is for the agent (Sutton et al. 2018). The former is expressed as a policy, $\pi(a|s)$, while the latter is represented in a value function, $v_\pi(s)$. The quality of actions can be expressed through expected returns, and the expected returns depend on what actions are performed.

Formally, a state-value function of a state s under policy π can be defined as the expected return when starting in s and follow π thereafter:

$$v_\pi(s) = \mathbf{E}[G_t|S_t = s] = \mathbf{E}[\sum_{k=1}^{\infty} \gamma^k R_{t+k+1}|S_t = s] \tag{2.2}$$

The action-value function $q_\pi(s, a)$ can also be defined as the expected return of doing action a in state s, and follow policy π thereafter:

$$q_\pi(s, a) = \mathbf{E}[G_t|S_t = s, A_t = a] = \mathbf{E}[\sum_{k=1}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a] \tag{2.3}$$

Equation 2.3 will be referred to as the Q-function going forward.

The optimal policy

The goal of a RL algorithm is to find the policy that will yield the highest possible return. The policy in question is called the optimal policy. The optimal policy has a related optimal state-value function, $v_*(s) = max_\pi v_\pi(s)$, and an optimal Q-function, $q_*(s,a) = max_\pi q_\pi(s,a)$. The values of q_*, called Q-values, are updated according to the following equation (coined as the Bellman equation):

$$q_*(s,a) = \mathbf{E}[R_{t+1} + \gamma max_{a'} q_*(s,a)]. \tag{2.4}$$

Exploration vs. Exploitation

The trade-off between exploration and exploitation is also a key aspect of RL. An agent has to *exploit* what has already been determined to get its reward, but also *explore* the rest of the environment to make better action selections in the future (Sutton et al. 2018). In order to get the balance between exploration and exploitation, an *epsilon greedy strategy* is used. An exploration rate, ϵ, is defined and initially set to 1. The exploration rate will gradually decay with some factor so that the agent will become more inclined to exploit the environment. A randomly generated number, $r \in (0,1)$, will decide the trade-off at each time step. If $r > \epsilon$, the agent will choose its next action based on exploitation, and if $r < \epsilon$, the agent bases its action on exploration.

Learning rate

Suppose an agent experiences a state-action pair it has been to previously. In that case, we want to update the Q-value (Eq. 2.4) to reflect the Q-value regarding the agent's perception of future returns for this particular state-action pair. However, the solution is not to overwrite the current state but instead introduce a portion of the new Q-value to the old Q-value. How quickly the agent updates this value is governed by the learning rate, $\alpha \in (0,1)$. The learning rate determines the step size, that is, the speed at which the model learns, and is a highly influential parameter in search of a policy.

Although RL methods have proved to be successful in various applications, the implementations are only suitable for fully observable, low-dimensional state spaces (Mnih et al. 2015). For real-world applications, we are dealing with high-dimensional sensory inputs required to be processed efficiently and precisely. In order to meet these requirements, deep reinforcement learning is introduced.

2.3 Deep Reinforcement Learning

In a vast state space like a manufacturing environment, it is more feasible to approximate the value and policy functions outlined in the previous section. The approach to such a problem can be made by introducing neural networks, specifically through deep reinforcement learning (DRL). DRL is the combination of deep neural networks with reinforcement learning methods.

Neural networks are models which learn to associate inputs and outputs patterns adaptively using learning algorithms. Deep learning extends this by introducing additional, hidden layers between the input and output, enabling agents to make decisions without human interference (Mnih et al. 2015). Inspired by the structure and information processing of the human brain, deep learning has shown remarkable results, especially in supervised learning, with image processing and speech recognition as the most prominent (LeCun et al. 2015).

Although today's top results in deep learning are shown with supervised learning, data in the real world is generally unlabeled, and the information's structure is discovered by observing the environment. Reinforcement learning is one of the areas where labeled data is not required. The combination of deep learning and reinforcement learning is still in its infancy, but with the emergence of DRL, it is predicted to dominate the future of deep learning (LeCun et al. 2015).

Architecture

The deep neural network is incorporated in the agent like a brain, as depicted in figure 2.3. Given the current environment state as input, DRL can evaluate the agent's current state and rank all possible actions from this state. The rank is based on previous experiences from the current state and estimates the total reward expected after an action is taken. The agent will choose the best possible action leading to the highest long-term reward.

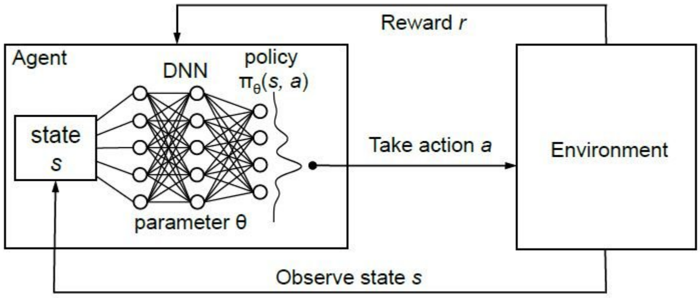

Figure 2.3: Agent and environment interaction in deep reinforcement learning (from Mao et al. 2016)

The memory of state-action values is acquired from nodes on which the neural network is built. With the same trial-and-error mindset as in pure reinforcement learning, the network nodes adjust their weights according to the backpropagated response received through the reward function.

Composite rewards

Dispatching rules in manufacturing shop floors often aim to optimize objectives simultaneously. For instance, the facility may strive towards increased throughput and equipment utilization while also desiring to reduce mean flow time, tardiness, and vehicle travel time. These objectives may sometimes be conflicting, so it is necessary to construct a global reward function to reflect all objectives simultaneously.

The construction of a global reward function with multiple objectives is feasible in theory. However, the practical implementation often tends to deviate from the theoretical global optimal solution due to variable perturbation (Deb et al. 2014). Thus, we are more concerned with reducing the variable sensitivity to find a robust solution in practice.

In DRL, parameters like product throughput, energy consumption, delay ratio, maintenance cost, and equipment utilization can be considered in the reward function since the reward is a scalar (Hu et al. 2020). As a result, the rewards can be combined based on multiple objectives into a *composite reward*. This feature has some clear benefits, as multi-objective optimization for practical purposes faces challenges regarding robustness and sensitivity (Deb et al. 2014). Since the reward is a scalar, the objectives can be weighted according to the practitioner's desire, giving a transparent overview of each reward's effect on the overall system performance.

The parameters outlined in section 2.2.1 have a significant impact on the policy. If the parameters are not controlled with care, it may lead to sudden catastrophic drops in performance. The performance is predominantly governed by the step size, i.e., how fast we replace the old policy value with the new policy value. To obtain an optimized policy, we need to choose a suitable procedure to update the policy.

Value function approximation vs. policy gradient methods

Traditionally, reinforcement learning approaches have been dominated by value function approximation for policy updating. Here, the agent's behavior is determined by the estimated long-term expected value of each action in a particular state (Beitelspacher et al. 2006). Various techniques have been introduced in the past decades, with deep Q-learning as one of the most prominent approaches due to its essential, model-free fashion (Wiering et al. 2012).

Although this widely used technique works well in game environments, it is poorly understood and fails on many simple problems on continuous state/action spaces (Schulman, Wolski, et al. 2017). Implementations with value function approximation often lead to deterministic policies, although the optimal policy is stochastic (R. S. Sutton et al. 2000).

In order to tackle these deficiencies, *policy-gradient methods* are often used as an alternative in stochastic environments because the methods can express stochastic optimal policies (Beitelspacher et al. 2006). Additionally, they are guaranteed to converge (R. S. Sutton et al. 2000). As the shop floor environments investigated in this thesis are continuous and stochastic with a medium-high degree of randomness and uncertainty, policy-gradient methods are a natural approach to consider.

2.3.1 Policy-gradient methods

Policy-gradient methods are key contributions for controlling reinforcement learning problems (Schulman, Wolski, et al. 2017). Unlike value function approximators that base the policy on a long-term reward estimate, policy-gradient methods work by directly computing an estimator of the gradient and use this in a stochastic gradient ascent algorithm. Gradient ascent algorithms are iterative methods to optimize an objective function (Bottou 2012). Policy-gradient methods directly update the policy according to the approximation of the gradient concerning the policy parameter (Richard Sutton et al. 1983). The estimated gradient is obtained by differentiating a loss function:

$$L^{PG}(\theta) = \hat{\mathbf{E}}_t[log\pi_\theta(a_t|s_t)\hat{A}_t] \tag{2.5}$$

The objective is to *maximize* equation 2.5. Here, $\hat{E}_t$ denotes the empirical expectation at time t, π_θ is a policy parameterized by θ, $(a_t|s_t)$ is the action-state pair, and $\hat{A}_t$ is the estimated advantage at time t. The advantage, A, can be considered as another version of the Q-value (Eq. 2.4) with lower variance by subtracting the state-value from the baseline $(A(s,a) = Q(s,a) - V(s))$.

A significant downside with traditional policy-gradient methods is the possibility of performance collapse if the step size gets too large or hopelessly slow progression with too small step sizes. The question is then how to update the new policy from the old policy.

Since the field of policy-gradient optimization has accelerated in the wake of deep neural networks, this thesis focus on a new, state-of-the-art family of policy-gradient methods for reinforcement learning called *proximal policy optimization* (PPO; Schulman, Wolski, et al. 2017).

Proximal Policy Optimization

Obtaining a successful DRL model is less pronounced than with other machine learning approaches as the parameters are highly tuning sensitive, and the effect of each parameter is hard to debug (Schulman, Wolski, et al. 2017). PPO tries to overcome these challenges by finding a suitable trade-off between implementation simplicity, sample complexity, and ease of tuning. Similar to the Trust Region Policy Optimization (TRPO; Schulman, Levine, et al. 2015), PPO aims to allow for the most significant possible improvement step without stepping so far that it causes performance collapse. In contrast to the TRPO, however, PPO presents a much simpler, first-order optimization with the same performance excellency (Schulman, Wolski, et al. 2017).

PPO algorithms use a hybrid approach of sampling data from the environment and optimization of an objective function using stochastic gradient ascent (Schulman, Wolski, et al. 2017). We denote the probability ratio between the old and the new policy as:

$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)} \tag{2.6}$$

Without a distance limitation between the new and old policy parameter θ, maximizing equation 2.5 may cause instability and destructively large policy updates. To avoid this issue, PPO introduces a simplified constraint by using a clipped objective; it forces $r(\theta)$ to stay within a small interval determined by a parameter, ϵ:

$$L^{CLIP}(\theta) = \hat{\mathbf{E}}_t[min(r_t(\theta) * \hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) * \hat{A}_t)] \tag{2.7}$$

The introduction of ϵ makes PPO a trust-region optimization method. More specifically, the ratio of taking a single action between the updated policy and the old policy is constrained to be no greater than $1+\epsilon$ ($A > 0$ increases action probability) and no less than $1-\epsilon$ ($A < 0$ decreases action probability)

Figure 2.4 shows the effect of the clip for a single timestep t. The red dot represents the starting point for the optimization ($r = 1$), and the clip depends on whether the advantage is positive or negative.

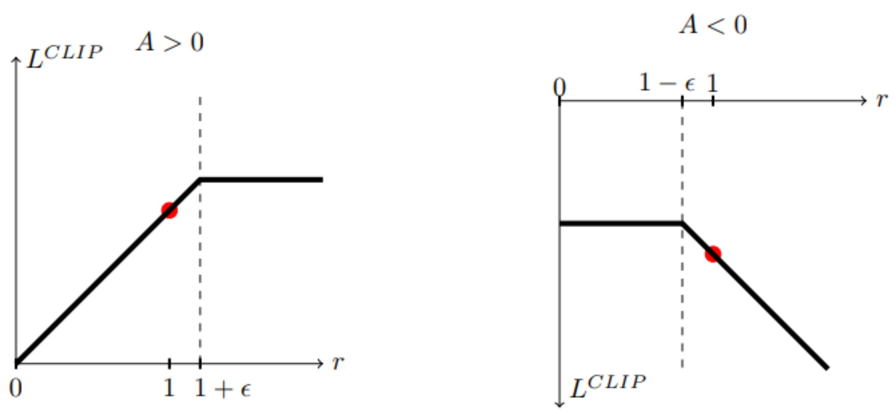

Figure 2.4: PPO clip parameter illustration fFrom Schulman, Wolski, et al. 2017)

Deep learning success depends on empirical choices of joint model structure tuning, data representation, and model optimization (Jaderberg et al. 2017). These components are controlled by numerous parameters. In a neural network setting these parameters are referred to as *hyperparameters* - a parameter whose value is used to control the learning process (Hutter et al. 2019).

PPO-clip provides the loss function, and its parameter, ϵ, affects the upper bound magnitude of resulting behavioral changes in the policy. In addition to this feature, it is necessary to simultaneously tune the hyperparameters involved in the learning process. Complex machine learning approaches like DRL often have many hyperparameters involved in the learning process. The complexity makes it necessary to automatically set these parameters in compliance with each other, known as automated hyperparameter optimization (HPO).

2.3.2 Automated hyperparameter optimization

The objective for automated HPO is to strike a sustainable trade-off between the performance and cost of a deep learning model. The non-stationary problems in DRL enlarge the scenario complexity level. As a result, the hyperparameters themselves are often non-stationary (Jaderberg et al. 2017). Automated HPO has essential implications for efficient policy generation, like reducing human effort, improve performance, and improve the reproducibility and fairness of scientific studies (Hutter et al. 2019). Two of the most widely used methods for automatic HPO are sequential optimization and parallel search[1].

Sequential optimization (Figure 2.5) uses knowledge from previous training runs to gradually tune the hyperparameters towards a satisfactory performance. This method starts by guessing an initial set of hyperparameter values. Then, these values are used in the model for training before the performance is evaluated. This process iterates until the performance converges to a satisfactory result. Although sequential optimization uses minimal computational resources, these methods have an apparent downside; due to their sequential nature, the process is tedious and slow, especially for expensive optimization processes (Jaderberg et al. 2017).

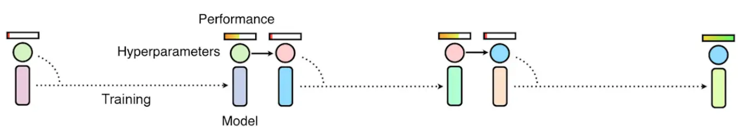

Figure 2.5: Sequential optimization

On the other hand, parallel search (Figure 2.6) can be viewed as a population of networks trained independently in parallel. When training stops, the researcher chooses the model with the highest performance. Parallel search is good at finding regions for sensitive hyperparameters, but the method is in danger of wasting

[1]Sequential optimization and parallel search are also referred to as hand-tuning and random search, respectively

computational power on inadequate combinations, making it inefficient for several applications (Jaderberg et al. 2017).

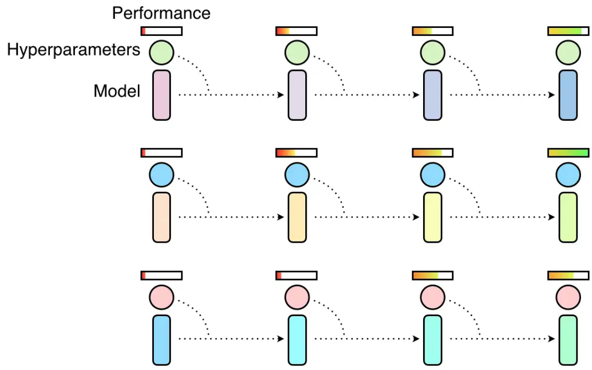

Figure 2.6: Parallel search

Jaderberg et al. 2017 suggests a different approach, aiming to bridge the gap from the shortcomings of sequential optimization and parallel search, namely population-based training (PBT).

Population-based training

PBT is a hybrid of the two techniques mentioned. Like parallel search, the hyperparameters are picked at random, and a population of concurrently running neural networks are trained in parallel. Inspired by genetic algorithms, the networks leverages information sharing across the population and *exploits* partial results from other promising training runs to refine the hyperparameters. Furthermore, the networks are capable of *exploring* new hyperparameters as training progresses by changing their values randomly (Figure 2.7).

The periodical process of exploiting the population and exploring new values secure training runs from a poor performance baseline while exploring the solution space consistently. This capability is critical in a reinforcement learning setting with significant non-stationary learning dynamics (Jaderberg et al. 2017).

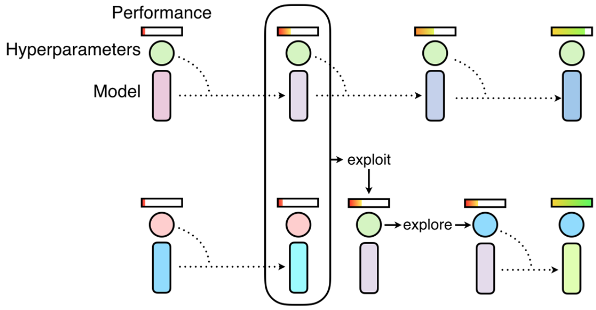

Figure 2.7: Population-based training

As for other common machine learning approaches, we want to optimize a set of hyperparameters, h, and corresponding weights, θ, of a model to maximize an objective. For this, we approximate the hyperparameters by introducing a function that evaluates the objective, $eval(\theta)$. Furthermore, the weights are updated iteratively, formulated by a step function, $\theta \leftarrow step(\theta|h)$. These steps are sequentially embedded that ideally converges to an optimal solution. Finally, we search over all possible hyperparameter values, $(h_t)_{t=1}^{T} = \mathbf{h}$, to arrive at an approximation:

$$\theta^* = optimize(\theta|\mathbf{h}^*), \text{ where } \mathbf{h}^* = \underset{h \in H^T}{\arg\max}\, eval(optimize(\theta|\mathbf{h})) \tag{2.8}$$

For an efficient calculation of equation 2.8, we form a population P where different hyperparameters optimize each model in the population. The methods *exploit* and *explore* are used to let h and θ adapt according to the population's performance (see Algorithm 1). These methods let us not only benefit from local optimization but also periodic model selection and hyperparameter refinement (Jaderberg et al. 2017).

Algorithm 1 Population Based Training

procedure TRAIN(P) ▷ Initial population P
    for $(\theta, h, p, t) \in P$ **do**
        while $r \neq 0$ **do** ▷ One step of optimization using hyperparameters h
            $\theta \leftarrow step(\theta|h)$ ▷ Current model evaluation
            $p \leftarrow eval(\theta)$
            if $ready(p, t, P)$ **then**
                $h', \theta' \leftarrow exploit(h, \theta, p, P)$ ▷ Use the rest of population P to find better solution
                if $\theta \neq \theta'$ **then**
                    $h, \theta \leftarrow explore(h', \theta', P)$ ▷ Produce new hyperparameters h
                    $p \leftarrow eval(\theta)$ ▷ New model evaluation
                end if
            end if
            update P with new $(\theta, h, p, t+1)$
        end while
    end for
    return θ with the highest p in P
end procedure

PBT does not require any population synchronization, meaning the computing processes are independent and run in parallel. Asynchronous behavior has a significant effect on data efficiency and improving the convergence rate (Ooi et al. 2015).

2.3.3 Why Deep Reinforcement Learning in a CMHS?

Despite the growing number of technologies in the wake of Industry 4.0, several opportunities are underexploited or neglected by companies (Moeuf et al. 2017). The research community has introduced bundles of sophisticated algorithms, but the main shortcomings of many solutions are extensive prior expert knowledge and human intervention. The lack of programmatic expertise within manufacturing companies makes it difficult to utilize all the benefits of these new technologies, and they need solutions that do not require expertise in this area. DRL can overcome these challenges because limited human interference and prior expert knowledge are necessary for the dispatch rule construction (Waschneck et al. 2017). Although the development phase itself may require highly skilled professionals, an abundance of third-party software is available for an effortless DRL implementation.

Other types of optimizers work well for offline dispatching in static environments, but they tend to overfit the data and consequently be biased towards specific situations (Priore, Gómez, et al. 2014). In contrast, DRL can adapt well to stochastic environments by responding immediately to changes. With the interconnectivity through an IPS, as one of the significant pillars in a CMHS, the environment can be monitored continuously and respond to changes dynamically.

As one of the emerging trends in Industry 4.0, big data is also under-exploited in the industry. From the literature study conducted by Moeuf et al. 2017, none of the companies reported any use of big data. Many companies still do not consider data as a source of added value (Bi et al. 2014).

A notable contributor to this perception is that capturing, cleaning, and managing significant data sources is often a comprehensive task. In contrast, DRL does not rely on predefined training data; as it exploits a learning-by-doing approach, the training data is generated in the simulation environment. The learning process is continuously improved based on newly discovered patterns that generate higher rewards. Moreover, due to the interconnectivity between simulation and reality with a CMHS, simulation can supplement learning purposes.

Deep learning methods often suffer from inefficient training. Learning an agent the environment dynamics and rules for a particular system can be a cumbersome task. Instead of starting all over again for each training run, pre-training the neural network to learn the underlying features can effectively speed up learning (Cruz et al. 2019). Pre-training of a DRL model has shown to be feasible and highly effective to speed up the learning (Anderson et al. 2015; Cruz et al. 2019; Abtahi et al. 2011). The effect of a pre-trained model in a CMHS allows for rapid response and high flexibility when major deviance in production occurs.

To summarize why DRL can be a good fit in the era of Industry 4.0 and the concept of CMHS, the following arguments are made:

1. DRL can learn its own dispatching rules with limited human interference or prior expert knowledge

2. It is not based on predefined training data, which makes it easier with implementation and maintenance

3. It has the flexibility to re-train a model within hours, and the agents can be pre-trained from existing pre-training

4. There exist quality tools for painless integration between simulation and training for dynamic dispatching

5. No research has been done with interconnectivity between materials and MHE with dynamic dispatching in the scale as CMHS facilitates. Therefore, it is possible to explore other applications of DRL that have not yet been identified.

Limitations

The key aspect of deep learning, in general, is that the layers of features are learned from data, not designed by humans (LeCun et al. 2015). Thus, the DRL model can be viewed as a black box with a lack of data transparency during training. It is not easy to understand why the agents make their choices in certain situations and how each parameter with corresponding weights affects the overall system performance. Hence, it is vital to monitor the effect of each parameter separately before composing them into a full-fledged reward function.

Even if the reward function is well-defined, it is often impossible to let agents interact freely in a real-world environment due to safety, cost, or time constraints. In real-world applications, François-Lavet et al. 2018 distinguish between two DRL limitations:

1. Agent is unable to interact due to dissimilarities between simulation and the real-world environment
2. Acquisition of new observations is not possible due to external, uncontrollable dependencies

In order to limit the impact of these limitations, it is crucial to develop an accurate simulation model that takes significant changes into account. Hence, generalization in the dispatching policy is vital to avoid overfitting (François-Lavet et al. 2018). These limitations are kept in mind during the model implementation presented in the next chapter.

Chapter 3

Methodology

This section describes the methods used to acquire knowledge regarding material handling in shop-floor manufacturing and this thesis's systematic approach to answer the research questions outlined in the introduction. The methodology is the justification for research methods - a systematic approach to solve a research problem (Goddard et al. 2014). The techniques presented let the reader assess the research's reliability and validity, enabling study replication.

Two main methods have formed the methodology used in this project: qualitative and quantitative. Quantitative research is based on the quantitative measurements of characteristics and is applicable when phenomenons can be expressed in terms of quantities (Goddard et al. 2014). On the other hand, qualitative research is primarily appropriate when the aim is to discover the underlying motives of behavior (Goddard et al. 2014).

A literature study contributes to the qualitative part of the research and is a key contributor to the answer to RQ1. The structure of the literature searches is elaborated in Section 3.1 and the results are presented in Chapter 4. A simulation model is constructed for quantitative research to answer RQ2 and RQ3. The justification and construction of the simulation model is found in Section 3.2 - 3.4, while the corresponding results are presented and analyzed in Chapter 5.

3.1 Literature study

A literature study is essential to understand existing research on the chosen topic. It can guide the research development to construct reasonable hypotheses and research questions in the initial research phase (Karlsson 2008). The purpose of the literature study is to identify research on material handling in shop floor manufacturing, especially research involving real-time positioning of material-handling equipment with or without Industry 4.0 applications, to answer RQ1.

Block searches were the dominating method in the literature study. Based on Sgarbossa et al. 2020, three categories were chosen that reflected Industry 4.0 applications in manufacturing connected to shop floor material handling:

- CPS/CM frameworks with real-time data collection (CPS)
- Shop floor machine scheduling (MS)
- Shop floor material handling (MH)

The CPS abbreviation stands for Cyber-Physical System and was used as the literary community use the terms CPS and CM interchangeably. The categories were chosen due to the diversity in problem formulations that tackled different shop floors and material handling processes. Thus, these three were selected as general bins to demonstrate the primary contribution value of the papers.

Scopus were the primary search database used for the block searches. Google Scholar were also used as a supplement to avoid overlooking relevant papers. The keywords used were:

- "Material handling" AND ("AGV" OR "AMR" OR "Shop floor*" OR "Scheduling" OR "Dispatching" OR "Fleet" OR "Industry 4.0" OR "Non-automated" OR ("Forklift" AND ("Manufacturing" OR "Production")))
- ("Cyber-physical system*" OR "Cloud manufacturing") AND ("Material handling" OR "Shop floor")
- "Material handling equipment" AND ("Shop floor" OR "Scheduling" OR "Dispatching")

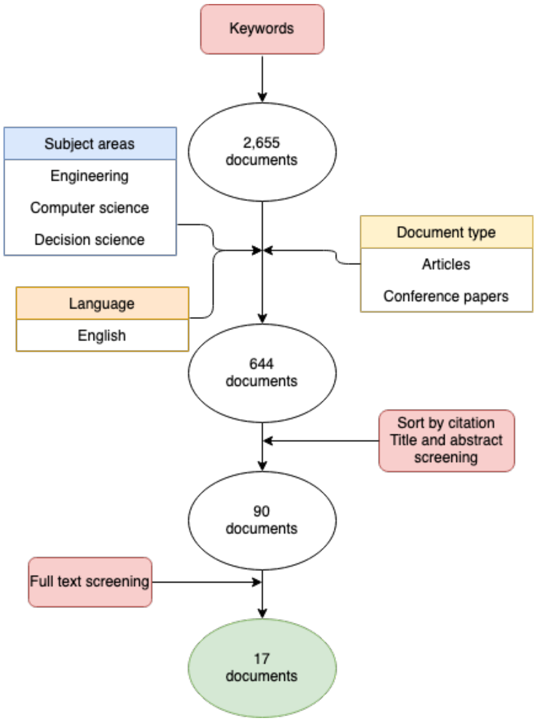

Figure 3.1: Filtering process for the literature study

The initial search gave 2,655 documents in Scopus. The documents were filtered to avoid outdated research, where only articles and conference papers in English over a time period from 2014 to 2021 were considered. "Engineering," "Computer Science," and "Decision Science" were chosen as subject areas to limit the results to papers that mainly focus on I4.0 technologies and decision-making techniques in manufacturing. After filtering, 644 documents remained.

The searches were sorted according to the number of citings in descending order, and each paper's title, abstract, and keywords were deemed relevant based on the following criteria:

- Does the document involve any type of MHE tracking for non-automated MHE? If not, is the MHE used either an AGV or an AMR?
- Does the document dedicate its focus to shop floor manufacturing?
- Does the document mention MHE scheduling/dispatching?

These criteria reduced the number of papers to 90. The documents were stored and handled by using BibTeX as a reference library. Promising articles were skimmed and noted in a logbook with key topics, methods, and a summary. After a full-text review and removal of duplicates, 17 documents were chosen for further analysis.

The documents were then used to identify the use and impact of Industry 4.0 in shop floor material handling and CM in particular. This analysis laid the foundation for how the CMHS addresses the current research gaps as a contribution to answer RQ1. The results from the literature study can be found in Chapter 4.

3.2 Simulation modeling

Modeling is a way to explore and understand the structure of a system that appears in the real world through a risk-free, virtual environment (Borschev 2013). The concept of modeling, particularly in manufacturing shop floors, is of great interest because finding solutions by experimenting with real objects is too expensive, dangerous, or simply impossible. It is highly impractical to model material handling activities for a job shop in a real environment, primarily if the movement of jobs in the environment only relies on the material handling equipment (Xie et al. 2015). Modeling simplifies the real-world environment to predict how a system behaves where there is room for mistakes and continuous iterations.

Pure analytical modeling is usually not suited for complex, dynamic environments because an analytic solution often does not exist or is hard to obtain (Borschev 2013). A manufacturing shop floor is an example of such dynamic environments, and to model dynamic behavior in these systems, simulation modeling is used. Borschev 2013 describes six advantages of simulation modeling:

1. Possible to analyze systems and find solutions where methods like analytic calculations and linear programming fail
2. Requires less intellectual efforts than analytical modeling - it is easily scalable, incremental, and modular
3. Reflects the structure of a real system
4. Possible to measure any value and track any entity
5. Ability to animate system behavior in time, making it easier to verify and debug the model
6. A more persuasive and explainable tool compared to spreadsheets with numbers

Simulation advantages for a CMHS

Simulation modeling can capture every logical aspect of a system. Nonetheless, manufacturing environments' complexity and uncertainty make such systems intractable to analytic modeling and simulation (J. Xu et al. 2015). As simulations in virtual environments simplify the real world, the main issue is to map the insights, findings, and simulation solutions back to the real shop floor environment. Advanced manufacturing schemes like the multi-agent system (MAS) have aimed to overcome this drawback. However, due to a lack of global coordination, these schemes have not been able to handle complex systems efficiently on their own (S. Wang et al. 2016).

On the other hand, a CMHS is equipped to tackle this transferability issue. Exploiting the real-time location of products and material-handling equipment with an IPS can allow for dynamic job assignments to available resources from a centralized cloud platform. This capability bridges the gap from simulation to reality; simulation can mirror reality more accurately with IoT-equipped materials and MHE utilized in a CMHS. Suppose the layout and material flow modeled are in accordance with the real-world environment. In that case, the simulation could reflect the cloud-enabled shop floor's exact image, as they are both relying on the same assignment logic. Hence, simulation can be a great tool to identify a CMHS's capabilities regarding reliability and validity.

3.2.1 Simulation software

For simulating complex systems like a manufacturing shop floor, a multi-purpose simulation tool is needed. As the simulation software market has been increasing rapidly in the wake of Industry 4.0 (Xie et al. 2015), there is a need to map software capabilities to the modeling method and purpose. When choosing the right software for our purpose, the following requirements were made:

- Possibility to extend the graphical modeling language with other well-established code languages
- Real-time 3D animation capabilities of manufacturing processes
- Support a variety of simulation methodologies
- Good documentation and customer support

With these objectives as a starting point, a simulation software survey from the October 2017 issue of the ORMS Today journal (ORMS-Today 2017) was used to map out the software vendors and compare the software tools and capabilities.

From this mapping, AnyLogic became the natural choice as one of the most widely used simulation software in the manufacturing industry (Borschev 2013). AnyLogic is a multi-method simulation modeling software that supports all the principal simulation methodologies.

Additionally, it extends its graphical modeling with Java code. These capabilities give flexible opportunities in the modeling, as code snippets with dynamic values can replace the standardized graphic value editor. Notably, this thesis's objectives require tailor-made logic for agent behavior and flexibility beyond what any application programming interface (API) can offer today. Moreover, AnyLogic provides several manufacturing process examples, guideline books for learning purposes, and a responsive support team.

3.2.2 Deep Reinforcement Learning software

There are two possibilities for implementing DRL to the simulation model: constructing the neural network with Java code from scratch or using a third-party extension to the AnyLogic software. The former approach allows for a more custom-made solution, offering higher flexibility in the development process. However, this approach might be excessive for this thesis's purposes, as AnyLogic cooperates with several companies specializing in DRL. The purpose of the simulation model is to demonstrate the simplicity of policy generation with CMHS, and the DRL software Pathmind was chosen due to its painless compatibility with AnyLogic.

Pathmind is a Software-as-a-Service (SaaS) that offers painless integration between simulation modeling and reinforcement learning. It is a web-based platform that provides easy access to state-of-the-art DRL and cloud computing. The capabilities of Pathmind manifest how the platform can be used in manufacturing for decision-making and optimization with limited prior expert knowledge:

1. It handles the integration between simulation and RL libraries automatically
2. No data science expertise is required
3. Policies generated can be used in real-world manufacturing facilities with real-time decision making

The policy is trained in the Pathmind software, extracted as a zip file, and deployed in AnyLogic. Figure 3.2 depicts the connection between the components.

Figure 3.2: Simplified Pathmind setup

Symbiotic relation between PPO and PBT in Pathmind[1]

Pathmind has integrated PPO and PBT simultaneously, where PBT controls parameter tuning while PPO governs the behavioral policy changes. The updated PPO policy results from many fine-grained changes among the different neural network weight dimensions. Whereas PPO provides the loss function, the actual loss landscape is transversed with an ADAM optimizer (Kingma et al. 2017), which uses adaptive learning rates based on the dimension-wise curvature approximation of the landscape. ADAM treats each neural network weight dimension separately so that the adequate step size can be small in some dimensions but large in others.

The learning rate parameter tuned via PBT is the α parameter of ADAM, which still is a free parameter that benefits from tuning (Loshchilov et al. 2019). ADAM's α parameter affects the magnitude of changes to individual neural network weight dimensions, whereas the PPO-clip parameter ϵ affects the upper bound magnitude of resulting behavioral changes in the policy.

As PBT explores different α parameters, PPO can safely compensate for inappropriately large α values through a well-tuned clip parameter. The PBT tunes many different hyperparameters, PPO-clip's ϵ and ADAM's α included.

[1] An explanation of PPO and PBT can be found in Section 2.3.1 and 2.3.2, respectively

3.3 Model implementation

The simulation model is a multi-agent system (MAS) in a fully cooperative, stochastic environment. All agents strive for a joint maximization of the same goal. The agents' returns cannot be maximized independently in such environments, as agents' behavior is correlated (Buşoniu et al. 2010). Multi-agent environments are often far more complex than single-agent environments, and hardwired behavior may be inappropriate due to dynamic, unpredictable changes over time. In practice, the agents in a MAS are heavily dependent on real-time communication with sensors and tracking, making a MAS suitable for CMHS validation and testing.

3.3.1 Scope

The simulation model's focus is on the cloud engine and the way it assigns material-handling jobs to MHE (see figure 1.2). Hence, the simulation model does not represent a physical implementation of the CMHS. That being said, the cloud engine follows the built-in simulation model logic.

The model will only consider material handling as transportation of materials from A to B on the manufacturing shop floor. Conveyor-based systems are not regarded as material handling in this interpretation and will not be considered going forward.

The research includes hybrid setups of different MHE types working together on the shop floor. However, the simulation model's focus will be on managing forklifts only. A hybrid setup introduces unnecessary complexity in the simulation model, and as the benefits are evident on a conceptual level, it has been left out for simplicity reasons. Still, it should be noted that the symbiotic relationship between different MHE types is investigated in the literature study and is considered an integral part of the CMHS concept.

3.3.2 Modeling technique

In order to model the system, appropriate modeling techniques are needed. Two modeling methods have been used in the simulation model: *discrete-event modeling* and *agent-based modeling*. The two methods are differentiated based on the model's abstraction level, i.e., the amount of information and details included. Figure 3.3 portrays a rough division of the two methods, where discrete-event is known for a low abstraction level. In contrast to discrete-event, agent-based modeling span various purposes and scopes, focusing on the autonomous, decentralized, and individual behavior of objects.

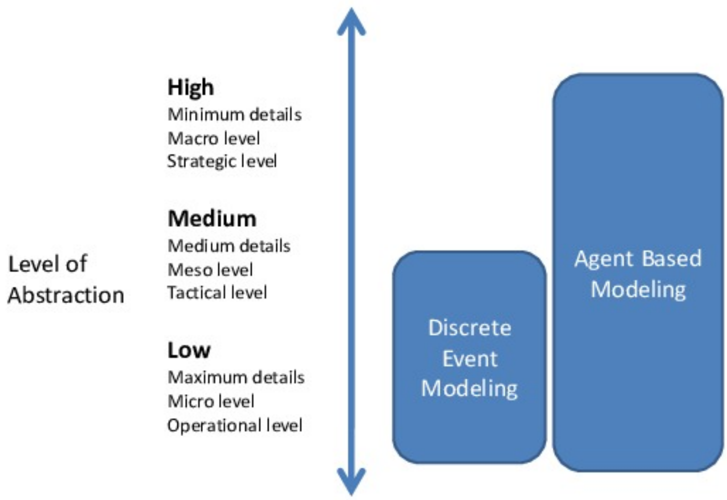

Figure 3.3: Level of abstraction for discrete-event and agent-based modeling (from Borschev 2013)

Since the objective is to investigate the utilization of MHE and product throughput in a manufacturing shop floor through a predetermined set of stages, the natural approach is to model the processing stages in the environment using discrete-event modeling. This method enables us to model the system as a *process* (sequence of operations) done on *entities* (materials) by *resources* (material-handling equipment).

However, the MHE involved in the process should also choose actions from a finite action space based on the state of the environment in which they are operating. A significant difference between discrete-event models and agent-based models is that neither the entities nor the process have behavior on their own in the discrete-event case (Borschev 2013). When implementing the CMHS, the system behavior is governed by a cloud-enabled shop floor that accounts for the environment's state, which changes dynamically in time. Agent-based modeling can be suitable when investigating individual object behavior and optimal policies through sophisticated heuristics and machine learning algorithms. Hence, autonomous behavior is also appreciated for this simulation model.

Many real-world environments are too complex to model with one method only (Borschev 2013). Thus, the model will be built according to the following multi-method approach:

- *Discrete-event* for the processes and the environment as a whole
- *Agent-based* for the decision-making and general behavior of the agents (MHE)

3.3.3 Shop floor layout

The layout is based on an Italian shop floor example where forklifts move products and supply in the facility, aiming to acquire the highest possible product throughput (Figure 3.4). The shop floor is modeled with dimensions 152 by 178 meters and thereby represents a large shop floor facility with considerable distances that need to be covered by the forklifts between the material handling jobs. The shop floor produces three different product types that undergo a series of steps until they are finished and transported away from the facility. Products cannot be processed without supply present at the workstation, but the type of supply provided is equal regardless of product type or workstation number.

The process stages are fixed for each product, the type of operations executed in the environment are predetermined, and the same kind of entities are used to perform material handling tasks. The processes have a starting point at the supermarkets (the source), where materials are injected into the system. Then, numerous processes (seizing transporters, moving materials, processing products) are executed discretely. When the required stages are completed, the product is removed from the model at an ending point (the sink).

The shop floor has four docks where forklifts can idle when no job is present. Depending on the model method and return policy after a material handling activity is performed, the forklift will navigate to a dock until a new job occurs. The forklifts carry out simple transportation tasks of product or supply from pickup-point to destination. The arrival rate of materials and supply ready for processing is scenario-dependent but is assumed to be stochastic in real life. The scenario parameters and corresponding results are outlined in Chapter 5.

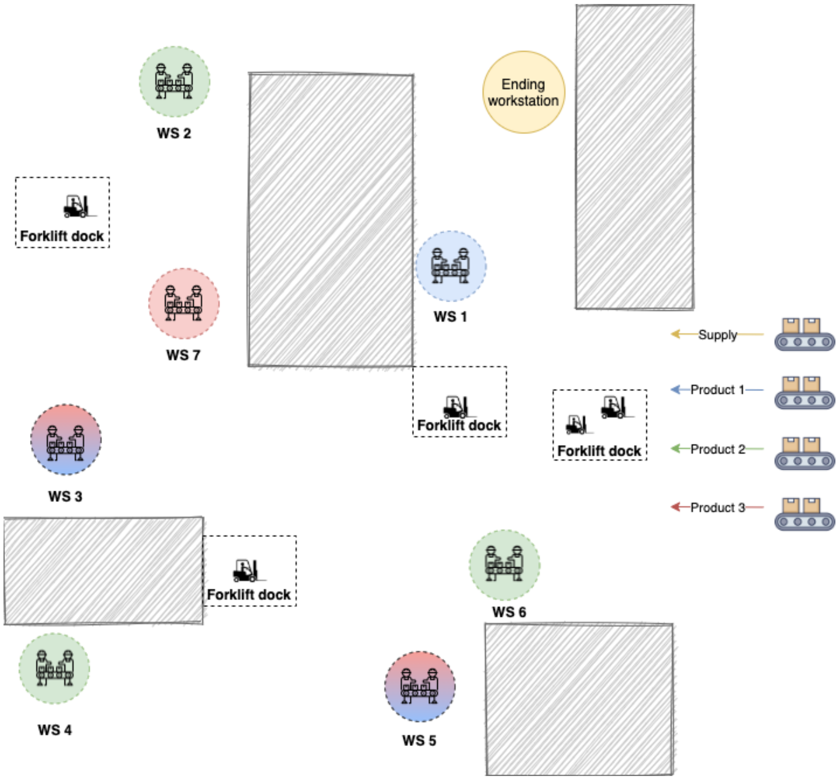

Figure 3.4: Layout

Each product is equally prioritized and has to move through three workstations in a predefined order. As illustrated in figure 3.5, some products share the same workstation in different completion stages of the material flow. When a product reaches the "Ending workstation," the throughput count is incremented by one and is considered completed.

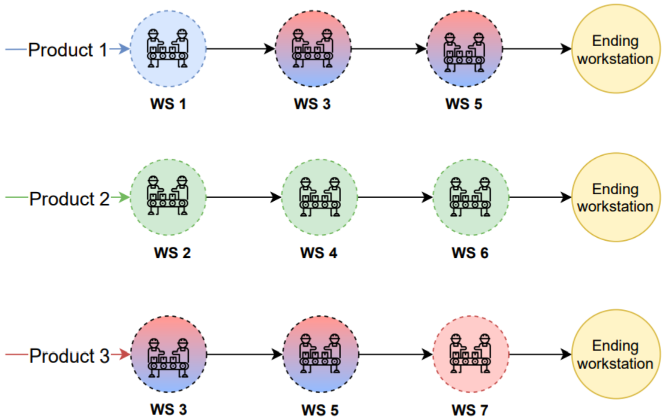

Figure 3.5: Product flow

The facility possesses seven workstations, where each workstation is responsible for some kind of product processing. The type of processing is not specified in the example provided. However, as the simulation model focuses exclusively on material handling activities, elaborating on processing specifics is unnecessary as we advance.

3.3.4 Stochastic shop floor processes

In most manufacturing shop floor environments, unpredictable real-time events like arrivals of urgent jobs or workstation failures may occur at any moment, making it challenging to stick to a pre-defined schedule or plan (Ouelhadj et al. 2009). Running a material handling operation under such conditions can result in low MHE utilization and poor manufacturing performance if the system struggles to adapt to real-time events (Tompkins et al. 2010). In order to mimic the stochastic shop floor, the simulation model introduces stochastic elements to the demand and maintenance requirements in scenarios that will be elaborated further in Chapter 5. However, the mathematical methods are explained here.

The rate of arrivals that simulate the system demand is expressed as a uniform discrete distribution bounded by a (min, max) input (Figure 3.6). In this case study, arrival rates can be perceived as unfinished products/supplies delivered to the facility. The distribution is easy to use and is beneficial for scenario testing as the input is only two variables. More complex distributions like the Weibull distribution (Rinne 2008) are often seen in manufacturing simulation models. However, as this study aims to demonstrate differences in control methods rather than simulating the day-to-day operation of a facility, the uniform distribution is more than sufficient.

The lower complexity is also helpful for precise result extraction as they are generated by Monte Carlo simulation (see section 3.4.4).

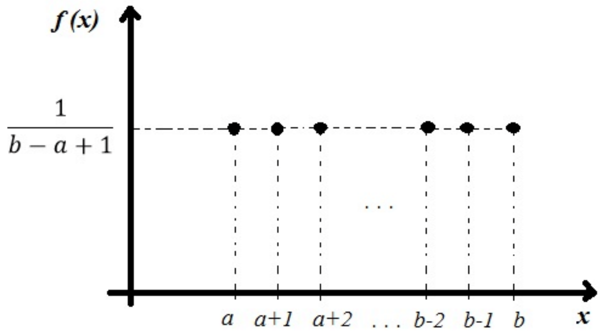

Figure 3.6: Uniform discrete distribution (from Sirjani 2017)

3.4 Modeling methods

As stated in the research objectives, three methods are chosen to investigate the capabilities of the CMHS: conventional non-CMHS version, heuristic CMHS, and deep reinforcement learning CMHS. The basis for comparison is scenario-testing, where the performance is assessed based on total product throughput and MHE utilization over an eight-hour workday.

3.4.1 Conventional method (non-CMHS)

In the conventional method, visual contact with the workstation is a prerequisite. The method demonstrates how forklift drivers in a traditional manufacturing environment behave without support from a cloud-enabled shop floor. Hence, forklifts are assigned to a cluster of workstations based on their home dock position as illustrated in figure 3.7. Consequently, the forklifts are unable to contribute anywhere else in the environment than in their assigned area.

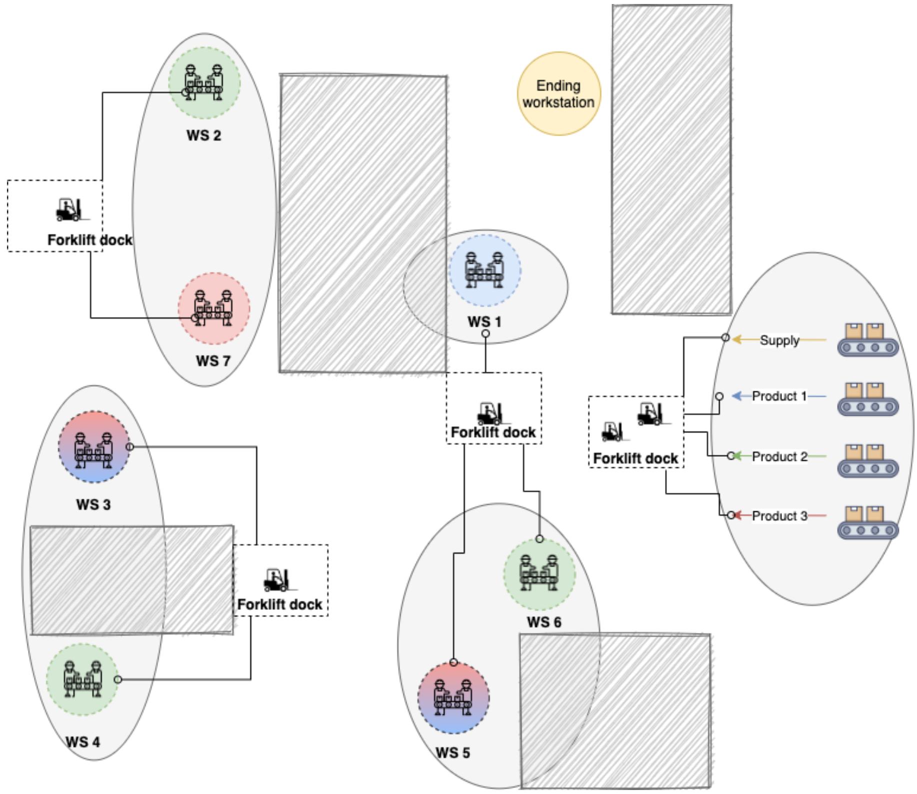

Figure 3.7: Conventional method restricted to visual sight

In the past, analytical models have been the most popular methods to determine fleet sizes in real-time manufacturing environments, but simulation offers an alternative way (Chawla et al. 2018). Thus, the number of forklifts in each dock is determined by a trial-and-error approach where higher material flow on a station requires a larger fleet size.

3.4.2 Heuristic method (w/CMHS)

The heuristic method is based on the same layout configuration as the previous method. However, the environment is assumed to be equipped with sensors for materials and MHE tracking components to simulate the CMHS. Thus, the visual line-of-sight requirement is no longer necessary, so the forklifts can move freely on the shop floor, which implicates that any forklift can perform a material handling job anywhere on the shop floor. Here, the forklifts are assumed to possess a notification device (e.g., tablet) that instructs where to move and which job to perform.

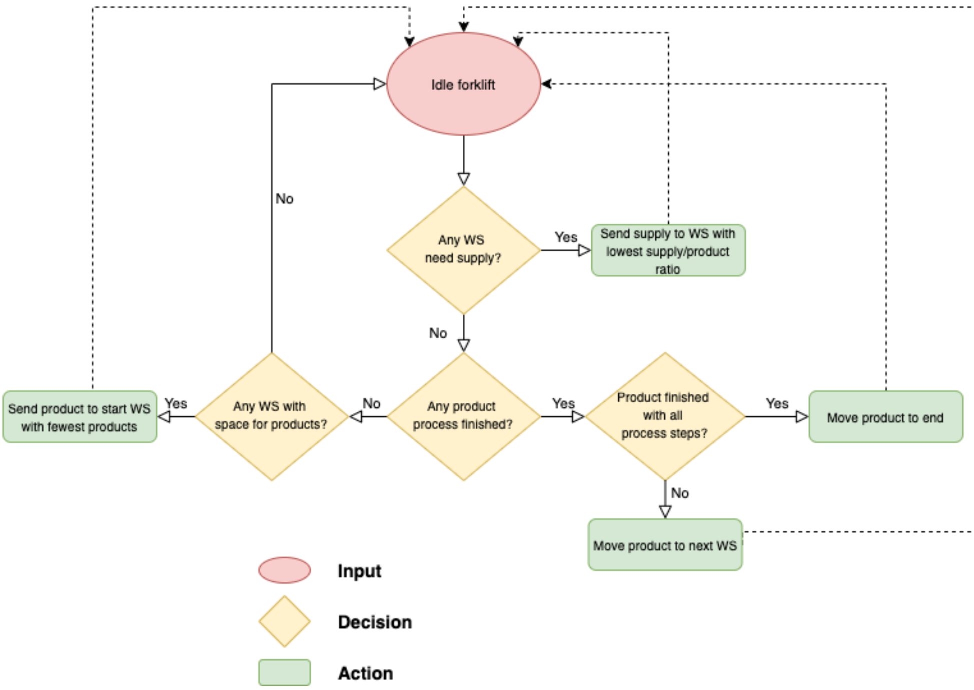

Figure 3.8: Action flowchart for heuristic method

Which action to perform is governed by the environment's state. As demonstrated in figure 3.8, the action space is structured as a decision tree hierarchically based on the essential parts necessary to maintain high performance and adequate material flow. Several factors determine the decisions: if the workstation is congested, if the throughput of each product type is imbalanced, and so on.

The heuristic method uses static dispatching and idling rules to control which MHE is assigned to what task. There are several rules to choose from, listed in Section 2.1. Inspecting table 2.1, the LIV and LUV dispatching rules are not applicable for testing due to the simulation model having a MHE-initiate action-loop. GQL and LWT are very similar as they use metrics from a given node's output in the system. During the trial-and-error phase of the simulation modeling, GQL performed worse than LWT and STD and was, as a result, eliminated from further evaluation.

Idling MHE also need to abide by a static rule in this heuristic method and the two most common *centralized positioning* (CP) and *point of release/nearest idle location* (NIL) (see Section 2.1). The CP rule instructs each idle MHE to the dock between WS1, WS5, WS6, and the supermarkets as these are the busiest zones on the shop floor. The NIL rule instructs idle MHE to the nearest dock after it has become idle.

The following heuristic dispatching rules will be tested:

Dispatch rule	Idle rule
STD	NIL
STD	CP
LWT	NIL
LWT	CP

Table 3.1: Heuristic dispatch and idle rules

Assessing the rule combinations in different scenarios allows for a gained insight into the optimal dispatching method for the given scenarios. Also, the impact of idling rules will become apparent as a result of the analysis.

3.4.3 Deep Reinforcement Learning (w/CMHS)

In contrast to the previous method, any specific decision hierarchy does not control the DRL method. The forklifts are trained to discover actions in certain states, which yields higher total product throughput. To obtain this capability, they need *observations* to see environment states relevant to the decision-making and *rewards* to recognize productive behavior.

Training of the DRL model is done in Pathmind and is based on the same architecture and principles as outlined in the theory section - parameter tuning with PBT (Section 2.3.2), policy changes governed with PPO (Section 2.3.1), and traversed optimization with ADAM (Section 3.2.2).

Observations

Observations provided to the agents consist of individual and global observations (Figure 3.9). Global observations are simply the components' states in the environment, like queue length, supply needs, and finished products. Since the Pathmind software used for dispatching policy generation is disconnected from the actual simulation model, the agents need information about their current location and who they are in the population. Additionally, as the simulation model portrays a collaborative multi-agent system, the agents need to observe other agents' current whereabouts and activities.

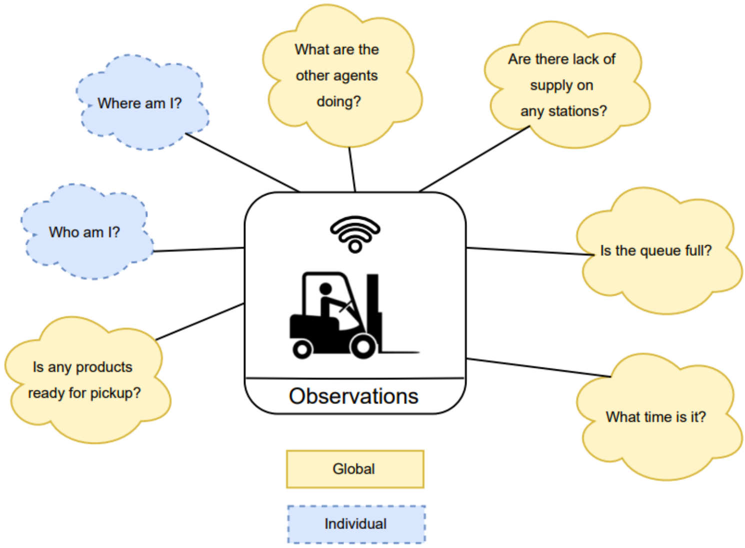

Figure 3.9: Environment observations

Rewards

Specifying a common goal is one of the most challenging tasks in reinforcement learning, multi-agent reinforcement learning in particular (Buşoniu et al. 2010). Although the main objective is to obtain the highest total throughput possible, the agents need to be rewarded on penalized in other parts of the procedure to understand how they can achieve higher throughput. Hence, numerous individual and global rewards and penalties shape the composite reward functions. Note that the reward functions will vary between scenarios, but they will all be a combination of the rewards/penalties portrayed in figure 3.10.

As explained in the theoretical background (see section 2.3), multiple, conflicting objectives often deviate from a global optimum in practice. Figure 3.10 portrays some contradictory objectives. For instance, agents should be rewarded for *higher throughput*, but not at the expense of *full queues* or significant *product imbalances.* The correct weighting of each reward or penalty is an art in itself. The resulting reward function used in this simulation model is based on an extensive trial-and-error methodology. Moreover, as seen in Chapter 5, these weights are tuned differently according to the scenario investigated.

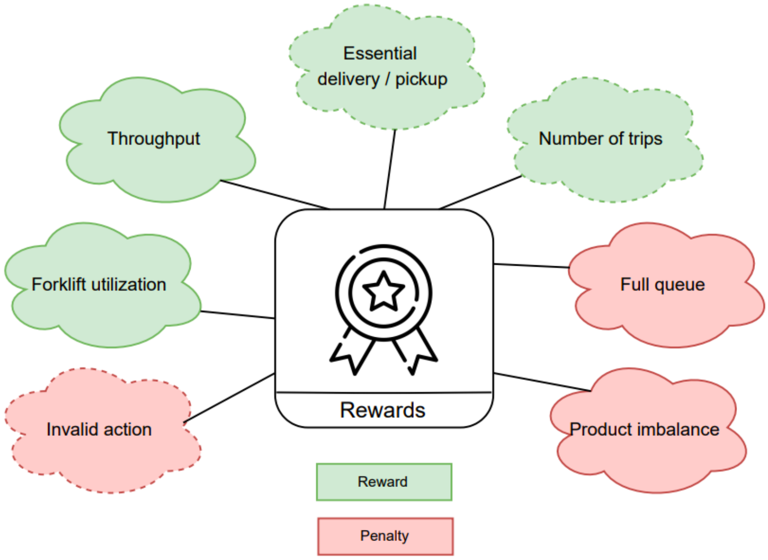

Figure 3.10: Rewards and penalties

Pathmind training

The DRL method is re-trained for each specific scenario investigated. The AnyLogic model is uploaded to the Pathmind service, and the reward function is specified. In order to evaluate the effect of reward variables, experiments are run in parallel, and the performance is monitored through a learning process graph (Figure 3.11). This graph depicts how the agents learn according to the defined reward function.

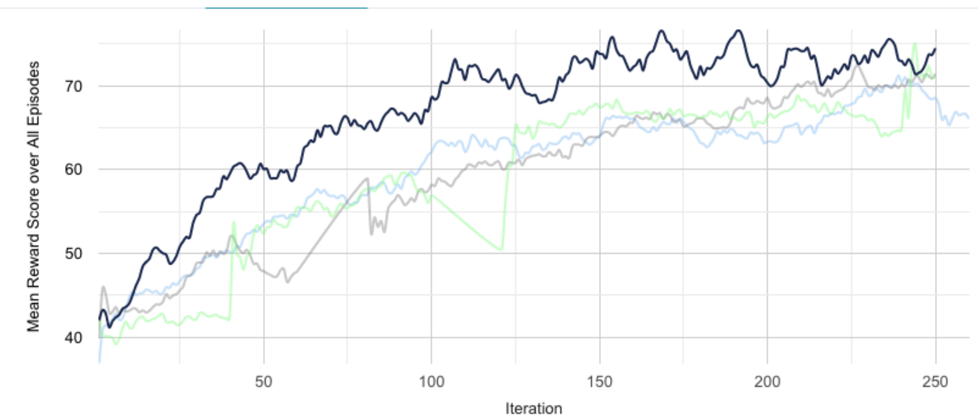

Figure 3.11: Policy training in Pathmind

When the policy is trained, Pathmind will play a single simulation run as many times as possible. Then, the final reward is averaged over all episodes and captured into a single iteration. This process continues until convergence is reached. As the figure above illustrates, Pathmind will run different hyperparameter combinations in parallel and use the experience-sharing capability of PBT to improve the policy gradually. After a fixed number of iterations—250 in this case—Pathmind chooses the run with the highest mean reward score overall episodes and transforms it into a full-fledged policy.

3.4.4 Performance evaluation

Key Performance Indicators (KPIs) are a set of quantifiable measurements used to evaluate the critical business objectives' overall performance (Parmenter 2015). For simulation models, KPIs can be used to evaluate and compare cases' performance in respective scenarios (Xie et al. 2015).

The main objective is to find a trade-off between minimized material handling equipment costs and maximized throughput. To measure the MHE costs, earlier simulation methods in job shops have emphasized MHE utilization to identify equipment productiveness (Xie et al. 2015). In the simulation model, MHE is only considered utilized when they are either on their way to a job or performing the job. Hence, both total product throughput and MHE utilization will be the basis of analysis for the results. As there are no interdependencies between the products on the evaluated shop floor, product imbalances are not considered a KPI when analyzing the performance results.

In order to ensure a viable basis of comparison, the performance will be evaluated from Monte Carlo (MC) simulation. MC is suitable for this model's purposes, as it is broadly used to model probabilities of different outcomes when random variables are involved. Although some of the scenarios include variables like arrival rates and processing times with randomly generated values, it is worth mentioning that each of the methods evaluated is based on the same randomness. Hence, the MC simulation comparison is credible, and the variability does not differ between methods.

Instructions on how to run the simulation model are explained in Appendix A.

3.4.5 Limitations

The main limitation of this thesis's simulation modeling is that we have not cooperated with any manufacturing companies. As a result, the simulation modeling has limited data available. Preferably, information about a company's characteristics, real-time data about mean processing times, arrival rates, and failure rates would have made the generated results more credible and more manageable for a company to compare its existing operations with the shop floor evaluated in the simulation model.

To reduce the impact of this limitation, the layout from the Italian job shop serves as a baseline. Additionally, each method will experiment with various scenarios based on educated assumptions about variables like arrival rates, processing times, and failure rates in cooperation with our supervisors. Although the data basis is limited, the novelty of CMHS will still benefit from this analysis before moving into an entire case study.

Although the use of third-party software often leads to a faster development process, it may also limit the flexibility of customized logic and solutions as the developer is bounded to the API services provided by the issuer. Even if AnyLogic provides snippets of Java code, the drag-and-drop palettes on which the model relies are static classes with few editing possibilities. The user is inevitably dependent on the provider who can jeopardize the model's stability, especially when two software services are intertwined like AnyLogic and Pathmind.

Minor transparency issues have also been encountered with Pathmind. Although the simple system and techniques utilized are known, the detailed implementation is not visible as this is the software's trade secret. This shortcoming is somewhat insignificant in practice, but it also prohibits us from identifying the exact parameter values.

Finally, we acknowledge that the solution in this thesis does not guarantee a perfect simulation model or DRL policy. This perfection requires expert knowledge and considerable effort so that some elements may have been overlooked.

Chapter 4

Literature Study Results and Analysis

A thorough understanding of how I4.0 has influenced the material handling research field is integral to assess the potential gains from a CMHS (RQ1). Gaining insights into what areas have received more, some or no attention by researchers allows for a direct approach where the CMHS is formulated to answer the gaps identified in the literature. With this process in mind, a literature study has been conducted with a base footing in I4.0 research relevant to shop floor material handling is presented. The CMHS is highly technology-dependent, and to understand the influence of I4.0 and CM in material handling, it is necessary to complete the research surrounding the topic.

Manufacturing has been an established field on its own in the past. Still, technological advancements have resulted in a research field that spans a wide range of problem formulations with a corresponding variety of solutions. The initial research phase revealed a lack of academic common ground, which necessitated a literature study to clarify the current state of material handling. Without this understanding, the theoretical improvements in flexibility and productivity of a CMHS are challenging to demonstrate. The following section (4.1) presents the results and associated research gaps. Then, the CMHS is discussed regarding these gaps and the theoretical background knowledge to answer RQ1.

4.1 Results and analysis

Table 4.1 lists the papers that match the selection criteria, categorized into the groups presented in 3.1: Cyber-Physical System (CPS), Machine Scheduling (MS), and Material Handling (MH).

Paper	**CPS**	**MS**	**MH**	**Comment**
Tao et al. 2017	Yes	No	No	CPS architecture
Lee et al. 2015	Yes	No	No	CPS architecture
Zhong et al. 2016	Yes	No	No	CPS architecture, RFID
Liu, L. Wang, et al. 2019	Yes	No	No	CPS/CM literature review
Mourtzis et al. 2018	Yes	Yes	No	RFID, CM, machine operators
Greis et al. 2019	Yes	Yes	No	RFID, CM, machine operators
Nielsen, Dang, et al. 2017	Yes	No	Yes	AMR part feeding, CPS
Rahman et al. 2019	Yes	No	Yes	MHE automation, AGV scheduling
De Ryck et al. 2020	Yes	No	Yes	Centralized vs. Decentralized AGV control
Y. Zhang, G. Zhang, et al. 2015	Yes	No	Yes	Real-time RFID CPS, optimization
Y. Zhang, Zhu, et al. 2017	Yes	No	Yes	CPS architecture, IoT, AGV control
Yao et al. 2018	Yes	No	Yes	CPS architecture, AGV dispatching
Wan et al. 2017	Yes	Yes	Yes	CPS architecture, IoT, AGV technology
Kumar et al. 2019	Yes	Yes	Yes	Integrated machine and MH scheduling, AGV automation
Pakpahan et al. 2018	Yes	Yes	Yes	Integrated job and MH scheduling
Sahin et al. 2017	No	Yes	Yes	MS+MH, AGV
Nielsen, Do, et al. 2017	No	Yes	Yes	MS+MH, RFID, 'circular' layout, AGV

Table 4.1: Research papers from the literature study

The I4.0 paradigm is frequently discussed in the literature, covering various industrial practices with great results in academic journals and databases. Recent advancements in cloud technology have spawned the CM paradigm that is aimed toward manufacturing operations (Li et al. 2010).

A common finding in the literature regarding I4.0 advancements in material handling is the requirement for a Cyber-Physical System (CPS)—also called a digital twin—which serves the same purpose as a CM service. The CPS term has been more frequently used in literature, and although CM is the preferred term for this thesis, it has been kept in this section to agree with the papers in question. Still, the reader should know that the research state on CPS's is relevant for CM, material handling, and the CMHS.

The CPS topic is still relatively new in the research community and lacks a solid base of actual implementations by practitioners. Thus, the focus on CPS research has been mainly on architectures to handle real-time data and support shop-floor manufacturing. Lee et al. 2015 presented a five-layer CPS architecture as the critical enabler for an intelligent manufacturing operation. Tao et al. 2017 proposed a CPS framework that connected the physical and virtual shop floor with a "Shop-floor Service System," enabled by sensors and other IoT like *Radio-frequency identification* (RFID). Similar concepts were also proposed in Zhong et al. 2016 and Liu, L. Wang, et al. 2019.

There is a clear consensus in all three papers that a CPS enabled by real-time data and other IoT technology has immense economic potential and will improve decision-making capabilities and shop floor flexibility on multiple levels. However, the papers approaching CPS from an architecture perspective marginalizes the inherent risk of investing in a CPS and does not provide clear guidelines for practitioners.

Moving a level deeper into the CPS and how it can support manufacturing operations, the question of how to track materials and equipment arises. Mourtzis et al. 2018 used a cloud-based CPS with real-time data capture of part movement with RFID to perform dynamic job scheduling of sensor-equipped machines. Wan et al. 2017, Kumar et al. 2019 and Pakpahan et al. 2018 investigated the same topic, but also included material handling in their architecture proposal. Greis et al. 2019 proposed a cloud-based demand-driven solution for scheduling by tracking human machine operators with RFID in a cellular manufacturing environment resulting in increased worker utilization and thus reducing the number of workers required to operate the machine cell.

However, real-time data capture has not been exclusive to manufacturing machines to use in job scheduling. The literature study has shown that real-time tracking of equipment and objects is essential to improve material handling efficiency. The majority of papers referencing a method for capturing this kind of data uses RFID. Y. Zhang, G. Zhang, et al. 2015 used RFID to track human-operated trolleys in a cellular manufacturing environment and optimized material handling by minimizing carbon emission and travel costs. Y. Zhang, Zhu, et al. 2017 proposed a CPS control model for material handling that equipped workstations and AGVs with RFID to enable them to communicate directly. Wan et al. 2017 proposed a context-aware shop floor control model where RFID was used to report material quantities at each workstation. Yao et al. 2018 tracked materials with RFID on the shop floor to optimize the scheduling of AGVs.

RFID has been among the favorite IoT technologies to enable real-time control in shop-floor manufacturing and material handling. However, RFID has some limitations; for instance, it requires an object to be physically "scanned" to update the CPS. The technology also struggles with signal distortion and requires a direct line of sight between tag and receiver (Sgarbossa et al. 2020). A faulty scan can lead to errors during the pick and delivery of materials and might be costly or cumbersome to fix. Thus, even though RFID has proven beneficial for manufacturing and material handling efficiency, these drawbacks may act as bottlenecks for further flexibility and productivity improvements on the shop floor.

The MHE is an essential part of any material handling operation, and developments in automation and robotized technologies have spawned AGVs and AMRs. The majority used AGVs as MHE of choice to enable automated material handling in their proposed CPS control model (De Ryck et al. 2020, Y. Zhang, Zhu, et al. 2017, Yao et al. 2018, Wan et al. 2017). Multiple papers were found investigating algorithms for AGV dispatching, but only Rahman et al. 2019 was chosen because of the limited relevance to the literature study. The paper proposed a genetic and greedy algorithm to optimize AGV dispatch in a job shop with a functional layout. Sahin et al. 2017 and Nielsen, Do, et al. 2017 investigated the integration of MH and MS, where AMRs were the only enablers of automation.

AGVs and AMRs are very prevalent in material handling research, mainly due to their inherent automation capabilities that promise increased manufacturing efficiency. However, non-automated MHE is more common among real practitioners. Although "non-automated" was included as a search term, no research papers were found that investigated these types of MHE in an I4.0 environment.

4.2 Research gaps addressed by a CMHS - RQ1

The literature study aimed to answer the following research question:

RQ1 *How can a CMHS improve the material handling activities in manufacturing in terms of flexibility and productivity performance?*

The results from the literature study in the previous section indicate that continuous acquisition of high-accuracy real-time data opens up a new range of possibilities to explore from a material handling perspective. The CPS is also a frequently discussed topic, but most academic research has been centered around architectural proposals that lack clear directions on implementation. The literature study also revealed that RFID had been the most popular Indoor Positioning Technology (IPT) to track equipment and material in real-time. The scanning process is a limiting factor as the position of MHE and materials cannot be tracked passively.

The CMHS solves this issue with an Indoor Positioning System (IPS), where the MHE and objects are equipped with tags broadcasting their locations to the centralized cloud-based CPS with high accuracy and frequency. A more detailed explanation of the IPS and IPTs can be found in Sgarbossa et al. 2020, but in short, it locates objects on the shop floor with triangulation and broadcasts the position data to the CPS several times per second. This approach allows for a more accurate and reliable positioning compared to RFID.

The literature study also revealed the importance of a CPS working as a centralized hub for interconnected equipment and material, acting as an enabler for automation in shop floor processes. Keeping records updated in real-time reduces the risk of misplacing or losing items and improves the ability to control the manufacturing shop floor. Moreover, making the CPS cloud-based improves its computational capabilities and is a crucial contributor to a scalable CMHS.

As conventional material handling techniques have been based on trial-and-error methods, the CMHS introduces a way of automating a manufacturing company's current MHE. As stated in Section 4.1, a lot of academic research on material handling has been concerned with AGV and AMR implementations, disregarding the fact that many industrial sectors and companies use non-automated MHE in their material handling activities.

Combining the theoretical background and the literature study results reveals several limitations regarding conventional material handling methods and recent Industry 4.0 applications. However, these limitations are more centered around what is *not* current literature's focal point, like the lack of automation focus for non-automated MHE. The list below suggests how the features of CMHS can address them:

1. Automation of capabilities for all MHE
2. Increased freedom of movement, not restricted to a predetermined shop floor area
3. Dynamic dispatching when and where MHE is needed
4. Material flow and layout adaptability
5. Preventing errors in the material picking- and delivering process
6. Smart positioning of idle MHE

Automation of multiple MHE types

The academic community's response to automation in material handling is seemingly biased towards investigating how AGVs and AMRs can be used in the CPS and algorithms to optimize their behavior. It is well known that many manufacturing companies still use carts, trolleys, and forklifts for their material handling. Replacing the entire MHE fleet with AGVs is not a sustainable choice from a cost or practicality perspective in all shop floor environments. Based on the literature study, AGVs work well when fixed guided paths can be established, but this also limits the shop floor flexibility as the AGV paths must be reconfigured if changes are made to the production layout or the material flows.

The CMHS provides a way to automate existing MHE in a manufacturing company by automating the dispatching process. Depending on the type of material handling job, the CMHS enables automatic and dynamic selection of the optimal available vehicle to dispatch. For example, the CMHS will know the difference between a worker walking on the shop floor with a trolley and a worker with a jack trolley that can transport pallets. Choosing the right MHE reduces complexity, as multiple MHE types can operate simultaneously in the manufacturing facility.

Increased freedom of movement

Conventional material handling in high material flow environments has been carried out by grouping/clustering MHE in groups where they are assumed to be most useful. In this non-Industry 4.0 environment, the MHE are restricted as they need to "see" (visually) when a new job is available.

The CMHS can solve this problem because it centrally manages the location of each MHE and the material handling jobs. The system introduces a lot more flexibility and removes the requirement to be in visual contact or restricted to a small area.

Dynamic dispatching

Another benefit of the CMHS is the ability to dispatch MHE while they are on the move. RFID tags are passive and need to be scanned by a reader to log position data. The result is a sparsely populated set of data meant to reflect where MHE are in real-time, making the optimization potential of MHE movement limited. Like a pipeline of position data, the IPS flows from the shop floor into the cloud engine continuously. The precision and frequency are very high, and the derived knowledge of where MHE is located can be leveraged to perform dynamic dispatching.

When the MHE completes a job, it will automatically be registered as available in the CMHS and made available for dispatch. The literature study has shown that higher utilization of MHE is an essential contributor to material handling efficiency. Efficient dispatching is, for that reason, the main objective that needs to be solved to maximize the potential of a CMHS, which will be the main focus for the rest of this thesis.

Material flow and layout adaptability

Evaluating process types and choosing a suitable layout has been an essential aspect of planning a manufacturing facility in the past but is also vulnerable to unpredictable demand and sudden changes (Section 4.1). Reconfiguration of production equipment and workflows can be a costly undertaking for a manufacturing company. The CMHS aims to improve material handling such that detailed layout becomes a less cumbersome process in the future.

Error prevention

As identified in 4.1, manual order picking is prone to common errors, like picking the wrong material type or material amount and transporting the material to the wrong location. Occurrences of such errors may be reduced or even eliminated in the CMHS. Due to the interconnectivity property between materials and equipment and the highly accurate IPS, the CMHS will assist the MHE in picking the suitable material with the right amount, transporting it to the right place at the right time with high accuracy.

Smart positioning of idle MHE

Another vital element for efficient material handling is to decide where to position idle MHE. Analogous to the Uber concept, idle MHE can navigate to places on the shop floor where the expected demand density is highest, i.e., where material handling jobs most likely occur. This demand-based navigation is enabled by the real-time positioning data of the MHE and intelligent cloud computing based on knowledge gained over time. The positioning of other MHE can also be considered to avoid MHE accumulation in places where the demand is already met. This approach allows MHE to be proactive even if idle, contributing to increased MHE utilization.

Table 4.2 summarizes current challenges concerning today's material handling in manufacturing shop floors, the consequences this leads to, and the corresponding CMHS capability that may solve these challenges.

Challenge	**Consequence**	**CMHS capability**
Only AGVs have automation capabilities	Manual solutions have to rely on outdated conventional assignment methods	Automation capabilities for all MHE
MHE restricted to predefined areas or flows	Low flexibility, increased risk of bottlenecks	Increased freedom of movement
Need to return back empty-handed after completed job	Low resource utilization	Dynamic dispatching
Reconfiguration of layouts	High reconfiguration costs	Operates independently of layout
Misplaced materials and picking errors	Cumbersome reallocation of materials and risk of stock-out	Accurate logging of MH jobs
Idle MHE returning to fixed idle locations	Longer travel distance to job	Dynamic allocation of idle locations

Table 4.2: Summary of how material handling challenges are addressed by a CMHS

Summary

The CMHS is a concept combining IPS and cloud technology to track MHE and objects in a shop floor manufacturing environment. This new paradigm introduces a new approach to material handling not yet seen in the academic community by giving MHE increased freedom of movement, automating several MHE types, allowing dynamic dispatching, and a higher degree of integration and flexibility. Many challenges have been identified for today's manufacturing shop floors, and the CMHS can utilize new technology to cope with these challenges. The literature study has shown a place for a CMHS in shop floor material handling, but yet to be explored is how dynamic dispatching can be utilized to the fullest.

Chapter 5

Simulation Results and Analysis

In this chapter, the results from the simulation scenarios will be presented and analyzed in succession to form a foundation for discussion of RQ2 and RQ3 in Chapter 6. A recap of the material handling methods used in the analysis is provided, and the stochastic parameters that define the simulation scenarios are presented. Each scenario is then explained in detail, along with the generated results and analysis.

5.1 Methods and stochastic parameters

The three material handling methods that have been selected to demonstrate the capabilities of a CMHS were presented in Section 3.4. The conventional method aims to mimic non-automated material handling and serves as a base of comparison for the different CMHS methods that are put to the test (RQ2). Four heuristic method permutations have also been tested to investigate whether or not a heuristic method outperforms the Pathmind policy (RQ3). Additionally, some scenarios are tested with policies trained for other scenarios and investigate if there is a need for re-training. In total, the methods are summarized in Table 5.1.

MH method	CMHS	Characteristic	Flexibility	# Forklifts
Conventional	No	Visual contact	Low	5
Heuristic STD/NIL	Yes	Static rule	Medium	3
Heuristic STD/CP	Yes	Static rule	Medium	3
Heuristic LWT/NIL	Yes	Static rule	Medium	3
Heuristic LWT/CP	Yes	Static rule	Medium	3
Pathmind policy/policies	Yes	DRL	High	3

Table 5.1: Methods evaluated in the analysis

Based on the shop floor configuration investigated for the scenarios, five forklifts were needed in the conventional method to maintain visual contact with the workstations. After simulating the heuristics with the same amount of forklifts, the performance was not affected considerably compared to a case with three forklifts only. Hence, the conventional method is evaluated with five forklifts while the CMHS methods only have three, i.e., a 40% reduction in fleet size.

Each method has been tested for scenarios developed to simulate different load and stochastic uncertainty levels for the material handling system to handle. The methods are then evaluated based on their achieved product throughput, meaning the number of products that complete all process stages and forklift utilization by running 100 Monte Carlo iterations for each method/scenario combination. The scenarios have been constructed by assigning parameters to specific values and distributions. These stochastic parameters are summarized in the list below:

- Hourly arrival rate of product and supply made available at the supermarket.
- Number of products passed through the workstation until maintenance is required
- Workstation maintenance/repair time

Assigning these parameters to different distributions makes it possible to simulate different levels of uncertainty on the shop floor. 11 scenarios have been formulated to capture as many stochastic aspects as possible of the material handling operations within the scope of this study. First, three scenarios without arrival variability and workstation downtime were constructed to mimic predictable material flow. Then, stochastic loads with different levels of variability were introduced to evaluate the material methods under these conditions. Finally, workstation failures and downtime were introduced together with stochastic loads to assess its influence on the shop floor for each method.

Uncertainty and arrival rate differences due to demand changes and unexpected workstation failures are chosen as the main stochastic elements for two reasons. First, as the case study information includes daily material flow data and a workstation location overview, the most relevant factors are demand- and workstation-related. These stochastic elements are also easy to implement and configure in the simulation model such that ensuring replicability over multiple experiments becomes feasible. One cannot rule out that production companies may have different stochastic processes in their facility that are not captured in this thesis. Still, unexpected demand and production breaks are common in manufacturing and highly correlated to the system's productivity (Sajadi et al. 2011).

5.2 Scenarios

The following list is an overview of each scenario:

(sc1) Unlimited product and supply level

(sc2) Flooding non-stochastic load

(sc3) Moderate non-stochastic load

(sc4) Flooding stochastic load

(sc5) High variance stochastic load

(sc6) Supply shortage

(sc7) Moderate stochastic load

(sc8) Flooding stochastic load w/ moderate failure

(sc9) Flooding stochastic load w/ frequent failure

(sc10) Moderate stochastic load w/ moderate failure

(sc11) Moderate stochastic load w/ high frequency failure

5.2.1 Scenario 1 – Unlimited product and supply level

`sc1` is the non-stochastic without any arrival rate in the supermarkets or failure rates in the workstations. New product ready to be distributed to its first production stage is always available, and the supply pickup supermarket never runs out. This scenario makes the shop floor environment predictable, and all workstation and material handling jobs could, in theory, be pre-planned without the need for re-planning or scheduling.

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	60.54	2.837	0.524	0.006
STD/NIL	71.13	2.707	0.975	0.009
STD/CP	71.55	2.743	0.98	0.007
LWT/NIL	70.27	2.799	0.981	0.007
LWT/CP	70.8	2.582	0.983	0.007
Pathmind	59.61	10.459	0.894	0.1

Table 5.2: Results `sc1`

The conventional and Pathmind methods are evenly matched on product throughput, while the heuristic methods have a near 100% utilization and even higher throughput. Pathmind performs worse than anticipated, but the high $\sigma_{Throughput}$ value might indicate that the learned policy struggles to prioritize what kind of material handling jobs to execute when the product and supply levels are unlimited.

Pathmind seems to prioritize the dispatching of MHE to the supermarkets with lower prioritization for products further down the processing stages. The consequence of this behavior is that the workstation capacity limit is reached, resulting in large queues. The conventional method can cope with these queues more efficiently as forklifts monitor all workstations across the shop floor, enabling more responsiveness to the queues. High throughput levels for the Pathmind method on par with the heuristic are achieved in some simulation runs, but the occasional poor result drastically reduced the average throughput performance.

The throughput deviance between the methods is further manifested by the highly skewed distribution for the Pathmind method, as depicted in figure 5.1. Although high performance (from 57 to 69 in total throughput) accounts for 88% of the simulation runs, extreme outliers significantly affect average performance.

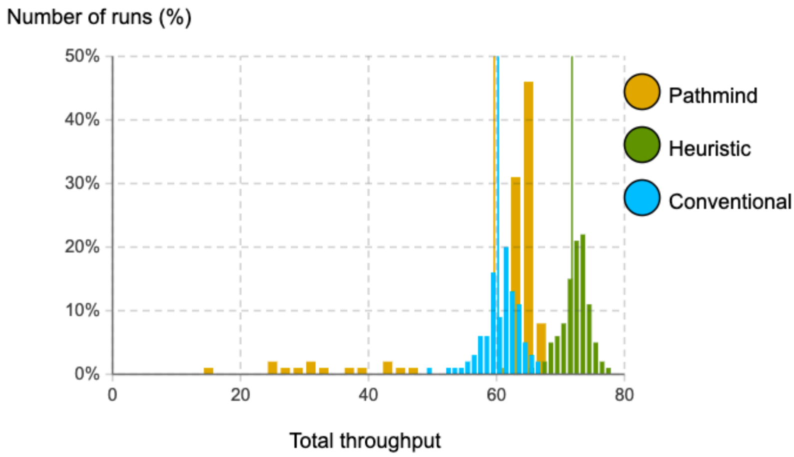

Figure 5.1: Total throughput histogram for `sc1`

5.2.2 Scenario 2 - Flooding non-stochastic load

`sc2` introduces a constant, non-stochastic arrival rate of 8 products per hour. In contrast to `sc1`, this scenario does not have unlimited amounts of products and supply, but the loads are still high relative to the workstation capacities. Hence, there is still a risk of flooding the system with products and generate workstation queues.

Parameter	Value
minArrivalRate	8 per hour
maxArrivalRate	8 per hour
arrivalRate	constArrival(8) per hour
supplyRate	3*constArrival(8) per hour

Table 5.3: Parameters `sc2`

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	45.46	7.047	0.458	0.037
STD/NIL	53.57	6.209	0.805	0.051
STD/CP	54.36	5.984	0.829	0.056
LWT/NIL	54.58	7.193	0.829	0.056
LWT/CP	54.79	6.854	0.807	0.06
Pathmind	56.44	6.204	0.809	0.077

Table 5.4: Results `sc2`

With a reduction in product arrivals, the conventional method has lower throughput and utilization than the other methods. Furthermore, the Pathmind method improves its performance drastically compared to `sc1` and is on a par with the heuristics. As the number of arrivals reduces, the workstations are flooded to a lesser degree than in `sc1`, giving the forklifts more space for strategic positioning and thoughtful task priorities. This scenario is in favor of more sophisticated knowledge like the Pathmind method, as substantiated by the measured performance depicted in table 5.4. However, these DRL capabilities are still not remarkably prominent in high product loads, and the heuristics will still be deemed sufficient in this scenario.

The distribution of total throughput is less skewed for the Pathmind method compared to `sc1`. However, simulation run outliers that perform considerably lower than both the conventional method and the heuristics are still observed for Pathmind, as demonstrated in figure 5.2. These outliers should be taken seriously since the shop floor greatly benefits stability for the dispatching policy. Simultaneously, as the material handling activities are handled by human-operated vehicles, one could argue that throughput collapses can be detected and manually overridden by the workers.

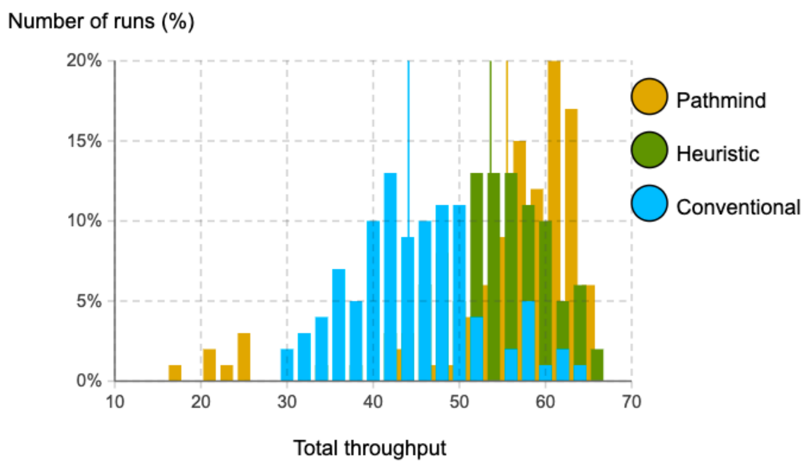

Figure 5.2: Total throughput histogram for `sc2`

5.2.3 Scenario 3 - Moderate non-stochastic load

`sc3` reflects a moderate scenario with a medium load of products introduced on the shop floor, and the material flows are still predictable without any stochastic elements. 'Moderate' in this context is related to the workstation capacities specific to this particular shop floor. Hence, this scenario will rarely encounter any overflow of products at the workstations.

Parameter	Value
minArrivalRate	4 per hour
maxArrivalRate	4 per hour
arrivalRate	constArrival(4) per hour
supplyRate	3*constArrival(4) per hour

Table 5.5: Parameters `sc3`

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	8.91	3.709	0.22	0.024
STD/NIL	10.62	5.701	0.436	0.037
STD/CP	10.11	5.587	0.395	0.038
LWT/NIL	10.63	5.425	0.439	0.039
LWT/CP	10.90	5.896	0.402	0.043
Pathmind	26.51	4.951	0.418	0.054

Table 5.6: Results `sc3`

With a more controlled flow of products, table 5.6 shows significant improvements for the Pathmind method with over 197% and 143% increase in total throughput compared to the conventional and LWT/CP heuristic, respectively. Although LWT/CP performs slightly better than the other heuristics, no emerging pattern of dispatch or idle rule that performs best is identified. Furthermore, the heuristics have marginally higher throughput than the conventional method with considerably higher $\sigma_{Throughput}$.

The forklifts' ability to position themselves strategically becomes increasingly crucial in this scenario due to the lower product load. The utilization is equal, or even lower in some cases, for the Pathmind method than the heuristics. This result indicates that freedom of movement independent of predetermined idle locations is critical to achieving the highest possible throughput performance. When material handling jobs are not available at all times, strategic positioning in the idling periods becomes especially important.

The results are further enhanced by the distribution in figure 5.3. All methods follow a normal distribution without any extreme outliers. The LWT/CP heuristic is the only method with a fatter tail to the right of the mean. This observation is as expected due to the high variance for the heuristic methods.

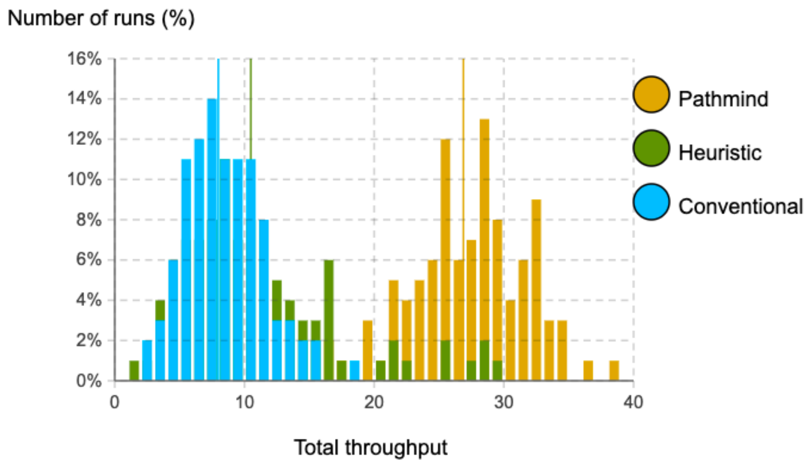

Figure 5.3: Total throughput histogram for `sc3`

5.2.4 Scenario 4 - Flooding stochastic load

`sc4` introduces a high load, stochastic arrival rate to the shop floor environment with the following parameters:

Parameter	Value
minArrivalRate	5 per hour
maxArrivalRate	9 per hour
arrivalRate	uniformDiscrete(5, 9) per hour
supplyRate	3*uniformDiscrete(5, 9) per hour

Table 5.7: Parameters `sc4`

The parameters are evaluated with the `uniformDiscrete(minArrivalRate, maxArrivalRate)` function where a number in the distribution is returned for each product. The supply has an arrival rate of 3∗`uniformDiscrete(minArrivalRate, maxArrivalRate)` as supply stockouts aren't investigated in this scenario.

The high load, stochastic arrival rate introduces the possibility of uneven distribution of new products on the shop floor, leading to extended pickup and delivery queues in the workstations. Hence, there is still a risk of flooding the system with products as in `sc2`.

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	34.12	16.106	0.391	0.09
STD/NIL	43.48	16.235	0.715	0.152
STD/CP	42.59	16.51	0.687	0.153
LWT/NIL	43.1	15.849	0.719	0.142
LWT/CP	41.98	15.531	0.703	0.147
Pathmind	48.41	8.435	0.687	0.11
Pathmind (sc1)	45.84	12.831	0.707	0.159

Table 5.8: Results `sc4`

As for the previous scenarios, the conventional method has the lowest average throughput. Moreover, its utilization is significantly lower than all other methods. Due to visual contact requirements, combined with an uneven product balance, a few forklifts may be significant contributors to lower utilization as material handling jobs occurs less frequently in their predetermined working area.

The heuristics are more or less evenly matched, although we see a slight increase in total throughput for the STD/NIL heuristic. However, this difference does not conclude which heuristic is the most suitable in this scenario, as the throughput deviance is correspondingly higher for the heuristics with higher throughput mean.

The scenario-trained Pathmind policy yields the highest average throughput and a significantly lower $\sigma_{Throughput}$ compared to the conventional and heuristic methods, indicating that the policy systematically handles the unpredictable arrivals more adequately. Low throughput deviation is especially favorable here as stability is highly appreciated for unpredictable scenarios. The Pathmind policy trained for `sc1` also operates well in this scenario, although with a higher $\sigma_{Throughput}$ and slightly lower throughput.

Comparing the throughput distribution in this scenario to the distribution in `sc2`, we see that the introduction of a stochastic element produces a significantly higher throughput deviation between the Pathmind method and the other methods in `sc4`; `sc2` gives an increase of 3% in total throughput for Pathmind compared to the best-performed heuristic, while `sc4` produce over 11% throughput increase for the same comparison. As hypothesized, Pathmind is more superior in a scenario with unpredictable material flow.

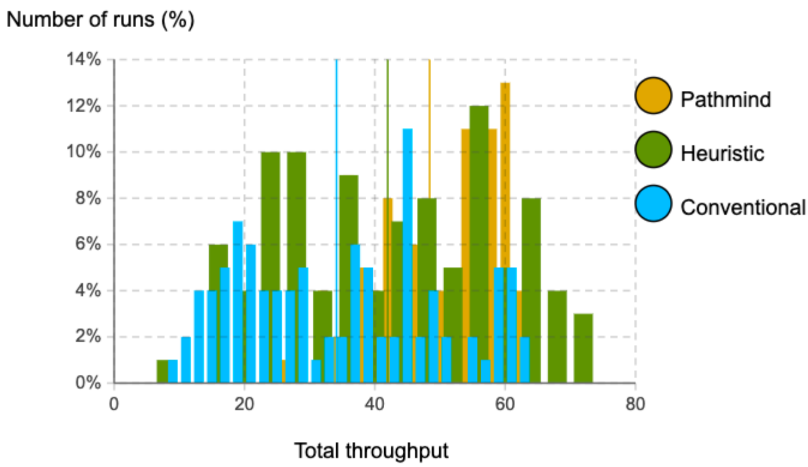

Figure 5.4: Total throughput histogram for `sc4`

5.2.5 Scenario 5 - High variance stochastic load

`sc5` is an extreme scenario where the shop floor environment is exposed to an overflow of arrivals with the following parameters:

Parameter	Value
minArrivalRate	3 per hour
maxArrivalRate	13 per hour
arrivalRate	uniformDiscrete(3, 13) per hour
supplyRate	3*uniformDiscrete(3, 13) per hour

Table 5.9: Parameters `sc5`

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	41.06	23.618	0.385	0.143
STD/NIL	47.92	25.491	0.764	0.225
STD/CP	47.46	25.349	0.743	0.241
LWT/NIL	45.19	23.929	0.745	0.217
LWT/CP	46.2	25.482	0.738	0.254
Pathmind	48.75	15.016	0.692	0.199
Pathmind(sc1)	42.78	19.007	0.665	0.244

Table 5.10: Results `sc5`

The conventional method is again the worst performer at 41 product throughput, although this is still better than the performance in `sc4`. An explanation for this difference may be that, on average, higher loads of products are injected into the system. This theory is further enhanced by the notably large $\sigma_{Throughput}$ for the conventional method.

The best heuristic dispatching rule is STD, suggesting that a distance-based rule is more applicable in highly stochastic shop floor scenarios. However, the idling rules seem to have little influence on the overall result.

The Pathmind policy performs marginally better on product throughput compared to the heuristic methods but has a significantly lower $\sigma_{Throughput}$ value. In a highly unpredictable shop floor environment, achieving the lowest possible deviation in throughput per day may have significant value even though the average throughput is the same.

The importance of low deviation is further enhanced by examining the minimum throughput value for the methods. For specific simulation runs, both the conventional method and the heuristics fail to throughput a single product during the eight-hour workday. In contrast, Pathmind produces a minimum of 18 products for the same scenario. The policy trained for `sc1` is not capable of maintaining the same throughput level and should thus be re-trained.

It is worth noticing that the throughput differences between the CMHS methods are not as evident in this scenario as in `sc4`. Due to large arrival deviations and unique simulation runs through random seeds, finding a dynamic policy that can handle all possible variations the shop floor may encounter is more complicated.

The distribution depicted in figure 5.5 shows how influencing high-variance stochastic loads have on the performance. When the product arrivals vary from 3 to 13 per hour, some simulation runs will naturally have more significant material flow, producing higher throughput. As a result, no precise distribution can be detected for any method.

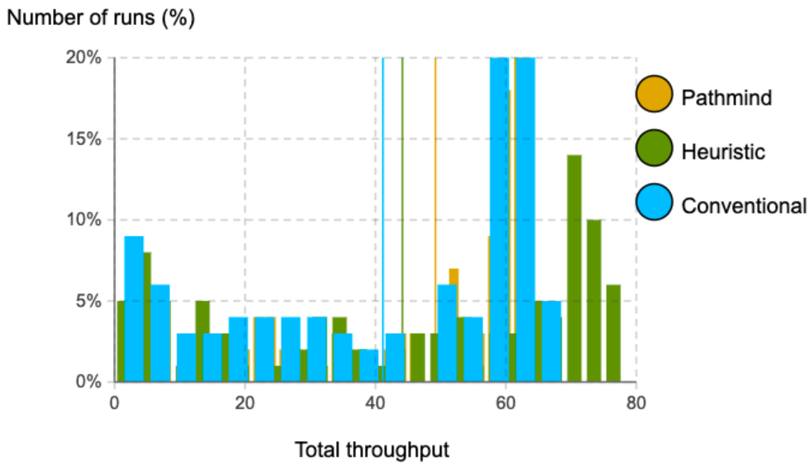

Figure 5.5: Total throughput histogram for `sc5`

5.2.6 Scenario 6 - Supply shortage

`sc6` tests the impact of supply shortage on the product arrival rates from `sc4`. While the product arrival rate is stochastic, the supply rate is constant, causing stock-outs in the supermarket resulting in workstations unable to process products.

Parameter	Value
minArrivalRate	5 per hour
maxArrivalRate	9 per hour
arrivalRate	uniformDiscrete(5, 9) per hour
supplyRate	20 per hour

Table 5.11: Parameters `sc6`

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	31.75	5.551	0.376	0.033
STD/NIL	42.24	7.452	0.687	0.05
STD/CP	40.07	7.443	0.647	0.052
LWT/NIL	41.35	7.084	0.691	0.053
LWT/CP	40.79	7.888	0.659	0.058
Pathmind	48.65	4.427	0.733	0.058
Pathmind(sc1)	41.38	8.652	0.651	0.094
Pathmind(sc4)	47.29	5.911	0.678	0.065

Table 5.12: Results `sc6`

With a supply shortage, the conventional method has the lowest product throughput but with a low deviation. The heuristic methods deliver ten additional products on average. The STD dispatching rule edges out LWT moderately, and the NIL idling rule performs best compared to CP. Idling at the nearest idling location seems to enable a more efficient response to supply shortages.

The Pathmind policy outperforms the other methods by a clear margin depicted by the low deviation and high utilization. Once again, the low deviation creates a more stable throughput and is further enhanced by the methods' minimum throughput value; 38 for Pathmind, 22 for the STD/NIL heuristic, and 16 for the conventional method.

Pathmind can react to stock-outs dynamically and prioritize workstations that are supplied already. The policy trained for `sc4` yields the same product throughput as the scenario trained policy, at the cost of a slight increase in deviation. Acquiring similar results with a not specifically trained policy for the scenario in question is highly relevant as re-training can be avoided, which is both cost-effective and time-saving.

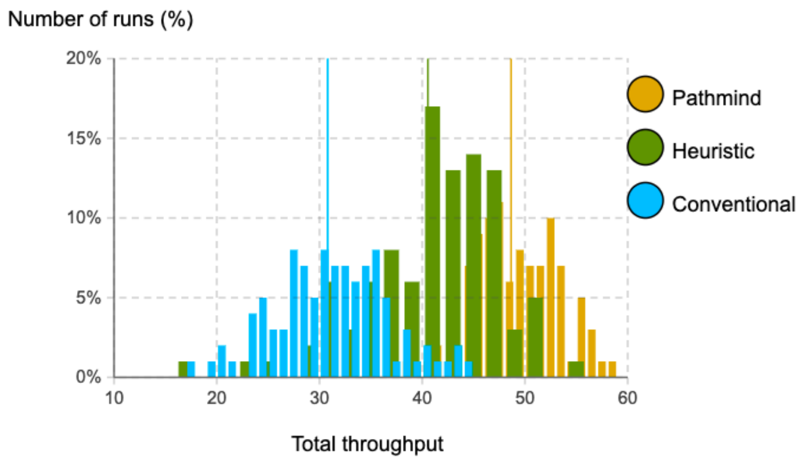

Figure 5.6: Total throughput histogram for `sc6`

5.2.7 Scenario 7 - Moderate stochastic load

`sc7` has a load of new product arrivals that are more in line with the operating capacity of the workstations, leading to less congestion and shortages. The scenario is similar to `sc3` in terms of product load magnitude but differs with the stochastic arrival rate.

Parameter	Value
minArrivalRate	3 per hour
maxArrivalRate	6 per hour
arrivalRate	uniformDiscrete(3, 6) per hour
supplyRate	3*uniformDiscrete(3, 6) per hour

Table 5.13: Parameters `sc7`

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	12.94	10.334	0.239	0.072
STD/NIL	17.17	12.872	0.483	0.107
STD/CP	17.38	14.295	0.444	0.12
LWT/NIL	16.82	13.776	0.477	0.129
LWT/CP	19.79	13.495	0.474	0.113
Pathmind	31.5	9.465	0.483	0.12

Table 5.14: Results sc7

The conventional method is again the worst performer, but the difference is lower than the heuristic methods. The $\sigma_{Throughput}$ is also lower for the conventional method with an almost constant utilization rate of 24%.

LWT with CP idling rule performs best in the Moderate stochastic load environment suggesting that a combination of prioritizing products that have waited for the longest to be picked up leads to gained throughput. Also, positioning idle forklifts in the most active area of the shop floor seems to be beneficial in this scenario.

The Pathmind method is the best performer in this scenario, yielding more than 11 products compared to the best heuristic method with a low deviation. By inspecting the methods' distributions, both the conventional and LWT/CP heuristic are skewed to the left; 60% and 51% of the simulation runs are below 15 in total throughput, respectively. In contrast, throughput with the Pathmind method never falls below 15.

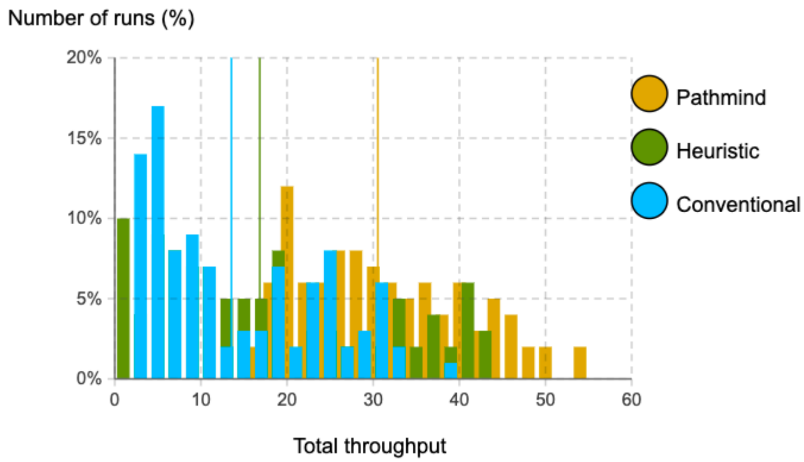

Figure 5.7: Total throughput histogram for `sc7`

5.2.8 Scenario 8 - Flooding stochastic load w/ moderate failure

`sc8` has the same arrival rate used in `sc4` but introduces a reasonable possibility of workstation failure such that repair or maintenance is required.

Parameter	Value
minArrivalRate	5 per hour
maxArrivalRate	9 per hour
arrivalRate	uniformDiscrete(5, 9) per hour
supplyRate	3*uniformDiscrete(5, 9) per hour
productsUntilFailure	20 + uniformDiscrete(0, 5) products
timeToRepair	2000 seconds

Table 5.15: Parameters `sc8`

The time until failure depends on the number of products that have passed through the workstation. This number is determined by a number varying between 20 and 25 products for each workstation. As a result, approximately four down-periods for each workstation occurs on average over eight hours. The repair time is set to 2000 seconds (33 minutes).

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	38.43	14.788	0.382	0.087
STD/NIL	41.52	13.645	0.706	0.118
STD/CP	42.24	13.97	0.634	0.124
LWT/NIL	40.63	15.11	0.698	0.131
LWT/CP	40.27	14.949	0.707	0.137
Pathmind	37.34	9.029	0.706	0.113

Table 5.16: Results sc8

The introduction of workstation failures decreases the throughput gap between the conventional and heuristic dispatching methods. The STD dispatching rule yields the best throughput performance and CP positioning of idle forklifts. The differences are more small yet prominent than most of the scenarios tested previously except sc5 and sc7. The results are inconclusive as to which idle policy works best, indicating a symbiotic relationship in the dispatching rule-pair.

The Pathmind method performs slightly worse conventional when workstation failures are introduced. A probable cause is that the training process cannot take the workstation failures into account before it teaches itself to yield throughput. The failure rates are throughput-dependent. Thus, at the beginning of the learning phase, no/few failures occur. The shortcoming is that the trained policy cannot correct this when flooding arrivals lead to high failure rates.

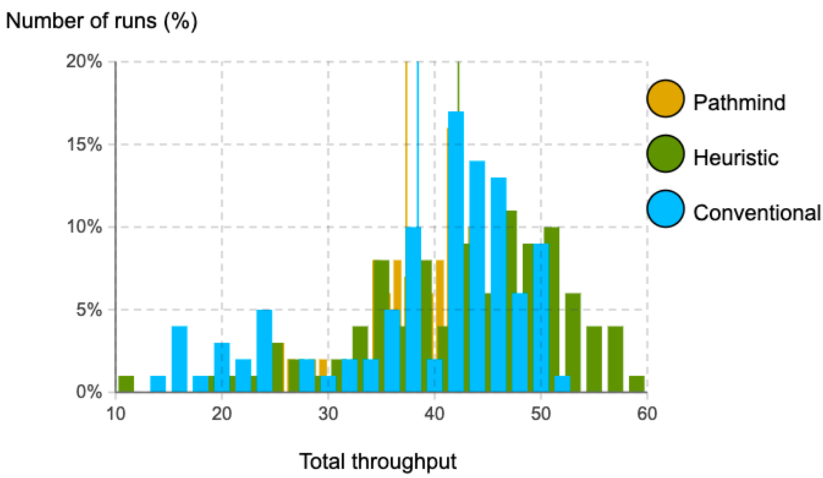

Figure 5.8: Total throughput histogram for sc8

5.2.9 Scenario 9 - Flooding stochastic load w/ frequent failure

`sc9` has the same arrival rate used in `sc4` but introduces the possibility of frequent workstation failures. Compared to `sc8`, the likelihood for workstation repair is almost doubled.

Parameter	Value
minArrivalRate	5 per hour
maxArrivalRate	9 per hour
arrivalRate	uniformDiscrete(5, 9) per hour
supplyRate	3*uniformDiscrete(5, 9) per hour
productsUntilFailure	10 + uniformDiscrete(0, 5) products
timeToRepair	2000 seconds

Table 5.17: Parameters `sc9`

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	37.41	12.465	0.375	0.079
STD/NIL	42.76	12.078	0.67	0.107
STD/CP	41.59	13.87	0.656	0.13
LWT/NIL	42.59	12.256	0.694	0.119
LWT/CP	42.13	12.688	0.67	0.128
Pathmind	38.55	8.421	0.653	0.106

Table 5.18: Results `sc9`

While the conventional and heuristic methods seem unaffected by the increased frequency of workstation failures, the Pathmind method responds poorly. The combined effect of flooding product arrivals and high failure rate appears to affect Pathmind's throughput performance significantly and may be caused by training bias as discussed in 5.2.9. The throughput differences between heuristic dispatching rules seem to diminish as the failure rate increases.

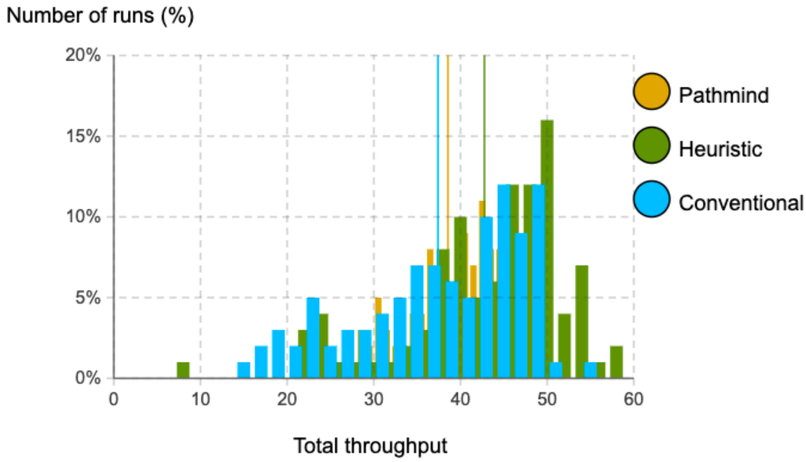

Figure 5.9: Total throughput histogram for `sc9`

5.2.10 Scenario 10 - Moderate stochastic load w/ moderate failure

`sc10` has the same arrival rate used in `sc7` with a moderate probability of workstation failure, approximately two times per workstation.

Parameter	Value
minArrivalRate	3 per hour
maxArrivalRate	6 per hour
arrivalRate	uniformDiscrete(3, 6) per hour
supplyRate	3*uniformDiscrete(3, 6) per hour
productsUntilFailure	20 + uniformDiscrete(0, 5) products
timeToRepair	2000 seconds

Table 5.19: Parameters `sc10`

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	19.52	10.59	0.242	0.067
STD/NIL	22.52	14.978	0.495	0.112
STD/CP	21.35	13.58	0.439	0.104
LWT/NIL	22.63	15.424	0.49	0.114
LWT/CP	23.64	14.174	0.447	0.117
Pathmind	27.08	8.759	0.486	0.119

Table 5.20: Results `sc10`

The conventional method has the lowest throughput, but the difference to heuristic methods has decreased, which seems to be a pattern for workstation failures. The best dispatching rule LWT for both idle rules, but CP is the most suited idling rule when machine failures occur.

Pathmind is the clear best performer in terms of yielded throughput and deviation. Workstation failures seem to be reduced when the system load is lower and, to some extent, matches production capacity.

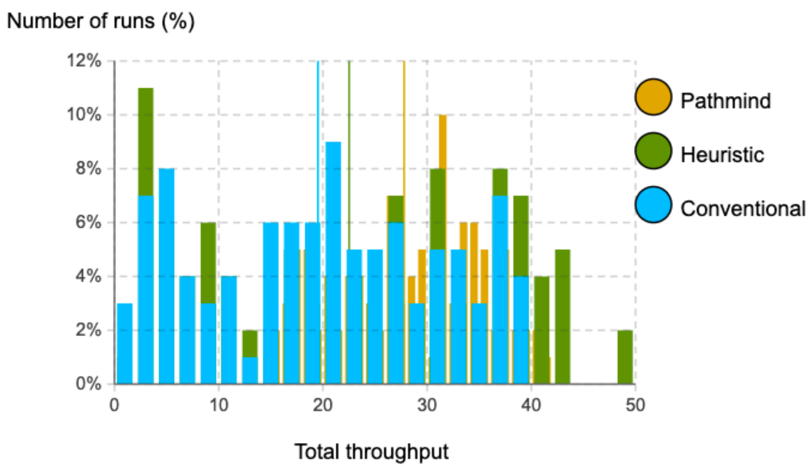

Figure 5.10: Total throughput histogram for `sc10`

5.2.11 Scenario 11 - Moderate stochastic load w/ frequent failure

`sc11` has the same arrival rate used in `sc7` with a very frequent probability of workstation failure, approximately 7 or 8 times per workstation.

Parameter	Value
minArrivalRate	3 per hour
maxArrivalRate	6 per hour
arrivalRate	uniformDiscrete(3, 6) per hour
supplyRate	3*uniformDiscrete(3, 6) per hour
productsUntilFailure	5 + uniformDiscrete(0, 5) products
timeToRepair	2000 seconds

Table 5.21: Parameters `sc11`

Method	Throughput (pcs/day)	$\sigma_{Throughput}$	Utilization (%)	σ_{Util}
Conventional	17.75	11.643	0.237	0.067
STD/NIL	21.96	13.974	0.465	0.107
STD/CP	22.96	14.011	0.445	0.111
LWT/NIL	22.27	13.34	0.481	0.101
LWT/CP	22.56	14.056	0.444	0.108
Pathmind	27.37	6.961	0.482	0.102
Pathmind(sc10)	26.99	6.625	0.432	0.092

Table 5.22: Results `sc11`

The same pattern identified in sc8, sc9, and sc10 continues for the performance difference between the conventional method and dispatching rules. The heuristics are also evenly matched with STD/CP as the best performer, but with no clear winner. Increased failure rates seem to even out the possibility of one dispatching rule-pair beating the others.

The Pathmind policy is the best performer and handles frequent failures well under moderate load, stochastic circumstances. The deviation is also 50% than the best heuristic. In addition, the policy trained in `sc10` performs equally well, indicating an ability to operate independently from workstation failure rates by responding to them dynamically.

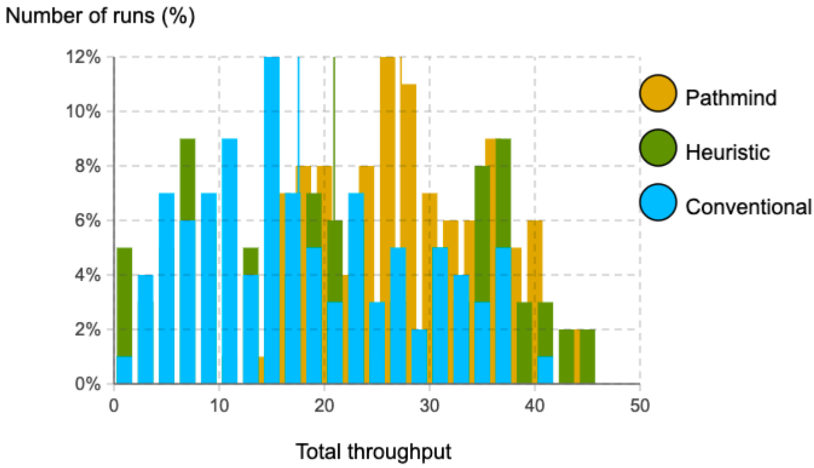

Figure 5.11: Total throughput histogram for `sc11`

5.3 Analysis related to RQ2

The simulation model aimed to answer two of the research questions outlined in the introduction. The second research question was formulated as follows:

RQ2 *In what scenarios should a CMHS be applied in manufacturing shop floors compared to traditional dispatching approaches?*

The simulation results clearly show the CMHS's ability to achieve higher productivity in terms of throughput compared to the conventional method—regardless of the dispatching method—while reducing the active fleet size from five to three MHE. In all the 11 scenarios investigated, there is no single case where the conventional method performs better than the best CMHS method. The increased freedom of movement enabled by the IPS and cloud engine allows a lower number of MHE to cover a larger area, unrestricted by the proximity needed in non-automated conventional material handling.

The CMHS's capabilities become evident in scenarios where the product loads are kept in balance relative to the workstations' capacities. This observation is particularly prominent for the CMHS supported by reinforcement learning, yielding a throughput increase of almost 200% for the moderate non-stochastic load introduced in `sc3`. The conventional approach struggles with high idle times as the forklifts cannot contribute where it is needed, and this observation is further supported by the low MHE utilization in nearly all scenarios investigated.

Simultaneously, the relative gain of a CMHS also seems to diminish as a function of increased load on the material handling system. `sc1` depicts the extreme case where unlimited amounts of products and supplies are available at all times. Although the best CMHS method yields a throughput improvement of 18%, the worst-performing method scores lower than the conventional method. It seems like the load is the determining factor for the method differences, as the performance gain increases to 24% when moderate load restrictions are introduced (`sc2`).

As the load uncertainty increases, the CMHS makes more sense, like the results in `sc4` indicate. However, the performance deviation is not as clear when the variance is extremely high (`sc5`). The shop floor seems to be in such a chaotic condition that even the CMHS struggles to adapt to the load changes. Nevertheless, the conventional method struggles to adapt to changing material flow dynamics more evidently than the CMHS methods, and the results substantiate the inflexibility of predetermined working areas. The high throughput deviation for the conventional method is particularly critical in scenarios where unpredictable flows and sudden workstation failures occur, where a considerable amount of simulation runs fail to produce a single product throughout the eight-hour workday.

The four heuristic methods evaluated in the simulation model have demonstrated mostly equal performance with a few exceptions. There is no evidence of a dispatch/idle rule pair that consistently outperforms the others, but the most common best performers are LWT/CP (4/11) and STD/NIL (4/11). However, only `sc5`, `sc7`, and `sc10` stand out when comparing the differences in total throughput. In `sc10`, the introduction workstation failures yields a slightly lower difference between heuristics compared to `sc7`.

5.4 Analysis related to RQ3

The third research question was concerned with how the CMHS might benefit from more sophisticated logic through reinforcement learning:

RQ3 *When should reinforcement learning support the decision-making process in a CMHS for dispatching the material handling activities?*

The simulation results have demonstrated that reinforcement learning is superior to the heuristics for constant and stochastic load scenarios when the workstations operate within their capacity, reducing the chance of long queues (`sc3`, `sc7`, and `sc10`). This observation is in line with the general analysis for the CMHS as a whole; the performance gain is governed by the relation between product load and workstation capacity.

The introduction of reinforcement learning has significant implications for the forklifts' productivity. The MHE utilization in `sc3` and `sc7` are similar for both Pathmind and the best heuristic, but the former method is far more productive with a throughput increase of 143% compared to the best-performing heuristic. This result indicates that Pathmind is better at strategic positioning of idle MHE under "moderate loads" compared to the predetermined idling areas used by the heuristics. When material handling jobs are not available at all times, strategic positioning in the idling periods becomes particularly important.

When the material handling system is flooded with new arrivals, the Pathmind policy prioritizes the dispatching of MHE to the supermarkets. Under extreme flooding like in `sc1`, the Pathmind policy achieves lower product throughput than the other methods. The relative gap is relatively small for the less severe flooding scenarios. Hence, heuristic methods are deemed sufficient when the workstations are operating close to their maximum capacity.

Pathmind is also able to adapt to stock-outs in supply needed to process products like the performance in `sc6` shows. It identifies which workstation needs to supply the most and, as a result, enables the shop floor to output more products at a stable rate. The literature study revealed that delayed or missing products or supplies are a common cause of problems in shop-floor manufacturing, and the scenario results indicate that Pathmind can extract a lot out of these situations.

The possibility of workstation failures in high load scenarios seems to reduce Pathmind productivity further, whereas both the conventional and heuristic methods have higher throughput in `sc8`. However, lower product loads indicate that workstation failures are less frequent, as observed in `sc10`. Still, the Pathmind performance is unaffected by a quadruple increase in workstation failures in `sc11` and performs significantly better than the other methods. Hence, the critical takeaway is that workstation failure has a smaller impact on productivity if the load levels handled by the Pathmind policy are kept under control.

The simulation analysis also aimed to demonstrate the need for re-training of Pathmind policies when the stochastic elements in the shop floor changed. Drastic changes in load parameters yield a worse performance in Pathmind throughput, facilitating a re-training. However, if the company using this method can quantify the average daily load and variance of time, Pathmind can train the policy to achieve satisfactory results. `sc7` demonstrates this capability as it can handle the uncertain loads much better than the alternative methods. Hence, using reinforcement learning is further enhanced for moderate load scenarios since re-training can be avoided.

Chapter 6

Discussion

This thesis is an in-depth investigation of an applied concept utilizing CM and I4.0 technologies in the shop floor environment. Manufacturers are looking to improve their ability to adapt to changes in demand or unforeseen occurrences in their production processes to stay competitive. While many companies still have not digitized their processes yet, there is a clear consensus that leveraging these technologies is essential to improve flexibility and, in turn, increase productivity.

The CMHS is proposed as a low-cost, low-complexity solution with the promise to increase the overall productivity of manufacturing operations from a material handling perspective. This chapter will discuss the findings from the literature study and simulation results related to the research questions.

The goal is to clarify the factors a potential adopter of a CMHS needs to consider when assessing their material handling operation and manufacturing productivity. Section 6.1 discuss the findings related to RQ1, emphasizing implications the CMHS has on automation in today's manufacturing shop floors. Section 6.2 and 6.3 provides a critical view of the analyzed results in light of RQ2 and RQ3. Practical considerations for a CMHS implementation is outlined in section 6.4, while section 6.5 intends to support the decision for whether or not to adopt the CMHS. Finally, suggestions for further work are presented.

6.1 RQ1 - CMHS and automation

A vital feature of the CMHS is the ability to automate existing, previously non-automated material-handling equipment. The CMHS gives companies another option to guided vehicles and robots, as it allows for preserving the previously non-automated MHE fleet. Humans have a better ability to gather tacit and implicit knowledge of the patterns in a shop floor compared to AGVs and AMRs. Thus, rather than replacing the workers, the CMHS equips already capable human operators with automation abilities. A CMHS implementation would also not require any changes to manufacturing processes or the shop floor layout.

However, there are also apparent reasons for using automated MHE. Unlike human workers, such equipment has a higher work capacity to operate the shop floor beyond the eight-hour workday. Thus, large companies with continuous activity and demand might not be suited to use humans for their material handling activities. That being said, the CMHS is still fit to support shop floors where automated MHE are in use. The necessity of utilizing dispatching methods and tracking both MHE and the environment is just as relevant for automated MHE as for human workers.

Companies with MHE for different types of material handling may use a hybrid solution of AGVs/AMRs and non-automated MHE. The implementation barriers seen in manufacturing companies make it less likely that AGVs or AMRs fulfill all material handling needs for all companies and that a hybrid solution is therefore of interest. CMHS also facilitates collaborative automation for all MHE types. By assigning attributes like weight capacity, vehicle speed, and other factors that play a role in the dispatching choice, distinguishing which MHE to carry out a specific job is a straightforward task for the cloud engine. Although there is appealing evidence for an easy hybrid MHE solution, issues like collision, collaboration, and safety may occur that are difficult to predict in advance.

The introduction of a CMHS will also impact the influence human workers have on the decision-making process. The tacit knowledge human workers possess is challenging to model and measure, and there is a risk that this knowledge may disappear when the cloud makes all decisions. Even though the transferability between simulation and reality with the CMHS is highly accurate in theory, it is always a risk for dissimilarities between the model and the real-world environment. How this will affect the overall performance and well-being of the workers are questions that should be investigated further.

6.2 RQ2 - Dispatching and productivity

The result analysis from all 11 scenarios favors the implementation of the CMHS compared to the conventional benchmark in terms of higher product throughput and efficient MHE utilization. However, the simulation results are obtained from a case study layout that benefits from the increased MHE mobility as there are few areas within line of sight of each other. Thus, the results are less conclusive for layout configurations where the MHE operators can visually observe and use their experience to know where the next job arrives. Simultaneously, as the CMHS operates independently of layout configurations, a CMHS implementation would at the very least be on par with a non-CMHS in terms of productivity and flexibility. The results of the CMHS predominantly outperform the conventional method, so one can argue that even such layouts will benefit from the system but with a lower productivity increase. Hence, the decision for implementing the CMHS boils down to a cost-benefit trade-off for layouts where visual observation and facility-specific knowledge are prominent.

The literature study has revealed that a significant motivator behind automation and digitization in material handling is driven by the intention to increase flexibility and the ability to respond to unforeseen events in real time. Thus, a dynamic material handling system should efficiently control the dispatch of MHE jobs and idle areas on the shop floor. Still, it is equally crucial for manufacturing companies to clearly understand how stochastic elements influence them in the shop floor environment.

In this thesis, uncertainty and load differences in demand and unexpected workstation failures were chosen as stochastic elements. One cannot rule out that production companies may have different stochastic processes in their facility that are not captured in this thesis. These elements may be business-specific, so the CMHS should avoid too much generalization and be implemented concerning the company at hand.

Generalized assumptions on what dispatching method to deploy have next to no value for manufacturing companies looking to extract the most potential from their CMHS. Thus, the simulation scenarios express the different types of stochastic shop floors such that companies can use their as-is situation to gauge which dispatching method works best. The layout and processes in the case study facilitated the use of MHE-initiated dispatching rules. Different dispatching/idling rules may also be appropriate for other configurations. For instance, in cases where machine scheduling is relevant, workstation-initiated dispatching rules can also be investigated.

The dispatching rule heuristics are simple, easy-to-implement approaches that govern which MHE is assigned to which job and where it should return to idle. Although the results are inconclusive on which rule-pair to use, it is evident that the differences increase with the level of uncertainty as long as the material handling system is not flooded with products. When there is much available product to handle, the idle positioning becomes less significant, and the optimal choice of the heuristic rule becomes a zero-sum game. Due to the marginal performance increases and somewhat inconclusive evidence, companies can be encouraged to use trial-and-error to find the optimal dispatching rule-pair over time. Regardless of dispatching rule choice, implementing the CMHS in some way or another is the most decisive factor to improve material handling productivity.

6.3 RQ3 - Dispatching with reinforcement learning

The results showed significant improvements with the introduction of reinforcement learning logic for the CMHS in most scenarios. This improvement was not only in terms of higher throughput and more strategic MHE positioning than the heuristic methods but also with lower deviations and performance consistency.

Simultaneously, some of the scenarios do not obtain this improvement. The reasoning behind the priorities made by a reinforcement learning policy is unknown, and no apparent deficiencies with the Pathmind method are discovered in these scenarios. These findings reveal one of the significant shortcomings with reinforcement learning; the actions can be viewed as a black box, which gives difficulties regarding behavior interpretation and data transparency. Heuristics, on the other hand, are easier to interpret as the rules are tangible and rigorous. Hence, a heuristic approach may be the best starting point for the company's CMHS implementation to evaluate if the system works in line with its purpose.

The results indicate that reinforcement learning policies seem to handle the stochastic load well, but if the arrivals of new products deployed in the material handling system get out of hand, the productivity decreases. There is a possibility of throughput collapse in these situations where the Pathmind policy achieves considerably lower throughput relative to the mean. However, a manual override can be implemented to combat this infrequent issue. Human operators can use their cognitive abilities to notify the cloud engine and trigger a reset, resulting in expected efficiency.

6.4 Practical considerations

A company contemplating whether or not to invest in a CMHS would need to consider the practicalities and cost benefits of the system. The literature study and identified reports from actual manufacturing companies have revealed that the technologies underlying the concept have reached a level of maturity that would enable a realistic implementation of a CMHS on a shop floor. Although cloud computing, third-party software, and real-time indoor tracking have become more accessible in recent years, the feasibility of a CMHS from a business perspective comes down to a cost analysis.

The following expense items need to be evaluated to acquire and maintain a CMHS:

- Purchase of an IPS
- Development or subscription/purchase of CMHS software (cloud CPS)
- Implementation and maintenance costs (consulting)

The IPS used in the Logistics 4.0 lab at NTNU cost around 50k NOK for a $\approx 100m^2$ area. A CMHS makes more sense in more extensive areas as size can be associated with a need to automate. Thus, a purchase cost in the 100K-200K NOK range is a low-end estimate that most potential adopters would need to be willing to invest in the IPS. Comparing these numbers to estimated purchasing costs of single AGVs or AMRs (from 100K to 500K NOK), it becomes clear that decision-makers must decide based on the number of MHE, shop floor size, and cost of labor to operate non-automated MHE. However, as the number of MHE increases, a CMHS becomes more sensible from an economic perspective.

Decision-makers must also take costs related to the development or purchase of CMHS-related software into account. Developing a fully operative CMHS for use in a single manufacturing company is a costly and perhaps risky endeavor due to the novelty of the system and the absence of established actors in the IT industry. On the other hand, there is a possible market segment to provide the CMHS as a Software-as-a-Service (SaaS) as a "full package" solution. A corporation able to offer implementation, policy generation, and maintenance would lift the burden of expertise and risk of the manufacturing companies, but at the expense of increased purchase/subscription costs. Still, the increases in productivity demonstrated by the simulation model and automation capabilities without expensive AGVs/AMRs argue that the CMHS can be an economically viable option in the future.

A proposed strategy for early adopters of the CMHS is to begin by using the low-complexity dispatching rules to assign MHE to jobs and, after some time, start exploring DRL as an option. The Pathmind software used for the policy generation in this thesis is free of use for small-scale experiments, making it possible to investigate its potential in the early stages. For larger enterprise implementations, the software price range from 5000 NOK to 10,000 NOK. Pathmind also necessitates the AnyLogic simulation software to build and deploy the model in the CMHS, costing at least 20,000 NOK per year.

Quantifying the savings or increased earnings is an intricate process as a CMHS can be deployed in a diverse set of shop-floor environments. However, fleet size reduction enabled by a switch from conventional to CMHS methods can yield significant savings while increasing manufacturing throughput. The simulation results revealed a 40% reduction in forklifts needed to serve the material handling operation. A smaller forklift fleet size leads to lower costs, and a CMHS can contribute to a significant cost reduction associated with running a fleet of MHE. If a forklift driver earns 400,000 NOK a year and ten workers are needed to operate five forklifts on two eight-hour shifts, a reduction from five to three forklifts will save the company 1,600,000 NOK each year. Thus, a company can, with some confidence, expect a return on their investment in a CMHS.

Summing up the essential cost components related to the CMHS, it is possible to determine its case-specific applicability with quantifiable variables. In order to quantify the savings, costs, and earnings related to a CMHS implementation, we suggest a generalized economic model to calculate the marginal profit expected for a business, P_b, when switching from a conventional non-automated system to a CMHS. Note that the equation is not used in the result analysis in the previous chapter due to limited case data on costs and product prices. With tangible operational cost parameters for a business, the following equation is proposed:

$$P_b = \sum_{i=1}^{i=n}(p_i * [T_{CMHS_i} - T_{op_i}]) + C_{op} + C_{MHE} * (N_{op} - N_{CMHS}) - (C_{sw} + C_{IPS} + C_c) \tag{6.1}$$

Here, p_i reflects the marginal profit for a single product in product type i, $(T_{CMHS_i} - T_{op_i})$ is the throughput increase for product type i with a CMHS compared to current operations, and C_{op} is cost savings for the alternative operational costs. $C_{MHE} * (N_{op} - N_{CMHS})$ is the cost savings with a reduced MHE fleet compared to current operations. Finally, $(C_{sw} + C_{IPS} + C_c)$ is additional CMHS-related costs like software subscriptions, physical IPS implementation, and consultant expenses. Note that the net profit in 6.1 is a measure between CMHS and conventional non-automated operations. Comparing the CMHS to a system with automated MHE will make the equation different in terms of MHE price, C_{MHE}.

6.5 CMHS evaluation checklists

The main goal of the research questions posed in this thesis was to identify a set of guidelines for manufacturing companies and provide them with a tool to evaluate if a CMHS is suited for their material handling needs. If deemed applicable, the next step was to identify the required level of dispatching complexity and choosing between simple dispatching rules and the DRL method. The problem statement was divided into two parts:

(1) *Should the company implement a Cloud Material Handling System?*

(2) *If so, how sophisticated does the logic need to be to obtain satisfactory results?*

Table 6.1 provides a checklist for a company to decide if the CMHS can be helpful in their operations. If the decision to implement the CMHS is made, Table 6.2 provides a checklist to see if their manufacturing and material flow characteristics will benefit from a deep reinforcement learning dispatching policy:

Situation	**Yes**	**CMHS**
MHE related		
Experiencing low MHE utilization and high idle rates		Enables MHE to take jobs outside a pre-defined area
AGVs/AMRs lack the capabilities required in the material handling operation		Workers can operate MHE
A shift to only AGVs/AMRs is too expensive		CMHS provides a cost-efficient alternative
Human-operated MHE used together with AGVs/AMRs		Automation of all MHE types
Shop floor related		
Material handling is required frequently ($> 50\frac{jobs}{day}$) in the shop floor		Flow is needed to increase overall productivity
MH jobs spanning an area ($> 200m^2$) with obstructed views		CMHS enables increases freedom of movement for MHE and productivity increase
Layout are sometimes reconfigured and material flows can change		Operates independently of layouts and material flows

Table 6.1: Checklist allowing companies to gain insights whether they should implement a CMHS to automate their material handling operation

Situation	Yes	Dispatching method
Constant arrivals of new products		Dispatching rules are sufficient. DRL policy is optimal
Unlimited arrivals of new product		Dispatching rules perform best
Able to avoid flooding the shop floor with new product		DRL works best when the production facility operates within capacity
Risk of supply stock-outs		DRL adapts well to stock-outs
Arrivals ranging from no activity to flooding		Dispatching rules are sufficient. DRL policy not worth the added costs
Frequent stops in production		DRL prioritizes well as long the system is not flooded

Table 6.2: Checklist allowing companies to determine if their operations will benefit from DRL

6.6 Further work

This thesis is the first practical investigation for using a CMHS regarding MHE flexibility and productivity in a manufacturing shop floor. Although the results provide manufacturing companies with pointers on when and how to implement the CMHS, the concept is still in its infancy, and further work is needed.

The information and data basis used for the simulation model is limited due to corporate confidentiality. Future validation and evaluation of the CMHS should be tested in a more practical business case, where real-time data about variables like processing times, arrival rates, and failure rates lays the basis of comparison.

Section 6.4 has given a rough estimate of the costs and savings that follows with a CMHS implementation. A generalized economic model is proposed, but these are estimates that are closely linked to the specific manufacturing processes and material handling operation of each company. Thus, a product earnings increase is only valuable to estimate using prices from the company in question. Extracting key operation parameters reduces the number of assumptions needed and can support a rigorous cost model development.

The investigation should be executed both through simulation and a physical CMHS implementation to confirm that the transferability between simulation and reality is, in fact, as accurate as this thesis indicates. Additional parameters that may affect the variability and uncertainty for the material handling activities are also relevant to implement following the present business case.

The simulation model only considers forklifts as the only MHE. The possibility of having mixed types of MHE in the fleet can be of great interest for companies reluctant to replace an existing MHE fleet or companies that need both human workers and automated vehicles to perform their day-to-day operations. Hence, incorporating multiple MHE types in the examination should be done in practice to investigate the interaction between humans and AGVs/AMRs.

Further investigation should also consider material handling and production scheduling in combination. The products in the example environment used in this thesis moved through three workstations in a predefined order. When considering a large-scaled environment, products may have multiple workstations performing the same tasks. There is a possibility of extending the dispatching policy/rules to contemplate machine scheduling regarding queue lengths, machine utilization, and other parameters relevant to overall productivity performance.

Reinforcement learning has proved to be a highly appropriate extension to the dispatching methods in the CMHS. The use of this particular machine learning method is further justified in literature, where multiple machine learning approaches for dynamic dispatching in material handling are tried out. However, as the CMHS is a relatively new concept, these approaches are not directly transferable to the CMHS. Hence, further work should include other types of machine learning approaches.

Chapter 7

Conclusion

With the introduction of Industry 4.0 technologies, manufacturers seek new opportunities to improve material handling in terms of flexibility and productivity by automating material handling activities. Few attempts have been made to automate manufacturing shop floors dominated by human-operated vehicles like forklifts and pallet trucks. This thesis investigates how real-time positioning of material-handling equipment (MHE) can enhance automation capabilities for non-automated MHE through a Cloud Material Handling System (CMHS) concept.

A literature study was conducted to identify research gaps addressed by a CMHS and how its capabilities can improve the material handling activities in manufacturing. The results show that CMHS addresses several aspects in combination not yet seen in academia, like automation capabilities of all MHE types, dynamic dispatching and allocation of idle locations for the MHE, and universal implementation independent of layout configuration.

A simulation model based on a case study was developed to determine in what scenarios the CMHS proved to be particularly beneficial. A conventional non-automated case, where visual line-of-sight to job arrivals was a prerequisite, served as a benchmark. The scenarios differed in variable product arrival rates, stochastic product loads, and uncertain workstation failures. Out of the 11 scenarios investigated, the CMHS achieved both higher throughput and MHE utilization for all cases with a 40% reduction in the number of MHE required. Most prominent were scenarios with moderate, non-stochastic loads with a throughput increase as much as 197%, suggesting that the CMHS particularly applies for shop floors where product loads are kept in balance relative to the system's capacity.

The simulation model also aimed to determine when sophisticated reinforcement learning policies further enhanced the CMHS implementation. The reinforcement learning method performed better than the heuristic methods in 8 out of 11 scenarios, often with significantly lower throughput deviation. These scenarios were characterized by moderate product loads and variable arrival rates, suggesting that reinforcement learning is more appropriate in these scenarios. However, as the shop floor was exposed to high product loads, which resulted in long workstation queues, the policy struggled notably with extreme throughput outliers. Hence, the heuristics were deemed sufficient in such scenarios.

This thesis is the first attempt to investigate the impact of a CMHS on the manufacturing shop floor. Further work should explore the CMHS from a cost perspective through a practical business case.

Appendix A

How to run the simulation model

This repository (`https://github.com/augustdahl/master`) is made in connection with our master's thesis in Engineering and ICT at NTNU. Please follow the instructions below to run the simulation model:

1. Download the AnyLogic University version (free 30-day trial): `https://www.anylogic.com/downloads/`
2. Download this repository as a zip file and un-zip the folder in your preferred location
3. Open AnyLogic University and the MasterThesis.alp file (located in /master/-MasterThesis/MasterThesis.alp)
4. Navigate to the Monte Carlo Experiment (MC: Main) and run the file
5. Choose one method (Conventional, Pathmind, Heuristic) and one of the 11 scenarios on the left-hand side
6. If the heuristic method is chosen, you are free to choose one of the four pickup/release policy combinations listed. Note that the pickup/release policy is only available for the heuristic method
7. Start the experiment in the bottom left corner. The performance statistics will appear in the histograms
8. If you want to compare different methods, you can do the following:

- Wait until the first experiment is finished
- Click on the pause button in the bottom left corner
- Change to the method you want to compare with and run the experiment again

It is also possible to run a single simulation run to analyze the forklifts' behavior. To do this, you follow the same procedure as above, but now with the Simulation Experiment class (Simulation: Main).

Bibliography

F. Abtahi and I. Fasel. "Deep belief nets as function approximators for reinforcement learning". In: *AAAI Workshop - Technical Report*. Vol. WS-11-15. 2011, pp. 2–7. DOI: 10.3389/conf.fncom.2011.52.00029.

C. W. Anderson, M. Lee, and D. L. Elliott. "Faster reinforcement learning after pretraining deep networks to predict state dynamics". In: *Proceedings of the International Joint Conference on Neural Networks*. Vol. 2015-September. 2015. DOI: 10.1109/IJCNN.2015.7280824.

Tuan Le-Anh and MBM De Koster. "On-line dispatching rules for vehicle-based internal transport systems". In: *International Journal of Production Research* 43.8 (2005), pp. 1711–1728. DOI: https://doi.org/10.1080/00207540412331320481.

Josh Beitelspacher et al. *Policy Gradient vs. Value Function Approximation: A Reinforcement Learning Shootout*. 2006. URL: https://www.researchgate.net/publication/228957531_Policy_Gradient_vs_Value_Function_Approximation_A_Reinforcement_Learning_Shootout.

Z. Bi and D. Cochran. "Big Data Analytics with Applications". In: *Journal of Management Analytics* (2014), pp. 249–265. DOI: 10.1080/23270012.2014.992985.

A. Borschev. *The Big Book of Simulation Modeling: Multimethod Modeling with Anylogic 6*. AnyLogic North America, 2013. ISBN: 0989573176.

Léon Bottou. "Stochastic Gradient Descent Tricks". In: ed. by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Springer Berlin Heidelberg, 2012, pp. 421–436. DOI: 10.1007/978-3-642-35289-8_25.

Lucian Buşoniu, Robert Babuška, and Bart De Schutter. “Multi-agent Reinforcement Learning: An Overview”. In: *Innovations in Multi-Agent Systems and Applications - 1*. Ed. by Dipti Srinivasan and Lakhmi C. Jain. Springer Berlin Heidelberg, 2010, pp. 183–221. DOI: 10.1007/978-3-642-14435-6_7.

P. Centobelli et al. “Layout and material flow optimization in digital factory”. In: *International Journal of Simulation Modelling* (2016), pp. 223–235. DOI: 10.2507/IJSIMM15(2)3.327.

V Chawla, A Chanda, and S Angra. “Automatic guided vehicles fleet size optimization for flexible manufacturing system by grey wolf optimization algorithm”. In: *Management Science Letters* 8.2 (2018), pp. 79–90. DOI: 10.5267/j.msl.2017.12.005.

Hyun-Seon Choi, Ji-Su Kim, and Dong-Ho Lee. “Real-time scheduling for reentrant hybrid flow shops: A decision tree based mechanism and its application to a TFT-LCD line”. In: *Expert Systems with Applications* 38.4 (2011), pp. 3514–3521. DOI: https://doi.org/10.1016/j.eswa.2010.08.139.

Gabriel V. de la Cruz, Yunshu Du, and Matthew E. Taylor. “Pre-training with non-expert human demonstration for deep reinforcement learning”. In: *The Knowledge Engineering Review* 34 (2019), e10. DOI: 10.1017/S0269888919000055.

M. De Ryck, M. Versteyhe, and F. Debrouwere. “Automated guided vehicle systems, state-of-the-art control algorithms and techniques”. In: *Journal of Manufacturing Systems* 54 (2020), pp. 152–173. DOI: https://doi.org/10.1016/j.jmsy.2019.12.002.

Kalyanmoy Deb and Kalyanmoy Deb. “Multi-objective Optimization”. In: *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer US, 2014, pp. 403–449. DOI: 10.1007/978-1-4614-6940-7_15.

Jacques Desrosiers et al. “Time constrained routing and scheduling”. In: *Handbooks in operations research and management science* 8 (1995), pp. 35–139. DOI: https://doi.org/10.1016/S0927-0507(05)80106-9.

Goran Dukic et al. “Simulation Analysis of Models for Estimation of Empty Travel Time of Vehicles in Non-automated Material Handling Systems”. In: *FME Transactions* 46 (2018), pp. 336–341. DOI: 10.5937/fmet1803336D.

Pius J Egbelu. "Positioning of automated guided vehicles in a loop layout to improve response time". In: *European Journal of Operational Research* 71.1 (1993), pp. 32–44. DOI: https://doi.org/10.1016/0377-2217(93)90258-O.

G. Fragapane et al. "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda". In: *European Journal of Operational Research* (2020), pp. 405–426. DOI: https://doi.org/10.1016/j.ejor.2021.01.019.

Vincent François-Lavet et al. "An Introduction to Deep Reinforcement Learning". In: *Foundations and Trends® in Machine Learning* 11.3-4 (2018), pp. 219–354. DOI: 10.1561/2200000071.

W Goddard and S Melville. *Research methodology: An introduction.* Springer, 2014. DOI: 10.1007/978-3-319-74173-4_1.

NP Greis et al. "Manufacturing-Uber: Intelligent operator assignment in a connected factory". In: *IFAC-PapersOnLine* 52.13 (2019), pp. 2734–2739. DOI: https://doi.org/10.1016/j.ifacol.2019.11.621.

William Hellmann et al. "Human, AGV or AIV? An integrated framework for material handling system selection with real-world application in an injection molding facility". In: *The International Journal of Advanced Manufacturing Technology* 101 (2019), pp. 815–824. DOI: 10.1007/s00170-018-2958-x.

Jeffrey W Herrmann. "Information flow and decision-making in production scheduling". In: *IIE Annual Conference and Exhibition 2004* (2004), p. 1. URL: https://www.researchgate.net/publication/238511805_Information_Flow_and_Decision-Making_in_Production_Scheduling.

Ying-Chin Ho and Hao-Cheng Liu. "A simulation study on the performance of pickup-dispatching rules for multiple-load AGVs". In: *Computers & industrial engineering* 51.3 (2006), pp. 445–463. DOI: https://doi.org/10.1016/j.cie.2006.08.007.

H Hu et al. "Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0". In: Elsevier Ltd, 2020. DOI: https://doi.org/10.1016/j.cie.2020.106749.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated Machine Learning - Methods, Systems, Challenges*. Springer, 2019, pp. 3–35. DOI: 10.1007/978-3-030-05318-5.

Illhoe Hwang and Young Jae Jang. "Q() learning-based dynamic route guidance algorithm for overhead hoist transport systems in semiconductor fabs". In: *International Journal of Production Research* 58.4 (2020), pp. 1199–1221. DOI: 10.1080/00207543.2019.1614692.

Max Jaderberg et al. *Population Based Training of Neural Networks*. 2017. arXiv: 1711.09846 [cs.LG].

G James et al. "An Introduction to Statistical Learning". In: Springer, 2013, pp. 26–27. DOI: https://doi.org/10.1007/978-1-4614-7138-7.

Lukasz Kaiser et al. "Model-Based Reinforcement Learning for Atari". In: *CoRR* abs/1903.00374 (2019). URL: http://arxiv.org/abs/1903.00374.

C Karlsson. *Research Operations Management*. Routledge, 2008. DOI: https://doi.org/10.4324/9780203886816.

Haejoong Kim, Dae-Eun Lim, and Sangmin Lee. "Deep Learning-Based Dynamic Scheduling for Semiconductor Manufacturing With High Uncertainty of Automated Material Handling System Capability". In: *IEEE Transactions on Semiconductor Manufacturing* 33.1 (2020), pp. 13–22. DOI: 10.1109/TSM.2020.2965293.

Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

Naveen Kumar, Pankaj Chandna, and Dheeraj Joshi. "Integrated scheduling of part, tool and automated guided vehicles in a flexible manufacturing system using modified genetic algorithm". In: *International Journal of Industrial and Systems Engineering* 32.4 (2019), pp. 443–468. DOI: https://doi.org/10.1504/IJISE.2019.101332.

Y LeCun, Y Bengio, and G Hinton. "Deep Learning". In: The MIT Press, 2015, pp. 436–444. DOI: https://doi.org/10.1038/nature14539.

Jay Lee, Behrad Bagheri, and Hung-An Kao. "A cyber-physical systems architecture for industry 4.0-based manufacturing systems". In: *Manufacturing letters* 3 (2015), pp. 18–23. DOI: https://doi.org/10.1016/j.mfglet.2014.12.001.

B.H. Li et al. "Cloud manufacturing: A new service-oriented networked manufacturing model". In: *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS* (2010), pp. 1–7. URL: https://www.researchgate.net/publication/289781997_Cloud_manufacturing_A_new_service-oriented_networked_manufacturing_model.

Yongkui Liu, Lihui Wang, et al. "Scheduling in cloud manufacturing: state-of-the-art and research challenges". In: *International Journal of Production Research* 57.15-16 (2019), pp. 4854–4879. DOI: https://doi.org/10.1080/00207543.2018.1449978.

Yongkui Liu and Xun Xu. "Industry 4.0 and cloud manufacturing: A comparative analysis". In: *Journal of Manufacturing Science and Engineering* 139.3 (2017). DOI: 10.1115/MSEC2016-8726.

Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

Hongzi Mao et al. "Resource Management with Deep Reinforcement Learning". In: *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. Association for Computing Machinery, 2016, pp. 50–56. DOI: 10.1145/3005745.3005750.

Vladimır Mařık and Jiřı Lažansk. "Industrial applications of agent technologies". In: *Control Engineering Practice* 15.11 (2007), pp. 1364–1380. DOI: https://doi.org/10.1016/j.conengprac.2006.10.001.

V. Mnih, K. Kavukcuoglu, and D. et al. Silver. "Human-level control through deep reinforcement learning". In: *Nature* 521 (2015), pp. 436–444. DOI: 10.1038/nature14236.

A Moeuf et al. "The industrial management of SMEs in the era of Industry 4.0". In: *International Journal of Production Research* (2017). DOI: 10.1080/00207543.2017.1372647.

M Mohri, A Rostamizadeh, and A Talwalkar. "Foundations of Machine Learning". In: The MIT Press, 2018, pp. 1–7, 379–403. ISBN: 978-0-262-01825-8.

Dimitris Mourtzis and Ekaterini Vlachou. "A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance". In: *Journal of manufacturing systems* 47 (2018), pp. 179–198. DOI: https://doi.org/10.1016/j.jmsy.2018.05.008.

KP Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012, pp. 1–10. ISBN: 9780262018029.

Izabela Nielsen, Quang-Vinh Dang, et al. "A methodology for implementation of mobile robot in adaptive manufacturing environments". In: *Journal of Intelligent Manufacturing* 28.5 (2017), pp. 1171–1188. DOI: 10.1007/s10845-015-1072-2.

Izabela Nielsen, Ngoc Anh Dung Do, et al. "Material supply scheduling in a ubiquitous manufacturing system". In: *Robotics and Computer-Integrated Manufacturing* 45 (2017), pp. 21–33. DOI: https://doi.org/10.1016/j.rcim.2016.08.009.

Semih Onut, Selin Soner Kara, and Sinan Mert. "Selecting the suitable material handling equipment in the presence of vagueness". In: *International Journal of Advanced Manufacturing Technology* 44 (2009), pp. 818–828. DOI: 10.1007/s00170-008-1897-3.

Beng Chin Ooi et al. "SINGA: A Distributed Deep Learning Platform". In: *Proceedings of the 23rd ACM International Conference on Multimedia*. Association for Computing Machinery, 2015, pp. 685–688. DOI: 10.1145/2733373.2807410.

ORMS-Today. *SIMULATION SOFTWARE COMPARISON*. Available at https://www.anylogic.com/resources/white-papers/simulation-software-comparison/ (01.05.2021). 2017.

D. Ouelhadj and S. Petrovic. "A survey of dynamic scheduling in manufacturing systems". In: *Journal of Scheduling* 12.4 (2009), pp. 417–431. DOI: https://doi.org/10.1007/s10951-008-0090-8.

Eka KA Pakpahan et al. "Job shop scheduling considering material handling process". In: *MATEC Web of Conferences*. Vol. 204. EDP Sciences. 2018, p. 07008. DOI: 10.1051/matecconf/201820407008.

D Parmenter. *Key performance indicators: developing, implementing, and using winning KPIs*. Vol. 29. Wiley, 2015. DOI: 10.1002/9781119019855.

Paolo Priore, Alberto Gómez, et al. "Dynamic scheduling of manufacturing systems using machine learning: An updated review". In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 28.1 (2014), pp. 83–97. DOI: 10.1017/S0890060413000516.

Paolo Priore, José Parreño, et al. "LEARNING-BASED SCHEDULING OF FLEXIBLE MANUFACTURING SYSTEMS USING SUPPORT VECTOR MACHINES". In: *Applied Artificial Intelligence* 24.3 (2010), pp. 194–209. DOI: 10.1080/08839510903549606.

PWC-Deutchland. *Digital Factories 2020 Shaping the future of manufacturing.* Available at https://www.pwc.de/de/digitale-transformation/digital-factories-2020-shaping-the-future-of-manufacturing.pdf (05.05.2021). 2017.

PWC-US. *Industrial MobilityHow autonomous vehicles can change manufacturing.* Available at https://www.pwc.com/us/en/industrial-products/publications/assets/pwc-industrial-mobility-and-manufacturing.pdf (05.05.2021). 2018.

Wei Qin et al. "Dynamic dispatching for interbay automated material handling with lot targeting using improved parallel multiple-objective genetic algorithm". In: *Computers Operations Research* 131 (2021), p. 105264. DOI: https://doi.org/10.1016/j.cor.2021.105264.

Humyun Fuad Rahman and Izabela Nielsen. "Scheduling automated transport vehicles for material distribution systems". In: *Applied Soft Computing* 82 (2019), p. 105552. DOI: https://doi.org/10.1016/j.asoc.2019.105552.

Horst Rinne. *The Weibull Distribution: A Handbook.* 2008, p. 808. DOI: https://doi.org/10.1201/9781420087444.

I Sabuncuoglu. "A study of scheduling rules of flexible manufacturing systems: a simulation approach". In: *International Journal of Production Research* 36.2 (1998), pp. 527–546. DOI: https://doi.org/10.1080/002075498193877.

Cenk Sahin et al. "A multi-agent based approach to dynamic scheduling with flexible processing capabilities". In: *Journal of Intelligent Manufacturing* 28.8 (2017), pp. 1827–1845. DOI: https://doi.org/10.1016/j.asoc.2012.02.001.

Seyed Mojtaba Sajadi, Mir Mehdi Seyed Esfahani, and Kenneth Sörensen. "Production control in a failure-prone manufacturing network using discrete event simulation and automated response surface methodology". In: *The International Journal of Advanced Manufacturing Technology* 53.1-4 (2011), pp. 35–46. DOI: https://doi.org/10.1007/s00170-010-2814-0.

Thomy Eko Saputro, Ilyas Masudin, and Babak Daneshvar Rouyendegh. “A literature review on MHE selection problem: levels, contexts, and approaches”. In: *FME Transactions* 46 (2015), pp. 336–341. DOI: https://doi.org/10.1080/00207543.2015.1005254.

John Schulman, Sergey Levine, et al. “Trust Region Policy Optimization”. In: (2015), pp. 1889–1897. URL: http://arxiv.org/abs/1502.05477.

John Schulman, Filip Wolski, et al. *Proximal Policy Optimization Algorithms*. 2017. arXiv: 1707.06347 [cs.LG].

Fabio Sgarbossa, Mirco Peron, and Giuseppe Fragapane. “Cloud Material Handling Systems: Conceptual Model and Cloud-Based Scheduling of Handling Activities”. In: *Scheduling in Industry 4.0 and Cloud Manufacturing*. Springer, 2020, pp. 87–101. DOI: 10.1007/978-3-030-43177-8_5.

Anupam Shukla and Ritu Tiwari. *Discrete Problems in Nature Inspired Algorithms*. CRC Press, 2017. DOI: https://doi.org/10.1201/9781351260886.

D. Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 362.6419 (2018), pp. 1140–1144. DOI: 10.1126/science.aar6404.

Reza Sirjani. “Optimal Capacitor Placement in Wind Farms by Considering Harmonics Using Discrete Lightning Search Algorithm”. In: *Sustainability* 9 (Sept. 2017). DOI: 10.3390/su9091669.

R. S. Sutton et al. “Policy gradient methods for reinforcement learning with function approximation”. In: *Advances in Neural Information Processing Systems*. 2000, pp. 1057–1063. DOI: 10.1.1.79.5189.

Richard Sutton, Satinder Singh, and David Mcallester. “Comparing Policy-Gradient Algorithms”. In: *IEEE Trans. on Systems, Man, and Cybernetics* (1983). DOI: 10.1.1.75.430.

RS Sutton and AG Barto. “Reinforcement Learning: An introduction”. In: The MIT Press, 2018, pp. 1–13. ISBN: 978-0-262-03924-6.

A. Tabatabaei, F. Torabi, and T. Paitoon. “Simultaneous scheduling of machines and automated guided vehicles utilizing heuristic search algorithm”. In: *2018 IEEE*

8th Annual Computing and Communication Workshop and Conference (CCWC). 2018, pp. 54–59. DOI: 10.1109/CCWC.2018.8301705.

Fei Tao and Meng Zhang. "Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing". In: *Ieee Access* 5 (2017), pp. 20418–20427. DOI: 10.1109/ACCESS.2017.2756069.

D Telgen et al. "Hierarchical management of a heterarchical manufacturing grid". In: (2014), pp. 825–832. DOI: 10.14809/faim.2014.0825.

James A Tompkins et al. *Facilities planning*. John Wiley & Sons, 2010. DOI: https://doi.org/10.1080/00207543.2011.563164.

Iris FA Vis. "Survey of research in the design and control of automated guided vehicle systems". In: *European Journal of Operational Research* 170.3 (2006), pp. 677–709. DOI: https://doi.org/10.1016/j.ejor.2004.09.020.

Jiafu Wan et al. "Context-aware cloud robotics for material handling in cognitive industrial Internet of Things". In: *IEEE Internet of Things Journal* 5.4 (2017), pp. 2272–2281. DOI: 10.1109/JIOT.2017.2728722.

S. Wang et al. "Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination". In: *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS* (2016), pp. 158–168. DOI: https://doi.org/10.1016/j.comnet.2015.12.017.

Y Wang et al. "Multi-Objective Workflow Scheduling With Deep-Q-Network-Based Multi-Agent Reinforcement Learning". In: Institute of Electrical and Electronics Engineers Inc., 2019, pp. 39974–39982. DOI: 10.1109/ACCESS.2019.2902846.

B Waschneck et al. "Optimization of global production scheduling with deep reinforcement learning". In: *International Journal of Production Research* (2017). DOI: https://doi.org/10.1016/j.procir.2018.03.212.

Marco Wiering and Martijn van Otterlo. *Reinforcement Learning*. Vol. 12. Springer, 2012. DOI: 10.1561/2200000071.

C. Xie and T. T. Allen. "Simulation and experimental design methods for job shop scheduling with material handling: a survey". In: *International Journal of Advanced Manufacturing Technology* 80 (2015), pp. 233–243. DOI: 10.1007/s00170-015-6981-x.

J. Xu et al. "Simulation optimization: A review and exploration in the new era of cloud computing and big data". In: *Asia-Pacific Journal of Operational Research* 32 (2015). DOI: https://doi.org/10.1142/S0217595915500190.

Xun Xu. "From cloud computing to cloud manufacturing". In: *Robotics and computer-integrated manufacturing* 28.1 (2012), pp. 75–86. DOI: https://doi.org/10.1016/j.rcim.2011.07.002.

Fengjia Yao et al. "Optimizing the scheduling of autonomous guided vehicle in a manufacturing process". In: *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. IEEE. 2018, pp. 264–269. DOI: 10.1109/INDIN.2018.8471979.

Xuejun Yue et al. "Cloud-assisted industrial cyber-physical systems: An insight". In: *Microprocessors and Microsystems* 39.8 (2015), pp. 1262–1270. DOI: https://doi.org/10.1016/j.micpro.2015.08.013.

F. Zangaro et al. "An optimization model for the storage assignment of the reference under ergonomics constraints". In: *Proceedings of the Summer School Francesco Turco* (2019), pp. 403–409. DOI: https://doi.org/10.1016/j.ifacol.2018.08.296.

Yingfeng Zhang, Geng Zhang, et al. "An optimization method for shopfloor material handling based on real-time and multi-source manufacturing data". In: *International Journal of Production Economics* 165 (2015), pp. 282–292. DOI: https://doi.org/10.1016/j.ijpe.2014.12.029.

Yingfeng Zhang, Zhengfei Zhu, and Jingxiang Lv. "CPS-based smart control model for shopfloor material handling". In: *IEEE Transactions on Industrial Informatics* 14.4 (2017), pp. 1764–1775. DOI: 10.1109/TII.2017.2759319.

Ray Y Zhong et al. "Visualization of RFID-enabled shopfloor logistics Big Data in Cloud Manufacturing". In: *The International Journal of Advanced Manufacturing Technology* 84.1-4 (2016), pp. 5–16. DOI: 10.1007/s00170-015-7702-1.

Tong Zhou et al. "Reinforcement Learning With Composite Rewards for Production Scheduling in a Smart Factory". In: *IEEE Access* 9 (2021), pp. 752–766. DOI: 10.1109/ACCESS.2020.3046784.

NTNU
Norwegian University of
Science and Technology